

**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E  
PROCESSOS INDUSTRIAIS**

**GEOVANE GRIESANG**

Desenvolvimento de um Mecanismo de Comunicação entre Agentes Pedagógicos em um  
Ambiente Virtual de Aprendizagem

Santa Cruz do Sul, janeiro de 2013.

**UNIVERSIDADE DE SANTA CRUZ DO SUL**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E**  
**PROCESSOS INDUSTRIAIS**

Por

**GEOVANE GRIESANG**

Desenvolvimento de um Mecanismo de Comunicação entre Agentes Pedagógicos em um  
Ambiente Virtual de Aprendizagem

Dissertação apresentada ao Programa de Pós-  
Graduação em Sistemas e Processos Industriais  
da Universidade de Santa Cruz do Sul – UNISC  
como requisito parcial para a obtenção do título  
de Mestre em Sistemas e Processos Industriais.

Orientadores:

Profa. Dra. Rejane Frozza

Prof. Dr. Rolf Fredi Molz

Santa Cruz do Sul, janeiro de 2013

*“As pedras no caminho?  
Vou guardá-las para construir um castelo.”*

Fernando Pessoa

*“Se você faz o que sempre fez, você obterá o  
que você sempre obteve.”*

Anthony Robbins

*“Deus não joga aos dados”.*

Albert Einstein

## **Agradecimentos**

Primeiramente, agradeço a Deus pela dádiva da vida e em poder realizar esta obra. Agradeço imensamente a todos os meus familiares, principalmente aos meus pais, Otavio e Marlice Griesang, pelo incentivo e colaboração nesses longos anos de estudo. Agradeço também as minhas irmãs, Gisele e Grasielle Griesang, pela dedicação para que eu pudesse concluir o mestrado. Agradeço carinhosamente à minha noiva, Taciane Alves Figueirola, uma pessoa muito especial, pela sinceridade e companheirismo sempre presentes, apesar de minha ausência em alguns momentos.

A todos os meus colegas do PPGSPI 2011/2012, mais do que colegas de classe, mais do que parceiros no estudo, irmãos que fizeram parte de um cotidiano.

Ao apoio recebido de todos os meus colegas e amigos, em especial, Gilberto Dessbesell Jr.. Sinceros agradecimentos à direção e aos colegas da empresa Cantú & Stange Ltda, pela contribuição, pela confiança e pelo grande apoio durante mais essa etapa.

Enfim, muito obrigado ao corpo docente e secretariado do Programa de Pós-Graduação em Sistemas e Processos Industriais - PPGSPI (Mestrado), que compartilharam de seus conhecimentos, especialmente minha orientadora, professora, Dra. Rejane Frozza e ao meu co-orientador, professor, Dr. Rolf Fredi Molz que me acompanharam no decorrer desse projeto, contribuindo para o sucesso do mesmo.

Também agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo apoio financeiro, que permitiu que essa etapa fosse alcançada. E a todos aqueles que contribuíram na busca deste ideal, meus mais profundos e sinceros agradecimentos.

## Resumo

A aprendizagem é uma das mais importantes habilidades do ser humano. Com isso, diversas táticas e tecnologias estão sendo usadas para motivar os estudantes na aquisição de novos conhecimentos. A tecnologia da informação está cada vez mais presente, como por exemplo, nos Sistemas Tutores Inteligentes (STI) que têm sido utilizados no processo de ensino-aprendizagem. O STI desenvolvido na UNISC (Universidade de Santa Cruz do Sul) é um projeto de pesquisa de professores, estudantes, e ex-estudantes da universidade. Esse ambiente vem recebendo diversas atualizações durante os últimos anos, como por exemplo, a inserção de dois agentes pedagógicos com emoções para interação com os estudantes. No entanto, não havia interação entre esses agentes. Portanto, o desenvolvimento de um mecanismo de comunicação entre esses agentes pedagógicos é o foco deste trabalho. Além dos agentes tutor e companheiro, já existentes no STI, também foi desenvolvido o agente facilitador, que se comunica com os agentes pedagógicos para determinar quem deve interagir com o estudante. Como o ambiente é reativo (gera uma resposta para um estímulo), o agente escolhido deve ter a habilidade de tratar o estímulo gerado pelo estudante no STI. Como mais de um agente pedagógico pode responder a um mesmo estímulo, uma variação do método da roleta (algoritmos genéticos) foi desenvolvida para determinar qual agente deve iniciar a interação com o estudante. O protocolo de comunicação/interação e o formato das mensagens trocadas entre os agentes foram modelados com base na fundamentação FIPA (*Foundation for Intelligent Physical Agents*). Por fim, cenários de interação foram gerados para a realização de testes, no conteúdo referente ao “Novo Acordo Ortográfico da Língua Portuguesa”, com objetivo de avaliar o mecanismo desenvolvido.

**Palavras-chave:** processo de ensino-aprendizagem, sistemas tutores inteligentes, sistemas virtuais de aprendizagem, mecanismo de comunicação entre agentes pedagógicos.

## **Abstract**

Learning is one of the most important skills of being human. With this, many tactics and technologies are being used to motivate students in the acquisition of new knowledge. Information technology is increasingly present, e.g. in Intelligent Tutoring Systems (ITS) that have been used in the teaching-learning process. The ITS developed at UNISC (University of Santa Cruz do Sul) is a research project for teachers, students, and ex-students of the university. This environment has received several upgrades in recent years, e.g. the insertion of two pedagogical agents with emotions to interact with students. However, there was no interaction between these agents. Therefore, development of a communication mechanism between those pedagogical agents is the focus of this work. Besides the tutor and companion agents, already existent in the ITS, also was developed the facilitator agent, which communicates with pedagogical agents to determine who must interact with the student. Because the system is reactive (generates a response to a stimulus), the chosen agent should have the ability to treat the stimulus generated by the student in ITS. How more of one pedagogical agent can respond to the same stimulus, a variation of the method of roulette (genetic algorithms) was developed to determine which agent should start the interaction with the student. The protocol communication/interaction and format of messages exchanged between agents were modeled based on the grounding FIPA (Foundation for Intelligent Physical Agents). Finally, interaction scenarios were generated for realization of testing, in the referent content the “New Portuguese Language Orthographic Agreement”, with objective to evaluate the mechanism developed.

**Keywords:** teaching-learning process, intelligent tutoring systems, virtual learning systems, communication mechanism among pedagogical agents.

## Sumário

Lista de figuras .....	9
Lista de tabelas .....	11
Lista de quadros.....	12
Lista de abreviaturas .....	13
1 Introdução.....	15
2 Fundamentação teórica.....	20
2.1 Agentes .....	20
2.1.1 Classificação de agentes .....	21
2.1.2 Propriedades de agentes.....	23
2.1.3 Sistemas multiagentes .....	27
2.1.4 Ambientes virtuais de aprendizagem.....	29
2.1.5 Sistemas tutores inteligentes.....	30
2.1.6 Agentes pedagógicos .....	33
2.1.7 Agentes de Interface .....	34
2.1.8 Agentes pedagógicos animados.....	35
2.2 Comunicação entre agentes .....	37
2.2.1 Linguagem de comunicação entre agentes (ACL) .....	38
2.2.2 Protocolo de comunicação entre agentes.....	45
2.2.3 FIPA <i>Request Interaction Protocol</i> (FIPA, 2002a).....	46
2.2.4 FIPA <i>Contract Net Interaction Protocol</i> (FIPA, 2002b).....	47
2.2.5 FIPA <i>Propose Protocol</i> (FIPA 2002c) .....	49
2.2.6 FIPA <i>Brokering Protocol</i> (FIPA, 2002d).....	50
2.2.7 FIPA <i>Recruiting Protocol</i> (FIPA, 2002e) .....	51
2.2.8 Quadro comparativo dos protocolos estudados.....	53
2.3 Heurística.....	54
2.3.1 Algoritmos genéticos .....	55
2.3.1.1 Método da roleta.....	56
2.4 STI usado como base do projeto de pesquisa.....	58
2.4.1 Os agentes pedagógicos tutor e companheiro .....	59
2.4.2 Arquitetura conceitual dos agentes pedagógicos inseridos no STI .....	61
2.4.3 Histórico de evolução do ambiente .....	64

2.5	Trabalhos relacionados .....	65
2.5.1	Plataforma COMPOR.....	65
2.5.2	Plataforma SACI .....	67
2.5.3	Arquitetura de um agente identificador de fatores motivacionais e afetivos em um AVA 71	
2.5.4	O papel dos Agentes Inteligentes nos STIs .....	74
2.5.5	Agentes inteligentes para o ensino de lógica.....	77
2.6	Considerações referentes aos trabalhos relacionados.....	78
2.7	Considerações.....	83
3	Desenvolvimento do mecanismo de comunicação.....	86
3.1	Metodologia.....	86
3.2	Desenvolvimento do mecanismo de comunicação.....	89
3.3	Aspectos de implementação .....	92
3.3.1	Java.....	92
3.3.2	PostgreSQL.....	93
3.4	Mecanismo de comunicação entre os agentes pedagógicos .....	99
3.4.1	Fluxograma de funcionalidade dos agentes.....	103
3.5	Formato das mensagens.....	108
3.5.1	Linguagem de comunicação usada.....	108
3.5.2	Linguagem de conteúdo usada .....	111
3.6	Heurística.....	114
3.7	Integração com o STI .....	115
3.8	Cenários de interação.....	119
3.9	Decisões tomadas pelos agentes .....	125
3.9.1	Questionário .....	127
3.9.2	Avaliação .....	128
3.10	Outros desenvolvimentos .....	135
3.11	Considerações.....	140
4	Conclusão .....	142
	Referências .....	147
	Anexo A: Exercícios sobre o novo acordo ortográfico .....	162
	Anexo B: Modelo er completo .....	165
	Anexo C: Questões do questionário .....	167
	Anexo D: Termo de consentimento livre e esclarecido.....	168



## Lista de figuras

Figura 1. Conceito de agentes – adaptado de Wooldridge (2009).....	20
Figura 2. Estrutura de um sistema multiagente – adaptado de Jennings (2000) .....	28
Figura 3. Arquitetura clássica de um STI - Goulart e Giraffa (2001) .....	31
Figura 4. Arquitetura de um STI proposta por Self (1999) - Goulart e Giraffa (2001).....	32
Figura 5. Camadas da linguagem KQML – adaptado de Finin et al. (1994a).....	40
Figura 6. Protocolo de interação <i>Request</i> – adaptado de FIPA (2002a).....	46
Figura 7. Protocolo de interação <i>ContractNet</i> – adaptado de FIPA (2002b).....	48
Figura 8. Protocolo de interação <i>Propose</i> – adaptado de FIPA (2002c) .....	49
Figura 9. Protocolo de interação <i>Brokering</i> – adaptado de FIPA (2002d) .....	51
Figura 10. Protocolo de interação <i>Brokering</i> – adaptado de FIPA (2002e) .....	52
Figura 11. Exemplo do algoritmo de seleção método da roleta (TANOMARU, 1995).....	57
Figura 12. Caricaturas em 2D e os estados emocionais da agente Dóris (SANTOS, 2001) ....	59
Figura 13. Caricatura em 2D do agente companheiro Dimi (SILVA, 2002) .....	60
Figura 14. Caricatura em 3D da agente tutora Dóris (FROZZA et al, 2009a) .....	61
Figura 15. Caricatura em 3D do agente companheiro Dimi (KÜNZEL, 2010) .....	61
Figura 16. Arquitetura conceitual dos agentes tutor e companheiro (FROZZA et al., 2007) ..	62
Figura 17. Arquitetura modificada do Eletrotutor IV – Moissa (2001) .....	71
Figura 18. Arquitetura do agente Percepção – Moissa (2001) .....	73
Figura 19. Arquitetura do ambiente – Moissa (2001) .....	73
Figura 20. Arquitetura para troca de mensagens – adaptado de Frigo et al. (2004).....	76
Figura 21. Comunicação entre agentes no ambiente OHALYEN - González et al. (2007).....	78
Figura 22. Fluxograma executivo do projeto de pesquisa .....	87
Figura 23. Comunicação de tarefas diferentes e simultâneas .....	89
Figura 24. Comunicação de tarefas iguais e simultâneas .....	90
Figura 25. Comunicação com empilhamento de mensagens.....	91
Figura 26. Modelo ER – Tabela Aluno .....	95
Figura 27. Modelo ER – Tabela Turma.....	96
Figura 28. Modelo ER – Tabela Aula.....	97
Figura 29. Modelo ER – Tabela Pagina .....	97
Figura 30. Modelo ER – Tabela Exercício .....	98

Figura 31. Cenário antigo da comunicação entre os agentes pedagógicos.....	99
Figura 32. Novo cenário da comunicação entre os agentes pedagógicos.....	100
Figura 33. Protocolo de comunicação/interação desenvolvido .....	102
Figura 34. Fluxograma do agente facilitador – estímulo recebido .....	104
Figura 35. Fluxograma do agente pedagógico – lista habilidade .....	106
Figura 36. Fluxograma do agente pedagógico – executar habilidade .....	106
Figura 37. Fluxograma do agente facilitador – atualiza interação .....	107
Figura 38. Formato das mensagens .....	109
Figura 39. Exemplo de destruição de agentes no método da roleta .....	115
Figura 40. Funcionamento das <i>threads</i> no protocolo desenvolvido.....	118
Figura 41. Cenário 01 – interação coordenada - Dimi .....	119
Figura 42. Cenário 01 – interação coordenada - Dóris.....	120
Figura 43. Cenário 02 – empilhamento de mensagens – estímulo 1 .....	121
Figura 44. Cenário 02 – empilhamento de mensagens – estímulo 2.....	122
Figura 45. Cenário 02 – empilhamento de mensagens – fim da interação.....	122
Figura 46. Cenário 06 - <i>Smartphones</i> executando o ambiente.....	124
Figura 47. Aula sobre o novo acordo ortográfico – histórico.....	125
Figura 48. Aula sobre o novo acordo ortográfico – hífen parte 1 .....	126
Figura 49. Aula sobre o novo acordo ortográfico – exercícios .....	126
Figura 50. Modelo ER – Tabela Questionario.....	127
Figura 51. Alunos de Lógica para Computação da UNISC realizando a avaliação.....	129
Figura 52. Gráfico com o resultado da questão 1 - questionário.....	129
Figura 53. Gráfico com o resultado da questão 2 - questionário.....	130
Figura 54. Gráfico com o resultado da questão 3 – questionário .....	130
Figura 55. Gráfico com o resultado da questão 4 - questionário.....	131
Figura 56. Gráfico com o percentual de acertos e erros - exercícios.....	132
Figura 57. Pergunta para estudante que pulou de página .....	133
Figura 58. Pergunta para estudante que está com dúvida.....	133
Figura 59. Pergunta específica realizada pela agente Dóris .....	134
Figura 60. Aproveitamento do estudante nos exercícios.....	134
Figura 61. Identificação do agente e caixa de diálogo sem o botão fechar .....	135
Figura 62. Mensagem para impedir que o estudante não responda uma pergunta.....	136
Figura 63. Erro ao não responder um exercício.....	136
Figura 64. Mensagem de aproveitamento do estudante .....	139
Figura 65. Gabarito do exercício .....	139

## Lista de tabelas

Tabela 1 – Parâmetros das mensagens FIPA-ACL .....	42
Tabela 2 – Atos comunicativos FIPA-ACL .....	42
Tabela 3 – Quadro comparativo dos protocolos.....	53
Tabela 4 – Evolução do STI no decorrer dos anos .....	64
Tabela 5 – Tabela comparativa dos trabalhos relacionados .....	78

## Lista de quadros

Quadro 1 – Exemplo 01: Mensagem FIPA-ACL .....	44
Quadro 2 – Exemplo 02: Mensagem FIPA-ACL .....	45
Quadro 3 – Algoritmo da roleta.....	58
Quadro 4 - Exemplo de regra de produção.....	<b>Erro! Indicador não definido.</b>
Quadro 5 – Exemplo de uma mensagem de consulta do agente Facilitador .....	110
Quadro 6 – Exemplo de mensagem de informação da agente Dóris.....	113
Quadro 7 – Exemplo de formato de mensagem usado no STI.....	117
Quadro 8 – Exemplo de registro da comunicação entre os agentes .....	138

## Lista de abreviaturas

2D	Duas dimensões
3D	Três dimensões
ACL	<i>Agent Communication Language</i>
AG	Algoritmos Genéticos
AIML	<i>Artificial Intelligence Markup Language</i>
API	<i>Application Program Interface</i>
AUML	<i>Agent Unified Modeling Language</i>
AVA	Ambiente Virtual de Aprendizagem
CFP	<i>Call For Proposal</i>
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
ER	Entidade/Relacionamento
FAQ	<i>Frequently Asked Questions</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
GEFT	<i>Group Embedded Figures Test</i>
HTML	<i>HyperText Markup Language</i>
IA	Inteligência Artificial
IEC	Inventário de Estilo Cognitivo
JADE	<i>Java Agent DEvelopment</i>
JAT	<i>Java Agent Template</i>
JEE	<i>Java Enterprise Edition</i>
JESS	<i>Java Expert System Shell</i>
JSE	<i>Java Standard Edition</i>
JVM	<i>Java Virtual Machine</i>
KQML	<i>Knowledge Query and Manipulation Language</i>
LDP	Linguagem de Descrição de Protocolos
MCT	Ministério da Ciência e Tecnologia
MIT	<i>Massachusetts Institute of Technology</i>
RMI	<i>Remote Method Invocation</i>

SACI	<i>Simple Agent Communication Infrastructure</i>
SGBDOR	Sistema Gerenciador de Banco de Dados Objeto Relacional do projeto
SL	<i>Semantic Language</i>
SMA	Sistemas Multiagentes
SO	Sistema operacional
STI	Sistema Tutor Inteligente
UNISC	Universidade de Santa Cruz do Sul
VKB	<i>Virtual Knowledge Base</i>
XML	<i>Extensible Markup Language</i>

## 1 Introdução

A utilização de computadores em sala de aula tem contribuído para maior motivação dos estudantes em seu processo de aprendizagem, pois oferece mais diversidade na maneira com que o estudante desenvolve o conhecimento. Portanto, o uso desses equipamentos na educação permite que novos programas educacionais sejam inseridos nos ambientes de ensino-aprendizagem, condizentes às expectativas dos estudantes atuais (CUTMORE et al., 2000; LEE et al., 2004; RAU et al., 2004).

A aprendizagem é uma das capacidades mais valiosas cultivadas pelo ser humano, devido à constante absorção de novos conhecimentos (AZEVEDO e TAVARES, 2001). Segundo Silva e Heck (2008), o uso de soluções tecnológicas nas atividades de ensino-aprendizagem tem sido exposto tanto por professores quanto por estudantes. Esses professores relatam que muitas das atividades realizadas com o uso de computadores não seriam possíveis, ou seriam muito difíceis de serem executadas em sala de aula, apenas com o auxílio do quadro-negro.

Portanto, uma tática de ensino pode apresentar um determinado conteúdo de diferentes formas, como por exemplo, em forma de texto ou imagens. Assim, uma das soluções mais usadas para que essa tática de ensino-aprendizagem ocorra são os chamados ambientes virtuais de aprendizagem (AVA), que são sistemas educacionais com o objetivo de auxiliar os estudantes durante o seu processo de aprendizagem (GUARDIA, 1997; FREEMAN, 2000; GAMBOA e FRED, 2001; VILLAREAL e GIRAFFA, 2001; LIONAZ, 2001; GAVIDIA e ANDRADE, 2003). Trabalhos sobre psicologia cognitiva foram realizados desde os anos 80, apresentando evidências que indicam que os estudantes que interagem com os STIs melhoram o seu próprio desempenho (STERNBERG, 2008; ANDERSON, 2010).

Os STIs são *softwares* educacionais que incorporam técnicas da Inteligência Artificial (IA), como o uso de agentes pedagógicos (GAVIDIA e ANDRADE, 2003). Portanto, a missão pedagógica de um STI está diretamente relacionada a esses agentes pedagógicos que, por sua vez, possuem como principal função a comunicação com o estudante (GIRAFFA, 1999). Assim, os agentes pedagógicos inteligentes são inseridos no sistema para aumentar sua eficiência e para motivar o estudante que interage com esse tipo de ambiente.

Essa motivação pode ser obtida com o uso de agentes animados que, por sua vez, podem expressar emoção através de gestos e expressões de face (JAQUES e VICARI, 2005). Segundo Perkins (1995), “... *as pessoas aprendem mais quando têm uma oportunidade razoável e uma motivação para o fazer*”.

Também é importante destacar que um agente pedagógico que se encontra sozinho dentro de um STI pode ter o controle total sobre o resultado de suas próprias ações. Neste caso, não há necessidade do uso de um mecanismo de comunicação. Porém, de acordo com Barbosa (2005), quando um agente é “*colocado em sociedade, haverá uma interferência social resultante do efeito da ação individual de um agente para o alcance dos objetivos de outros, de modo que serão necessários processos complexos de cooperação, coordenação e negociação*”.

Barbosa (2005) ainda destaca que essa interferência social altera os objetivos que um agente pode atingir, fazendo com que esse agente possa alcançar aqueles objetivos que ele não conseguiria atingir sozinho, com base em uma relação de dependência de outro agente. A teoria da dependência propõe um procedimento complexo, onde um agente delega uma ação para outro agente, para que um determinado objetivo possa ser alcançado (CONTE e CASTELFRANCHI, 1992; CASTELFRANCHI et al, 1992; SICHMAN, 1995; REZENDE et al, 2003).

Os Sistemas Multiagentes (SMA) são *softwares* computacionais com vários agentes interagindo uns com os outros, como em uma sociedade de agentes. Entretanto, cada agente visa satisfazer suas próprias metas para que um objetivo maior e em comum possa ser atingido (WOOLDRIDGE, 2009).

Desta forma, é possível que em um STI haja mais agentes inteligentes interagindo com o estudante. Neste caso, é de fundamental importância que esses agentes também interajam entre si, independentemente se essa interação ocorre de forma direta (via troca de mensagens específicas entre os agentes) ou indireta (via ambiente dos agentes). Com isso, os agentes passam a não mais tomar suas próprias decisões de forma isolada, e sim, passam a interagir para tomar uma decisão em conjunto.



Deste modo, o objetivo deste projeto de pesquisa foi desenvolver um mecanismo de comunicação entre agentes pedagógicos de um STI, com o objetivo de permitir a interação entre eles. Para isto, o STI desenvolvido no Departamento de Informática da UNISC (projeto de estudantes e professores) foi utilizado para a validação da proposta. Portanto, o mecanismo de comunicação procurou atender as necessidades do STI em questão.

O STI desenvolvido teve seus primeiros trabalhos realizados em 1998 (SANTOS et al., 2001) e sua escolha se justifica pelo fato deste sistema estar em constante aperfeiçoamento e estudo (SANTOS et al., 2001; SILVA, 2002; MERGEN e SCHREIBER, 2005; MAINIERI et al., 2005; FROZZA et al., 2007; 2009a; 2009b; 2011; SILVA et al., 2010; BORIN, 2010; KÜNZEL, 2010; KÜNZEL et al., 2011; KÜHLEIS, 2011).

Além disso, os agentes pedagógicos inseridos no STI base se comunicavam com o estudante, no entanto, era necessário que ocorresse uma comunicação entre os próprios agentes. A falta de um mecanismo de comunicação estava causando muita confusão, pois os agentes pedagógicos estavam sugerindo ao estudante tarefas diferentes, conflitantes, simultâneas ou repetidas. O problema mencionado estava ocorrendo pelo fato dos agentes pedagógicos não saberem que o estudante já estava interagindo com outro agente.

Uma parte importante do mecanismo de comunicação são os chamados protocolos de comunicação (interação) entre agentes, que podem ser definidos como um conjunto de padrões especificados e projetados para permitir a comunicação entre agentes. O uso de protocolos de interação não é obrigatório, mas se forem utilizados, a comunicação entre os agentes deve ser consistente com o protocolo escolhido (MAMADOU, 2000).

Contudo, faz-se necessário também a definição de uma linguagem de comunicação, que é um formalismo, com a finalidade de codificar as mensagens trocadas entre os agentes, independente da linguagem de programação usada na aplicação. Essa linguagem de comunicação deve ser usada em conjunto com o protocolo de interação entre agentes (VICARI et al, 2009).

Portanto, o objetivo principal deste projeto de pesquisa é modelar e implementar um mecanismo de comunicação entre agentes pedagógicos em um STI, para que o agente com características de tutor e o agente com características de companheiro pudessem interagir e tomar decisões em relação às atividades dos estudantes.

Para isto, são definidas regras e estruturas do mecanismo de comunicação, sempre levando em consideração as necessidades do STI usado como base para o projeto e pesquisas bibliográficas realizadas em todas as fases do trabalho. Além disso, também é determinado um protocolo para comunicação (interação) entre os agentes, assim como as linguagens de comunicação e conteúdo.

Outro objetivo específico é a definição de um mecanismo para a tomada de decisão, cujo objetivo era decidir qual dos agentes pedagógicos deveria interagir com o estudante. Neste caso, é necessário o desenvolvimento de um novo agente (facilitador) para realizar a comunicação com os agentes pedagógicos. Propor o desenvolvimento de uma heurística para essa tomada de decisão também faz parte deste objetivo.

Após a definição das estruturas, regras e algoritmos pertencentes ao mecanismo de comunicação, é necessária a integração de todas essas técnicas ao STI. Todos os conceitos são desenvolvidos na mesma linguagem de programação do sistema base.

Outro objetivo específico é preparar cenários de interação, com a finalidade de testar o funcionamento do mecanismo de comunicação desenvolvido. Para isso, 4 (quatro) questões sobre a interação dos estudantes com os agentes pedagógicos são definidas para serem respondidas pelos próprios estudantes que participaram do processo de avaliação. O assunto “Novo Acordo Ortográfico da Língua Portuguesa” é o conteúdo da base de conhecimento do STI usado nesta etapa.

Também é importante destacar que diversas outras implementações são realizadas com a finalidade de melhorar, corrigir e agregar funcionalidades ao sistema. A correção mais significativa é a reestruturação da implementação do Dimi (agente companheiro), que em alguns momentos parava ou não iniciava uma interação com o estudante, ficando inativo. As melhoras vão desde a identificação dos agentes (nome das caixas de diálogos) até problemas

de implementação (com caixas de erros genéricas na tela). Já como inovação pode-se citar a exibição do aproveitamento do estudante na execução dos exercícios, exibição do gabarito das respostas e a desenvolvimento do questionário.

Por fim, a organização do texto está dividida em capítulos que contemplam a evolução do trabalho. O capítulo 2 trata da fundamentação teórica envolvendo os estudos utilizados para o desenvolvimento do mecanismo de comunicação. O capítulo 3 traz a metodologia aplicada neste trabalho, assim como, apresenta as características e as funcionalidades implementadas, assim como a homologação das tecnologias desenvolvidas, resultados da avaliação realizada, outros desenvolvimentos realizados. No último capítulo são apresentadas as conclusões obtidas com o desenvolvimento da presente dissertação de mestrado e sugestão para trabalhos futuros.

## 2 Fundamentação teórica

A revisão de literatura abordada neste projeto de pesquisa está dividida em cinco tópicos principais. O primeiro compreende conceitos relacionados aos agentes inteligentes, cujo foco está direcionado para a área da educação. A seguir, o segundo tópico aborda a questão da comunicação entre agentes, apresentando, inclusive, algumas linguagens e protocolos de comunicação. O terceiro tópico fala sobre o STI usado como base para o projeto, enquanto o quarto tópico apresenta alguns trabalhos correlatos (relacionados). Por fim, o quinto e último tópico apresenta a proposta de desenvolvimento deste projeto de pesquisa.

### 2.1 Agentes

De acordo com Wooldridge e Jennings (1995), Maes (1996), Russell e Norvig (2004), Perucia et al. (2005) e Wooldridge (2009), um agente é um *software* computacional que, por sua vez, pode perceber o ambiente em que ele está inserido através de sensores (entrada). Além disso, esses agentes também podem tomar decisão e agir de forma independente (autônoma) no ambiente, executando uma ação de saída por meio de atuadores. A Figura 1 apresenta a interação do agente com o ambiente, onde o agente percebe as informações de entrada pelos sensores e executa uma ação de saída através dos atuadores.



Figura 1. Conceito de agentes – adaptado de Wooldridge (2009)

Segundo Russell e Norvig (2004), em um agente robótico, os sensores poderiam ser representados pelas câmeras de filmagem e detectores de faixa de infravermelho, enquanto os atuadores poderiam ser os motores e braços mecânicos. Para Torreão (2005), um agente poderia ser comparado a um ser humano, onde seus sensores seriam os olhos, ouvidos, tato e olfato. Desta forma, os atuadores seriam a boca, mãos e pernas.

Bradshaw (1997) descreve um agente como uma entidade de *software* que trabalha em um ambiente de forma contínua e independente, interagindo de forma flexível e inteligente com outros agentes, sem influência humana. Além de conseguir se comunicar e cooperar com outros agentes, um agente inteligente também deve aprender com base em suas experiências.

De acordo com Ferber (1999),

“...um agente é uma entidade real ou virtual, imersa num ambiente sobre o qual é capaz de agir, que dispõe de uma capacidade de percepção e de representação parcial deste ambiente, que pode se comunicar com outros agentes e que possui um comportamento autônomo, consequência de suas observações, de seu conhecimento e das suas interações com os outros agentes”.

Existem vários tipos de agentes que, por sua vez, apresentam diferentes características. Assim, o próximo tópico aborda a classificação de agentes conforme suas propriedades.

### **2.1.1 Classificação de agentes**

Segundo Ferber (1999), os agentes podem ser classificados de acordo com habilidades cognitivas. Portanto, os próximos tópicos apresentam as características que os agentes devem possuir para fazerem parte do grupo de agentes reativos e cognitivos. É importante destacar que agentes que possuam ambas as características podem ser classificados como híbridos.

#### **2.1.1.1 Agentes reativos**

Agentes do tipo reativo são classificados com base nos seus modelos de organização biológica ou etológica, como por exemplo, no comportamento de uma comunidade de cupins ou formigas. Uma formiga só não pode ser analisada como uma entidade inteligente. Por outro lado, a complexa organização de uma sociedade de formigas em um formigueiro pode ser considerada inteligente, pois esse comportamento surge da necessidade de atividades em grupo para a busca e estocagem de alimentos, além da manutenção dos berçários para reprodução (SICHMAN et al., 1992; COSTA e BITTENCOURT, 1997; BITTENCOURT, 1998).

Esses agentes também estão baseados no modelo de funcionamento estímulo-resposta (percepção e ação), no qual não traçam suas futuras ações e não fazem consulta a informações

históricas do passado, pois não possuem memória. Além disso, não se comunicam com outros agentes, ou seja, apenas tomam conhecimento das ações e comportamentos dos outros agentes através de alterações no ambiente. Deste modo, se dois ou mais eventos do mesmo tipo forem recebidos ou percebidos pelo agente reativo em momentos distintos, será executado um comportamento idêntico (BITTENCOURT, 1998; FERBER, 1999; WEISS, 1999).

### ***2.1.1.2 Agentes cognitivos***

Agentes do tipo cognitivo são baseados em organizações sociais dos seres humanos, como por exemplo, grupos, hierarquias e mercados (BITTENCOURT, 1998). Com isso, esses agentes adquirem seu conhecimento através da comunicação direta com outros agentes, assim como através de percepção das mudanças ocorridas no ambiente e raciocínio a respeito do comportamento dos demais agentes (SICHMAN et al., 1992; BITTENCOURT, 1998; FERBER, 1999).

Como esses agentes possuem memória, podem fazer consulta a dados históricos do passado. Portanto, os agentes considerados cognitivos utilizam suas habilidades de raciocínio (aprendizagem automática) para planejar de que maneira poderão conquistar seus objetivos, levando em consideração o conhecimento que adquiriram. Entretanto, isto pode resultar em mais de uma possibilidade, o que leva o agente a fazer uma escolha, possível graças a sua capacidade de tomar decisões (SICHMAN et al., 1992; BITTENCOURT, 1998; FERBER, 1999).

O número de agentes cognitivos em uma comunidade ou sociedade, em geral, pode ser considerado pequeno, ficando restrito há algumas dezenas. No entanto, esses agentes podem apresentar comportamento inteligente, estando em um grande grupo ou sozinhos (SICHMAN et al., 1992).

Além disso, agentes cognitivos precisam de mecanismos complexos de coordenação e protocolos de alto nível para interação/comunicação (FERBER, 1999). Esses agentes elevam a qualidade do sistema sob o ponto de vista pedagógico, pois permitem gerar um sistema mais perceptivo com autonomia, flexibilidade, colaboração e adaptação (FRIGO et al., 2004). Tais características são estudadas no próximo tópico.

### 2.1.2 Propriedades de agentes

De acordo com Wooldridge e Jennings (1995), um agente pode ser definido como um componente de *hardware* ou um *software* computacional com as seguintes propriedades ou características essenciais:

- **Autonomia ou independência:** os agentes trabalham sem a interferência externa (THIRY, 1999), ou seja, sem a influência humana ou de outros agentes. Além disso, essa característica permite que o agente possua algum tipo de controle sobre as suas ações e o estado interno (CASTELFRANCHI, 1995). Desta forma, a autonomia é uma propriedade que define o agente, onde sua falta caracteriza um *software* computacional qualquer, não um agente (FRANKLIN e GRAESSER, 1996). No entanto, Reis (2003) afirma que é comum o uso de agentes que se comportem de forma autônoma e também possam acatar ordens e instruções de humanos, quando isto for desejável.
- **Reatividade:** os agentes percebem seu ambiente e respondem às alterações ocorridas nele. Tais mudanças podem ocorrer por intervenção humana via interface gráfica, outro agente, a Internet, ou alguma coisa que provoque um estímulo no ambiente (WOOLDRIDGE e JENNINGS, 1995; THIRY, 1999; MARTINS, 2002). Para Garcia e Sichman (2003), a habilidade de reagir às alterações do ambiente ocorre a partir do reconhecimento de um contexto já conhecido. Portanto, a reatividade é mais uma característica necessária para que o sistema seja considerado um agente. Esta propriedade permite que o agente possa responder rapidamente às alterações no ambiente onde ele está inserido (FRANKLIN e GRAESSER, 1996).
- **Pró-atividade:** os agentes não precisam ficar limitados a tomar uma decisão apenas quando houver alguma mudança no ambiente (reatividade). Com isso, os agentes podem tomar a iniciativa e também podem exibir um comportamento direcionado por objetivos (WOOLDRIDGE e JENNINGS, 1995).
- **Comunicabilidade ou habilidade social:** é a habilidade do agente se comunicar por troca mensagens com outros agentes ou com outras entidades, como por exemplo,

humanos, objetos ou o próprio ambiente (GENESERETH e KETCHPEL, 1994; GARCIA e SICHTMAN, 2003). Essa comunicação tem como finalidade a conclusão de objetivos, sendo estes do próprio agente ou de uma determinada entidade. Neste caso, o agente pode ajudar a entidade a completar sua meta. É uma característica essencial quando há mais de um agente envolvido, ou quando é necessário o envio ou recebimento de uma informação (GARCIA e SICHTMAN, 2003).

Ainda sobre a característica de comunicação dos agentes, Thiry (1999) afirma que essa propriedade é considerada a mais importante sempre que houver mais de um agente envolvido na realização de uma determinada tarefa. Essa comunicação ocorre por troca de mensagens, através de uma linguagem de comunicação de alto nível (GENESERETH e KETCHPEL, 1994). Além disso, outras características são importantes para o sucesso dessa comunicação, como por exemplo, conhecimento, cooperação, negociação, coordenação, entre outros tipos de propriedade que exijam o uso de comunicação (MARTINS, 2002). O tópico 2.2 - Comunicação entre agentes aborda essa propriedade com mais detalhes.

O equilíbrio da habilidade social com as capacidades pró-ativas e reativas é muito importante, principalmente na realização de um trabalho cooperativo por uma sociedade de agentes que visam um mesmo objetivo. Com isso, cada um dos agentes precisa equilibrar a reação a mudanças no ambiente, com a decisão individual para realizar tarefas individuais, e com a habilidade social necessária para desempenhar tarefas em conjunto (REIS et al., 2001).

Além das características consideradas essenciais por Wooldridge e Jennings (1995), os agentes também podem apresentar outras características, tão importantes quanto às principais (na definição dos autores). Portanto, existem muitas propriedades que podem ser atribuídas aos agentes, dependendo do tipo da aplicação que se deseja desenvolver:

- **Mobilidade:** é a habilidade dos agentes se movimentarem de uma máquina para outra (FRANKLIN e GRAESSER, 1996), semelhante ao que ocorre no interior de uma rede de computadores (WHITE, 1994). Desta forma, os agentes podem ser transmitidos para outros lugares, para que possam executar suas tarefas e depois retornar. No entanto, isso gera mais complexidade, uma vez que há a necessidade de adaptação a outras plataformas e sistemas (MARTINS, 2002).



- **Coordenação:** as ações individuais de cada agente devem estar coordenadas para que um objetivo em comum possa ser atingido. Em muitos casos, apenas um agente não é suficiente para a realização de uma determinada atividade em um Sistema Multiagente (SMA), pois o agente pode não possuir conhecimento ou habilidade suficiente para isto. Entretanto, quando os agentes trabalham em conjunto, podem compartilhar conhecimento para a execução de uma atividade. Para facilitar futuras interações, cada agente pode armazenar o “modelo” de uma interação executada com sucesso, para que a mesma possa ser usada novamente (WEISS, 1999). De acordo com Weiss (1999), a coordenação é essencial em ambientes compartilhados e pode ser dividida em negociação e cooperação.
- **Negociação:** Weiss (1999) afirma que a negociação é uma coordenação entre agentes antagônicos ou “egoístas” (*self-interested*). Deste modo, a negociação está relacionada à coordenação de agentes que competem entre si, ou seja, podem possuir objetivos conflitantes. Usualmente, protocolos de negociação são utilizados para que os agentes possam negociar e chegar a um consenso. Para isto, algumas regras de negociação e atributos devem ser definidas e utilizadas pelos agentes durante esse processo de negociação (GREEN et al., 1997; WEISS, 1999).
- **Cooperação ou colaboração:** é a habilidade dos agentes trabalharem em conjunto para a conclusão de uma determinada atividade que, em muitos casos, podem possuir grande grau de complexidade. Desta forma, o sucesso na execução dessas atividades é benéfico para todos os agentes envolvidos. Para isto, o agente com essa característica precisa possuir um forte espírito de cooperação e colaboração com os demais agentes. A propriedade de cooperação ou colaboração é muito importante para uma sociedade de agentes, pois é essa característica que vai permitir o agente interagir com os demais agentes (WOOLDRIDGE e JENNINGS, 1995; DECKER e SYCARA 1997). Weiss (1999) destaca que a cooperação é uma coordenação entre agentes não antagônicos e não “egoístas”, pois não possuem objetivos que podem entrar em conflito.
- **Continuidade temporal ou persistência:** é a habilidade do agente manter um estado interno conciso através do tempo, ou seja, um processo executando de forma contínua,

sem que ele seja interrompido por uma casualidade (BELGRAVE, 1995; FRANKLIN e GRAESSER, 1996; MARTINS, 2002).

- Confiabilidade ou segurança: para Lee et al. (1998), um SMA pode ser considerado estável quando o mesmo volta ao equilíbrio de forma voluntária após um período de perturbação (turbulência), que pode ter sido causado por um agente inoperante ou outra falha qualquer.
- Verdade ou veracidade: os agentes sempre devem comunicar a informação verdadeira, nunca devem repassar uma informação falsa de propósito (GALLIERS, 1998).
- Benevolência: os agentes sempre devem tentar fazer aquilo que lhes é determinado, onde agentes não possuem objetivos em conflito (ROSENSCHEIN e GENESERETH, 1985).
- Planejamento: é a habilidade de um agente passar do modelo para a execução de uma ação, onde esse agente pode determinar qual é a melhor decisão a ser tomada naquele instante (AUER, 1995; BELGRAVE, 1995).
- Organização: é um conjunto de relações entre agentes, no qual produzem um sistema ou unidade complexa, com qualidades desconhecidas ao nível do usuário humano (WEISS, 1999).
- Sociabilidade: é a habilidade de o agente participar em múltiplas relações, onde ele pode interagir com vários outros agentes, ao mesmo tempo ou em tempos distintos (YU, 2001).
- Flexibilidade: é a habilidade dos agentes poderem escolher de forma dinâmica suas ações e a ordem de execução das mesmas (AUER, 1995; FRANKLIN e GRAESSER, 1996).

- Obrigação ou responsabilidade: os agentes podem expressar disponibilidade para realização de uma tarefa, entretanto, ao fazerem isto, eles estarão assumindo um compromisso e ficarão responsáveis por esta atividade até que ela seja executada (WOOLDRIGE e JENNINGS, 1995).
- Raciocínio: os agentes procuram agir de uma maneira que seus objetivos possam ser alcançados da melhor forma possível. Para isto, em cada momento, os agentes tomam suas decisões com base nas suas capacidades e conhecimentos (GALLIERS, 1998).
- Aprendizagem ou adaptatividade: é uma das características que define um agente inteligente (THIRY, 1999). Portanto, a aprendizagem também pode ser definida como a capacidade dos agentes modificarem ou adaptarem o seu comportamento baseado nas suas experiências e no resultado de ações realizadas no ambiente (LIU, 2001; SPERB, 2002).
- Inteligência: este tipo de capacidade pode ser conceituado como uma coleção de recursos, atributos e características que dão a possibilidade do agente escolher quais ações serão executadas. A inteligência está relacionada a muitas outras propriedades, como por exemplo, aprendizagem, raciocínio, comunicação, entre outras. De acordo com Gilbert e Manny (1996), a capacidade do agente de raciocinar é uma das características da inteligência que diferem agentes inteligentes de agentes comuns, considerados “robóticos”.

Contudo, quando mais um agente interage dentro de um mesmo ambiente significa que esse sistema é multiagente. Desta forma, o próximo tópico aborda a questão dos sistemas que possuem múltiplos agentes.

### **2.1.3 Sistemas multiagentes**

Um Sistema Multiagente (SMA) pode ser definido como um *software* computacional composto de dois ou mais agentes. Desta forma, essa sociedade de agentes deve interagir e/ou executar tarefas em conjunto para que seus objetivos possam ser atingidos (LESSER, 1999; WOOLDRIDGE, 2009). Portanto, os agentes precisam interagir uns com os outros, seja para

alcançar seus objetivos individuais ou para gerenciar dependências resultantes pelo fato dos agentes estarem inseridos em um ambiente comum (JENNINGS, 1993; CASTELFRANCHI, 1998).

Na maioria dos casos, os agentes devem agir para alcançar seus objetivos, sendo em nome de pessoas, empresas ou por parte de alguma iniciativa mais ampla de resolução de problemas. Assim, quando agentes interagem, há tipicamente algum contexto organizacional subjacente entre eles (GESSER, 1991; FERBER, 1999). Neste contexto é definida a natureza da relação entre os agentes (por exemplo, eles podem ser colegas que trabalham juntos em uma equipe ou um podem ser o gerente do outros agentes) e, conseqüentemente, influencia o seu comportamento (JENNINGS, 2000).

A estrutura de um SMA é ilustrada na Figura 2, onde o contexto organizacional (organização) é representado pelas elipses tracejadas que, por sua vez, envolvem os agentes pertencentes àquela organização. Ainda é possível observar a interação dos agentes, dentro e fora das organizações.

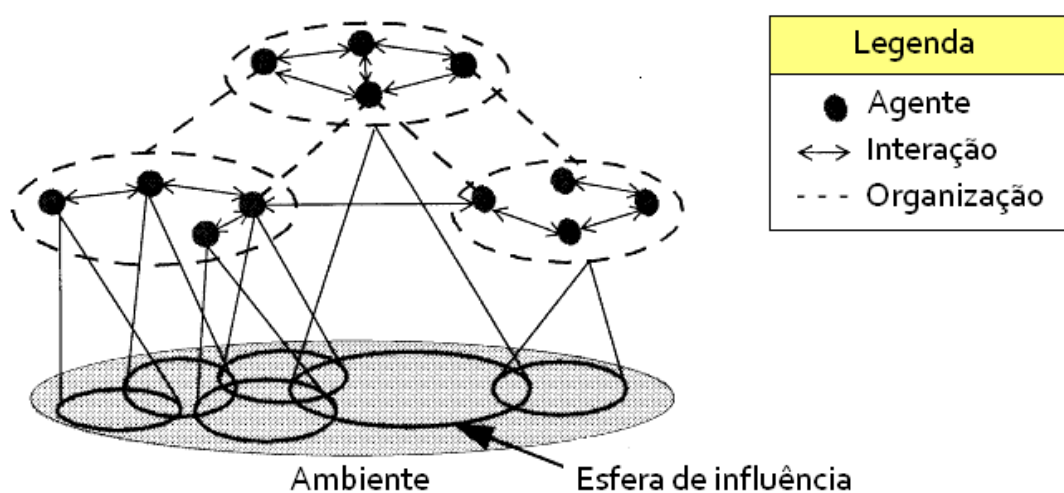


Figura 2. Estrutura de um sistema multiagente – adaptado de Jennings (2000)

Portanto, um SMA pode possuir múltiplos agentes, onde cada um pode ter diferentes capacidades de percepção e ação no mundo. Cada agente tem uma esfera de influência distinta sobre o ambiente, ou seja, cada agente possui a capacidade de influenciar diferentes partes do ambiente (JENNINGS, 2000).

Ainda com base na Figura 2, é possível observar que as esferas de influência podem coincidir dependendo das relações existentes entre os agentes. Portanto, Reis (2003) faz uso do seguinte exemplo para explicar esta situação:

“... na geração de horários de dois departamentos, os agentes responsáveis pela geração de cada horário terão como esfera de influência os horários dos docentes que lecionam no seu departamento, das turmas dos cursos desse departamento e das salas alocadas a esse departamento. Esses agentes irão interagir no caso de efetuarem alocações em docentes, turmas ou salas que lhes sejam comuns. Interações entre agentes que se encontram em um mesmo ambiente podem ocorrer de diversas formas e como tal interessa estudar os tipos e características de cada possível interação”.

O próximo tópico aborda um SMA específico para área da educação, onde o ambiente virtual é frequentado por alunos e estudantes.

#### **2.1.4 Ambientes virtuais de aprendizagem**

A tecnologia está em constante avanço e o seu desenvolvimento contribui para o surgimento e aperfeiçoamento de novas formas de ensino e aprendizagem. O intenso ritmo do mundo globalizado aliado à necessidade das pessoas estarem sempre se atualizando, faz com que muitas delas procurem outras formas de aprendizado (PEREIRA, 2007). Desta forma, os computadores oferecem diversos recursos e benefícios, para que as próprias pessoas possam construir o seu conhecimento, de forma autônoma e independente (MARTINS et al., 1999).

Segundo Behar et al. (2007), um Ambiente Virtual de Aprendizagem (AVA) “... é constituído por uma grande infraestrutura tecnológica e por todas as relações estabelecidas pelos sujeitos participantes, tendo como foco principal a aprendizagem”. No contexto dos ambientes de aprendizagem, Peters (2003) conceitua que:

“... os estudantes não devem ser objetos, mas, sim, sujeitos do processo de aprendizagem. Por isso devem ser criadas situações de ensino e aprendizagem nas quais eles mesmos possam organizar seu estudo (princípio do estudo autônomo). O próprio estudo não é iniciado e dirigido por eventos expositivos e receptivos ritualizados, mas, sim, por meio de discussão e interação (princípio do estudo por meio de comunicação e interação)”.

Já Harasim et al. (2005) aborda as redes de aprendizagem como grupos de pessoas se comunicam através do computador de forma a aprenderem juntos, o que considera mais

atraente e eficaz do que aprender sozinho, uma vez que cada pessoa possui o seu ritmo de aprendizagem, assim como, horário e local próprio para a realização das atividades. Portanto, Harasim et al. (2005) define “redes de aprendizagem” como:

“... uma palavra que descreve os espaços compartilhados formados por computadores interligados em todo o mundo por sinais de telefone e de satélite. Com o auxílio das redes, os educadores podem criar ambientes de aprendizagem eficazes, nos quais os professores e alunos em locais diferentes constroem juntos o entendimento e as competências relacionadas a um assunto particular”.

Em 2002, Moran já fazia uma espécie de “antecipação” ou “previsão” de como seriam as atividades de ensino e aprendizagem nos dias atuais. Portanto, segundo Moran (2002):

“Os alunos estão prontos para a Internet. Quando podem acessá-la vão longe. O professor vai percebendo que, aos poucos, a Internet está passando de uma palavra da moda à realidade em alguns colégios e nas suas famílias. Nestes próximos anos viveremos a interligação da Internet, com o cabo, com a televisão. Imagem, som, texto e dados se integrarão em um vasto conjunto de possibilidades. Ver-se e ouvir-se à distância se tornará corriqueiro. Pedir a um colega que dê aula comigo, mesmo que esteja em outra cidade ou país, ao vivo, será plenamente viável. As possibilidades da Internet no ensino estão apenas começando”.

Portanto, as redes virtuais podem juntar em um único lugar diversas oportunidades de aprendizado, como por exemplo, atividades interdisciplinares, tarefas *on-line*, conferências, projetos, cursos ou novas ideias (FERNANDES, 2007).

Neste contexto, pode-se destacar que uma das soluções mais utilizadas para que essas táticas de ensino-aprendizagem obtenham sucesso são os AVAs e, entre eles, é possível destacar os Sistemas Tutores Inteligentes. Esse tipo de sistema é abordado no próximo tópico.

### **2.1.5 Sistemas tutores inteligentes**

De modo geral, os chamados Sistemas Tutores Inteligentes (STI) podem ser definidos como sistemas educacionais que, por sua vez, fazem uso de técnicas de Inteligência Artificial (IA) para auxiliar os estudantes durante o seu processo de aprendizagem (GUARDIA, 1997; FREEMAN, 2000; GAMBOA e FRED, 2001; VILLAREAL e GIRAFFA, 2001; LIONAZ, 2001; GAVIDIA e ANDRADE, 2003).

Os STIs surgiram na década 70, quando Carbonell (1970) propôs uma nova forma de arquitetar os sistemas virtuais de educação (aprendizagem-computadorizada). Neste contexto, o autor resolveu considerar a dinâmica existente na relação estudante-professor. Deste modo, ele queria modelar em um sistema educacional algumas das situações possíveis em sala de aula, como por exemplo, o fato do professor poder observar o que está acontecendo durante o andamento das aulas e a forma com que o professor estrutura, organiza e passa a matéria.

Carbonell (1970) também propôs uma arquitetura que foi considerada clássica para os STIs. Esta arquitetura ficou conhecida como tradicional e “funcional tripartida”. O termo “tripartida” se refere às funções relacionadas aos módulos domínio, aluno e tutor. No entanto, a modelagem deste modelo não é uma tarefa trivial, pois possui três módulos fundamentais: aluno, tutor e domínio. A Figura 3 ilustra essa arquitetura clássica (GOULART e GIRAFFA, 2001).

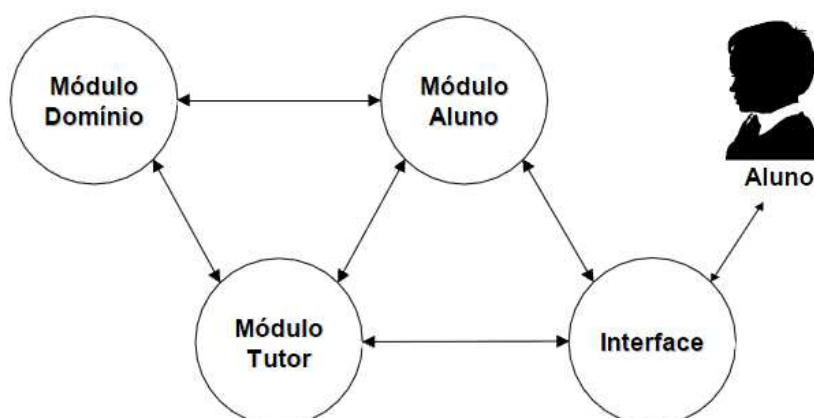


Figura 3. Arquitetura clássica de um STI - Goulart e Giraffa (2001)

Neste contexto, o módulo aluno armazena as características individuais do estudante. O módulo tutor guarda o conhecimento referente às estratégias e táticas que serão usadas de acordo com as características do aluno (representadas no módulo aluno). Já o módulo domínio possui o conhecimento sobre o conteúdo ou disciplina na forma de estereótipos, regras de produção, entre outros formatos. Por fim, a interface é responsável por permitir a interação entre o aluno e o tutor (CARBONELL, 1970).

A arquitetura proposta por Carbonell (1970) foi revisada por Self (1999) que, por sua vez, agregou à arquitetura clássica modelos de interação, de permissão e da situação. A Figura 4 ilustra a arquitetura de um STI proposta por Self (1999) (GOULART e GIRAFFA, 2001).

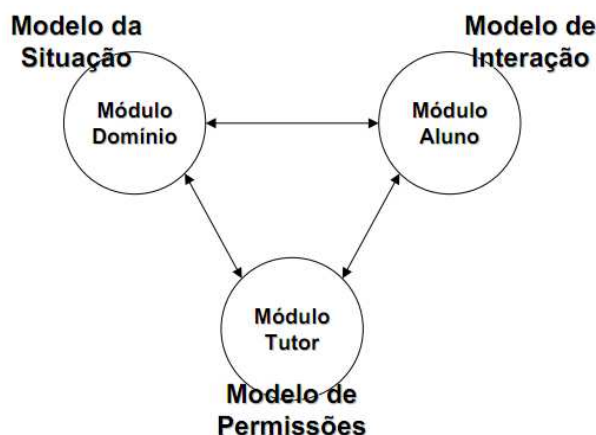


Figura 4. Arquitetura de um STI proposta por Self (1999) - Goulart e Giraffa (2001)

É importante destacar que essa nova arquitetura conceituada por Self (1999) tornou-se um grande avanço, pois conseguiu separar o domínio da sua forma de manipulação. Com isso, estratégias de ensino diferentes começaram a ser associadas conforme o modelo do estudante existente no STI. Desta forma, os modelos conceituados por Self (1999) são listados a seguir:

- **Modelo da situação:** está associado ao módulo domínio, que serve para modelar o conhecimento de acordo com o domínio (conteúdo) em estudo, permitindo o acesso ao conhecimento por parte do estudante no momento em que ele interage com o STI. Este módulo determina ainda, a complexidade e a maneira com que esse conteúdo pode ser apresentado ao estudante (SELF, 1999). No entanto, a melhor maneira de representar, armazenar e ter acesso ao conhecimento vai depender do domínio ensinado. Portanto, isso acaba reforçando a natureza interdisciplinar dos STI (SELF, 1993).
- **Modelo de interação:** está associado ao módulo aluno. Os tutores humanos fazem uso de vários recursos para conseguirem identificar o conhecimento e as necessidades dos seus alunos, como por exemplo, expressões de face, postura do corpo, tom da voz e características afetivas. Isso apenas é possível através da experiência de ensino do professor. Desta forma, fica mais fácil do tutor fornecer auxílio e ensino individual para o estudante (WOOLF, 2007). Neste contexto, os STIs precisam fazer inferência sobre o conhecimento do estudante, armazenando no módulo aluno as informações obtidas através das ações e interações dos estudantes. Com isso, é possível adaptar o processo de ensino-aprendizagem deste estudante, com o propósito de torná-lo mais efetivo (SELF, 1999; WOOLF, 2007).



- Modelo de permissões: está associado ao módulo tutor, que é fundamental e determina quando deve intervir no aprendizado, através de um conjunto de estratégias de ensino contidas no tutor (SELF, 1999). Contudo, os módulos domínio e aluno dependem do módulo tutor para conseguirem adaptar a resposta deste tutor a um estudante em particular (WOOLF, 2007). Portanto, de uma maneira mais ampla, o módulo tutor é o responsável por conduzir o estudante de acordo com os objetivos e desafios que o ambiente proporciona a ele (SELF, 1999).

De acordo com Silva (2006), os STIs também podem ser usados em conjunto com as aulas tradicionais, através da Internet, com o objetivo de:

“... aumentar o nível de interação do aluno em cursos a distância, à medida que implementam estratégias para supervisionar as suas ações e propor alternativas dinamicamente, de acordo com os princípios e a proposta pedagógica do curso em questão, em uma atuação que não sobrepõe a atuação do tutor, mas que visa apenas agregar valor e qualidade às interações em um ambiente virtual de aprendizagem”.

Segundo Loinaz (2001), para que um STI seja desenvolvido com sucesso, é necessário que técnicas da área de ciência da computação sejam usadas, como por exemplo, conceitos da IA. Além disso, o conteúdo a ser apresentado para o estudante deve ser exposto de maneira que possa contribuir para sua motivação em frente ao computador. Por fim, parâmetros de educação ou de ciência da educação que possuem o objetivo de reiterar e sustentar o processo de ensino e aprendizagem mediante as novas tecnologias também devem ser considerados.

Com a evolução no desenvolvimento dos ambientes virtuais voltados para educação, o uso de Agentes Pedagógicos e Agentes de Interface podem contribuir para a aprendizagem do estudante. Os próximos tópicos abordam as características destes tipos de agentes inteligentes.

### **2.1.6 Agentes pedagógicos**

Os agentes são considerados pedagógicos quando estão inseridos em sistemas que fazem uso do paradigma de agentes desenvolvidos com a finalidade de educar, possuindo como objetivo fundamental auxiliar os estudantes em seu processo de ensino-aprendizagem (GÜRER, 1998; GIRAFFA, 1999; KAMPFF et al., 2005). Desta forma, os agentes pedagógicos provêm melhor interação e dinâmica aos ambientes educacionais (GIRAFFA,

1999), além de facilitar e/ou melhorar a aprendizagem dos estudantes (GÜRER, 1998). Já Kampff et al. (2005) destaca que esse tipo de agente se caracteriza por apresentar conteúdos e estratégias mais adaptadas ao perfil do estudante.

De acordo com Pereira (1997), integrar esses agentes pedagógicos dentro de sistemas educacionais proporciona grande melhoria nos aspectos pedagógicos destes ambientes, pois possibilitam uma interação mais intensa e natural, graças à aproximação entre estudantes e sistema. Para Moran (1998), promover o conhecimento é o mesmo que compreender todas as dimensões da realidade, fazendo com que seja possível captar e expressar essa realidade por completo.

Para Maes (1994), “personagens virtuais” podem contribuir para o aperfeiçoamento do processo de interação entre usuários humanos e o computador, caso estes sejam introduzidos nos ambientes virtuais. A autora relaciona o uso desses personagens ao ambiente de trabalho de uma pessoa comum, onde essa pessoa também recebe a ajuda de um agente pessoal (chefe ou colega). Esses personagens virtuais seriam os chamados agentes de interface, abordados no próximo tópico.

### **2.1.7 Agentes de Interface**

Os Agentes de Interface são conhecidos por darem suporte e assistência aos usuários de um determinado sistema. Além disso, esse tipo de agente também é capaz de “aprender”, observando e monitorando as ações do usuário dentro do ambiente. Desta forma, esses agentes possuem a capacidade de indicar maneiras mais competentes do usuário realizar uma determinada atividade (LAUREL, 1997). Sendo assim, o agente de interface pode agir como se fosse uma espécie de “assistente pessoal autônomo”, onde o próprio agente define a melhor maneira de cooperar com o usuário na realização uma determinada atividade (LENOX et al., 2001).

Desta maneira, Nwana (1996) compreende que a autonomia e a aprendizagem são duas das características mais marcantes deste tipo de agente. Inclusive, Maes (1996) conceitua um agente de interface como um agente de comportamento autônomo, responsável por gerenciar a interação entre um usuário e o computador. Com isso, a autora destaca que os

agentes de interface se diferenciam de uma interface tradicional justamente por terem a capacidade de alterar seu comportamento e ações de forma autônoma, com base no que esse agente aprendeu e através do comportamento do usuário durante o processo de interação com o ambiente.

Portanto, uma ideia interessante seria desenvolver esses agentes de interface dentro de dos sistemas de educação, com o principal objetivo de auxiliar o estudante no seu processo de aprendizado. Com isso, esses agentes poderiam assumir a figura do professor, ou então, a do companheiro de estudo do estudante (MARSELLA et al., 2000; HAYES-ROTH, 2001; RANK, 2005).

Segundo Nunes (2002), um sistema eficiente do ponto de vista pedagógico deve saber negociar com o estudante as decisões a serem tomadas, ou seja, não deve simplesmente impor essas decisões ao estudante. Portanto, os chamados Agentes Pedagógicos Animados podem ser usados para estruturar um sistema que possa decidir o que ensinar ao estudante (NUNES, 2002). O próximo tópico aborda as características desse tipo de agente pedagógico.

### **2.1.8 Agentes pedagógicos animados**

Nunes (2002) diz que os agentes pedagógicos animados são “personagens vivos” que habitam um ambiente de estudo, onde eles se comunicam, motivam e prendem a atenção do estudante. Portanto, a inclusão de “personagens virtuais” em AVAs é um fato importante a ser analisado, pois do ponto de vista do estudante, essa ação pode ser benéfica para as suas experiências de aprendizagem (LESTER et al., 1997). Contudo, a “vida” dos agentes contribui para o processo de imaginação dos estudantes, o que pode motivar e fazer com que estes estudantes fiquem animados na realização das tarefas propostas (TOWNS et al., 1998).

Neste contexto, espera-se que os agentes pedagógicos animados possam, a qualquer momento, auxiliar de forma individual todos os estudantes inseridos no ambiente. Com isso, é possível que cada estudante possa aprender no seu ritmo de aprendizagem, o que pode trazer os seguintes benefícios (LESTER et al., 1997):

- O estudante pode ter a impressão que o agente pedagógico animado está trabalhando junto com ele, ou seja, realmente está interessado e preocupado com o seu progresso de aprendizagem, o que pode motivar o estudante a continuar se esforçando.
- O agente pedagógico pode ser sensível ao progresso de aprendizagem do estudante, ou seja, o agente pode tentar recuperar o interesse do estudante antes que ele perca o foco da atividade que está realizando. Isto pode ser importante sempre que o estudante fica frustrado na realização de alguma tarefa.
- Para que o estudante não fique desanimado ou perca o ânimo na realização de alguma atividade, o agente pedagógico animado pode interferir e demonstrar entusiasmo na execução da tarefa em questão. Com isso, o estudante pode se sentir contagiado com o ânimo do agente e continuar a realização da atividade.
- Um agente pedagógico animado pode fazer com que o progresso de aprendizagem do estudante seja mais alegre e divertido. Para isto, é necessário que o agente possua uma personalidade rica e repleta de detalhes, o que pode fazer com que o estudante fique mais animado e interessado ao interagir com o agente. Conseqüentemente, isso pode fazer com que este estudante fique mais tempo no AVA.

Jaques (2004) afirma que os agentes pedagógicos animados são usados em AVAs, principalmente por possuírem outras formas de interagir com o estudante, como por exemplo, por meio de gestos e expressões de face (antropomorfismo). Segundo Vygotsky (1962), Izard (1984) e Goleman (1995), a motivação e a afetividade são duas características fundamentais que devem ser consideradas no processo de interação do usuário humano com o ambiente, pois contribuem para o progresso de aprendizagem do estudante.

Deste modo, os agentes pedagógicos animados também podem expressar emoção que, por sua vez, é uma importante característica da área da “computação afetiva”. Esta expressão foi criada por Picard (1997), com a finalidade de evoluir a computação tradicional (lógica) para uma computação que leve em consideração as emoções. Portanto, Picard (1997) concluiu e demonstrou que, se os computadores possuem a capacidade de reconhecer, expressar e até

possuir emoções, além de serem inteligentes, eles também podem se tornar mais humanos. É neste sentido que a afetividade pode contribuir para o processo de aprendizagem do estudante.

Entretanto, em uma sociedade de agentes, cada indivíduo possui suas características, com diferentes metas e visões sobre uma determinada situação. Deste modo, alguns conflitos podem ocorrer, no entanto, esses problemas devem ser resolvidos entre os próprios agentes (negociação). É importante destacar que, embora os agentes possuam metas diferentes, há um plano em comum, onde todos devem estar engajados na sua conclusão (TORSUN, 1995).

Portanto, em um SMA, os agentes precisam se comunicar para que consigam concluir seus objetivos, assim como, os objetivos do sistema ou sociedade no qual estão inseridos (SCHWAMBACH, 2004). Com isso, o próximo tópico aborda a questão da comunicação entre agentes.

## **2.2 Comunicação entre agentes**

Por possuírem um alto nível de abstração, é cada vez mais comum o uso de técnicas multiagentes para o desenvolvimento de sistemas considerados complexos (JENNINGS, 1999; 2001; ALMEIDA et al., 2004; ZAMBONELLI e PARUNAK, 2004), como os STIs. Contudo, o sucesso na execução de cada uma das tarefas de um SMA depende do poder de comunicação de cada agente (JENNINGS, 1996).

Portanto, uma das mais importantes etapas no desenvolvimento de um SMA é a maneira com que os agentes interagem entre si e as regras que envolvem essa interação, denominadas protocolos de comunicação (interação). Além disso, para que essa comunicação seja possível, faz-se necessário o uso de uma linguagem que ambos os agentes entendam. Normalmente, essas interações ocorrem com base em linguagens de alto nível, conhecidas como ACL (*Agent Communication Language* - Linguagem de Comunicação entre Agentes) (GLUZ e VICARI, 2003).

Com base nisso, o próximo tópico apresenta mais conceitos referentes às ACLs, assim como aborda algumas ACLs existentes.

### 2.2.1 Linguagem de comunicação entre agentes (ACL)

Segundo Vicari et al. (2009), a ACL pode ser definida como um formalismo, com a finalidade de codificar as mensagens trocadas entre os agentes, independente da linguagem de programação usada na aplicação (VICARI et al, 2009).

Fonseca (2000) afirma que a troca de informações entre agentes é indispensável para que os objetivos de um SMA possam ser atingidos. Neste contexto, é fundamental que os agentes sejam capazes de se comunicar com os demais, sem que isso resulte em um custo computacional muito grande. No geral, as ACLs apresentam uma notação formal, com sintaxe e semântica reduzida, além de serem bem definidas (FONSECA, 2000).

A maior parte das ACLs foi criada com base na Teoria dos Atos da Fala (*Speech Acts*), escrita em 1969 por Searle. Nesta teoria, o foco de estudo é a linguagem como veículo de comunicação, ou seja, a linguagem utilizada durante a fala (conversa) para a troca de informações (conhecimento). Portanto, essa linguagem foi criada para entender e descrever como as pessoas usam a linguagem diariamente (SEARLE, 1979; 1981).

Neste contexto, os indivíduos de uma sociedade de agentes “falam” entre si para que seus conhecimentos, planos e desejos possam ser compartilhados com os demais integrantes desta comunidade. Entretanto, a escolha de um protocolo de comunicação de alto nível para ser usado em um SMA não é uma tarefa trivial (FONSECA, 2000). Com isso, alguns autores preferem desenvolver seus próprios protocolos (RABELO, 1997; FONSECA, 2000), justamente pelas complicações de se implantar uma comunicação já padronizada e conhecida.

Russell e Norvig (1995) afirmam que comunicação é a “*troca intencional de informação através da produção e percepção de sinais de acordo com um sistema convencionado de símbolos*”. No entanto, Genesereth e Ketshpel (1994) usam as ACLs para definir um agente, onde uma entidade pode ser considerada um agente apenas se ela se comunica com outra através de uma ACL. Assim, o próximo tópico abordará conceitos e características de duas das ACLs mais conhecidas.

### 2.2.1.1 KQML

Inicialmente, a linguagem KQML (*Knowledge Query and Manipulation Language* - Linguagem de consulta e manipulação de conhecimento) foi desenvolvida pelo grupo KSE<sup>1</sup> que, por sua vez, recebeu patrocínio de importantes órgãos do governo norte americano, como por exemplo, DARPA<sup>2</sup>, ASOFR<sup>3</sup>, CNRI<sup>4</sup> e NSF<sup>5</sup> (PATIL et al, 1992).

Essa linguagem foi projetada com o objetivo de permitir a interação entre agentes e a interoperabilidade entre os sistemas (FININ et al., 1992; 1993; 1997), sendo necessário para isto o estabelecimento de padrões de comunicação para um compartilhamento padronizado de conhecimento (PATIL et al, 1992). Desta forma, a troca de informações ocorre por meio de uma comunicação de alto nível (GENESERETH e KETCHPEL, 1994; HÜBNER e SICHMAN, 2003).

De acordo com Finin et al. (1994b), KQML nasceu com a finalidade de comportar uma grande diversidade de arquiteturas de agentes, implementados em diferentes linguagens de programação e paradigmas. Assim, KQML é uma linguagem de comunicação orientada a mensagens e um protocolo para troca de informações de alto nível. Neste contexto, pode-se dizer que essa linguagem não necessita de mecanismos de transporte (TCP/IP, SMTP), não precisa se preocupar como o conteúdo da linguagem é representado (KIF, SQL, Prolog), e não depende da ontologia assumida pelo conteúdo, ou seja, qualquer tipo de dado pode ser enviado em uma KQML (FININ et al., 1992; 1994b).

Como já mencionado, um agente pode transmitir uma mensagem KQML sem a necessidade de especificar a linguagem utilizada (de livre escolhida), pois o conteúdo é “encapsulado” na própria sentença a ser enviada (BENECH et al., 1997). Benech et al. (1997) ainda destaca que, além de transmitir uma informação, as mensagens KQML também podem enviar ações, como por exemplo, uma afirmação, requisição, pergunta ou resposta.

---

<sup>1</sup> KSE: Knowledge Sharing Effort – Esforço de compartilhamento de conhecimento

<sup>2</sup> DARPA: *Defense Advanced Research Projects Agency*- Agência de Projetos de Pesquisa Avançada de Defesa.

<sup>3</sup> ASOFR: *Air Force Office of Scientific Research*-Escritório de Pesquisa Científica da Força Aérea.

<sup>4</sup> NRI: *Corporation for National Research Initiative*-Corporação para Iniciativas de Pesquisa Nacional.

<sup>5</sup> NSF: *National Science Foundation* - Fundação Nacional da Ciência.

Neste contexto, as primitivas de uma mensagem KQML são geralmente chamadas de “performativas” (*performatives*), onde são determinadas as ações ou os tipos de comunicação, assim como os parâmetros das mensagens. Portanto, a performativa estabelece o significado da mensagem e a forma como o seu conteúdo será interpretado (BENECH et al., 1997).

Conceitualmente, a linguagem KQML pode ser dividida em três camadas: conteúdo, mensagem e comunicação. Essa arquitetura de camadas pode ser observada na ilustração da Figura 5 (FININ et al., 1994a).

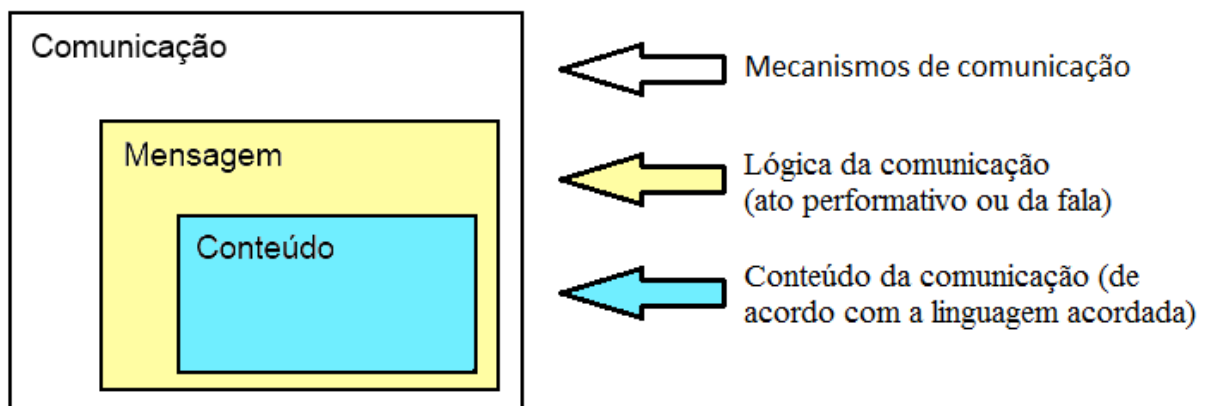


Figura 5. Camadas da linguagem KQML – adaptado de Finin et al. (1994a)

Ainda com base na arquitetura de três camadas da linguagem KQML pode-se destacar os seguintes pontos (FININ et al., 1994a):

- Camada de conteúdo: é a camada mais interna da arquitetura e armazena a informação real da mensagem, em qualquer linguagem de conteúdo, como por exemplo: KIF ou Prolog. No entanto, para que dois ou mais agentes possam se comunicar é necessário que ambos concordem em utilizar a mesma linguagem de conteúdo.
- Camada de mensagem: é a camada de que constitui o núcleo da linguagem. É ela que determina os tipos de interações que os agentes podem ter com uma KQML. A principal função desta camada é identificar o protocolo a ser usado para transmitir a mensagem e a performativa (ou ato da fala) que o transmissor anexa ao conteúdo. A performativa significa que o conteúdo pode ser uma afirmação, uma consulta, um comando, uma requisição, ou outra performativa conhecida.



- Camada de comunicação: é a camada mais externa da arquitetura e é através dela que os agentes trocam pacotes. Portanto, essa camada é responsável por codificar uma série de características para aceitar parâmetros de baixo nível, como por exemplo, a identificação do agente emissor e receptor da mensagem, assim como uma identificação única para a comunicação.

De acordo com Faraco e Gauthier (1998), todas as performativas da KQML estão relacionadas às bases de conhecimento dos agentes, onde geralmente as trocas de mensagem estão vinculadas às metas de cada agente. No entanto, essa base de conhecimento pode ser virtual (VKB - *Virtual Knowledge Base*), ou seja, não precisa ter a mesma estrutura que as bases reais. Neste contexto, cada agente fica responsável pela sua própria VKB. Wooldridge (2009) considera essa questão essencial para o entendimento das performativas.

Entretanto, esta seção apenas apresenta um estudo sobre a linguagem de comunicação KQML. Portanto, a linguagem de comunicação FIPA-ACL, descrita no próximo tópico, foi selecionada para ser usada neste trabalho.

### **2.2.1.2 FIPA-ACL**

A Fundação para Agentes Físicos Inteligentes (FIPA - *Foundation for Intelligent Physical Agents*) é responsável por criar padrões para a implementação da comunicação entre agentes. Ou seja, FIPA é uma organização sem fins lucrativos com o objetivo de garantir a interoperabilidade entre agentes heterogêneos (FIPA, 2013). Geralmente uma ACL é padronizada e disponibilizada para que qualquer SMA possa usá-la (VICARI et al., 2009).

A estrutura das mensagens da linguagem de comunicação FIPA-ACL (FIPA - *Agent Communication Language*) é bastante semelhante à KQML. Cada mensagem é formada por um identificador obrigatório, que especifica o tipo de ato de comunicativo. Em seguida vem uma série de *slots* opcionais. Geralmente as mensagens também possuem um emissor, um receptor e um campo de conteúdo. Os *slots* podem ser atribuídos em qualquer posição dentro da mensagem, sendo ainda possível a inserção de novos *slots* às mensagens, caso o prefixo “x-” seja utilizado (FIPA, 2002f).

Segundo a especificação da FIPA (2002f), os parâmetros nativos da FIPA-ACL são divididos em: tipo do ato comunicativo, participantes da comunicação, conteúdo de mensagem, descrição do conteúdo e controle de conversação. A lista completa dos parâmetros e categorias são apresentadas na Tabela 1 (GLUZ, 2005).

Tabela 1 – Parâmetros das mensagens FIPA-ACL

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

Fonte: Retirado de Gluz (2005) - Adaptado de FIPA (2002f)

Na medida do possível, os atos comunicativos devem representar os atos da fala definidos na Teoria dos Atos da Fala de Searle (1979; 1981). A Tabela 2 apresenta os atos comunicativos padrões da linguagem FIPA-ACL (FIPA, 2002g; GLUZ, 2005).

Tabela 2 – Atos comunicativos FIPA-ACL

Parâmetro	Descrição
<i>Accept-Proposal</i>	Informa a aceitação de uma proposta prévia para a execução de uma determinada ação. O conteúdo da mensagem deve conter a ação a ser feita e a condição aceita.

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

Parâmetro	Descrição
<i>Reject-Proposal</i>	Representa a ação de rejeitar a execução de alguma ação, durante uma negociação. O conteúdo é composto da ação rejeitada e da explicação do por que da rejeição.
<i>Request</i>	O agente emissor solicita ao receptor que ele execute alguma ação (possivelmente outro ato comunicativo). O conteúdo da mensagem é uma ação a ser feita.
<i>Request-When</i>	O agente emissor solicita ao receptor que ele execute alguma ação, quando uma dada proposição for verdadeira. O conteúdo da mensagem é composto da ação a ser feita e da proposição.
<i>Request-Whenever</i>	O agente emissor solicita ao receptor que ele execute alguma ação, assim que uma dada proposição for verdadeira e que a continue executando cada vez que ela se tornar verdadeira novamente. O conteúdo da mensagem é composto da ação a ser feita e da proposição.
<i>Subscribe</i>	Solicita a notificação do valor das atualizações no valor de uma dada referência. O conteúdo da mensagem é composto de uma expressão referencial (uma descrição do valor a ser notificado).

Fonte: Retirado de Gluz (2005) - Adaptado de FIPA (2002f)

Para melhor compreensão dos atos comunicativos, Gluz (2005) apresenta um exemplo de uso da linguagem de comunicação FIPA-ACL com a linguagem de conteúdo SL (*Semantic Language* – Linguagem semântica), onde o agente i cliente quer saber quais serviços de reserva de viagens um agente j assistente de viagens pode disponibilizar. O Quadro 1 lista a estrutura desta mensagem.

Quadro 1 – Exemplo 01: Mensagem FIPA-ACL

```
(query-ref
  :sender (agent-identifier :name i)
  :receiver (agent-identifier :name j)
  :content ((all ?x (available-service j ?x)))
  :language FIPA-SL
  :ontology travel-reservations-ontology)
```

Fonte: Gluz (2005).

No mesmo contexto do exemplo anterior, o agente j responde que faz reservas de passagens para viagens de trem e avião e pode também reservar (alugar) automóveis, assim como pode ser visto no Quadro 2 que, por sua vez, apresenta uma mensagem em resposta à mensagem anterior.

Quadro 2 – Exemplo 02: Mensagem FIPA-ACL

```
(inform:sender (agent-identifier :name j)
  :receiver (agent-identifier :name i)
  :content ((= (all ?x (available-service j ?x))
    (set (reserve-ticket train)
      (reserve-ticket plane)
      (reserve automobile))))
  :language FIPA-SL
  :ontology travel-reservations-ontology)
```

Fonte: Gluz (2005).

O próximo tópico aborda a comunicação entre os agentes para o padrão FIPA.

### 2.2.2 Protocolo de comunicação entre agentes

Segundo Odell et al. (2000), “... *um protocolo de comunicação entre agentes descreve um padrão de comunicação através de uma sequência válida de mensagens entre os agentes* ...”. O padrão FIPA possui um conjunto de protocolos que definem diversos modos possíveis de interação entre agentes através de troca de mensagens ACL (FIPA-ACL) (FIPA, 2013).

Na descrição de cada um dos protocolos, o agente *iniciador* inicia a comunicação (interação). Já o agente *participante* é a quem o agente *iniciador* se dirige. Em alguns casos, o *iniciador* é o agente que precisa de um serviço e o *participante* é quem fornece o serviço. Em outras situações pode haver mais *participantes*. Cada tipo de protocolo é identificado através do *protocolo* de uma mensagem ACL. Cada interação utilizando um protocolo é identificada através de um parâmetro de mensagem ACL único, não nulo e global, chamado *identificador da conversação*. Este parâmetro é definido pelo *iniciador* e é utilizado em todas as mensagens trocadas durante a comunicação (FIPA, 2013; CARVALHO, 2008).

Nos próximos tópicos, os protocolos são ilustrados através de um tipo de diagrama, similar com um diagrama de sequência UML, proposto em (ODELL et al, 2001).

### 2.2.3 FIPA Request Interaction Protocol (FIPA, 2002a)

Neste protocolo de interação, o *iniciador* precisa de um serviço e sabe a qual agente solicitar. A Figura 6 ilustra o comportamento deste protocolo, onde tanto o *iniciador* quanto o *participante* conhecem todas as pré-condições para a prestação do serviço. Por isso, nesta situação não há negociação, há simplesmente a *solicitação* seguida da prestação do serviço. Entretanto, este protocolo está longe de ser semelhante a um modelo cliente-servidor, pois a prestação do serviço pode não ocorrer devido ao *participante* (GLUZ, 2005; CARVALHO, 2008):

- Simplesmente ignorar a solicitação, já que é autônomo como todo agente.
- Não entender o que foi solicitado, podendo pedir que o *iniciador* repita a solicitação.
- Recusar-se a prestar o serviço, informando o motivo da recusa.

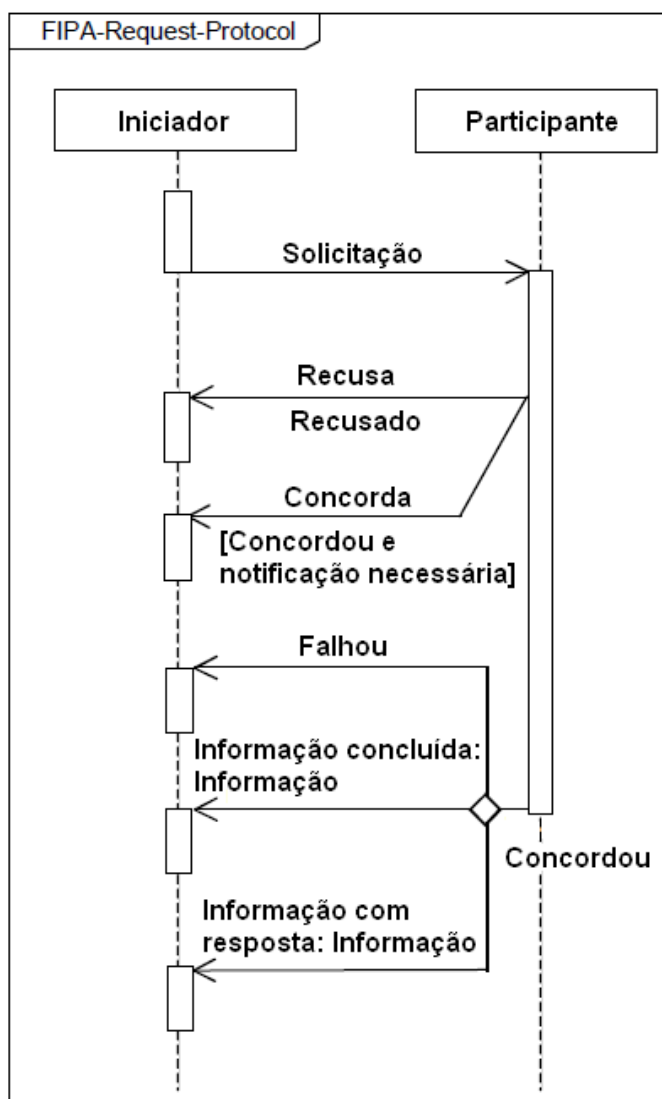


Figura 6. Protocolo de interação *Request* – adaptado de FIPA (2002a)

Ainda assim, o *participante* pode ter a intenção de prestar o serviço, mas não obter êxito devido a uma falha. Neste caso ele informa isto ao iniciador. Caso o *participante* aceite prestar o serviço, ele envia uma mensagem indicando a sua concordância, no entanto esta mensagem é opcional (GLUZ, 2005; CARVALHO, 2008).

Quando tudo ocorre normalmente e o serviço é prestado, duas respostas são possíveis por parte do *participante*: uma simples mensagem de confirmação, dizendo que o serviço foi realizado como esperado (*Informação concluída*), ou uma mensagem semelhante à anterior seguida de algum resultado (*Informação com resposta*) (GLUZ, 2005; CARVALHO, 2008).

#### **2.2.4 FIPA Contract Net Interaction Protocol (FIPA, 2002b)**

Neste protocolo de interação, o *iniciador* necessita de um serviço, sabe o que deseja, mas não sabe exatamente quem lhe prestará o serviço. Além disso, ele tem consciência de que pode obter o melhor serviço, se pesquisar e negociar. O *iniciador* envia uma mensagem a um determinado número de agentes *participantes* solicitando que enviem suas propostas de prestação de serviço. Esta mensagem inicial, chamada de CFP (*call for proposal* – chamada para proposta), contém todas as informações do serviço que o agente cliente deseja obter (CARVALHO, 2008).

Eventualmente um ou mais agentes *participantes* que receberam a mensagem CFP não a respondem. Isto ocorre porque tal agente simplesmente ignorou a mensagem, ou então porque não houve tempo para que a respondesse antes do tempo hábil definido pelo agente *iniciador* para o envio das respostas (*deadline*).

Uma quantidade  $j$  de agentes *participantes*, que receberam a mensagem CFP e se “candidataram” a prover o serviço, enviam suas propostas ao iniciador. Este valor é igual ao total de agentes que responderam à mensagem CFP,  $n$ , subtraído pelo total de agentes que enviaram como resposta a recusa de uma proposta. Esta quantidade é representada pela letra  $i$  (CARVALHO, 2008).

O *iniciador*, após receber propostas, escolhe a melhor e informa aos agentes candidatos a sua decisão, enviando ao agente cuja proposta foi aceita, uma mensagem de

*proposta aceita*, ou *proposta rejeitada* aos agentes cujas propostas não foram aceitas. É possível também que o *iniciador* deseje utilizar mais de um serviço, assim ele envia mais de uma mensagem de *proposta aceita* (CARVALHO, 2008). A Figura 7 apresenta o fluxo do protocolo de interação *ContractNet*.

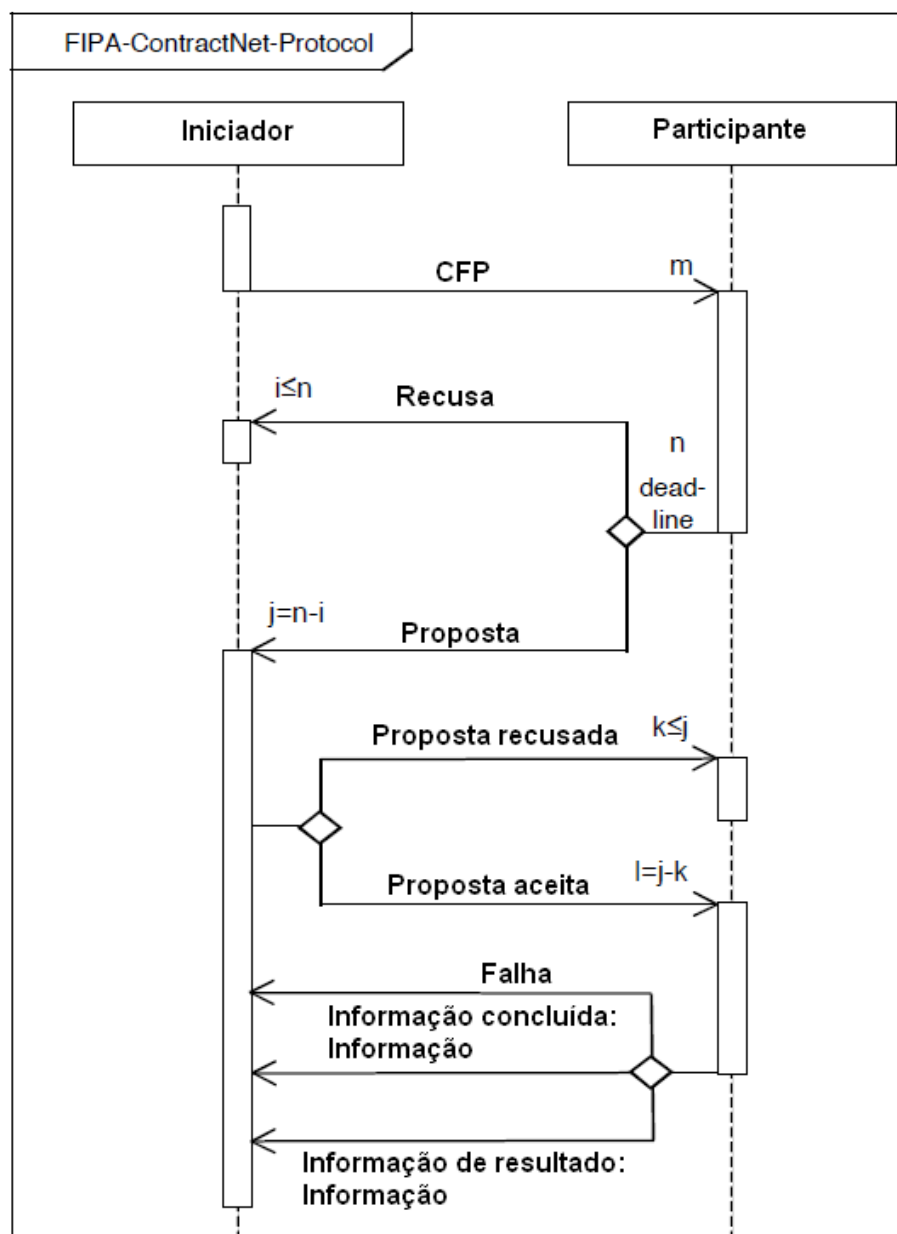


Figura 7. Protocolo de interação ContractNet – adaptado de FIPA (2002b)

Por fim, a negociação se encerra quando o *participante* que proverá o serviço recebe uma mensagem *proposta aceita*. Tal agente, então, realiza o que foi acordado e envia uma mensagem de confirmação, dizendo que o serviço foi realizado como esperado (*informação concluída*), ou uma mensagem semelhante à anterior seguida de algum resultado (*informação*



*com resultado*). Caso o agente provedor do serviço não consiga realizar sua tarefa, então ele envia uma mensagem de *falha* contendo o motivo de não ter obtido êxito (CARVALHO, 2008).

### 2.2.5 FIPA Propose Protocol (FIPA 2002c)

Através deste protocolo, o *iniciador* envia uma mensagem contendo uma proposta a outro agente, o *participante*. Esta proposta normalmente solicita o consentimento do agente *participante* para que uma determinada tarefa seja executada. Se este concordar, através de uma mensagem de *resposta aceita*, o que foi proposto será feito. A proposta pode ser rejeitada através de uma mensagem de *resposta rejeitada* (CARVALHO, 2008).

Este protocolo é semelhante ao *Contract Net*, no entanto, o serviço é realizado sem que haja a solicitação de um agente através de um CFP. Além disso, o serviço executado não precisa ter efeito direto nos desejos ou necessidades do agente “cliente”, caracterizando, portanto, um pedido de permissão para a execução de serviço, e não uma oferta de serviço. Um efeito disso é que, diferentemente dos outros protocolos, o agente iniciador é justamente o que executa o serviço, e não o *participante*. A Figura 8 apresenta o funcionamento deste protocolo (CARVALHO, 2008).

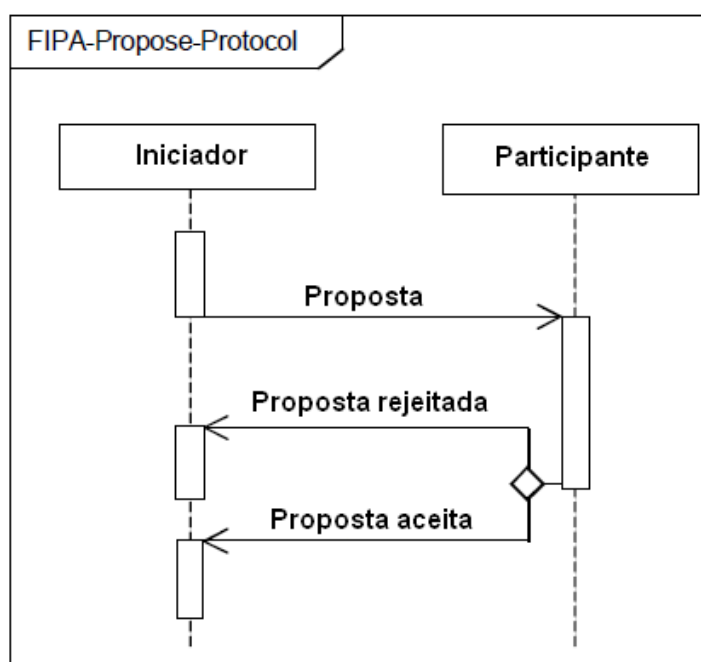


Figura 8. Protocolo de interação *Propose* – adaptado de FIPA (2002c)

### 2.2.6 FIPA *Brokering Protocol* (FIPA, 2002d)

Este protocolo conta com a participação de um agente *broker*, onde esse termo refere-se a um componente de *software* responsável por mediar o relacionamento entre um cliente e um servidor, ou entre um repositório e um elemento que solicita acesso a tal repositório. Esta palavra não foi traduzida em razão de não haver um termo correspondente em português que confira o mesmo significado (CARVALHO, 2008).

O agente *broker* oferece serviços que facilitam a comunicação com outros agentes através de seu conhecimento sobre requisitos comuns em um problema específico, e sobre a capacidade de alguns agentes em atender a estes requisitos, oferecendo uma solução ao problema. Quando um agente *iniciador* solicita a um *broker* para localizar um ou mais agentes que possam prestar um determinado serviço, o *broker* determina o conjunto de agentes apropriados para prestar o serviço, envia a solicitação a tais agentes, e em seguida encaminha a resposta ao agente que fez a solicitação (CARVALHO, 2008).

Tal esquema confere eficiência e transparência de localização durante a prestação do serviço, já que o *iniciador* não sabe e nem precisa saber onde estão os agentes que poderiam lhe prestar o serviço. O uso de agentes *broker* pode ainda simplificar significativamente a tarefa de interação entre agentes.

Adicionalmente, agentes *broker* também tornam o sistema mais adaptável e robusto em situações dinâmicas, oferecendo inclusive escalabilidade e segurança, assim como qualquer sistema que possua alta coesão e baixo acoplamento oferece. Este protocolo é chamado de macro protocolo, pois utiliza outro protocolo, chamado, sub-protocolo, para interagir com os agentes que de fato realizaram os serviços solicitados. A Figura 9 ilustra este protocolo (CARVALHO, 2008).

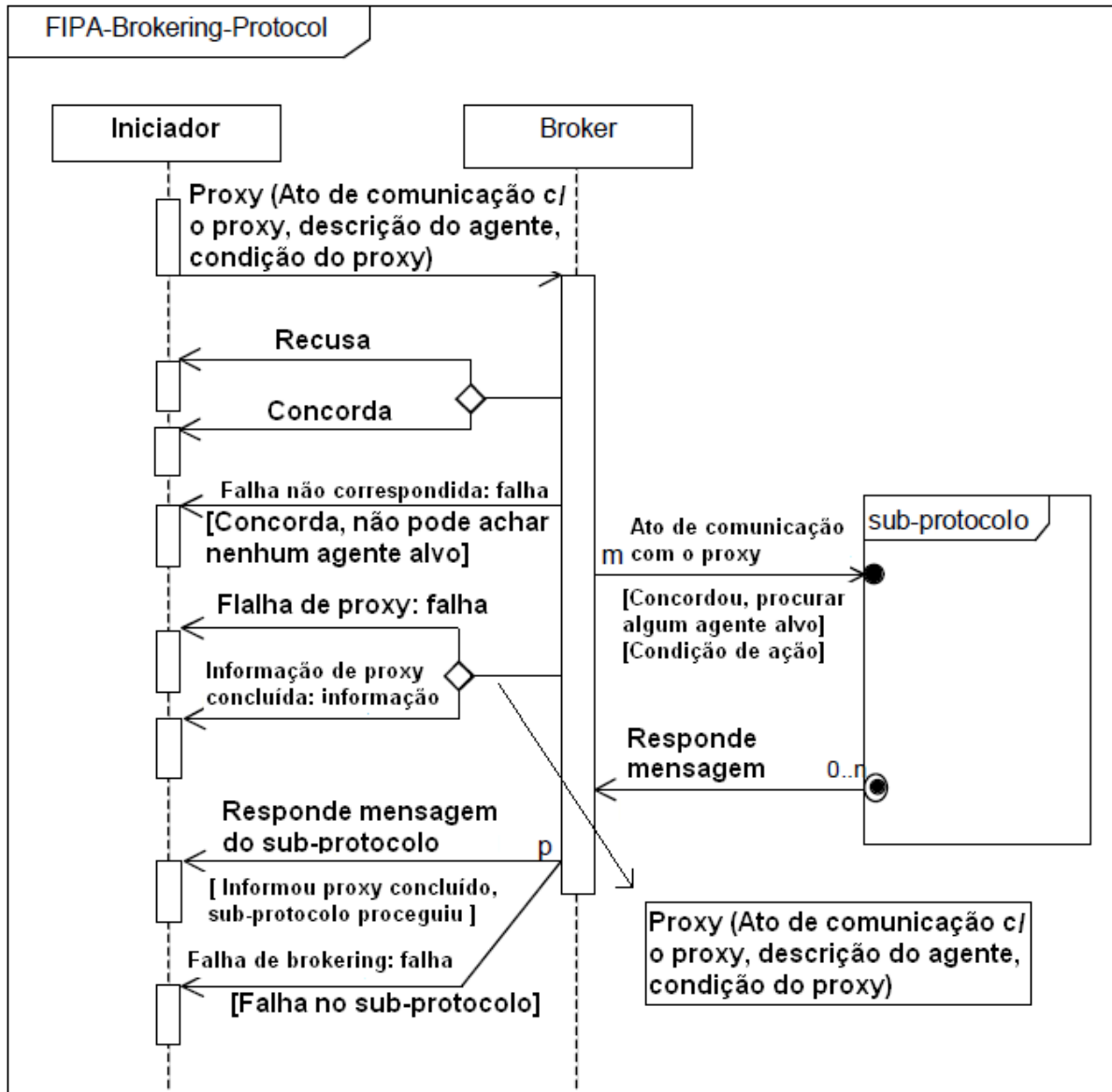


Figura 9. Protocolo de interação *Brokering* – adaptado de FIPA (2002d)

### 2.2.7 FIPA Recruiting Protocol (FIPA, 2002e)

Este protocolo de interação entre agentes é uma variação do protocolo *Brokering*. Neste caso, ao invés de intermediar a prestação do serviço, o agente *broker*, agora chamado de *recruiter* (recruta), irá localizar os agentes que prestarão o serviço, mas não irá encaminhar as respostas dos serviços executados pelos agentes localizados. Tais respostas serão entregues pelo agente que prestou o serviço diretamente ao agente *iniciador* ou qualquer outro indicado por ele. A Figura 10 ilustra este protocolo (CARVALHO, 2008).

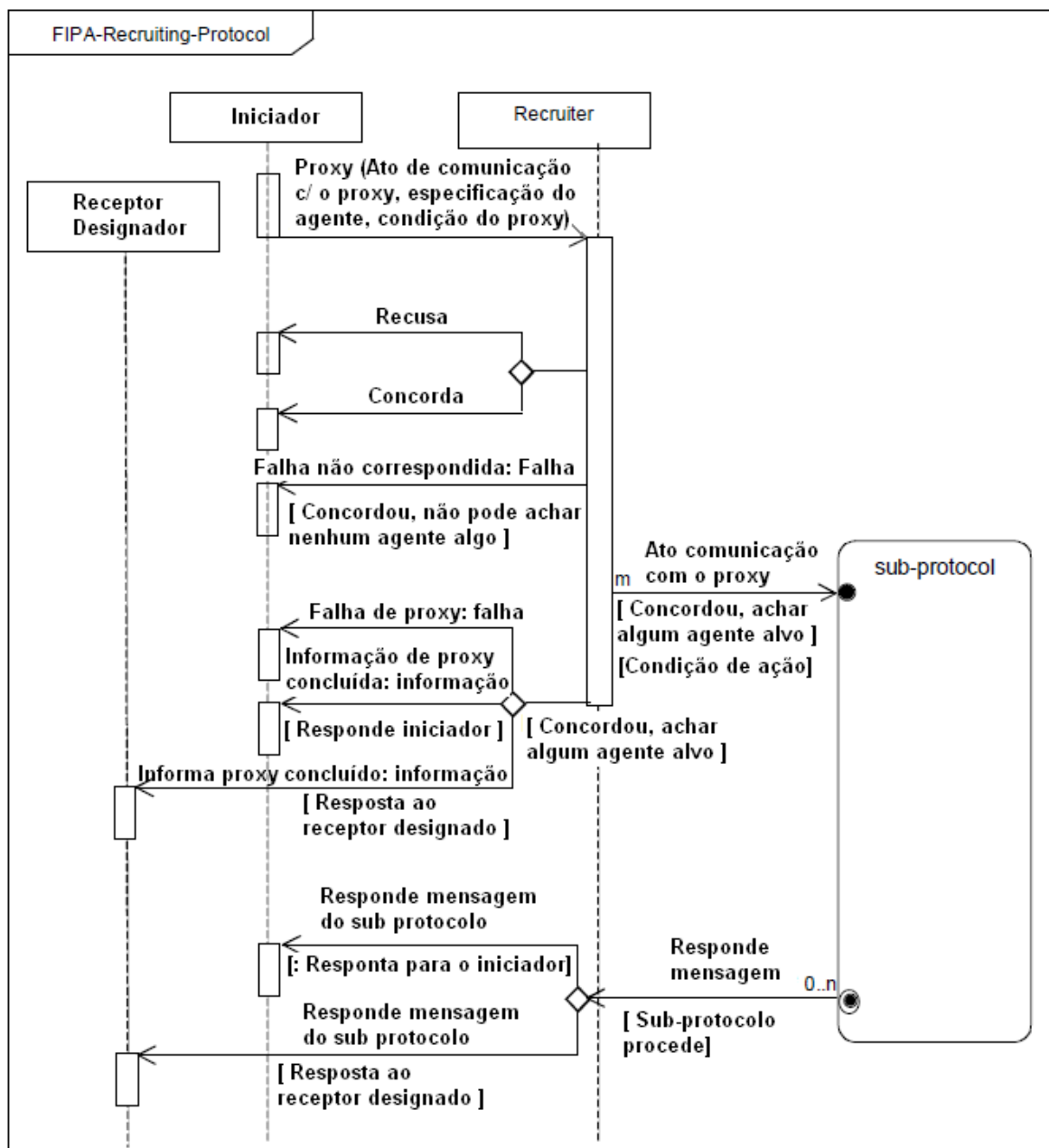


Figura 10. Protocolo de interação *Brokering* – adaptado de FIPA (2002e)

No entanto, para que se possa definir um protocolo de comunicação/interação a ser utilizado em algum projeto, deve-se conhecer antes as características do ambiente que irá usar este protocolo. Portanto, após a busca das necessidades do sistema, é possível determinar qual o melhor protocolo a ser utilizado em cada caso. Com isso, a próxima seção apresenta uma descrição comparativa dos protocolos estudados.

## 2.2.8 Quadro comparativo dos protocolos estudados

Com base nos protocolos de comunicação/interação estudados neste trabalho, foi possível a elaboração de um quadro comparativo entre esses protocolos. A Tabela 3 apresenta uma breve descrição dos protocolos usados para interação entre os agentes.

Tabela 3 – Quadro comparativo dos protocolos

<b>Protocolo</b>	<b>Descrição</b>
<i>FIPA Request Interaction Protocol (FIPA, 2002a)</i>	É um protocolo de pedido de execução de ações. Portanto, é usado para solicitar a execução incondicional de uma ou mais ações. Inicia com uma mensagem de requisição ( <i>request</i> ) que, por sua vez, pode ser recusada ou aceita pelo agente participante.
<i>FIPA Contract Net Interaction Protocol (FIPA, 2002b)</i>	Usado nos casos em que um agente solicita a um ou mais agentes que lhe apresentem propostas para a execução de uma determinada ação. Assim, o protocolo inicia pelo envio da mensagem CFP e prossegue, exceto quando uma mensagem é recusada ou não compreendida pelo agente participante.
<i>FIPA Propose Protocol (FIPA 2002c)</i>	Este protocolo é o mais simples. Usado na negociação entre agentes, onde um agente se coloca à disposição para realizar uma ação em determinadas condições. Desta forma, esta interação é iniciada pela mensagem <i>propose</i> (proposta). Contudo, essa proposta pode ser aceita ou rejeitada pelo agente que participa da negociação.
<i>FIPA Brokering Protocol (FIPA, 2002d)</i>	Usado para delegação de pedidos, onde o agente iniciador pede ao agente intermediário ( <i>broker</i> ) para este encontrar um ou mais agentes que possam responder a uma determinada pergunta. Assim, o intermediário encontra os agentes capazes de responder, passa-lhes a pergunta, recebe a resposta e envia a resposta para o agente iniciador. Este protocolo é iniciado pelo envio de uma mensagem <i>proxy</i> .
<i>FIPA Recruiting Protocol (FIPA, 2002e)</i>	Protocolo semelhante ao <i>FIPA Brokering Protocol</i> , portanto, também é usado para delegação de pedidos. A diferença é que a resposta dos agentes contatados pelo agente intermediário vai direto para o agente iniciador do protocolo ou para um conjunto especificado de agentes. Ele também é iniciado pelo envio de uma mensagem <i>proxy</i> . No entanto, o protocolo em questão é o único da família de protocolos definidos pela FIPA que tem mais que dois papéis: iniciador, recrutador e receptor designado.

Fonte: do autor

Dependendo do problema apresentado no projeto, é possível a utilização de partes de um ou mais protocolos. Neste caso, deve-se ter mente que todos os agentes devem entender o protocolo em questão. Portanto, além de definir um protocolo de interação a ser utilizado em um sistema, também é possível a escolha de mecanismos para a tomada de decisão. Assim, esses mecanismos podem trabalhar em conjunto com o protocolo definido. Tais mecanismos são as heurísticas, conforme descrito na próxima seção.

### 2.3 Heurística

Segundo o Novo Dicionário Aurélio, a palavra heurística pode ser conceituada como:

“Conjunto de regras e métodos que conduzem à descoberta, à invenção e à resolução de problemas.”. “Procedimento pedagógico pelo qual se leva o aluno a descobrir por si mesmo a verdade que lhe querem inculcar.”. “Ciência auxiliar da História, que trata da pesquisa das fontes.”. “Informação, metodologia, ou algoritmo, usado para resolver problemas por métodos que, embora não rigorosos, geralmente refletem o conhecimento humano e permitem obter uma solução satisfatória.” (FERREIRA, 2009).

Embora Ferreira (2009) já tenha fundamentado “heurística” na informática, também há outras definições para esse termo na área computacional. Desta forma, em 1971, Nicholson definiu “heurística” como um procedimento para resolver determinados problemas por um enfoque intuitivo, em geral racional, no qual a estrutura do problema passa a ser interpretada e explorada inteligentemente para obter uma solução razoável.

Por sua vez, Fernandes (2003) complementa Nicholson (1971) afirmando que heurísticas “... são métodos ou princípios para decidir, entre diversos cursos de ação alternativos, aquele que parecer mais efetivo para atingir um determinado objetivo”. Portanto, “... a Busca Heurística proporciona um meio de solucionar problemas complexos, para os quais não há disponível uma abordagem mais direta nem uma estrutura na qual qualquer técnica direta disponível possa ser inserida.” (FERNANDES, 2003).

Por outro lado, Reeves (1993) conceituou heurística como uma técnica que busca boas soluções (não necessariamente soluções ótimas) com um custo operacional razoável/aceitável, sem garantir soluções consideradas factíveis ou até ótimas e, em muitos casos, não é capaz de afirmar quão próximo uma solução factível está da solução ótima. Deste modo, pode-se optar pelo uso de uma heurística em problemas específicos, onde não se conhece qual é o melhor caminho a seguir, entretanto, o uso desta técnica poderia levar a uma solução (RUSSELL e NORVIG, 2009).

Portanto, pode-se concluir que as heurísticas estão diretamente ligadas à área de IA. Sendo assim, têm-se alguns modelos de IA para heurísticas, conforme listados a seguir: Algoritmo Genético, Programação Genética e Redes Neurais; Programação Evolutiva, Lógica

*Fuzzy*, Sistemas baseados em Conhecimento e Raciocínio Baseado em Casos (GANASCIA, 1993). Dentre os modelos citados, o algoritmo genético é descrito na próxima seção.

### 2.3.1 Algoritmos genéticos

Os Algoritmos Genéticos (AG) são técnicas para busca e otimização de soluções. São inspiradas no princípio *Darwiniano* de seleção natural e reprodução genética (metáfora da teoria da evolução – evolução das espécies). Contudo, essa técnica foi proposta por Darwin (1859) que, por sua vez, defendia que os indivíduos mais adaptados ao ambiente no qual estão inseridos seriam os que teriam mais chance de sobrevivência (maior longevidade) e, por consequência, teriam maior probabilidade de reprodução (perpetuar seus códigos genéticos para as futuras gerações). Por outro lado, os indivíduos menos adaptados ao meio teriam menos oportunidades de propagar seus códigos genéticos (DARWIN, 1859).

Desta forma, em meados de 1975, Holland criou os algoritmos genéticos para emular operadores genéticos (exemplo: *crossover*, mutação e reprodução) da mesma maneira como é observado na natureza. Com isso, Holland tinha como objetivo simular matematicamente todo o mecanismo da evolução biológica, com todas suas características e vantagens (HOLLAND, 1992). Por sua vez, Goldberg (1989), ex-aluno de Holland, foi responsável por difundir o princípio, obtendo sucesso na primeira aplicação industrial desenvolvida com o uso dos AGs. Depois disto, os AGs são usados para solucionar problemas de otimização e aprendizado de máquinas (GOLDBERG, 1989).

Portanto, os AGs podem ser definidos como modelos de computação que procuram copiar os mecanismos da “evolução natural das espécies” para a resolução de problemas de otimização considerados complexos, sendo mais robustos do que os métodos tradicionais de busca direta (MICHALEWICZ, 1998). Sendo assim, Goldberg (1989) afirma que, apesar da solução não ser garantida em um tempo finito, os AGs provaram ser um algoritmo de busca eficiente para a maioria das aplicações reais.

Neste contexto, primeiramente os AGs processam uma população de soluções iniciais (cromossomos) que, por sua vez, são evoluídas através de várias operações até que se consiga uma solução que melhor atenda a algum critério específico de avaliação (MITCHELL, 1998).

Para que isto seja possível, a cada geração, cromossomos são avaliados de acordo com uma função que verifica o nível de aptidão desse cromossomo, conhecida como função de *fitness*. Em AGs um cromossoma é uma estrutura de dados que representa uma das possíveis soluções do espaço de busca do problema. Desta forma, os cromossomos que tiverem o melhor valor de *fitness* são escolhidos para darem origem à próxima geração, através de operações como: seleção, cruzamentos e mutações.

Sendo assim, é normal que a cada geração o conjunto de soluções vá sendo melhorado e evoluindo, até que se consiga um resultado que atenda os objetivos desejados. Por fim, após vários ciclos de evolução, a população deverá possuir indivíduos mais aptos (GOLDBERG, 1989). Ainda no contexto dos AGs, a próxima seção descreve uma das técnicas de seleção de indivíduos (cromossomo) aptos (*fitness*) conhecida como método da roleta.

### 2.3.1.1 Método da roleta

Com a população já organizada, um algoritmo de seleção de indivíduos deve ser usado para escolher os que deverão fazer parte da próxima geração. Contudo, cada indivíduo deve receber uma probabilidade de ser selecionado, de acordo com o seu valor de *fitness* (função de aptidão). Sendo assim, uma das formas de seleção mais simples e extremamente utilizada, é o algoritmo conhecido como método da roleta (HOLLAND, 1992).

Neste contexto, o método da roleta pode ser comparado a uma roleta fictícia, que deve realizar um sorteio para escolher os indivíduos que deverão fazer parte de uma nova geração. Sendo assim, cada cavidade ou “casa” da roleta representa um indivíduo que, por sua vez, é proporcional ao seu valor de *fitness*. Desta forma, esse *fitness* está diretamente relacionado ao tamanho da cavidade da roleta, onde o indivíduo com mais aptidão terá uma probabilidade maior de ser o selecionado para transferir suas características aos seus filhos (próxima geração de indivíduos). Por outro lado, os indivíduos com menor de *fitness* tendem a ser excluídos da roleta (HOLLAND, 1992; GOLDBERG, 1989; TANOMARU, 1995).

A Figura 11 ilustra um exemplo do método da roleta, onde indivíduos são distribuídos proporcionalmente na roleta em questão (TANOMARU, 1995). Com base nessa figura, pode-se observar que a área de cada “casa” é diferente para cada indivíduo.



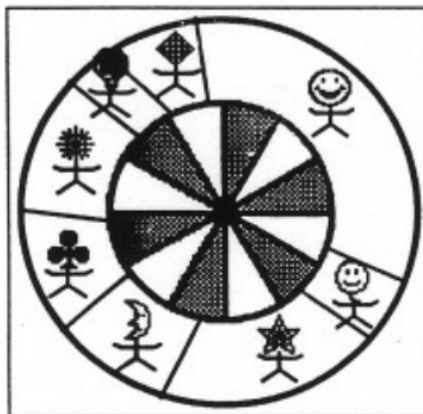


Figura 11. Exemplo do algoritmo de seleção método da roleta (TANOMARU, 1995).

Portanto, a cada giro da roleta fictícia, um indivíduo é escolhido. Assim, o cruzamento consiste em misturar o material genético de dois indivíduos da população, produzindo dois novos indivíduos (filhos) que herdam características dos pais. Por sua vez, a operação de mutação evita a convergência prematura do algoritmo, introduzindo na busca novas regiões do espaço de soluções (OLIVEIRA, 1998).

Contudo, Lacerda e Carvalho (2009) afirmam que, a probabilidade de seleção  $p_i$ , de um cromossomo  $s_i$  com aptidão  $f_i$ , é dada por:  $p_i = \frac{f_i}{\sum_{i=1}^N f_i}$ , sendo N o número de cromossomas e “i” um contador.

Sendo assim, o Quadro 3 apresenta, em português estruturado, o algoritmo baseado no método da roleta. Com base nesta função, é possível verificar que a seleção pode ser feita calculando-se uma coluna de aptidões acumuladas (valor atribuído à variável “total”). Logo, um número randômico é gerado com base em um intervalo entre os valores 0 (zero) e a variável “total” (soma de todas as aptidões).

Por sua vez, esse valor aleatório é armazenado na variável “rand”. Então, a variável “total\_parcial” e o contador “i” recebem o valor 0 (zero). Assim, o cromossomo escolhido é o que possui aptidão maior, “rand”, e uma cópia desse cromossomo é alocada na população intermediária. Por fim, esses passos são seguidos até que essa população intermediária fique com N cromossomos (LACERDA e CARVALHO, 1999).

Quadro 3 – Algoritmo da roleta

```

total ←  $\sum_{i=1}^N f_i$ ;
rand ← randômico(0, total);
total_parcial ← 0;
i ← 0;
repetir
    i ← i + 1;
    total_parcial ← total_parcial + fi;
até total_parcial ≥ rand;
retornar cromossomo Si;

```

Fonte: Lacerda e Carvalho (1999).

A próxima seção descreve as funcionalidades e os diversos trabalhos realizados sobre o STI escolhido como base para o presente trabalho.

## 2.4 STI usado como base do projeto de pesquisa

O mecanismo de comunicação entre agentes pedagógicos desenvolvido neste trabalho foi aplicado no STI desenvolvido na UNISC, projeto de estudantes e professores do departamento de informática da universidade, que teve seus primeiros trabalhos realizados em 1998 (SANTOS et al., 2001).

Este STI é um trabalho do grupo de pesquisa “Sistemas Computacionais de Apoio à Educação” do Departamento de Informática da UNISC, cadastrado no site do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq). O projeto de pesquisa vinculado a este grupo de pesquisa é “O Uso de Estilos Cognitivos e de Agentes Pedagógicos como auxílio ao Processo de Ensino-Aprendizagem em Sistemas Tutores Inteligentes”.

Contudo, o ambiente foi eleito para o projeto por estar em constante aperfeiçoamento e estudo (SANTOS et al., 2001; SILVA, 2002; MERGEN e SCHREIBER, 2005; MAINIERI et al., 2005; FROZZA et al., 2007; 2009a; 2009b; 2011; SILVA et al., 2010; BORIN, 2010; KÜNZEL, 2010; KÜNZEL et al., 2011; KÜHLEIS, 2011). Além disso, o STI em questão

possui dois agentes pedagógicos, onde ambos podem interagir com o estudante. No entanto, o sistema não possui uma interação entre os próprios agentes, sendo este o foco de pesquisa deste trabalho.

As próximas seções destacam algumas informações sobre o STI, como por exemplo, a modelagem dos agentes pedagógicos e a arquitetura conceitual de desenvolvimento dos agentes inseridos no STI. Além disso, também é apresentado o objetivo deste projeto.

### 2.4.1 Os agentes pedagógicos tutor e companheiro

Os agentes pedagógicos inseridos no *software* educacional são os agentes tutor e companheiro, onde o agente tutor possui função parecida a de um professor, com capacidade de identificar características relativas à aprendizagem do estudante. Por outro lado, o agente companheiro possui a função de atuar no ambiente como parceiro do estudante, ajudando-o nas tarefas propostas pelo ambiente (FROZZA et al., 2011).

O primeiro agente pedagógico desenvolvido no STI foi o agente tutor, chamado de Dóris. Além de auxiliar na modelagem do perfil do estudante, a professora virtual também guia e monitora o estudante dentro do ambiente, auxiliando-o em caso de dúvidas (SANTOS, 2000). Como a Dóris possui conhecimento específico em relação à aula apresentada, a mesma pode interagir com o estudante realizando perguntas sobre o assunto estudado.

Inicialmente, a Dóris possuía uma aparência em 2D (duas dimensões) e apresentava três estados emocionais, assim como pode ser observado na Figura 12.



Figura 12. Caricaturas em 2D e os estados emocionais da agente Dóris (SANTOS, 2001)

Ainda com base nos dados da Figura 12, pode-se observar que a caricatura mais da esquerda demonstra a Dóris em um estado de tristeza, já as imagens centrais exibem a agente em seu estado de felicidade. Por fim, os desenhos mais da direita expõem o estado de companheirismo da tutora virtual (SANTOS, 2001).

Já no ano de 2002 foi adicionado ao STI o agente companheiro, chamado Dimi. Esse agente passou a ser o colega virtual do estudante dentro do ambiente. Além de ajudar o estudante nas atividades do ambiente, Dimi também era responsável por obter as características cognitivas do estudante (SILVA, 2002).

Assim como a Dóris, Dimi também possuía uma representação em 2D, ilustrado na Figura 13. No entanto, ao contrário da agente tutora Dóris, o agente companheiro Dimi não possuía estados emocionais. Com isso, o agente companheiro não tinha habilidade para expressar emoções (SILVA, 2002).



Figura 13. Caricatura em 2D do agente companheiro Dimi (SILVA, 2002)

Em 2009 a agente tutora Dóris foi modelada em 3D (três dimensões), além de ter seu conjunto de estados emocionais alterado para expressar um total de 7 emoções: Tristeza, Alegria, Surpresa, Indignação, Expectativa, Atenção e Dúvida. Esses estados emocionais são explicados mais detalhadamente no tópico 3.1.2 – Computação afetiva. A Figura 14 apresenta a Dóris modelada em 3D (FROZZA et al, 2009a).



Figura 14. Caricatura em 3D da agente tutora Dóris (FROZZA et al, 2009a)

Em 2010 foi a vez do agente companheiro Dimi ganhar uma caricatura no formato 3D. Enquanto a agente Dóris apresenta características físicas semelhantes a uma mulher, Dimi possui propriedades masculinas. A Figura 15 demonstra o agente Dimi modelado em 3D (KÜNZEL, 2010).



Figura 15. Caricatura em 3D do agente companheiro Dimi (KÜNZEL, 2010)

#### **2.4.2 Arquitetura conceitual dos agentes pedagógicos inseridos no STI**

Os agentes pedagógicos contidos no STI em questão possuem como característica a interação com o estudante, assim como, a comunicação com os demais módulos do Ambiente Virtual de Aprendizagem (AVA). É importante destacar que, cada agente pedagógico atua de forma independente no sistema, não havendo interação entre eles. A arquitetura conceitual dos agentes pedagógicos está ilustrada na Figura 16 (FROZZA et al., 2007).

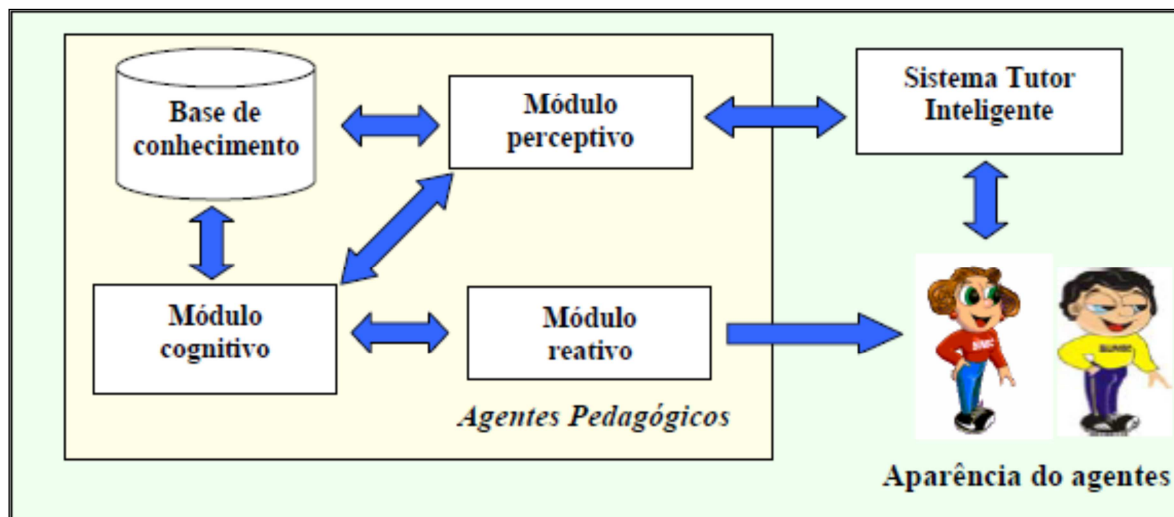


Figura 16. Arquitetura conceitual dos agentes tutor e companheiro (FROZZA et al., 2007)

Com base na Figura 16, é possível observar que o módulo perceptivo se comunica com os demais módulos, inclusive com os agentes pedagógicos e a base de conhecimento. Portanto, esse módulo é o responsável por extrair e armazenar dados referentes à interação do estudante com o STI. Além disso, o monitoramento das ações do estudante também é função deste módulo. Assim como, a verificação dos seguintes elementos (FROZZA et al., 2007):

- Hora de início e hora de término da interação entre o estudante e o STI; as páginas visitadas pelo estudante e tempo de permanência em cada uma destas páginas.
- Dificuldades e dúvidas apresentadas pelo estudante durante a realização da aula, além da opinião do estudante sobre os elementos usados na preparação da lição. Com isso, se pretende saber se o estudante realmente gostou da aula.
- Preferências do estudante sobre os elementos a serem aplicados nas aulas seguintes (o sistema pode apresentar o conteúdo de forma diferente dependendo do perfil do estudante).

Segundo Frozza et al. (2011), o uso de um STI “... proporciona a interdisciplinaridade (já que se destina a diferentes áreas de atuação), adaptando-se ao estilo cognitivo do estudante, com o apoio dos agentes pedagógicos tutor e companheiro, cujos comportamentos são determinados por percepções advindas do ambiente de interação com o estudante; e disponibilização de um ambiente de ensino-aprendizagem dinâmico e eficiente”.

Ainda sobre a arquitetura apresentada na Figura 16, pode-se observar que o módulo cognitivo possui a função de executar as inferências sobre a base de conhecimento, sendo possível determinar as ações que devem ser realizadas pelo agente pedagógico, sempre com base nas suas percepções (módulo perceptivo). Portanto, este módulo também faz a escolha das mensagens que serão emitidas pelo agente na sua interação com o estudante, assim como, entra em ação nas seguintes situações (FROZZA et al., 2007):

- Seleção de mensagens a serem enviadas ao estudante em momentos esporádicos (dicas e lembretes); quando o estudante deixa de visitar uma página; ou, quando o agente for ativado ou desativado (mensagens de boas vindas e de despedida).
- Seleção de perguntas para verificar se o estudante está com dificuldades; está gostando da interação e/ou dos elementos usados na apresentação das aulas.
- Seleção de respostas do agente às respostas do estudante.

O módulo reativo estabelece a *interface* entre os agentes pedagógicos e o estudante, onde são exibidas as mensagens para os estudantes. Além disso, este módulo também executa as ações indicadas pelo módulo cognitivo (FROZZA et al., 2007).

A Dóris ainda possui um módulo de tomada de decisão, que é ativado por regras de produção, onde as ações geradas por essa decisão constituem as táticas de ensino, enquanto as condições para essa tomada de decisão são as características do estudante. Com isso, o material apresentado para o estudante está relacionado ao seu perfil, onde as redes *bayesianas* estão relacionadas à modelagem do estilo cognitivo do estudante (FROZZA et al., 2007; 2009b).

De acordo com Frozza et al. (2009a), a Dóris emotiva foi desenvolvida para expressar emoções em 3D, com o objetivo de auxiliar os estudantes que interagem com o STI. Contudo, acredita-se que as emoções expressas pelos agentes pedagógicos podem influenciar de forma positiva na aprendizagem dos estudantes, já que a interação com esse tipo de agente se aproxima da realidade, onde a interação ocorre entre agentes reais.

O objetivo do uso deste tipo de agente pedagógico é demonstrar emoções de acordo com as ações do estudante. Com isso, a agente Dóris pode manifestar felicidade em

momentos que o usuário executa alguma atividade proposta. Por outro lado, a Dóris pode expressar sentimentos de tristeza quando o conteúdo apresentado não é percebido. Em 2009 a agente Dóris passou a expressar um total de sete emoções: tristeza, alegria, surpresa, indignação, expectativa, atenção e dúvida (BORIN, 2010; BORIN et al., 2012).

Já os estímulos que o ambiente gera através das ações dos estudantes referem-se a: pulo de página, permanência em cada página (pouco ou muito tempo - definido pelo especialista do conteúdo), volta a uma página anterior de conteúdo, ou mensagens aleatórias (exibidas quando o estudante permanece algum tempo sem interagir com o ambiente).

### 2.4.3 Histórico de evolução do ambiente

Com base nos dados apresentados pela Tabela 4, pode-se perceber que o projeto iniciou com apenas com um agente, a Dóris 2D, que expressava três emoções e tinha como uma das funções guiar o estudante dentro do ambiente. Depois disto, o agente companheiro Dimi 2D foi inserido no sistema, com o objetivo de ser um colega do estudante. Ambos agentes possuíam um conhecimento de domínio geral. Já nos anos 2005 e 2007 o ambiente passou a contar com outros trabalhos, como por exemplo, a coleta de informações para a detecção do perfil do estudante através de redes *bayesianas* e teste de *Ross*, respectivamente.

A partir do ano de 2009 o ambiente passou a contar com várias alterações, onde os agentes pedagógicos passaram para o formato 3D com a expressão de emoções, de acordo com a interação do estudante no STI.

Tabela 4 – Evolução do STI no decorrer dos anos

Característica	2000	2002	2005	2007	2009	2010	2011	2012
Agente Dóris	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Agente Dimi	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Guiar Aluno	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Coletar Informações	-	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Rede <i>Bayesiana</i>	-	-	Sim	Sim	Sim	Sim	Sim	Sim
Teste de <i>Ross</i>	-	-	-	Sim	Sim	Sim	Sim	Sim
Emoções Dóris	3	3	3	3	7	7	7	7
Emoções Dimi	-	Fixa	Fixa	Fixa	Fixa	2	2	2
Domínio de Conhecimento Dóris	Geral	Geral	Geral	Geral	Geral	Geral	Geral e específico	Geral e específico



Característica	2000	2002	2005	2007	2009	2010	2011	2012
Domínio de Conhecimento Dimi	-	Geral	Geral	Geral	Geral	Geral	Geral	Geral
Aparência Dóris	2D	2D	2D	2D	3D	3D	3D	3D
Aparência Dimi	-	2D	2D	2D	2D	3D	3D	3D
Interação entre agentes pedagógicos	-	-	-	-	-	-	-	Foco do projeto

Fonte: Adaptado de KÜHLEIS (2011)

A próxima seção apresenta alguns trabalhos correlatos, tanto desenvolvidos sobre um STI quanto outras áreas que, por sua vez, também precisam de protocolos de comunicação entre agentes.

## 2.5 Trabalhos relacionados

O estudo de trabalhos relacionados foi necessário, pois tais pesquisas contribuiriam no desenvolvimento de um mecanismo de comunicação entre agentes, proposto neste trabalho. Experiências de outros autores auxiliam nas decisões. Portanto, os próximos tópicos apresentam brevemente alguns trabalhos relacionados à comunicação entre agentes.

### 2.5.1 Plataforma COMPOR

O primeiro trabalho relacionado é intitulado “*Especificação e Implementação de Protocolos de Interação entre Agentes para a Plataforma COMPOR*”, e foi desenvolvido pelos seguintes pesquisadores: Glauber Vinícius Ventura de Melo Ferreira, Hyggo Oliveira de Almeida, Angelo Perkusich e Evandro de Barros Costa.

Este trabalho foi publicado na Biblioteca Digital Brasileira de Computação (BDBComp), que possui parceria com Sociedade Brasileira de Computação (SBC). Além disso, a pesquisa também foi publicada na InfoComp: Revista de Ciência da Computação da Universidade Federal de Lavras (UFLA), Lavras (MG), em novembro de 2004 (FERREIRA et al., 2004).

Os pesquisadores iniciam o artigo justificando o uso de uma plataforma baseada em componentes (COMPOR) para a criação de SMAs, no entanto, essa estrutura não implementa

os protocolos de comunicação, o que fez com que os autores desenvolvessem essa parte do trabalho. Resumo da obra:

“A abordagem multiagentes tem sido apontada como adequada para o desenvolvimento de sistemas complexos devido ao alto nível de abstração que oferece, sendo a interação entre os agentes sua característica mais importante. Dentro deste contexto, a infra-estrutura COMPOR para a construção de Sistemas Multiagentes utiliza a abordagem baseada em componentes para prover flexibilidade e reutilização no desenvolvimento de tais sistemas. Porém, esta infra-estrutura não implementa mecanismos que facilitem a definição de protocolos de interação entre os agentes do sistema. Neste artigo apresenta-se a definição e a implementação de mecanismos para interação entre agentes da infra-estrutura COMPOR, levando-se em conta protocolos e linguagens de comunicação com base nos padrões definidos por FIPA (FERREIRA et al, 2004)”.

Segundo Ferreira et al. (2004), a plataforma COMPOR é um projeto de Engenharia de *software* para SMAs, cujo trabalho é fruto de uma parceria entre a Universidade Federal de Campina Grande (UFCG) e a Universidade Federal de Alagoas (UFAL) (ALMEIDA et al., 2004). O COMPOR faz parte de um projeto maior, portanto, o COMPOR-Environment (COMPOR-E) é o módulo responsável pelo desenvolvimento dos SMAs (ALMEIDA et al., 2004).

Com o objetivo de facilitar a definição dos protocolos de interação/comunicação, Ferreira et al. (2004) desenvolveram e integraram ao módulo COMPOR-E, um componente para comunicação entre agentes por meio da linguagem ACL e uma ferramenta para especificação de protocolos de interação com base no padrão FIPA. Esses novos componentes estão disponíveis no COMPOR-E em uma paleta de componentes.

Em razão das diferenças existentes na estrutura das mensagens FIPA-ACL e KQML, Ferreira et al. (2004) optaram em implementar o componente apenas para a linguagem FIPA-ACL, pois ela segue o padrão FIPA e é mais usada no desenvolvimento de SMAs (GLUZ e VICCARI, 2003).

Para que a troca de mensagens ACL entre os agentes seja possível, é necessário o uso de arquivos XML (*Extensible Markup Language* - linguagem de marcação extensível) com as

especificações dos protocolos de interação/comunicação. Deste modo, a segunda ferramenta desenvolvida por Ferreira et al., (2004) possui justamente essa finalidade, ou seja, ela permite que os protocolos de comunicação sejam especificados neste componente que, posteriormente, altera ou cria os arquivos XML. Desta forma, esse *software* pode ser usado como um repositório de protocolos, já que os dados ficam armazenados no sistema. Durante o desenvolvimento dos trabalhos, os pesquisadores especificaram de forma completa os protocolos FIPA: *Request* e *Contract Net*.

Maiores especificações sobre os componentes desenvolvidos podem ser encontrados em Ferreira et al. (2004). O próximo tópico aborda a comunicação entre agentes sobre outra plataforma de desenvolvimento conhecido como SACI.

### **2.5.2 Plataforma SACI**

O SACI (*Simple Agent Communication Infrastructure* - Infraestrutura simples para comunicação de agentes) (SACI, 2011) é uma ferramenta desenvolvida no Laboratório de Técnicas Inteligentes (LTI) da Universidade de São Paulo (USP) e se trata de um ambiente para a construção e execução de Sistemas Multiagentes.

De acordo com Hirata et al. (2005),

“... o ambiente provê uma infra-estrutura que serve de base para a implementação do comportamento social dos agentes, fornecendo meios para que estes possam se conhecer e se comunicar de maneira transparente. Basicamente, o ambiente fornece dois recursos principais: uma API<sup>6</sup> (*Application Program Interface*) para compor, enviar e receber mensagens, e ferramentas que auxiliam o projetista a superar dificuldades em relação à programação em ambientes distribuídos (HÜBNER e SICHMAN, 2000)”.

Portanto, a ferramenta SACI foi criada com base na especificação da linguagem de comunicação KQML (LABROU e FININ, 2000) e foi desenvolvida em Java e XML.

Os dois próximos tópicos apresentam trabalhos desenvolvidos sobre o ambiente SACI, sendo que o primeiro implementou uma biblioteca para o reconhecimento e monitoramento de

protocolos de comunicação/interação, além de uma API para manipular as interações (troca de mensagens) entre os agentes, inclusive o ambiente (HIRATA et al., 2005). Já o segundo adicionou uma ferramenta para especificação e execução de protocolos de comunicação em SMAs, assim como um *software* com uma interface gráfica para manutenção dos protocolos de comunicação (GIMÉNEZ-LUGO et al., 2006).

### 2.5.2.1 Implementação de protocolos de comunicação

Este trabalho relacionado possui o seguinte título: “*Implementação de protocolos de comunicação no ambiente SACI*”, e foi desenvolvido pelos seguintes pesquisadores: Issao Hirata, Jomi Fred Hübner, Jaime Simão Sichman. O artigo foi publicado no XIV Seminário de Computação, em 2005 (HIRATA et al., 2005).

O ambiente SACI permite que os agentes de uma sociedade troquem mensagens de forma transparente, no entanto, isto ocorre de forma não estruturada. Para tanto, deve-se especificar uma linguagem de descrição de protocolos (LDP), assim como uma arquitetura em paralelo para que ela possa processar essa LDP. Neste contexto, os pesquisadores tiveram como objetivo melhorar o *software* SACI, através da criação de uma nova biblioteca para que essa plataforma pudesse reconhecer e monitorar os protocolos de comunicação que fazem parte dos mecanismos de comunicação/interação entre os agentes (HIRATA et al., 2005).

A especificação da LDP utilizada no trabalho de Hirata et al. (2005) foi elaborada com base nos diagramas AUML (*Agent Unified Modeling Language* – Linguagem de modelagem unificada de agentes) e também na proposta descrita em Populaire et al. (1993). Assim como Ferreira et al. (2004), Hirata et al. (2005) também desenvolveram um componente com uma interface amigável para que o programador possa modelar com mais facilidade os protocolos de comunicação/interação e, em seguida, gerar o arquivo XML com essas especificações. Isso faz com que o desenvolvedor não precise se preocupar com os detalhes da LDP usada.

O ambiente SACI já contava com uma API chamada de *mBox* (*MessageBox*), que possuía a função de transmitir e receber mensagens KQML entre os agentes, além de possuir uma interface para a comunicação entre o agente e o ambiente (HÜBNER e SICHMAN,

---

<sup>6</sup> API, traduzido para o português: Interface de programação de aplicativos.

2000). Com isso, as mudanças desenvolvidas no SACI foram focadas na criação de uma nova API, chamada de *ProtocolBox*, responsável por manipular as interações (troca de mensagens) de agentes de maneira semelhante ao *mBox* (Hirata et al., 2005).

Segundo Hirata et al. (2005), o novo componente desenvolvido tem praticamente as seguintes funcionalidades: “... ler e interpretar as informações de cada protocolo contidas nos arquivos XML; a de gerenciar e controlar a lógica das conversas (baseadas nos protocolos); a de oferecer uma API com métodos que auxiliam a programar tais interações nos agentes”.

Neste contexto, o *ProtocolBox* é uma extensão do *mBox*, onde ele possui os dados de as conversas em que o agente está participando e oferece métodos para que ele possa manipular esta conversa. Por fim, Hirata et al., (2005) encerra o artigo com a implementação do protocolo *FIFA Contrat Net* para exemplificar e validar o *software* desenvolvido.

O próximo tópico aborda mais um trabalho realizado sobre a plataforma SACI, cujas novas implementações e componentes foram disponibilizados.

### **2.5.2.2 Ferramenta para especificação e execução de protocolos em SMAs**

Este trabalho relacionado possui o seguinte título: “*Um Enfoque para a Especificação e Execução Flexível de Protocolos em Sistemas Multiagentes*”, e foi desenvolvido pelos seguintes pesquisadores: Gustavo Alberto Giménez-Lugo, I. Jeukens, Regis Faria e Jomi F. Hübner. O artigo foi publicado na revista *Gestão & Conhecimento* da FATEC (Faculdade de Tecnologia), em 2006 (GIMÉNEZ-LUGO et al., 2006).

De acordo com Giménez-Lugo et al. (2006), o principal objetivo do seu trabalho é a implementação de uma arquitetura para a criação, armazenamento e execução de protocolos em SMAs. Portanto, os pesquisadores propuseram um modelo formal para implementação e implantação dinâmica de protocolos, inclusive com organização e reorganização. Esse modelo é uma extensão da pesquisa de Lin et al. (1999), pois além de trabalhar com as ações dos agentes, o modelo também trata os papéis assumidos por eles. Portanto, o trabalho dos autores

visa permitir a execução de protocolos com tais características, onde um “ator” é um agente que desempenha um conjunto de papéis.

Giménez-Lugo et al. (2006) optaram em implementar o componente para execução de protocolos de comunicação em Java, para garantir a interoperabilidade entre os sistemas, já que o ambiente SACI também foi desenvolvido nesta linguagem de programação. Como os protocolos precisam ser transmitidos numa sociedade de agentes, os pesquisadores também escolheram XML como “linguagem de transporte” e “formato de armazenamento”, para da mesma forma, manter a interoperabilidade atrelada ao formato de envio de mensagens. Essa implementação insere uma nova camada de funcionalidades ao SACI, com a finalidade deixar “quase-transparente” a execução de protocolos de comunicação, permitindo que os projetistas possam se dedicar apenas aos objetivos dos sistemas a serem desenvolvidos (GIMÉNEZ-LUGO et al., 2006).

Além do modelo formal proposto para a especificação de protocolos, os pesquisadores também apresentaram um sistema para manutenção dos mesmos, onde uma interface gráfica foi implementada para permitir a captura de um protocolo de comunicação entre agentes. Esse *software* pode gerar um arquivo no formato XML. No entanto, para não desenvolver um componente desde o início, o ambiente *Ptolemy II* (DAVIS et al., 2001) foi usado como base. Essa ferramenta também foi implementada em Java e tem uma interface gráfica amigável, que possibilita a edição de esquemas com base em componentes, além de ser um ambiente de para modelar sistemas, permitindo a realização de simulação de protocolos antes de usá-los no ambiente SACI (GIMÉNEZ-LUGO et al., 2006).

Logo após o desenvolvimento da ferramenta proposta, protocolos de comunicação foram especificados nesse novo componente, como por exemplo, *FIPA Contract Net*.

Maiores especificações sobre os componentes desenvolvidos sobre o ambiente SACI podem ser encontrados em Giménez-Lugo et al. (2006) e Hirata et al. (2005). O próximo tópico aborda outro trabalho correlato ao presente projeto de mestrado, agora voltando a plataformas de STI.

### 2.5.3 Arquitetura de um agente identificador de fatores motivacionais e afetivos em um AVA

Este trabalho relacionado é intitulado “*Arquitetura de um Agente Identificador de Fatores Motivacionais e Afetivos em um Ambiente de Ensino e Aprendizagem*”. Foi um trabalho de dissertação desenvolvido por Harry Erwin Moissa, na Universidade Federal do Rio Grande do Sul (UFRGS), em 2001. Essa dissertação de mestrado contou com a orientação da Prof<sup>a</sup>. Dra. Rosa Maria Vicari.

Moissa (2001) desenvolveu um protótipo baseado na arquitetura de SMA autônomos desenvolvidos no “*Eletrotutor III*” (BICA, 1999), que depois foi adaptado para o “*Eletrotutor IV*” (SILVEIRA, 2001). A principal mudança realizada na arquitetura do STI foi a inserção de um novo agente, denominado Percepção. O novo agente foi inserido junto ao, já existente, agente Interface, conforme ilustra a Figura 17.

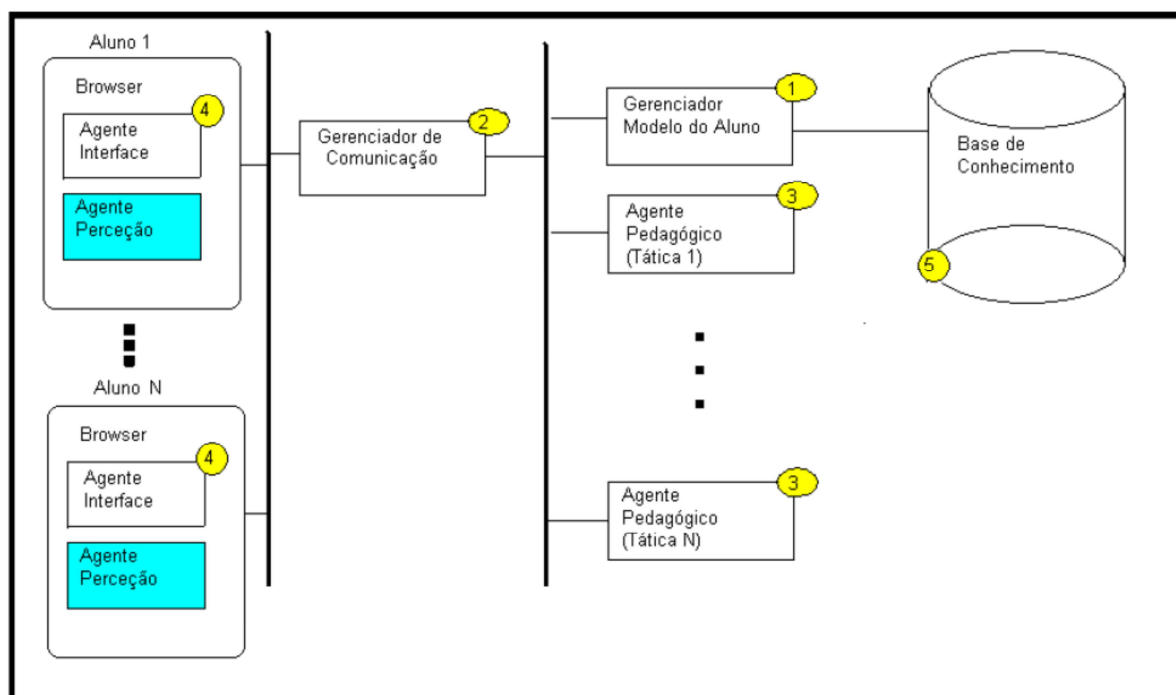


Figura 17. Arquitetura modificada do Eletrotutor IV – Moissa (2001)

Com base na arquitetura proposta por Moissa (2001), sua modificação está focada no agente marcado com o número 4, ou seja, o agente Percepção. Os demais agentes ficam iguais ao trabalho desenvolvido por Silveira (2001), onde o agente 2 (Gerenciador de comunicação)

é o responsável pela gerência de toda comunicação que ocorre entre o Agente de Interface e os demais agentes do sistema (mensagens são recebidas e enviadas através do Java RMI<sup>7</sup>).

Por sua vez, o agente 3 (agente pedagógico) é responsável pela tática de ensino. O agente 1 (gerenciador do modelo do aluno) preza pela criação e manutenção de uma base do conhecimento (5), que mantenha a representação cognitiva de cada um dos estudantes que estejam ou já estiveram conectados ao sistema. Também foi incorporado a esta arquitetura um novo conjunto de funções que darão suporte aos estados motivacionais identificados pelo agente percepção, que era um dos objetivos de Moissa (2001).

Segundo Moissa (2001), *“Para que o agente Percepção possa desempenhar sua tarefa com sucesso faz-se necessário que o STI apresente uma série de requisitos que serão classificados em quatro grupos: requisitos quanto à organização do domínio, requisitos quanto ao sistema de ajuda, requisitos quanto às estratégias de ensino e requisitos quanto aos eventos relevantes, ou seja, um modelo de estudante com afeto”*.

O agente Percepção é responsável por monitor toda a comunicação que ocorrem entre a interface e o STI. Os dados relevantes são armazenados em um histórico (*log*), onde são aplicadas regras que identificam os fatores motivacionais e afetivos do estudante, durante o tempo em que o Tutor é usado. Toda modificação ocorrida no sistema é enviada para os esquemas afetivos (modelo do aluno).

Portanto, com base nos dados estudados por Moissa (2011), a Figura 18 ilustra a arquitetura do agente Percepção. O sistema de comunicação usado no STI é fundamental para o funcionamento do agente Percepção, pois ele usa o conteúdo das mensagens com sua fonte de informação (MOISSA, 2001).

---

<sup>7</sup> RMI: *Remote Method Invocation* ou Invocação de métodos remotos.



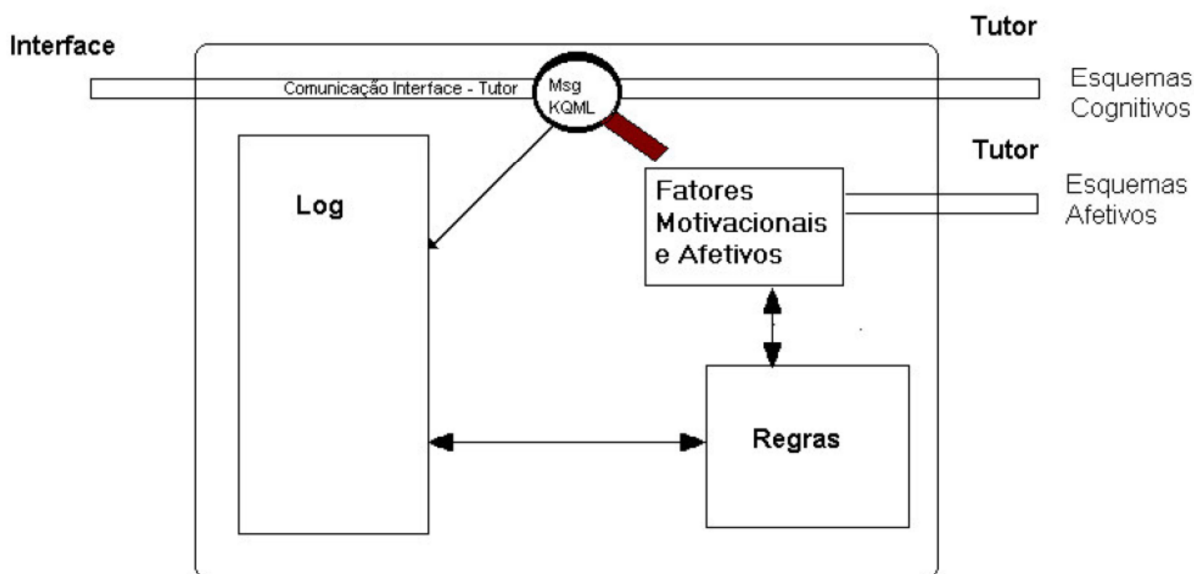


Figura 18. Arquitetura do agente Percepção – Moissa (2001)

O “Eletrotutor IV” usa de dois métodos de comunicação: o *JDBCThin* da *Oracle* que é utilizado na comunicação entre os agentes e o banco de dados, e o Java RMI<sup>8</sup> que é utilizado para a comunicação entre agentes. Com isso, a Figura 19 é uma adaptação da arquitetura do ambiente do “Eletrotutor III”, apresentado em Bica (1999) e Silveira (2001).

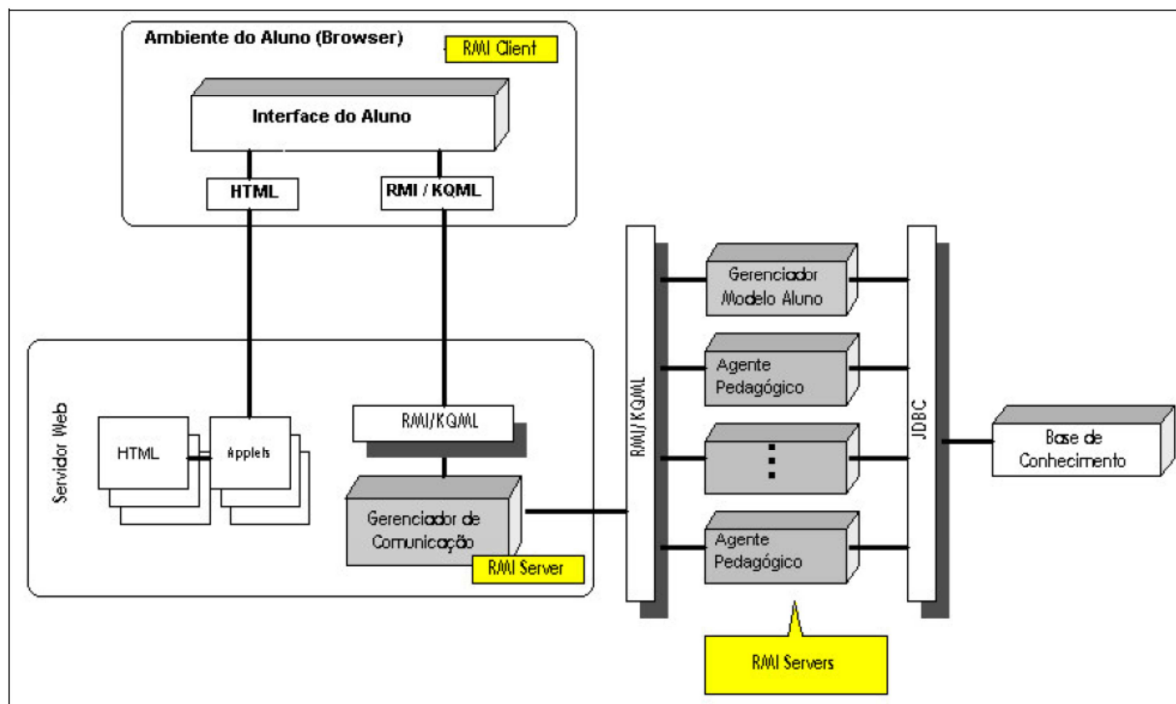


Figura 19. Arquitetura do ambiente – Moissa (2001)

Ainda com base na Figura 19 pode-se verificar que todas as comunicações ou troca de mensagens feitas entre o agente Interface e os demais agentes do *Eletrotutor* são do modelo ponto-a-ponto, que é mais seguro. Também pode-se perceber que RMI Java e um protocolo baseado em KQML são usados na comunicação (MOISSA, 2001).

Segundo Wooldridge (1995), no modelo ponto a ponto as mensagens são enviadas para um endereço específico que é conhecido pelo transmissor, onde esse transmissor sabe para quem a mensagem está sendo enviada. Assim sendo, os controles referentes à segurança são mais fáceis de desenvolver. Outra característica a ser destacada no trabalho é o Java RMI que implementa o protocolo KQML, o que deixa o sistema de comunicação flexível, padronizado e com bastante capacidade de distribuição (MOISSA, 2001).

O próximo tópico trata da arquitetura geral de comunicação do *MathTutor* (FRIGO e BITTENCOURT, 2002), outro STI com características relacionadas ao projeto de mestrado em questão.

#### **2.5.4 O papel dos Agentes Inteligentes nos STIs**

O presente trabalho relacionado é intitulado “*O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes*” e foi desenvolvido pelos seguintes autores: Luciana Bolan Frigo, Eliane Pozzebon e Guilherme Bittencourt. O artigo foi publicado nos anais do WCETE - *World Congress on Engineering and Technology Education*, São Paulo/SP, 2004.

De acordo com Frigo et al. (2004), *MathTutor* é um STI em desenvolvimento na Universidade Federal de Santa Catarina (UFSC) (FRIGO e BITTENCOURT, 2002), que pretende auxiliar no ensino de Fundamentos da Estrutura da Informação para os alunos de Engenharia de Controle e Automação.

Contudo, antes de focar na comunicação dos agentes, é necessário o entendimento da arquitetura e funcionamento do ambiente *MathTutor*. A seguir são especificados os quatro módulos deste sistema: o módulo do especialista contém os dados a respeito do conhecimento do conteúdo a ensinar; o módulo do estudante armazena os dados referentes à compreensão

---

<sup>8</sup> RMI: *Remote Method Invocation* ou Invocação Remota de Métodos.

do estudante sobre o domínio de conhecimento; o módulo pedagógico contém as regras para a tomada de decisão que possibilitam determinar o quanto o estudante está aprendendo; e o módulo da interface apresenta ao estudante o AVA criado pelo especialista (FRIGO et al., 2004).

Além disso, o *MathTutor* usa a tecnologia dos agentes cognitivos para a criação de um sistema mais perceptivo, o que aumenta a qualidade sob o ponto de vista pedagógico (FRIGO e BITTENCOURT, 2002). Portanto, esse tipo de agente pode decidir a ação a ser tomada, com base em um sistema de regras. Os agentes também adotam um modelo de cooperação, no qual ocorre o compartilhamento das tarefas a serem resolvidas, onde cada agente fica responsável por uma parte do problema. Eles também podem negociar uma informação (FRIGO et al., 2004).

Segundo Frigo et al. (2004), “troca de mensagens” foi a arquitetura de comunicação escolhida para o projeto. Com isso, os agentes podem se comunicar de forma direta, desde que respeitem a hierarquia do sistema que, por sua vez, possui dois níveis: (a) nível superior: formado pelo Agente de Interface (agente facilitador); e (b) inferior: constituído pelos agentes fornecedores de serviços. Entre esses agentes fornecedores de serviços não existe hierarquia, todos eles podem conversar, negociar e trocar informações dependendo dos seus interesses.

O *MathTutor* é composto por quatro agentes, onde dois deles são responsáveis pelo conteúdo teórico e os outros dois pelo prático, segundo a modelagem do domínio. Portanto, os agentes e a base de conhecimento são essenciais, assim como o compartilhamento de conhecimento e o aprendizado das preferências do estudante (FRIGO et al., 2004).

De acordo com os pesquisadores, o ambiente que permite a troca de mensagens entre os agentes é conhecido como *JATLite*<sup>9</sup> (*Java Agent Template Lite*) (JEON et al., 2000). Esse ambiente realiza essas transferências através da linguagem KQML, onde o conteúdo destas mensagens são “fatos” que serão inseridos na máquina *JESS*<sup>10</sup> (*Java Expert System Shell*) (JESS, 2011), permitindo que o sistema tome decisões e mude o seu comportamento. Por fim, a comunicação do estudante com o sistema ocorre através da interface, que é uma página

---

<sup>9</sup> *JATLite*: plataforma para agentes distribuídos.

<sup>10</sup> *JESS*: Sistemas baseados em Regras (em Java).

HTML (*HyperText Markup Language*) com conteúdo gerado por *Servlets*. Já a comunicação entre os agentes é feita através de performativas (FRIGO et al., 2004).

O *JATLite* pode ser definido como um conjunto de classes Java que fornecem uma arquitetura básica para a construção de agentes que se comunicam através da Internet. A transmissão das mensagens entre os agentes pode ocorrer por um mecanismo de *pooling* onde o agente emissor verifica se o receptor está conectado para enviar a mensagem. As conexões entre os agentes ficam ativas até que o agente encerre a conexão ou um tempo máximo de ociosidade seja atingido (*timeout*). A Figura 20 ilustra a arquitetura de troca de mensagens usada no projeto *MathTutor* (FRIGO et al., 2004).

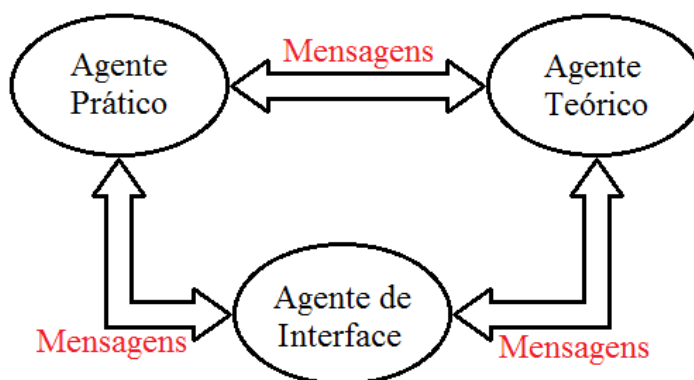


Figura 20. Arquitetura para troca de mensagens – adaptado de Frigo et al. (2004)

Frigo et al. (2004) fazem uma observação importante sobre o *JESS*, onde ele:

“... é utilizado em diversas aplicações, mas o uso do *JESS* com a tecnologia dos *Applets* deixa o sistema muito pesado. Por isso quando a ideia é utilizar aplicações com *JESS* via navegador, devemos considerar o uso do *JESS* do lado do servidor, como o que ocorre no caso dos *Servlets*, dispensando o usuário de carregar grande parte do sistema para sua máquina o que torna a interação com o sistema bastante lenta e entediante do ponto de vista do usuário”.

Portanto, o trabalho de Frigo et al. (2004) descreve a utilização de agentes cognitivos num sistema tutor inteligente cujo objetivo é auxiliar no ensino de Fundamentos da Estrutura da Informação para os alunos de Engenharia de Controle e Automação UFSC.

### 2.5.5 Agentes inteligentes para o ensino de lógica

Este trabalho relacionado possui o seguinte título: “*Agentes Inteligentes no Ambiente Virtual de Ensino de Lógica Halyen*” e foi desenvolvido por Sahudy Montenegro González, Annabell del Real Tamariz, Eduardo Coelho Carneiro e Jamile Souza de Almeida. O artigo foi publicado na Conferência IADIS Ibero-Americana WWW/Internet 2007, Vila Real, Portugal, em 2007.

Segundo González et al. (2007), o objetivo do ambiente é fazer com que a escolha da estratégia pedagógica seja dinâmica, com base no perfil do estudante. O HALYEN é voltado para o ensino de Lógica Matemática, por ser essencial para o ensino de matérias de algoritmos e programação, onde grande parte dos alunos de computação e cursos afins possui muitas dificuldades e muitas vezes não conseguem absorver o conteúdo por não ter um raciocínio lógico bem desenvolvido.

O ambiente O HALYEN, foi desenvolvido na linguagem Java, com o auxílio do *Framework JADE*<sup>11</sup> (*Java Agent Development framework*) (GYURJYAN et al., 2003), que possui uma infraestrutura de suporte ao desenvolvimento de SMAs. JADE segue as especificações FIPA e conta com um conjunto de ferramentas gráficas para auxiliar o desenvolvimento dos projetos. Neste contexto, o modelo de comunicação escolhido foi a “troca de mensagens”, onde cada agente sabe o nome e endereço de todos os agentes que formam o sistema. As mensagens seguem o padrão FIPA-ACL, que usa performativas para definir o protocolo de comunicação entre os agentes (GONZÁLEZ et al., 2007).

Por fim, as tarefas no ambiente são feitas através da comunicação entre os agentes. No entanto, nenhum agente se comunica de forma direta. Toda requisição é passada para o agente coordenador, que intermedia as ações entre os agentes. Ele recebe as requisições do ambiente e passa a mensagem para o agente responsável pela execução da tarefa. A Figura 21 ilustra essa comunicação entre agentes no O HALYEN (GONZÁLEZ et al., 2007).

---

<sup>11</sup>JADE: Framework Java para o desenvolvimento de agentes.

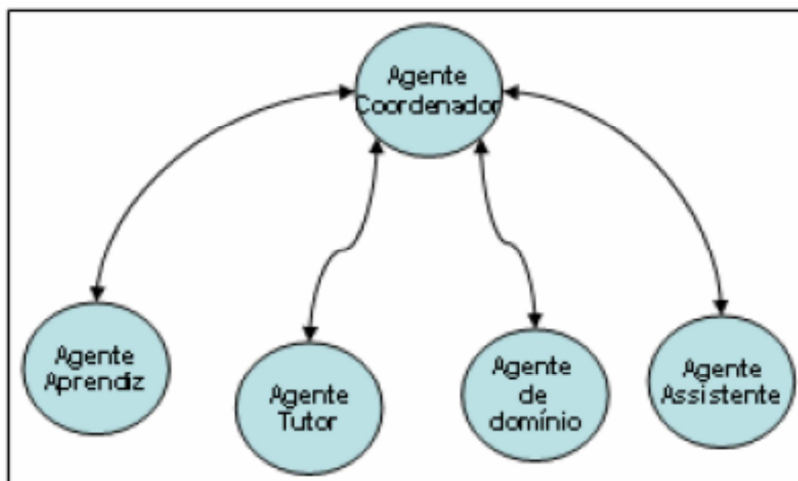


Figura 21. Comunicação entre agentes no ambiente OHALYEN - González et al. (2007)

## 2.6 Considerações referentes aos trabalhos relacionados

Com base nos trabalhos relacionados e através dos estudos realizados, a proposta do trabalho foi desenvolver um mecanismo de comunicação entre agentes pedagógicos em um AVA.

A Tabela 5 apresenta as características de alguns trabalhos relacionados que auxiliaram para escolha do protocolo de comunicação que foi desenvolvido entre os agentes pedagógicos do ambiente virtual de aprendizagem.

Tabela 5 – Tabela comparativa dos trabalhos relacionados

<b>Autores</b>	<b>Descrição</b>	<b>Comunicação</b>
Ferreira et al. (2004)	Especificação e Implementação de Protocolos de Interação entre Agentes para a Plataforma COMPOR.	Linguagem FIPA-ACL. Arquivos XML com especificações dos protocolos. Protocolos FIPA <i>Request</i> e <i>Contract Net</i> .
Giménez-Lugo et al. (2006)	Um Enfoque para a Especificação e Execução Flexível de Protocolos em Sistemas Multiagentes	Linguagem KQML. Arquivos XML com especificações dos protocolos. Protocolos FIPA <i>Contract Net</i> .
Hirata et al. (2005)	Implementação de protocolos de comunicação no ambiente SACI	Linguagem KQML. Arquivos XML com especificações dos protocolos. Protocolos FIPA <i>Contract Net</i> .
Moissa (2001)	Arquitetura de um Agente Identificador de Fatores Motivacionais e Afetivos em um Ambiente de Ensino	Linguagem de comunicação KQML. Troca de mensagens baseadas no modelo ponto a ponto e uso do Java RMI para a

Autores	Descrição	Comunicação
	e Aprendizagem	comunicação entre agentes.
Frigo et al. (2004)	O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes	Linguagem de comunicação KQML. Arquitetura “Troca de mensagens” para o envio e recebimento de mensagens, implementado na ferramenta JATLite e JESS.
González et al. (2007)	Agentes Inteligentes no Ambiente Virtual de Ensino de Lógica Halyen	Linguagem de comunicação FIPA-ACL. Arquitetura “Troca de mensagens” para o envio e recebimento de mensagens, implementado no <i>framework</i> JADE.
Griesang, Geovane (2013)	Desenvolvimento de um Mecanismo de Comunicação entre Agentes Pedagógicos em um Ambiente Virtual de Aprendizagem	Arquitetura “Troca de mensagens” para o envio e recebimento de mensagens; Linguagem de comunicação FIPA-ACL; Linguagem de conteúdo com formato próprio; Protocolo de comunicação/interação com base no FIPA <i>Contract Net Interaction Protocol</i> .

Fonte: Autor (2013).

Os três primeiros trabalhos relacionados (Ferreira et al., 2004; Hirata et al., 2005; Giménez-Lugo et al. 2006) focaram na criação de ferramentas para aperfeiçoar as plataformas de desenvolvimento de SMAs estudadas (COMPOR-E e SACI), oferecendo recursos para que o programador possa desenvolver sua atividade de forma mais eficiente. Isto foi possível porque tais componentes visaram simplificar e facilitar a tarefa dos programadores em relação à modelagem e especificação dos protocolos de comunicação, assim como das linguagens de comunicação (trocas de mensagens). Portanto, essas novas ferramentas contribuíram para que detalhes de implementação ficassem transparentes ao projetista, permitindo o foco na ideia central do trabalho pretendido.

Ainda com base nos três primeiros trabalhos da Tabela 5, pode-se observar que todos usam arquivos XML para especificação dos protocolos de comunicação. Assim como foram usados protocolos do padrão FIPA para testes e validações das ferramentas implementadas. A diferença entre a plataforma SACI e COMPOR-E foi justamente a linguagem de comunicação usada, onde COMPOR-E usa FIPA-ACL e SACI utiliza KQML.

O trabalho de Moissa (2001) é bastante interessante, onde as funções de comunicação devem respeitar e se adaptar à arquitetura proposta, inclusive, levando em consideração os requisitos necessários. Isto faz com que seja necessário o desenvolvimento de um protocolo de comunicação próprio, que se adapte às necessidades do sistema proposto. Isto é muito semelhante ao que ocorre com o STI da UNISC, que possui características próprias do ambiente, o que também motiva a criação de um protocolo de comunicação que atenda às necessidades do ambiente em questão.

Já Frigo et al. (2004) descreveram a utilização de agentes cognitivos usados no STI *MathTutor*. Portanto, os autores destacaram que este STI trabalha sobre a arquitetura “troca de mensagens”, onde os agentes se comunicam por mensagens KQML. O *JATLite* e *JESS* são componentes que permitem a troca de mensagens entre os agentes. Portanto, este trabalho relacionado apresenta outra proposta para comunicação entre agentes, contando com duas ferramentas para a criação dos agentes com o objetivo de se comunicarem pela internet.

O trabalho de González et al. (2007) também usa o modelo de comunicação “troca de mensagens”, no entanto, segue as especificações do padrão FIPA. Desta forma, a FIPA-ACL é usada como linguagem de comunicação. Contudo, para o desenvolvimento dos SMAs é utilizada a linguagem de programação Java com o auxílio do *framework* JADE.

Por fim, o mecanismo de comunicação desenvolvido neste trabalho usa a arquitetura de troca de mensagens para realizar a comunicação entre o agente Facilitador e os agentes pedagógicos. Desta forma, as mensagens trocadas são formatadas de acordo com a linguagem de comunicação FIPA-ACL, onde algumas das performativas desta linguagem foram usadas para montar a mensagem. Entretanto, uma linguagem de conteúdo (formato) própria foi utilizada para formatar o conteúdo da mensagem.

Além disso, um protocolo de comunicação próprio foi desenvolvido. No entanto, esse protocolo foi baseado no protocolo FIPA *Contract Net Interaction Protocol* (FIPA, 2002b). Os protocolos baseados no padrão FIPA consideram falhas que podem ocorrer nas trocas de mensagem. Contudo, essas falhas não foram tratadas no protocolo desenvolvido, uma vez que as informações são trocadas internamente dentro do próprio ambiente. Portanto, não há comunicação externa entre agentes.



Neste contexto, é possível destacar diferenças e semelhanças entre os trabalhos estudados e o trabalho desenvolvido. No geral, os objetivos dos trabalhos são diferentes, mas todos estão relacionados à comunicação entre agentes de um SMA.

Ferreira et al. (2004) teve como meta o desenvolvimento da ferramenta COMPOR-E (paleta de componentes integrada à plataforma COMPOR) para a especificação de protocolos de comunicação no padrão FIPA. Esses protocolos ficam armazenados na ferramenta e podem ser exportados para arquivos XML que, por sua vez, podem ser usados para a comunicação entre agentes de outros sistemas.

Por sua vez, o presente trabalho teve como objetivo o desenvolvimento de todo um mecanismo de comunicação entre agentes já implementados em um STI. Portanto, o uso de arquivos XML para troca de mensagens não foi atraente, pois os agentes foram desenvolvidos dentro de um mesmo ambiente, não sendo necessária uma comunicação externa entre eles. Por outro lado, o trabalho de Ferreira et al. (2004) se assemelha com este projeto pelo uso da fundamentação FIPA, tanto para linguagem de comunicação quanto para os protocolos.

Já o trabalho de Hirata et al. (2005) teve como objetivo o desenvolvimento de uma biblioteca para ser usada na plataforma SACI. Com isso, essa plataforma passou a monitorar e reconhecer protocolos de comunicação, permitindo que as informações contidas nos arquivos XML pudessem ser gerenciadas de forma mais amigável, pois “esconde” alguns detalhes de implementação.

Os autores Giménez-Lugo et al. (2006) também trabalharam sobre a plataforma SACI e tiveram como finalidade o desenvolvimento de uma arquitetura para o gerenciamento de protocolos de comunicação. Entretanto, optaram por desenvolver os componentes em Java, já que a plataforma SACI também foi desenvolvida nesta linguagem de programação. Essas novas funcionalidades também possuem o objetivo de fazer com que os projetistas dos protocolos possam se preocupar mais com os objetivos do sistema ao invés da comunicação.

Desta forma, os trabalhos de Hirata et al. (2005) e Giménez-Lugo et al. (2006) se diferenciam deste trabalho, pelo fato da plataforma SACI usar a linguagem de comunicação KQML ao invés da FIPA-ACL. Além disso, diferentemente deste trabalho, a implementação

dos autores também usam arquivos XML para a troca de mensagens entre agentes, o que não é necessário em comunicações realizadas dentro de um mesmo sistema (sem comunicação externa de agentes).

Moissa (2001) trabalhou com o STI *Eletrotutor*, onde o agente Percepção foi inserido no sistema, assim como, um conjunto de funcionalidades para oferecerem suporte aos estados motivacionais identificados por este novo agente. Portanto, o objetivo do autor foi fazer com que o agente Percepção pudesse monitorar toda comunicação realizada entre a interface e o *Eletrotutor*.

Portanto, esse agente Percepção tem finalidade semelhante ao agente Facilitador deste trabalho, pois também monitora os estímulos do ambiente. Além disso, o agente Facilitador participa efetivamente da comunicação entre os agentes pedagógicos do STI. É importante destacar que, o *Eletrotutor* usa linguagem de comunicação e protocolos baseado no padrão KQML, diferentemente deste trabalho. Assim como neste trabalho, o STI *Eletrotutor* também trabalha com troca de mensagens ponto-a-ponto.

O STI usado neste trabalho pode ser usado para o ensino-aprendizagem de qualquer conteúdo, armazenado na base de conhecimento do sistema. Entretanto, diferentemente deste STI, os STIs usados por Frigo et al. (2004) e González et al. (2007) possui conteúdo específico. Segundo Frigo et al. (2004), o STI *MathTutor* é usada para auxiliar os estudantes sobre os fundamentos da estrutura da informação para os estudantes de Engenharia de Controle e Automação. Assim, O STI HALYEN usado por González et al. (2007) é destinado ao ensino de Lógica Matemática, muitos usado por estudantes da computação.

Frigo et al. (2004) não desenvolveram o STI conhecido como *MathTutor*, apenas descreveram a utilização dos agentes conectivos em um STI e, apresentaram as características do sistema estudado. Com foco no STI, pode-se perceber que o *MathTutor* se assemelham por usaram a arquitetura de troca de mensagens, entretanto, se diferenciam pelo fato do STI *MathTutor* usar as padrão KQML para a comunicação de seus agentes.

Segundo González et al. (2007), o HALYEN foi desenvolvido com base na plataforma de desenvolvimento JADE que, segue as especificações FIPA (linguagem e protocolo de

comunicação). Contudo, este trabalho também se baseia no padrão FIPA, entretanto, o STI da UNISC não foi desenvolvido com auxílio de *framework*, como o JADE. Todo o ambiente foi desenvolvido em Java, assim como seus agentes e após este trabalho, a comunicação entre eles. Uma semelhança importante entre os trabalhos está relacionada à figura de um agente centralizador, chamado de agente Facilitador neste trabalho e, agente Coordenador no STI HALYEN. Com isso, ambos os agentes gerenciam as mensagens recebidas e enviadas pelos demais agentes.

## 2.7 Considerações

Os estudos descritos neste capítulo contribuíram para a definição das técnicas utilizadas neste trabalho. Com isso, pôde-se propor o desenvolvimento de um mecanismo de comunicação, onde se optou em usar a arquitetura “troca de mensagens” para a comunicação entre os agentes pedagógicos.

No entanto, nenhuma plataforma para desenvolvimento da interação entre os agentes foi usada, já que os agentes pedagógicos já estão desenvolvidos. Portanto, a troca de mensagens entre os agentes ocorre no próprio ambiente já desenvolvido, não sendo necessário o uso de mecanismos externos ao ambiente para a troca de informação entre os agentes. Mecanismos externos, como arquivos XML, seriam necessários se existisse algum agente externo ao ambiente que, por sua vez, precisasse se comunicar com os agentes internos do sistema.

Definiu-se o desenvolvimento de um protocolo comunicação baseado nos protocolos da FIPA. Nenhum desses protocolos foi utilizado por completo, apenas algumas ideias foram aproveitadas para a definição do protocolo usado neste trabalho. Portanto, pode-se afirmar que as mensagens usadas para negociação entre os agentes foram baseadas no protocolo FIPA *Contract Net Interaction Protocol* (FIPA, 2002b). Como no STI não existe troca externa de informações, alguns cuidados com falha nos protocolos de comunicação da FIPA não foram necessários.

Ainda se optou pela fundamentação FIPA-ACL como linguagem de comunicação, já que ela apresenta vantagem em relação ao KQML, uma vez que o padrão FIPA é utilizado

amplamente no desenvolvimento de SMAs (GLUZ e VICCARI, 2003). Assim, algumas performativas dessa linguagem foram usadas para a definição do formato das mensagens.

Contudo, como a troca de mensagens ocorre dentro do próprio STI, uma linguagem de conteúdo (formatação do conteúdo) foi desenvolvida conforme as necessidades do STI. Desta forma, o conteúdo da mensagem possui um formato próprio, com as informações necessárias para a correta comunicação entre os agentes.

Sobre o estudo de heurística e AGs, pode-se aproveitar a fundamentação do Método da roleta, estudada na seção Método da roleta deste capítulo. Esse método possui o objetivo de selecionar indivíduos que deverão fazer parte de uma futura geração de indivíduos. Entretanto, esse método foi usado para selecionar o agente pedagógico que deverá interagir com o estudante. Neste caso, os agentes Dimi e Dóris são adicionados à roleta fictícia para o sorteio possa ser realizado. O agente escolhido é selecionado para iniciar a interação com o estudante.

Entretanto, o método da roleta será executado apenas quando mais de um agente puder tratar um determinado estímulo. É importante destacar que, o número de interações que o agente já realizou com o estudante é determinante para a definição de quem irá interagir com o estudante. Neste caso, o agente que interagiu menos vezes terá uma probabilidade maior de ser o escolhido. Além disso, a contagem do número de interações considera todas as interações que o agente realizou com o estudante, mesmo nos casos em que apenas um dos agentes possui a habilidade de tratar um determinado estímulo. No contexto deste trabalho, apenas essa possibilidade foi validada. Entretanto, pode-se ajustar o algoritmo de contagem do número de interações para considerar apenas as interações realizadas quando mais de um agente possui a possibilidade de interagir com o estudante.

Com base nos trabalhos relacionados, observou-se que Ferreira et al. (2004) optaram em usar a linguagem de comunicação FIPA-ACL, pois são mais usadas nos SMAs. Inclusive, os protocolos de comunicação do padrão FIPA foram escolhidos para serem especificados e desenvolvidos na ferramenta desenvolvida pelos autores. Já González et al. (2007) desenvolveram seu trabalho na linguagem Java, com o auxílio do Framework JADE, que por

sua vez, segue as especificações FIPA. Conseqüentemente, a linguagem FIPA-ACL também foi utilizada.

Por outro lado, os autores Giménez-Lugo et al. (2006) e Hirata et al. (2005) optaram em usar KQML como linguagem de comunicação. Mesmo assim, os protocolos de comunicação foram especificados em seus trabalhos com base na fundamentação FIPA. Portanto, os conceitos referentes ao padrão FIPA foram aproveitados e utilizados neste trabalho.

A arquitetura de troca de mensagens foi utilizada nos trabalhos dos autores Moissa (2001), Frigo et al. (2004) e González et al. (2007). Essa tecnologia foi utilizada nos trabalhos dos pesquisadores pelo fato da comunicação entre os agentes ser direta. Portanto, os conceitos desta arquitetura também foram aproveitados neste trabalho.

Por fim, foi possível analisar que o trabalho de González et al. (2007) possui o agente coordenador, que intermedia as ações entre os agentes contidos no sistema. Desta forma, este trabalho aproveitou a ideia para desenvolver o agente Facilitador que, por sua vez, possui a função de controlar a comunicação entre os agentes pedagógicos, além de decidir o agente que deve iniciar a interação com o estudante. O próximo capítulo apresenta a metodologia usada no desenvolvimento do trabalho, desde a pesquisa das necessidades do STI até a avaliação do mecanismo de comunicação desenvolvido.

### 3 Desenvolvimento do mecanismo de comunicação

O mecanismo de comunicação desenvolvido é descrito neste capítulo. Desta forma, a próxima seção inicia destacando a metodologia de desenvolvimento usada neste trabalho. Em seguida, as demais seções apresentam ferramentas e tecnologias utilizadas na implementação do mecanismo de comunicação desenvolvido.

#### 3.1 Metodologia

Este trabalho refere-se a uma pesquisa aplicada de natureza descritiva, pois descreve a situação atual do STI em relação ao mecanismo de comunicação entre agentes. Contudo, o estudo também é de natureza exploratória, já que se baseou em estudos teóricos para o desenvolvimento de um mecanismo de comunicação entre agentes pedagógicos.

Segundo os autores Cervo e Bervian (1996), a pesquisa de natureza descritiva visa “... descobrir, com a precisão possível, a frequência como um fenômeno ocorre, sua relação e conexão, com os outros, sua natureza e características, correlacionando fatos ou fenômenos sem manipulá-lo...”.

Já o autor Gil (1991) descreve a pesquisa exploratória como sendo realizada sobre um problema que, no geral, são questões com pouco (ou nenhum) estudo anterior. Desta forma, a principal finalidade de estudos desta natureza é procurar padrões, ideias ou hipóteses. As técnicas geralmente usadas para a pesquisa exploratória são estudos de caso, observações ou análises históricas. Assim, seus resultados no geral, geram dados quantitativos ou qualitativos. Portanto, esse tipo de pesquisa avalia quais conceitos existentes podem ser aplicados a um problema ou se novas teorias devem ser desenvolvidas.

Inicialmente, foi realizado um intenso estudo teórico sobre o STI desenvolvido pelo Departamento de Informática da UNISC, projeto de estudantes e professores da universidade. Essa pesquisa teve por objetivo expandir o referencial teórico já realizado sobre este ambiente. Além disso, essa etapa foi muito importante para que o mecanismo de comunicação pudesse ser modelado de maneira correta, ou seja, de forma que atendesse as necessidades do STI em questão.

Em seguida, este estudo foi expandido para os protocolos de comunicação (interação) em si, assim como as demais fundamentações usadas na presente dissertação de mestrado. O objetivo destas fases foi fundamentar a importância e as contribuições que as pesquisas nesta área proporcionaram para o projeto. Essa etapa foi muito importante, pois determinou as tecnologias e a metodologia usada para a elaboração do mecanismo de comunicação proposto.

Após o devido levantamento das pesquisas bibliográficas relacionadas ao trabalho, foi possível iniciar o seu desenvolvimento. A Figura 22 ilustra o fluxograma executivo da dissertação de mestrado.

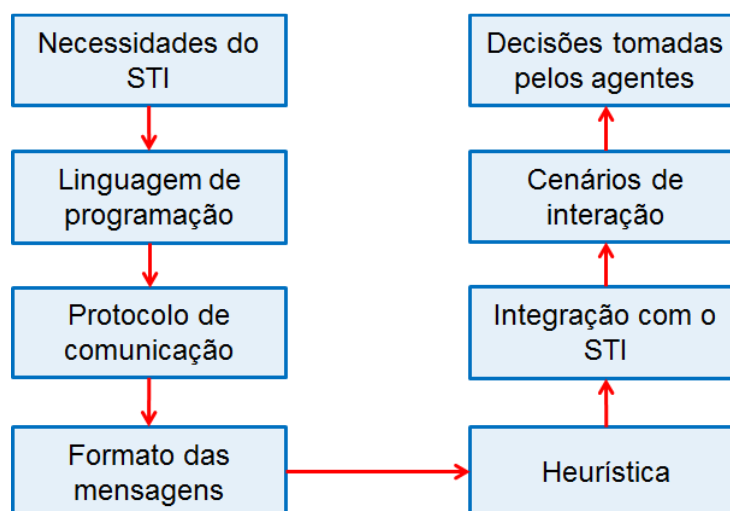


Figura 22. Fluxograma executivo do projeto de pesquisa

Portanto, os procedimentos metodológicos apresentados na Figura 22 foram definidos e descritos, conforme listados a seguir:

- **Necessidades do STI base:** Esse procedimento teve por objetivo definir os problemas a serem solucionados com o uso do mecanismo de comunicação.
- **Linguagem de programação:** Essa etapa teve como finalidade determinar a ferramenta ou a linguagem de programação usada durante o desenvolvimento do mecanismo de comunicação.

- Protocolo de comunicação: Esse procedimento teve como objetivo a definição das regras, estruturas e tecnologias usadas no desenvolvimento do projeto. Foi nesta fase que ficou definida a utilização de um protocolo de comunicação próprio, baseado no padrão FIPA.
- Formato das mensagens: Essa etapa teve como finalidade determinar o formato das mensagens trocadas entre os agentes, ou seja, os parâmetros a serem usados durante a comunicação. Portanto, essa fase definiu as linguagens de comunicação e conteúdo usadas no mecanismo de comunicação.
- Heurística: Esse procedimento teve como objetivo a definição de uma heurística para a tomada de decisão. A função básica desta etapa foi determinar o agente pedagógico que deveria interagir com o estudante.
- Integração com o STI: Essa fase teve como finalidade a integração e implementação de todas as tecnologias definidas nas etapas anteriores. É importante destacar que todo o desenvolvimento foi realizado dentro do STI estudado.
- Cenários de interação: Esse procedimento teve como objetivo a elaboração de cenários para verificar a interação dos agentes. Com isso, foi possível verificar se o resultado das interações ficou de acordo com o que se esperava.
- Decisões tomadas pelos agentes: Essa etapa teve como finalidade avaliar a troca de mensagens entre os agentes. Essa fase visou homologar o mecanismo de comunicação proposto. Um eventual fracasso no resultado deste procedimento significava o reinício do fluxo executivo até a obtenção de sucesso desta etapa.

Pretendeu-se com esta metodologia alcançar os objetivos referentes à proposta deste trabalho. Assim, o mecanismo de comunicação foi desenvolvido baseado nas necessidades do ambiente (STI) usado como base para este trabalho. As necessidades deste ambiente, as tecnologias utilizadas para o desenvolvimento, as etapas do processo de desenvolvimento para



a solução proposta e as estratégias para validação da atuação dos agentes são descritos detalhadamente nas próximas seções deste capítulo.

### 3.2 Desenvolvimento do mecanismo de comunicação

É importante destacar que os agentes pedagógicos vão interagir com o estudante sempre que novos estímulos forem gerados no uso do ambiente, como por exemplo, o estudante pular uma página, voltar para outra página a partir da página de exercícios, permanecer muito tempo ou pouco tempo em uma página. Além desses estímulos, os agentes também podem gerar perguntas aleatórias sobre o assunto (aula) tratado no ambiente, caso o estudante fique algum tempo na página de exercícios.

Portanto, inicialmente foram realizados diversos testes para levantar as necessidades do STI em questão, especialmente nos problemas relacionados à comunicação dos agentes pedagógicos. Os erros encontrados estavam diretamente relacionados à comunicação dos agentes com o estudante, conforme ilustrado na Figura 23.

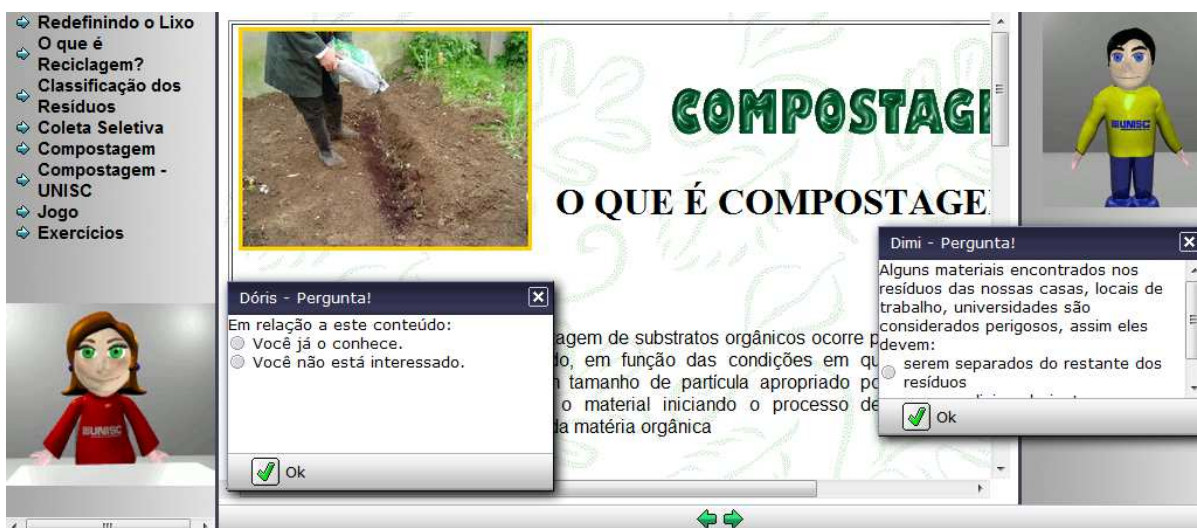


Figura 23. Comunicação de tarefas diferentes e simultâneas

Com base na Figura 23, pode-se observar que a agente tutora Dóris (à esquerda) e o agente companheiro Dimi (à direita) estão se comunicando com o estudante, entretanto, essa interação está ocorrendo ao mesmo tempo e cada um dos agentes pedagógicos está tratando de um assunto diferente. Enquanto a agente tutora está perguntando ao estudante se o mesmo já conhece ou não está interessado no assunto, o agente companheiro está fazendo uma

pergunta referente a esse assunto/aula (compostagem e reciclagem de lixo). Isto pode fazer com que o estudante fique confuso, não sabendo com qual dos agentes ele deve interagir. O mesmo pode ocorrer quando o estudante se depara com a situação ilustrada na Figura 24.



Figura 24. Comunicação de tarefas iguais e simultâneas

Com base na Figura 24, pode-se observar que além dos agentes pedagógicos estarem interagindo com o estudante ao mesmo tempo, ambos os agentes também estão sugerindo a mesma atividade para o estudante (estudo de um material complementar ao assunto). Com isso, pode-se concluir que tarefas simultâneas não são adequadas na comunicação com o estudante.

Essa comunicação simultânea dos agentes pedagógicos com o estudante ocorre justamente pelo fato dos agentes não conseguirem se comunicar entre si. Neste caso, nem o agente tutor e nem o agente companheiro sabem se o estudante já está interagindo com “alguém” no ambiente.

Portanto, a solução encontrada para resolução deste tipo de problema foi o desenvolvimento de um mecanismo de comunicação para os agentes pedagógicos inseridos no ambiente. Essa comunicação entre os agentes pedagógicos também elimina o problema das mensagens/perguntas repetidas e diferentes, pois cada um dos agentes interage separadamente com o estudante.

Por outro lado, durante o uso do ambiente, também foi detectado outro problema em relação à comunicação. Ao executar muitos estímulos no ambiente, que por sua vez encadeava uma série de ações dos agentes pedagógicos, podia ocorrer um empilhamento de perguntas/mensagens, conforme ilustrado na Figura 25.

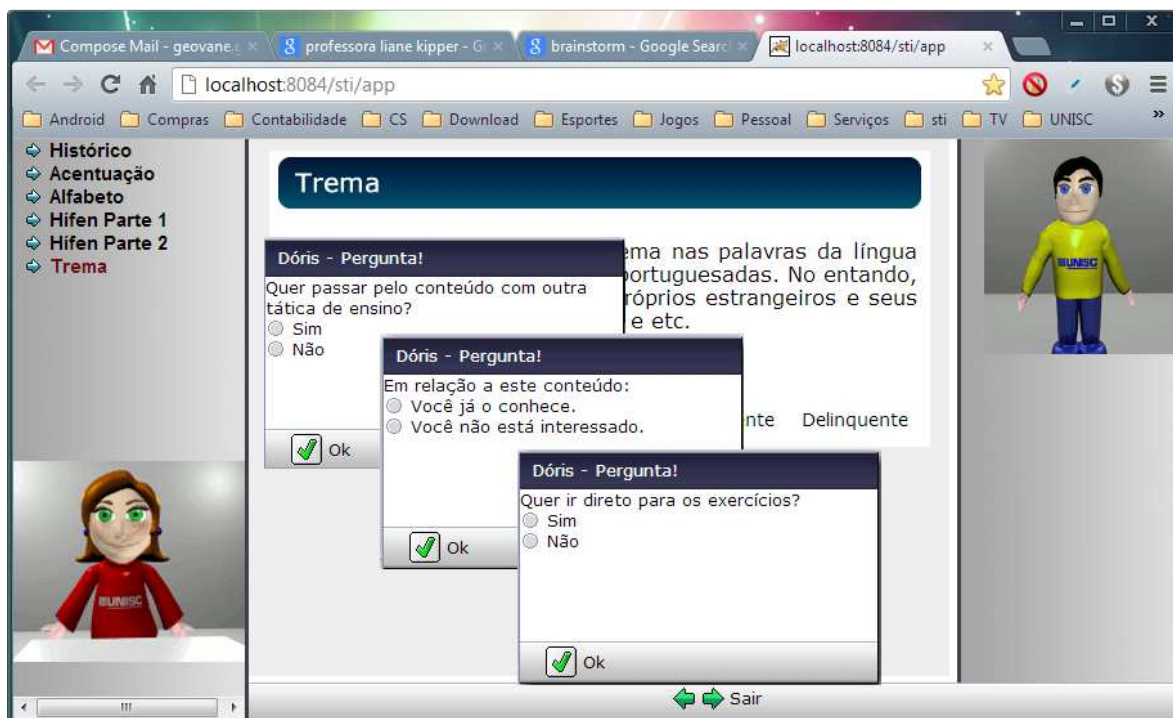


Figura 25. Comunicação com empilhamento de mensagens

Com base na Figura 25 pode-se observar que os diversos estímulos gerados no STI podem resultar em diversas novas comunicações com o estudante, fazendo com que o mesmo tenha que responder várias mensagens, uma atrás da outra. Para piorar, entre as diversas comunicações podiam ocorrer perguntas/mensagens repetidas.

Neste caso, um algoritmo para armazenar apenas os dois últimos estímulos (não repetidos) foi desenvolvido. Assim, o objetivo deste método é evitar o “empilhamento de mensagens”. Esse problema é causado por uma fila extensa de estímulos a serem tratados. Portanto, para evitar que o sistema passe a impressão de ter entrado em *loop* (repetição), os agentes pedagógicos precisariam tratar apenas o “estímulo atual” e o “último estímulo” gerado pelo estudante.

Além dos problemas de comunicação, outros problemas também foram solucionados e especificados nesta dissertação de mestrado, que serão detalhados na Outros desenvolvimentos. Já a próxima seção trata detalhadamente as tecnologias usadas no desenvolvimento do mecanismo de comunicação, como por exemplo, a linguagem de programação e o banco de dados usados no projeto.

### 3.3 Aspectos de implementação

Após a análise das necessidades do STI, a linguagem de programação e as ferramentas para o desenvolvimento do mecanismo de comunicação foram escolhidas. Como o STI e os agentes pedagógicos foram implementados em Java, optou-se pela mesma linguagem de programação. Por este mesmo motivo também se definiu PostgreSQL como o Sistema Gerenciador de Banco de Dados de Objeto Relacional do projeto (SGBDOR). Mais informações sobre as tecnologias usadas são apresentadas nas próximas seções.

#### 3.3.1 Java

Além de ter sido usada no desenvolvimento do STI base do projeto, pode-se destacar que Java é uma linguagem de programação gratuita e amplamente usada em todo o mundo. Segundo site oficial Java:

“Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela *Sun Microsystems* em 1995... Há muitos aplicativos e *sites* que funcionam somente com o Java instalado, e muitos outros aplicativos e sites são desenvolvidos e disponibilizados com o suporte dessa tecnologia todos os dias... A tecnologia Java pode ser encontrada em *laptops*, *datacenters*, *consoles* de jogo, computadores pessoais e científicos, celulares e na Internet.” (JAVA, 2012).

Contudo, Java SE ou JSE (*Java Standard Edition* – Plataforma Java, Edição Padrão) é uma plataforma amplamente utilizada para a programação na linguagem Java. Em termos práticos, JSE consiste de uma máquina virtual (JVM – *Java Virtual Machine*), o qual deve ser utilizado para executar programas Java, juntamente com um conjunto de bibliotecas ou APIs necessários para permitir a utilização de sistemas de arquivos, redes, interfaces gráficos, e outros (JAVA, 2012).

O STI estudado foi desenvolvido para ser acessado através da internet, portanto, usa a tecnologia JEE (*Java Enterprise Edition*– Plataforma Java, Edição Empresarial). JEE é uma

plataforma Java desenvolvida sobre a JSE, cujo principal objetivo é expandir seus serviços para o desenvolvimento de sistemas baseados no modelo cliente-servidor. Com isso, JEE visa facilitar o desenvolvimento de aplicações voltadas para internet. Assim, JEE possui APIs especialmente desenvolvidas para o acesso a servidores, a sistemas de *e-mail*, a banco de dados, entre outros. Por essas características, o JEE foi criado para suportar uma grande quantidade de usuários simultâneos (JAVA, 2012).

Portanto, Java foi a linguagem de programação selecionada para a implementação do mecanismo de comunicação proposto, devido às características especificadas nos parágrafos anteriores e pelo fato do STI já ter sido desenvolvido nesta tecnologia.

### **3.3.2 PostgreSQL**

O PostgreSQL é um SGBFOR de código aberto iniciado em 1986, na Universidade de Berkeley, na Califórnia, nos Estados Unidos da América. A comunidade PostgreSQL destaca que o SGBFOR em questão é o gerenciador de bando de dados de código aberto mais avançado e amplamente usado em todo o mundo (POSTGRESQL, 2012).

Portanto, o STI em estudo usa o PostgreSQL como banco de dados para o projeto. A estrutura das tabelas existentes no ambiente em questão está detalhadamente descrita na próxima seção.

#### **3.3.2.1 Estrutura das tabelas do STI**

O STI usado no projeto possui diversas tabelas, entretanto, não há registros em artigos e trabalhos sobre a estrutura e relacionamento entre elas. Com isso, um estudo foi realizado para definir o modelo ER (Entidade/Relacionamento) das tabelas usadas pelo STI. O modelo em questão foi desenvolvido inicialmente no trabalho de BENETTI (2011). Entretanto, com as atualizações no sistema, as estruturas das tabelas foram alteradas e esse modelo foi atualizado.

O Anexo B ilustra o modelo ER (visão geral) com as tabelas e seus relacionamentos. Logo, pode-se observar que as tabelas com mais relacionamentos são as tabelas Aluno, Aula,

Turma, Pagina e Exercicios. A tabela Aluno possui os dados de cadastro do estudante, assim como informações de seu primeiro acesso, identificação de perfil do estudante, entre outros.

A tabela Pagina possui informações referentes às páginas HTML inseridas no ambiente. Neste caso, é indispensável que o endereço da página seja informado corretamente, caso contrário, esta página não será encontrada pelo sistema. Já as tabelas Aula e Turma definem respectivamente a aula e a turma em que o estudante está matriculado. Desta forma, se o estudante não estiver cadastrado nessas tabelas, ele não poderá acessar as páginas da aula.

Contudo, a tabela Exercicios é responsável por armazenar os exercícios de cada aula. Se houver alguma informação nesta tabela, exercícios referentes a esses dados serão exibidos ao estudante que, por sua vez, deverá respondê-los.

Porém a Figura 26 ilustra a parte do modelo ER que apresenta os relacionamentos com a tabela denominada Aluno. Pode-se observar que essa tabela está relacionada com as tabelas: LogTrocaPerfil, Perfil, Turma, LogTrocaAula, RespostaAluno, RespostaExercicioTexto, LogExercicio e Duvida. As tabelas que iniciam com o prefixo “Log” foram criadas para guardar as ações realizadas pelo estudante durante sua interação com o ambiente.

Ainda com base Figura 26, pode-se observar que a tabela Perfil visa armazenar o perfil do estudante, que é detectado nas interações do estudante com o STI. Por sua vez, a tabela Duvida grava dúvidas do estudante referente à aula. Já as tabelas RespostaExercicioTexto e RespostaAluno guardam as respostas dos estudantes em cada exercício, tanto as questões descritivas quanto as questões objetivas. Ainda é possível destacar que cada estudante precisa estar cadastrado em alguma turma para participar da aula (tabela Turma).

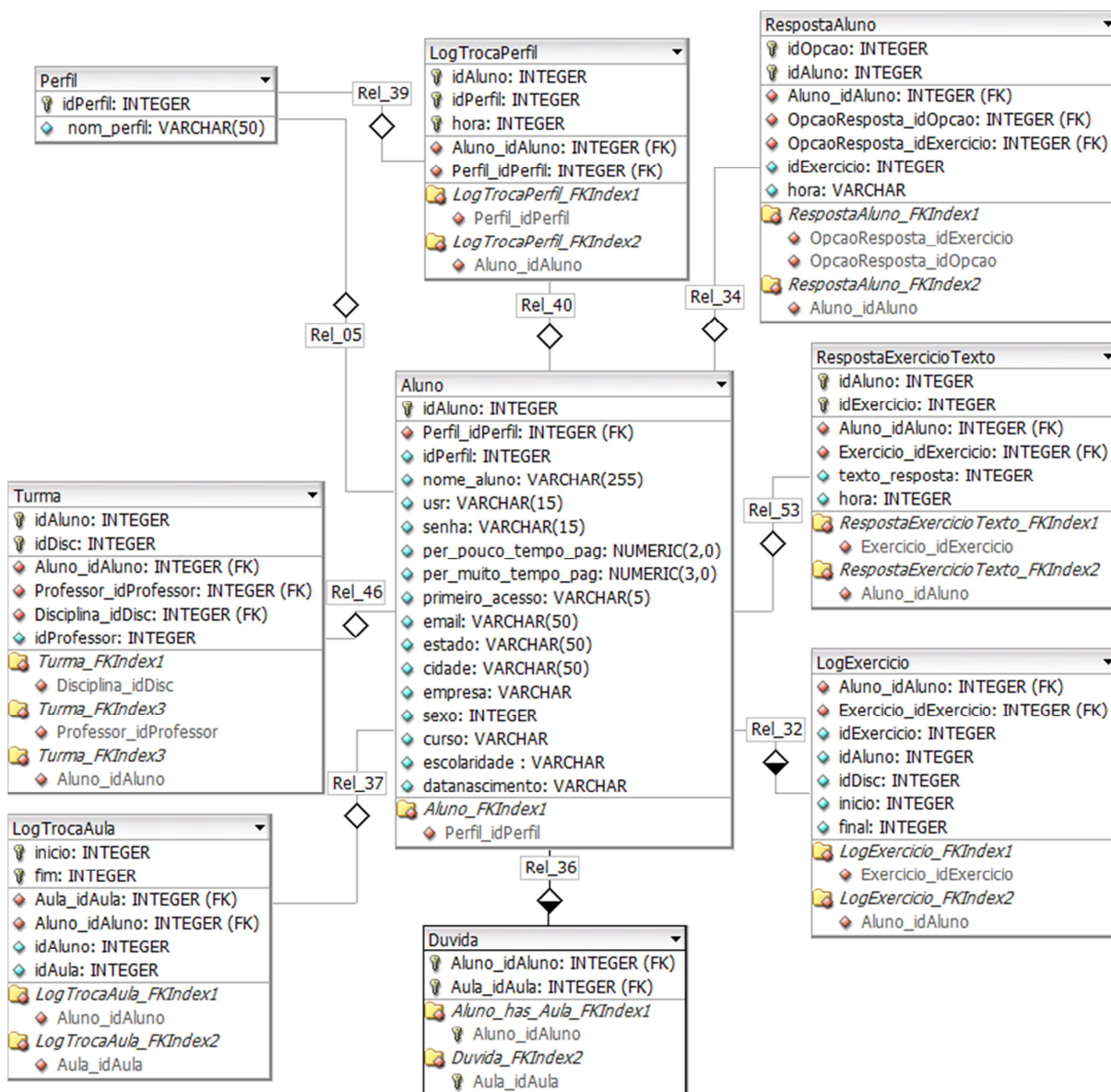


Figura 26. Modelo ER – Tabela Aluno

Por outro lado, a Figura 27 ilustra a tabela Turma e seus relacionamentos. Como já mencionado antes, as tabelas que começam com o prefixo *Log* foram criadas para guardar as ações realizadas pelo estudante dentro do ambiente (LogDicaPagina, LogPagina e logMc – *log* do material complementar). Também, a tabela Professor indica o professor da turma, a tabela Aluno indica o estudante matriculado na turma e a tabela Disciplina indica a disciplina que a turma em questão pertence.

As tabelas MaterialComp e mcAula não estão ligadas diretamente à tabela Turma, mas estão ligadas à tabela logMc e possuem informações sobre o material complementar da aula.

Neste caso, pode-se inclusive cadastrar um *link* para o material complementar. Sempre que o estudante estiver com dúvida ele pode acessar esse material extraclasse.

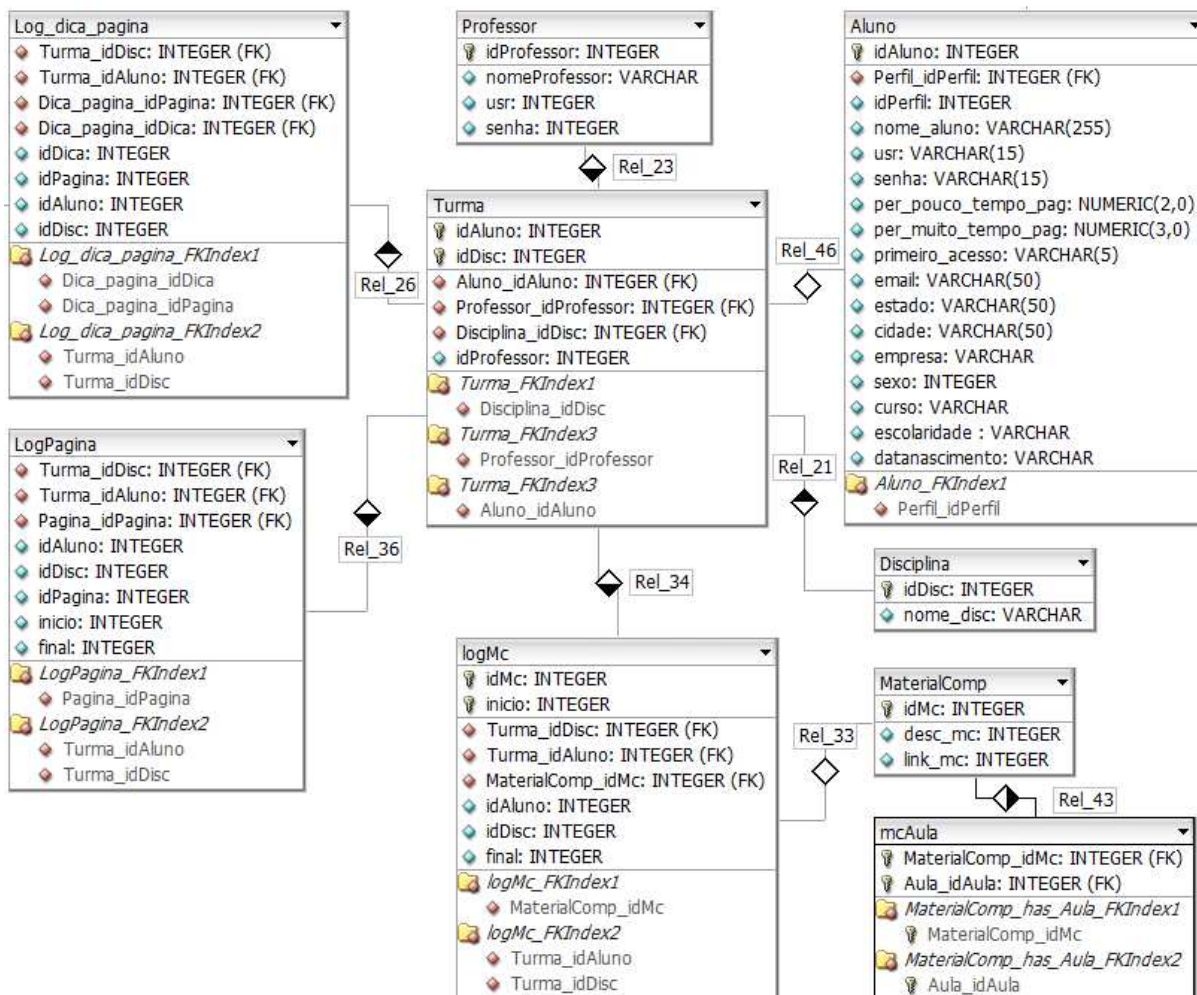


Figura 27. Modelo ER – Tabela Turma

Já a Figura 28 ilustra a tabela Aula e o seu relacionamento com as tabelas Dúvida, Disciplina, Pagina\_Aula, LogTrocaAula, mcAula e Exercício\_Aula. A tabela Aula possui a identificação e o nome da aula, assim como a identificação da disciplina cuja aula em questão está vinculada (tabela Disciplina). Assim como as demais tabelas de log, a LogTrocaAula visa guardar informações de navegação/interação do estudante no ambiente, desta vez em relação às aulas.

Ainda com base nos dados apresentados na Figura 28, pode-se verificar que a tabela Dúvida é usada pelo estudante para o cadastro de alguma dúvida sobre a aula. Além disso, a



tabela mcAula possui informações sobre o material complementar da aula. Logo a tabela Pagina\_Aula contém os dados relativos às páginas HTML a serem exibidas em cada aula. Por fim, a tabela Exercicio\_Aula possui os exercícios a serem apresentados ao estudante.

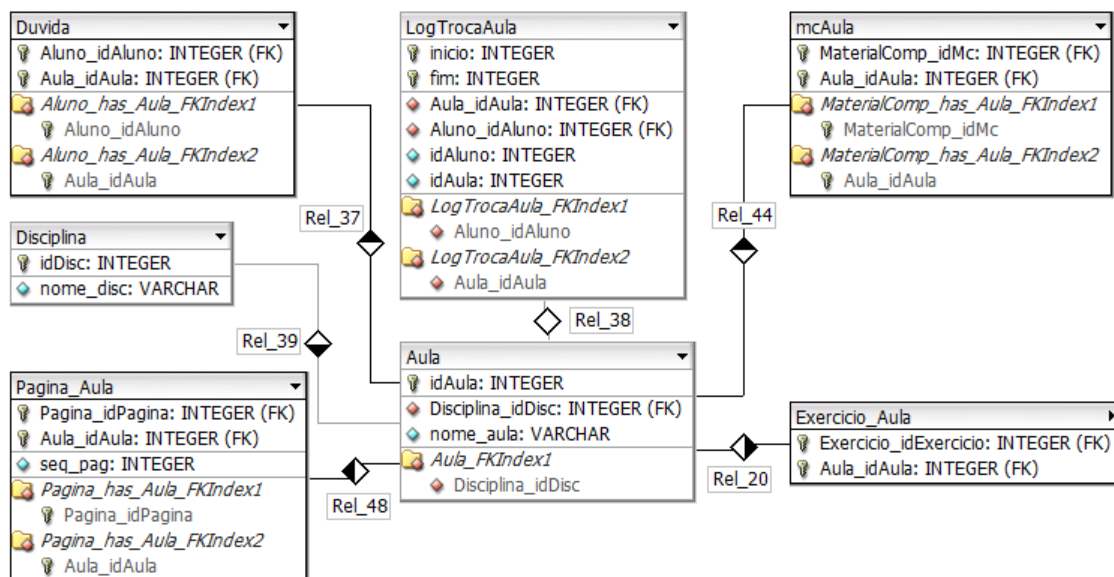


Figura 28. Modelo ER – Tabela Aula

Já o relacionamento da tabela Pagina com as tabelas Tatica, logPagina, Pagina\_Aula e Dica\_pagina pode ser observado na ilustração da Figura 29.

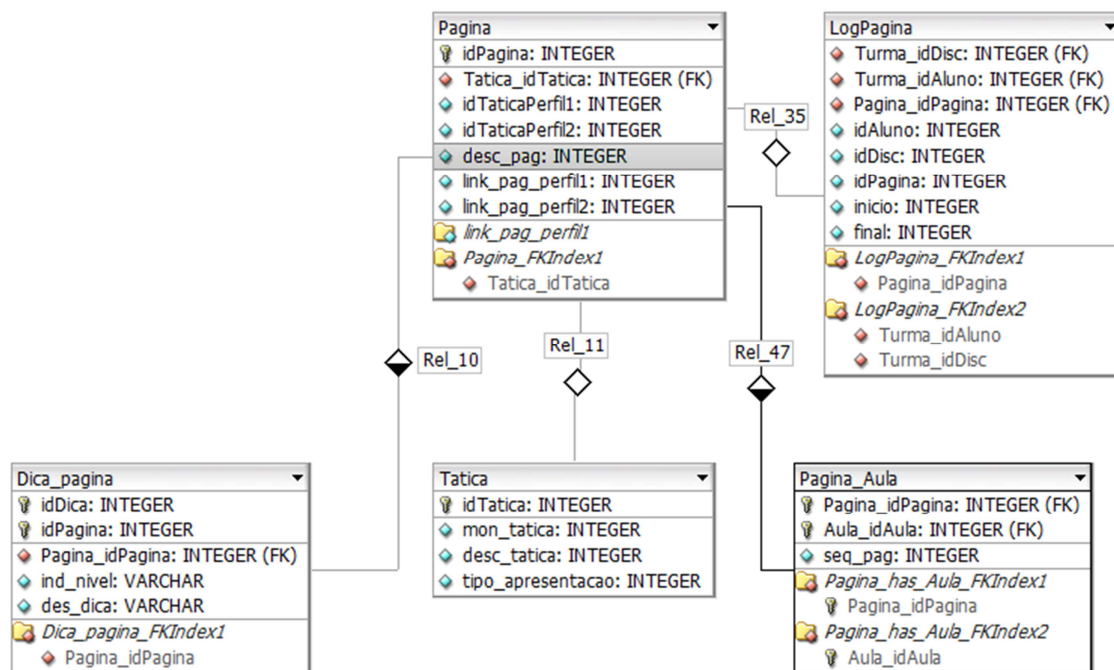


Figura 29. Modelo ER – Tabela Pagina

Ainda com base na Figura 29, o *log* referente à tabela Pagina é gravado na tabela LogPagina. No entanto, a tabela Pagina\_Aula possui a relação entre os dados da aula com as informações da página a ser exibida para o estudante. Também se pode perceber que a tabela Pagina guarda as informações sobre a página, independente da aula que ela pertença. Inclusive, uma mesma página pode ser exibida em aulas e disciplinas diferentes, não precisando refazê-la ou recadastrá-la. Sendo assim, as táticas de ensino cadastradas na tabela Tatica são da página, assim como as dicas da página gravadas na tabela Dica\_Pagina.

Por fim, a Figura 30 apresenta o relacionamento da tabela Exercicio com as tabelas LogExercicio, RespostaExercicioTexto, OpcaoResposta, Exercicio\_Aula e Dica\_Exercicio. O *log* referente à tabela Exercicio é gravado na tabela LogExercicio. Porém, a tabela nomeada RespostaExercicioTexto armazena o texto das respostas do estudante, referente às questões descritivas.

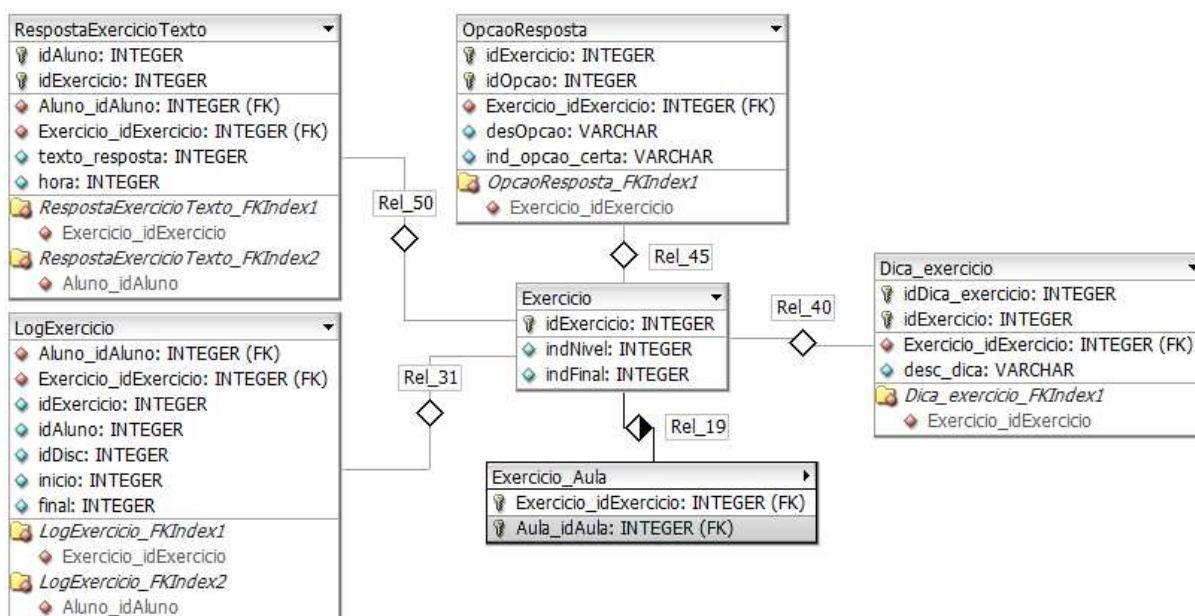


Figura 30. Modelo ER – Tabela Exercicio

Ainda com base nas informações apresentadas pela Figura 30, percebe-se que a tabela OpcaoResposta possui as opções de resposta dos exercícios, inclusive o resultado correto da questão. Já a tabela Dica\_exercicio possui dicas para a resolução dos exercícios. Por sua vez, a tabela Exercicio\_Aula armazena os exercícios que devem ser apresentados em cada aula. Sendo assim, os mesmos exercícios podem ser usados em diferentes aulas.

A próxima seção apresenta o protocolo de comunicação/interação usado no desenvolvimento do mecanismo de comunicação. Para isto, a seção inicia comparando o cenário antigo (antes do mecanismo) com o cenário novo (depois do mecanismo). Além disso, um novo agente foi criado para facilitar a interação entre os agentes pedagógicos.

### 3.4 Mecanismo de comunicação entre os agentes pedagógicos

Para a implementação do mecanismo de comunicação entre os agentes pedagógicos, foi necessário também o desenvolvimento de um protocolo de comunicação/interação entre esses agentes. Porém, para isto, foi necessário o entendimento de como o sistema funcionava. A Figura 31 ilustra o cenário de funcionamento do sistema de comunicação dos agentes antes do desenvolvimento do mecanismo de comunicação.

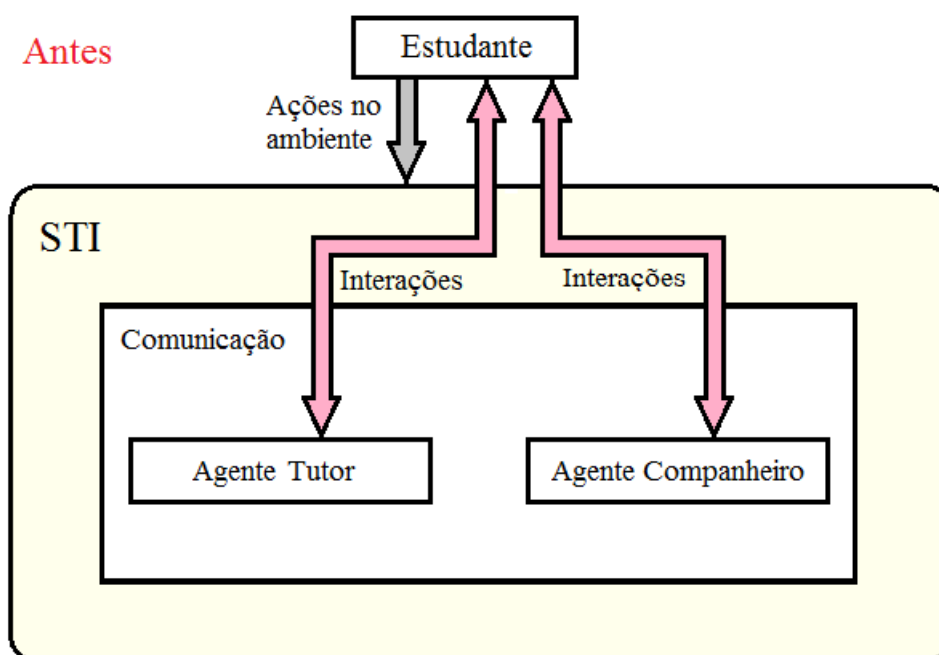


Figura 31. Cenário antigo da comunicação entre os agentes pedagógicos

Pela Figura 31, pode-se observar que o estudante interage com o STI através de ações que ele realiza no ambiente. Como o sistema é reativo, uma ação encadeia uma reação, que no contexto da comunicação, pode ser uma interação com o estudante.

No entanto, neste cenário anterior a comunicação ocorria de forma simultânea, com o agente tutor e o agente companheiro se comunicando com o estudante ao mesmo tempo.

Para que essa simultaneidade na interação dos agentes pedagógicos com o estudante pudesse ser corrigida, foi criado o mecanismo de comunicação ilustrado no novo cenário da Figura 32. Com isso, o objetivo dessa comunicação foi fazer com que os agentes pedagógicos interagissem entre si para decidir quem iria iniciar uma comunicação com o estudante.

Desta forma, o estudante continua gerando ações no sistema, entretanto, apenas um dos agentes pedagógicos se comunica com este estudante a cada vez. Para isto, essa troca de informações entre os próprios agentes ocorre por troca de mensagens, onde cada agente envia e recebe um “pacote” de informações referentes à forma com que a comunicação com o estudante deve proceder.

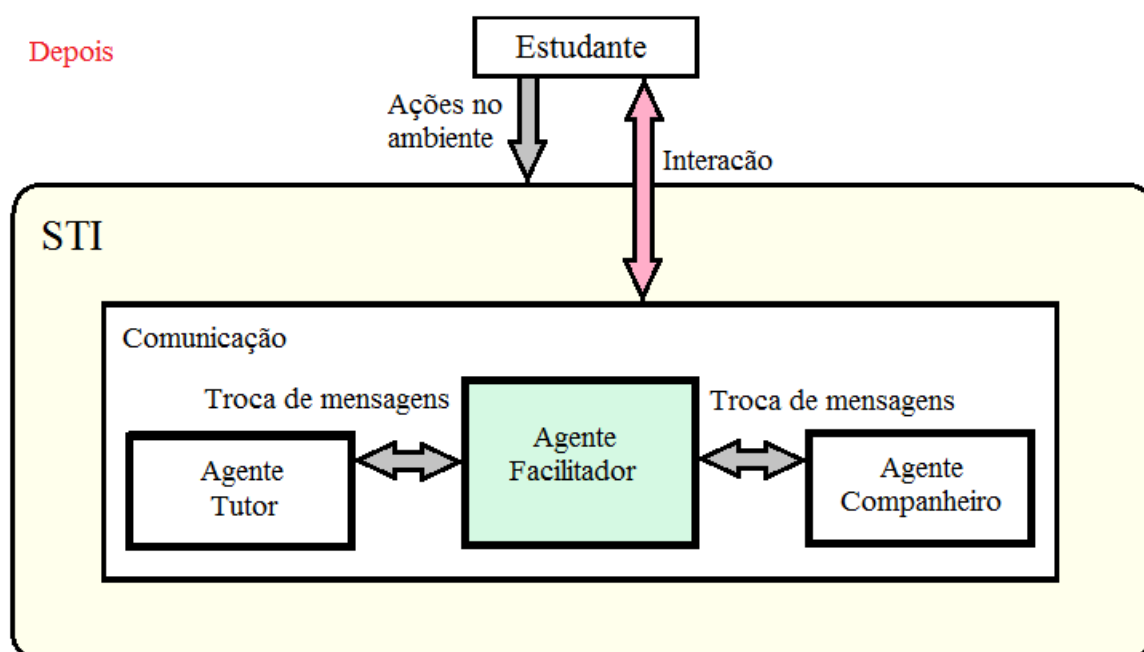


Figura 32. Novo cenário da comunicação entre os agentes pedagógicos

Pode-se observar na Figura 32 que há um novo agente no ambiente, chamado de agente facilitador. Este agente, por sua vez, possui a função de auxiliar os agentes pedagógicos em sua comunicação com o estudante.

Os agentes pedagógicos não interagem diretamente entre si, essa comunicação sempre é facilitada pelo agente facilitador. Sempre que o estudante interage com o ambiente, o agente facilitador é responsável por tratar esse estímulo, trocando informações com os agentes pedagógicos.

Os agentes já existentes no ambiente são agentes reativos, por suas características de reagir a estímulos recebidos por meio das ações dos estudantes durante a interação. Estes estímulos são recebidos e respondidos através do AVA. Assim, decidiu-se desenvolver um processo de comunicação entre os agentes, que não causasse uma alteração na estrutura dos agentes pedagógicos Dóris e Dimi, mantendo uma comunicação através do ambiente que, no caso deste trabalho, foi permitida pelo agente facilitador (responsável pela comunicação com cada agente e pelo processo de decisão). Todos os agentes se comunicam com o agente facilitador que, por sua vez, leva em consideração as mensagens enviadas pelos agentes pedagógicos para a tomada de decisão.

Como os agentes pedagógicos foram desenvolvidos para atuarem dentro do próprio STI, não há necessidade de troca de informação externa ao ambiente para a realização da comunicação entre esses agentes. No entanto, essas interações devem seguir um determinado protocolo de comunicação/interação para a troca de mensagens.

Portanto, os protocolos da FIPA foram estudados para verificar se algum deles poderia ser utilizado no sistema. Contudo, o protocolo que mais se aproximou das necessidades do STI foi o FIPA *Contract Net Interaction Protocol* (FIPA, 2002b). Desta forma, o protocolo desenvolvido foi baseado neste protocolo, entretanto, algumas alterações foram realizadas, principalmente pelo fato dos agentes pedagógicos estarem implementados dentro do mesmo sistema, onde os objetos dos agentes pedagógicos apenas são instanciados.

Com isso, não há possibilidade dos agentes não receberem uma mensagem ou ocorrer alguma falha, assim como previsto no protocolo FIPA *Contract Net Interaction Protocol* (FIPA, 2002b). O protocolo de comunicação/interação desenvolvido é ilustrado na Figura 33.

Pode-se destacar que o agente iniciador é o agente facilitador, enquanto os agentes participantes são os agentes pedagógicos. Neste caso, o agente iniciador da interação manda uma mensagem de consulta para os agentes participantes, com o objetivo de saber se este agente possui a habilidade para tratar o estímulo recebido. Em seguida, os agentes respondem se possuem ou não tal habilidade. Juntamente com essa resposta, também é enviada a informação “número de interações”.

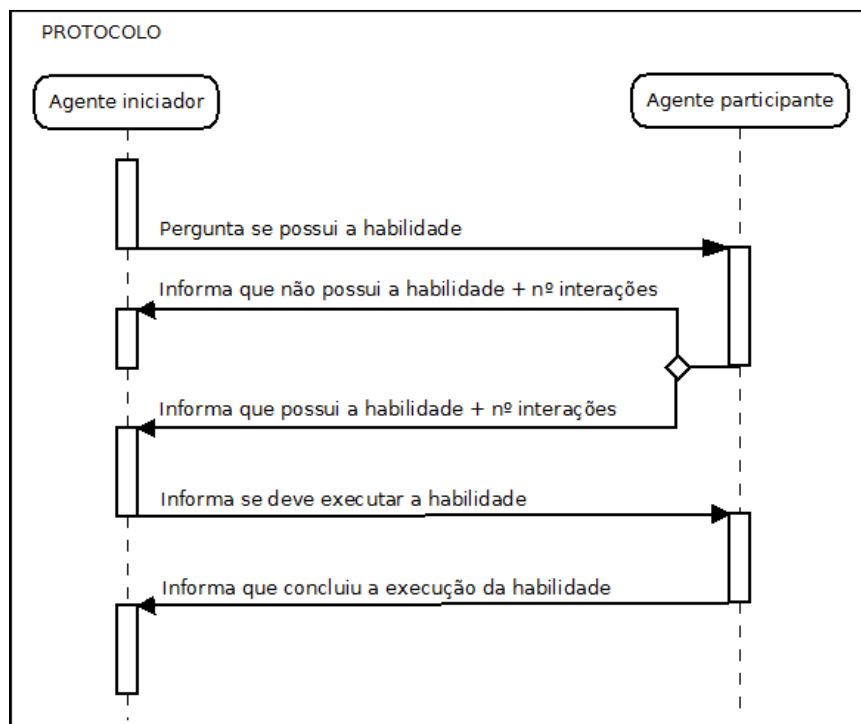


Figura 33. Protocolo de comunicação/interação desenvolvido

Esse número de interações corresponde à quantidade de vezes que o agente que responde a mensagem já interagiu com o estudante. Conseqüentemente, essa quantidade de interações é usada pela heurística, para determinar qual dos agentes deve interagir com o estudante. Assim, o agente que interagiu menos vezes com o estudante terá uma probabilidade maior de conseguir iniciar uma nova interação.

Sendo assim, a habilidade dos agentes pedagógicos está relacionada à capacidade de um agente tratar um determinado estímulo que, por sua vez, pode ser gerado por interações do estudante dentro do ambiente. Os possíveis estímulos a serem gerados são descritos a seguir:

- Pulo de página: estímulo gerado sempre que o estudante não seguir a sequência das páginas, pressionando um item posterior à página seguinte.
- Volta de página: estímulo gerado sempre que o estudante estiver na página de exercícios e resolver voltar para uma página de conteúdo anterior.
- Muito e pouco tempo: estímulos gerados sempre que o estudante ficar muito ou pouco tempo em uma determinada página. O tempo mínimo e máximo de permanência em

cada página é determinado pelo especialista humano (por exemplo, o professor). Entretanto, este tempo deve ser informado dentro do próprio sistema.

- Mensagens aleatórias: estímulo gerado em um determinado intervalo de tempo. Neste caso, se o estudante ficar algum tempo sem utilizar o sistema, algumas mensagens serão exibidas ao estudante, solicitando que o mesmo as responda. Essa habilidade é tratada pela agente Dóris, pois as questões apresentadas para o estudante são específicas ao conteúdo das aulas.

No ambiente em questão, o agente tutor (Dóris) possui a habilidade para tratar todos os estímulos gerados pelo ambiente. Entretanto, o agente companheiro (Dimi) não possui habilidade para tratar as mensagens aleatórias, uma vez que esse agente não possui uma base de conhecimento específica (apenas possui uma base de conhecimento geral). Portanto, as perguntas aleatórias são realizadas com base no conteúdo da aula que está sendo apresentado para o estudante. A próxima seção apresenta as funções dos agentes inseridos no ambiente, onde tais funcionalidades são descritas em forma de fluxogramas.

### 3.4.1 Fluxograma de funcionalidade dos agentes

Pode-se dizer que o mecanismo de comunicação/interação desenvolvido possui quatro passos a serem seguidos pelos agentes (conforme o protocolo de comunicação desenvolvido). O primeiro deles é executado pelo agente facilitador que, por sua vez, recebe um determinado estímulo e executa uma sequência de passos, conforme fluxograma ilustrado na Figura 34.

Antes de iniciar a comunicação com os agentes pedagógicos, foi implementado um algoritmo para evitar o “empilhamento de mensagens”, conforme descrito na **Erro! Fonte de referência não encontrada.** deste capítulo. A ideia do algoritmo é manter armazenado o “estímulo atual”, que está sendo tratado no momento da interação. Além disso, o algoritmo também guarda o “último estímulo”, que é executado logo após a conclusão do “estímulo atual”. Com isso, evitam-se as diversas interações dos agentes pedagógicos (pois não havia limite para o número de mensagens). Portanto, a primeira parte deste algoritmo pode ser verificada no início da Figura 34.

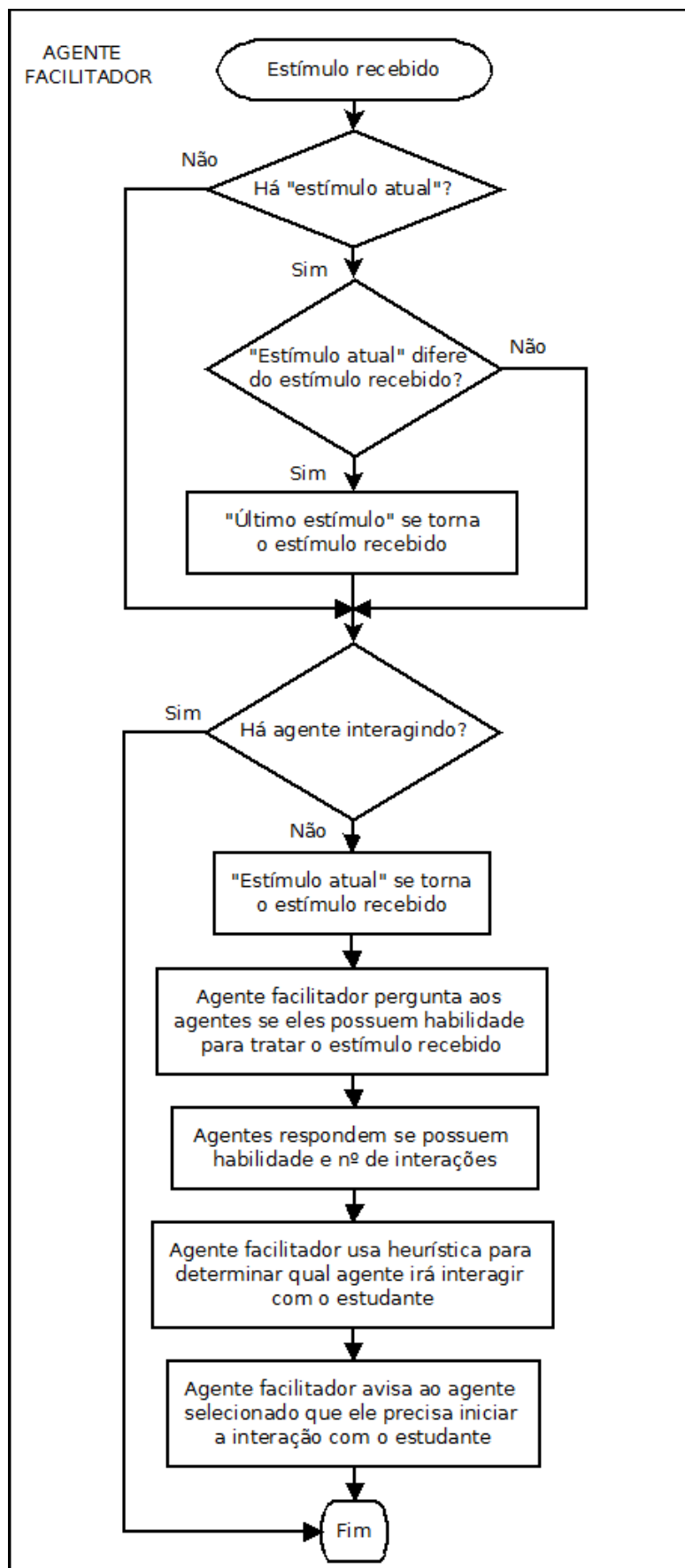


Figura 34. Fluxograma do agente facilitador – estímulo recebido



Ainda com base na Figura 34, é importante destacar que o algoritmo também obriga que o “estímulo atual” e o “último estímulo” sejam diferentes, evitando, assim, repetição de mensagens. Além disso, se o “último estímulo” estiver esperando para ser executado e nesse mesmo momento ocorre mais um estímulo, o “último estímulo” é eliminado e esse novo estímulo se torna o “último estímulo”.

Desta maneira, quando o agente facilitador recebe um estímulo, verifica-se a existência de um “estímulo atual”. Caso afirmativo, observa-se se este estímulo recebido é diferente do “estímulo atual” e, se for, o “último estímulo” se torna o estímulo recebido. Entretanto, pode-se verificar que, se não há “estímulo atual” ou se esse “estímulo atual” for igual ao estímulo recebido, nenhuma ação ocorre e o algoritmo é executado normalmente.

Em seguida, verifica-se se algum agente está interagindo com o estudante, caso haja, o algoritmo é encerrado (neste caso, o protocolo de comunicação/interação não é executado, justamente para evitar interação simultânea dos agentes pedagógicos). Entretanto, se não há interação, o estímulo recebido passa a ser o “estímulo atual” e a primeira parte do protocolo de comunicação/interação é executado. Logo, o agente facilitador inicia a comunicação com os agentes pedagógicos, enviando uma mensagem para saber quais dos agentes possuem a habilidade para tratar o “estímulo atual”.

Após receber todas as respostas dos agentes, o agente facilitador usa uma heurística para determinar qual dos agentes pedagógicos deve iniciar a interação com o estudante. Essa heurística será descrita com maiores detalhes na 3.6 deste capítulo. Esta heurística utiliza o número de interações entre o estudante e os agentes pedagógicos para determinar qual desses agentes deve interagir com o estudante. Por fim, o agente facilitador envia uma mensagem ao agente escolhido, informando que ele deve iniciar a interação com o estudante.

A Figura 35 ilustra o algoritmo executado pelos agentes pedagógicos no momento em que recebem uma mensagem de consulta do agente facilitador, que pergunta a cada agente pedagógico se eles possuem uma determinada habilidade. Portanto, ao receberem a pergunta, os agentes pedagógicos verificam se eles possuem tal habilidade em sua lista de habilidades. Em seguida, verificam o número de interações que tiveram com o estudante e enviam uma mensagem informando ambos os dados ao agente facilitador.

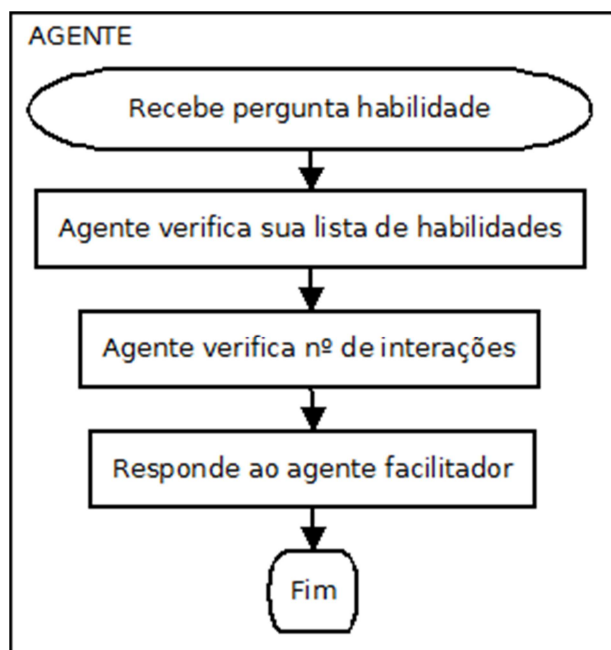


Figura 35. Fluxograma do agente pedagógico – lista habilidade

No momento em que o agente pedagógico selecionado para interagir com o estudante recebe a mensagem para executar essa interação, o algoritmo ilustrado na Figura 36 entra em execução. Desta forma, o agente escolhido trata o estímulo recebido e inicia uma interação com o estudante, avisando o agente facilitador que a interação começou.

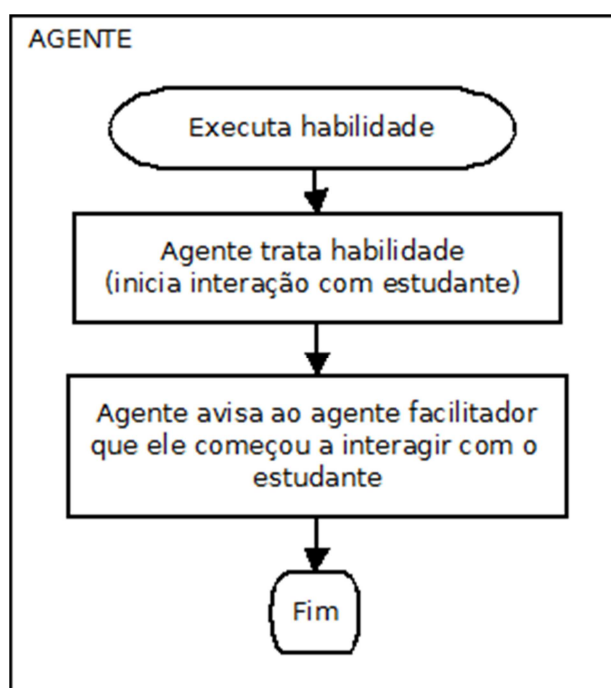


Figura 36. Fluxograma do agente pedagógico – executar habilidade

Finalmente, o algoritmo ilustrado na Figura 37 é executado para atualizar o *status* de interação dos agentes pedagógicos com o estudante. Contudo, ao receber uma atualização, o agente facilitador atualiza a situação da interação. Sendo assim, o algoritmo criado para evitar o empilhamento de mensagens continua a ser executado. Portanto, verifica-se se algum dos agentes pedagógicos está interagindo com o estudante e, caso afirmativo, a execução deste fluxograma é encerrada.

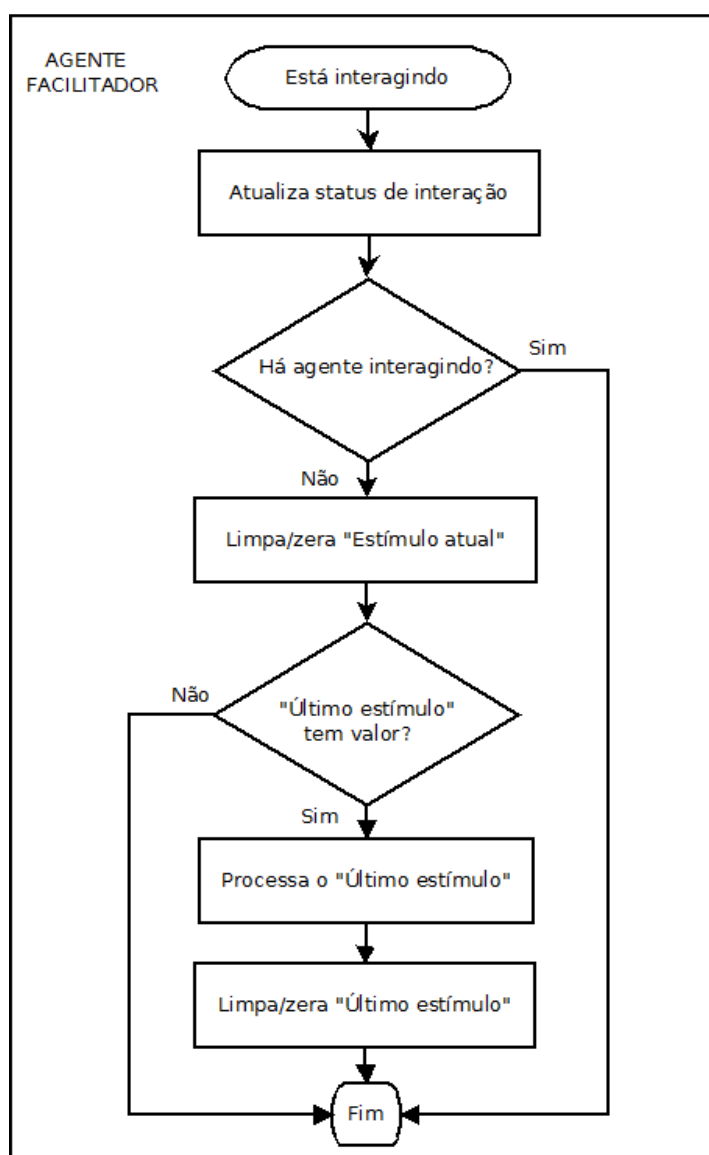


Figura 37. Fluxograma do agente facilitador – atualiza interação

Por outro lado, se não há interação com o estudante, o algoritmo limpa o “estímulo atual” e verifica se o “último estímulo” possui algum estímulo para ser executado. Caso não possua mais estímulo a ser executado, o algoritmo é encerrado. Entretanto, se ainda possuir

um estímulo a ser tratado, o “último estímulo” é executado e limpo, uma vez que o “estímulo atual” passa a ser o “último estímulo”. É importante destacar que, se alguma mensagem for enviada para informar ao agente facilitador que não há mais interações com o estudante, o primeiro ciclo do protocolo de comunicação/interação terá sido encerrado. Sendo assim, um novo estímulo ou a execução do estímulo que estava esperando na fila (“último estímulo”) inicia todo o processo novamente.

É importante destacar que o protocolo de comunicação/interação desenvolvido troca informações entre os agentes. No entanto, para que isso possa ocorrer, os agentes devem trocar mensagens entre si (agente facilitador e agentes pedagógicos). Sendo assim, essas mensagens devem possuir um formato padrão (único), para que ambos os agentes possam compreender.

Desta forma, a próxima seção descreve o formato das mensagens usadas nas trocas de informação entre os agentes. Além disso, é necessário que ambos os agentes saibam tratar as mensagens e, para isto, é necessário que uma mesma linguagem de conteúdo seja usada nas mensagens e compreendida pelos agentes envolvidos.

### **3.5 Formato das mensagens**

As mensagens que são trocadas entre os agentes do ambiente seguem o protocolo de comunicação/interação desenvolvido, entretanto, essas mensagens precisam ter um formato em comum, para que todos os agentes possam entender o seu conteúdo. As estruturas das mensagens usadas no desenvolvimento do mecanismo de comunicação estão descritas com maiores detalhes a seguir.

#### **3.5.1 Linguagem de comunicação usada**

Através do estudo realizado sobre as linguagens de comunicação, optou-se em utilizar uma estrutura de mensagens baseada na linguagem de comunicação FIPA-ACL. Além disso, a FIPA-ACL é usada com frequência na comunicação entre agentes e é homologada pela FIPA (FIPA, 2013).

Não são gerados arquivos no formato XML para a troca de informação entre os agentes, uma vez que os agentes estão implementados dentro de um mesmo sistema, não sendo necessária uma troca externa de informação. Por este mesmo motivo nenhuma plataforma ou *framework* para agentes, como JADE, foi utilizado no desenvolvimento da comunicação, já que os agentes já estavam modelados e desenvolvidos no sistema. A estrutura usada para o formato das mensagens é ilustrada na Figura 38.

Remetente	Destinatário
Performativa	
Conteúdo	
Linguagem	Ontologia

Figura 38. Formato das mensagens

Com base nos dados apresentados na Figura 38, pode-se perceber que seis parâmetros são usados para a composição da mensagem. É importante destacar que todas as informações que compõem a mensagem devem ser informadas para que a mensagem seja válida. Uma breve descrição dos parâmetros é listada a seguir:

- Remetente: Contém a informação do agente que está enviando a mensagem.
- Destinatário: Possui a informação do agente que deve receber a mensagem.
- Performativa: Define a mensagem que é enviada para o destinatário, ou seja, determina se a mensagem é uma consulta ou se é uma informação. Uma consulta pode ser usada para perguntar algo para um determinado agente, como por exemplo, se o agente em questão possui a habilidade para tratar certo estímulo. Uma informação, por sua vez, pode ser enviada para ordenar que o destinatário realize uma ação (executar o tratamento de um estímulo) ou simplesmente informar algo a um agente (término de uma interação com o estudante).
- Conteúdo: Possui o conteúdo, ou melhor, a informação que a mensagem deve levar até o destinatário da mensagem. Por exemplo, o conteúdo da mensagem pode possuir a informação de quantas vezes um determinado agente já interagiu com o estudante.

- Linguagem: Contém a informação de que linguagem é usada para processar o conteúdo da mensagem. Esta linguagem de conteúdo pode ser uma linguagem própria ou uma linguagem já existente. Apenas é indispensável que ambos agentes conheçam a mesma linguagem.
- Ontologia: Possui a informação de que ontologia está sendo usada na mensagem. É usada para dar significado à expressão contida no conteúdo. Podem-se usar ontologias diferentes para tratar de assuntos distintos. Por exemplo, pode-se usar uma ontologia para tratar assuntos gerais do ambiente (referente ao sistema em si) e, outra ontologia para tratar de assuntos específicos do ambiente (referente ao conteúdo das aulas).

Para a melhor compreensão, o Quadro 4 apresenta um exemplo prático de mensagem trocada entre os agentes, onde o agente Facilitador consulta a agente Dóris para saber se ela possui a habilidade de tratar um determinado estímulo. Tanto a linguagem de conteúdo quanto a ontologia são específicas para o STI em questão (próprias do ambiente). Portanto, apenas mensagens que possuam a linguagem e a ontologia esperada pelo sistema serão tratadas.

Quadro 4 – Exemplo de uma mensagem de consulta do agente Facilitador

Remetente: Agente Facilitador
Destinatário: Agente Dóris
Performativa: Consulta
Conteúdo:
Tipo: Consulta habilidade e número de interações
Código da habilidade: 1 (pulo de página)
Código da mensagem aleatória:
Número de interações:
Possui habilidade:
Agente interagindo: Não
Número da mensagem: 12
Linguagem: UNISC-STI-LANGUAGE
Ontologia: UNISC-STI-ONTOLOGY

Fonte: do autor

Desta forma, pode-se observar como o conteúdo da mensagem é formado. O número da mensagem corresponde ao código da mensagem, neste caso, essa é a 12ª mensagem que é trocada entre o agente Facilitador e os agentes pedagógicos. Ainda com base no conteúdo da mensagem, pode-se verificar que o tipo de mensagem trocada entre os agentes é uma consulta de habilidade e número de interações. Essas informações são essenciais ao agente Facilitador, para que o mesmo possa definir o agente pedagógico que deve iniciar uma interação com o estudante.

Além disso, o conteúdo da mensagem apresenta o código 1 (um) código da habilidade. Portanto, o agente Facilitador quer saber se a agente Dóris possui a habilidade de tratar um pulo de página (além do número de interações que ela já fez com o estudante). O código da mensagem aleatória não é informado, pois não é esse estímulo que foi gerado no ambiente. Assim, também é possível observar que as informações de “possui habilidade” e “número de interações” não foram informados no conteúdo da mensagem, pois o agente Facilitador não interage com o estudante.

Por fim, se essa mensagem está sendo enviada para a agente Dóris, significa que o estudante não está interagindo com os agentes no momento em que a comunicação entre eles está sendo realizada.

### **3.5.2 Linguagem de conteúdo usada**

Nenhuma linguagem de conteúdo existente foi usada para examinar o conteúdo das mensagens trocadas entre os agentes, pois como já mencionado anteriormente, os agentes já estão implementados dentro de um mesmo sistema. Portanto, desenvolver um mecanismo complexo para leitura e escrita de uma linguagem de conteúdo dentro do STI seria inviável e desnecessário, uma vez que os agentes se comunicam apenas internamente no ambiente.

Com base nisso, foi usada uma fundamentação própria como linguagem de conteúdo. Neste caso, o conteúdo das mensagens segue uma estrutura condizente com essa convenção. Portanto, o conteúdo das mensagens possui as seguintes informações:

- Número da mensagem: Indica o código da mensagem e serve para determinar se uma mensagem faz parte de uma mesma comunicação. Portanto, todas as mensagens com o mesmo número pertencem ao mesmo ciclo de comunicação do protocolo.
- Código da habilidade: Contém o número da habilidade que representa o tratamento de um determinado estímulo.
- Código da mensagem aleatória: Indica o número da mensagem/pergunta aleatória que deve ser apresentada para o estudante. Neste caso, o “código da habilidade” deve ser igual ao código de tratamento das mensagens aleatórias (perguntas realizadas em um determinado intervalo de tempo).
- Possui habilidade: Sinaliza se o agente remetente possui habilidade para tratar certo estímulo.
- Número de interações: Indica o número de interações que o agente remetente já teve com o estudante.
- Agente interagindo: Indica ao agente destinatário se ainda há interação com o estudante.
- Tipo da mensagem: Determina qual mensagem que está sendo enviada ou recebida.

Há quatro tipos de mensagens disponíveis no protocolo de comunicação:

- Consulta habilidade: Usada para perguntar ao agente destinatário se ele possui habilidade para tratar o estímulo enviado no conteúdo da mensagem.
- Informa habilidade e número de interações: Usada para que o agente remetente possa informar ao agente destinatário o número de interações que ele já fez com o estudante e, se esse agente remetente possui a habilidade para tratar o estímulo enviado no conteúdo da mensagem.
- Executa habilidade: Usada para que o agente remetente possa informar ao agente destinatário que ele foi escolhido para executar a habilidade em questão, ou seja, o agente remetente ordena que o agente destino trate o estímulo passado no conteúdo da mensagem.
- Informa interação: Usada para informar ao agente destinatário que a interação com o estudante iniciou ou que a interação já terminou.



É importante destacar que todas as mensagens possuem informação no parâmetro conteúdo. As mensagens devem ser trocadas de forma completa, com os seis parâmetros da linguagem de comunicação e com todos os dados do conteúdo preenchido. Caso não haja informação a ser enviada em um determinado campo, o mesmo deve ser enviado vazio.

Para a melhor compreensão, o Quadro 5 apresenta um exemplo prático de mensagem trocada entre os agentes, onde a agente Dóris envia uma mensagem para o agente Facilitador. Essa mensagem é uma resposta à consulta realizada no Quadro 4 da Linguagem de comunicação usada, deste mesmo capítulo. Com isso, pode-se observar que a performativa dessa mensagem é de informação (tipo de mensagem), onde a agente Dóris pretende informar ao agente Facilitador se ela possui a habilidade para tratar o estímulo de pulo de página, além do número de interações que ela já fez com o estudante. Assim, também se pode observar que a linguagem de conteúdo e a ontologia são específicas para o STI em questão (próprias do ambiente).

Quadro 5 – Exemplo de mensagem de informação da agente Dóris

Remetente: Agente Dóris
Destinatário: Agente Facilitador
Performativa: Informação
Conteúdo:
Tipo: Informação de habilidade e número de interações
Código da habilidade: 1 (pulo de página)
Código da mensagem aleatória:
Número de interações: 8
Possui habilidade: Sim
Agente interagindo: Não
Número da mensagem: 12
Linguagem: UNISC-STI-LANGUAGE
Ontologia: UNISC-STI-ONTOLOGY

Fonte: do autor

Ao analisar o conteúdo da mensagem, pode-se observar que o número da mensagem é 12 (doze), mesmo código apresentado no Quadro 4 da Linguagem de comunicação usada,

deste mesmo capítulo. Isto demonstra que ambas as mensagens pertencem ao mesmo ciclo de mensagens trocadas entre os agentes (mesmo estímulo). Além disso, a agente Dóris irá informar ao agente Facilitador que ela possui a habilidade de tratar o estímulo consultado, assim como irá avisar que ela já interagiu o com estudante 8 (oito) vezes, entretanto, não está mais interagindo no momento com ele.

Por fim, a agente Dóris também envia ao agente Facilitador o código da habilidade e o código da mensagem aleatória. Essas informações são reenviadas ao agente Facilitador para que ele saiba que as informações enviadas pela agente Dóris realmente condizem com que o foi consultado (solicitado) por ele.

Portanto, após definir o protocolo de comunicação/interação, o formato das mensagens e o conteúdo, foi necessário determinar uma heurística para a tomada de decisão que, por sua vez, deveria escolher um dos agentes pedagógicos para iniciar uma interação com o estudante. A próxima seção descreve a heurística desenvolvida no presente trabalho.

### **3.6 Heurística**

Quando apenas um dos agentes possui a habilidade de tratar o estímulo gerado no STI, não é necessária a utilização de métodos para determinar o agente pedagógico que deverá iniciar uma interação com o estudante. Entretanto, no momento em que mais de um agente pedagógico possui a habilidade de tratar o estímulo, é indispensável o uso de alguma técnica para a tomada de decisão. Neste caso, uma heurística foi implementada para auxiliar o agente facilitador na escolha do agente pedagógico que deverá interagir com o estudante.

Assim, um algoritmo baseado no método da roleta, descrito na 2.3.1.1 do Fundamentação teórica, foi desenvolvido para contribuir na escolha do agente pedagógico que deverá interagir com o estudante. Para isto, foi usado o número de interações que cada um desses agentes fez com o estudante. Assim, a informação do número de interações é fornecida pelos próprios agentes. A ideia foi fazer com que o agente que interagiu menos vezes com o estudante pudesse ter mais chances de interagir com ele no futuro (na próxima interação).

Desta forma, fazendo uma analogia com a roleta física, o algoritmo implementado visou dar mais “casas” para o agente que interagiu menos vezes com o estudante pudesse ter mais chance de se consagrar “vencedor”. Por outro lado, o agente com menos “casas” possui uma probabilidade menor de ser escolhido. Entretanto, mesmo tendo um número menor de cavidades, o agente que interagiu mais vezes com o estudante pode ser eleito mais uma vez.

Por exemplo, se a agente tutora (Dóris) interagiu seis vezes com o estudante e o agente companheiro (Dimi) interagiu apenas duas vezes, o método desenvolvido inverte tais valores, fazendo com que Dimi possua seis cavidades a seu favor e Dóris possua duas. A Figura 39 ilustra como os agentes ficariam distribuídos na roleta fictícia.

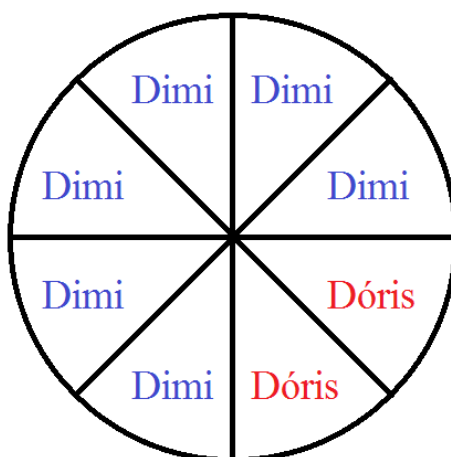


Figura 39. Exemplo de destruição de agentes no método da roleta

### 3.7 Integração com o STI

Após a definição da estrutura do protocolo de comunicação/interação, formato das mensagens e heurística para tomada de decisão, foi necessário o desenvolvimento de cada uma dessas funcionalidades. Juntamente com o protocolo foram implementados o conceito de habilidades, o agente facilitador e o algoritmo para eliminar o empilhamento de mensagens.

Portanto, a integração com o STI teve como objetivo a implementação e união de todas as tecnologias estudadas anteriormente. O primeiro passo no desenvolvimento do mecanismo de comunicação foi determinar quais habilidades cada agente pedagógico atenderia. Em seguida, o conceito de habilidade foi implementado e as habilidades de cada agente foram atribuídas aos mesmos. Logo depois, o agente facilitador foi integrado ao

sistema, sendo ele o responsável por receber o estímulo gerado pelo estudante. Contudo, no próprio agente facilitador também foi desenvolvido o algoritmo para evitar o empilhamento das mensagens.

Antes do desenvolvimento protocolo de comunicação/interação foi necessário realizar uma implementação interna para melhorar as classes dos agentes pedagógicos que, por sua vez, estavam implementados de forma separada (uma classe para cada agente). Portanto, os agentes foram implementados em apenas uma classe com todos os métodos comuns a eles. Em seguida, cada objeto da classe agente (agente pedagógico) foi instanciado separadamente e suas particularidades foram setadas. A vantagem dessa implementação foi o fato de não ser mais necessário ficar repetindo métodos e, conseqüentemente, futuros agentes pedagógicos podem ser inseridos no ambiente apenas instanciado novos objetos desta mesma classe.

Ainda no agente facilitador, foi implementada a heurística para a tomada de decisão. É importante lembrar que esta heurística apenas é executada se mais de um agente possuir a habilidade de tratar o estímulo gerado pelo estudante no ambiente.

O formato das mensagens e o seu conteúdo foram simples de serem implementados, pois apenas devem ser enviados de um agente para outro. O Quadro 6 apresenta um exemplo de mensagem montada dentro do STI. Essa mensagem forma um “pacote” e é colocada na fila de entrada do destinatário que, por sua vez, deve desmembrar essa mensagem e realizar o processamento necessário. O agente destinatário pode descompactar a mensagem, pois ambos agentes possuem a mesma linguagem de conteúdo.

Ainda com base nas informações apresentadas no Quadro 6, pode-se perceber que a mensagem a ser enviada é uma consulta de habilidade a ser realizada pelo agente facilitador. Com isso, essa mensagem será enviada a um dos agentes pedagógicos, seguindo a linguagem de conteúdo e ontologia adotada no sistema. Além disso, são enviadas as informações para compor o conteúdo, como o código da mensagem, do estímulo a ser tratado e da mensagem aleatória. Como o agente facilitador não interage com o estudante, os campos que indicam se o agente possui habilidade e se ele está interagindo são falsos, assim como o número total de interações com o estudante sempre é zero.

Quadro 6 – Exemplo de formato de mensagem usado no STI

```

ConteudoMensagem wConteudoMensagem = new ConteudoMensagem();
wConteudoMensagem.setTipo(MensagemFIPAACL.CONSULTA_HABILIDADE);
wConteudoMensagem.setCodHabilidade(codAviso);
wConteudoMensagem.setCodMsgAleatoria(codMsgAleatoria);
wConteudoMensagem.setPossuiHabilidade(false);
wConteudoMensagem.setNumeroInteracoes(0);
wConteudoMensagem.setAgenteInteragindo(false);
wConteudoMensagem.setNroMsg(nroMsg);

MensagemFIPAACL wMensagemFIPAACL = new MensagemFIPAACL();
wMensagemFIPAACL.setSender(Constants.AGENTE_FACILITADOR);
wMensagemFIPAACL.setReceiver(i++);
wMensagemFIPAACL.setPerformative(MensagemFIPAACL.QUERY_REF);
wMensagemFIPAACL.setContent(wConteudoMensagem);
wMensagemFIPAACL.setLanguage(MensagemFIPAACL.UNISC_STI_LANGUAGE);
wMensagemFIPAACL.setOntology(MensagemFIPAACL.UNISC_STI_ONTOLOGY);

```

Fonte: do autor.

Por outro lado, o desenvolvimento do protocolo de comunicação/interação foi mais complexo. Inicialmente esse protocolo foi implementado de forma sequencial, onde o agente facilitador enviava aos agentes pedagógicos uma pergunta, com o propósito de saber se os agentes em questão possuem a habilidade de tratar o estímulo recebido. Depois disto, o agente facilitador ficava esperando todos os agentes pedagógicos responderem para ordenar que o agente escolhido executasse o tratamento do estímulo em questão.

Portanto, para evitar o sequenciamento das mensagens, assim como a espera dessas mensagens de resposta, foram desenvolvidas *threads* com o objetivo de monitorar as filas de entrada dos agentes, tanto nos agentes pedagógicos como no agente facilitador. A Figura 40 ilustra o funcionamento da troca de mensagens entre os agentes.

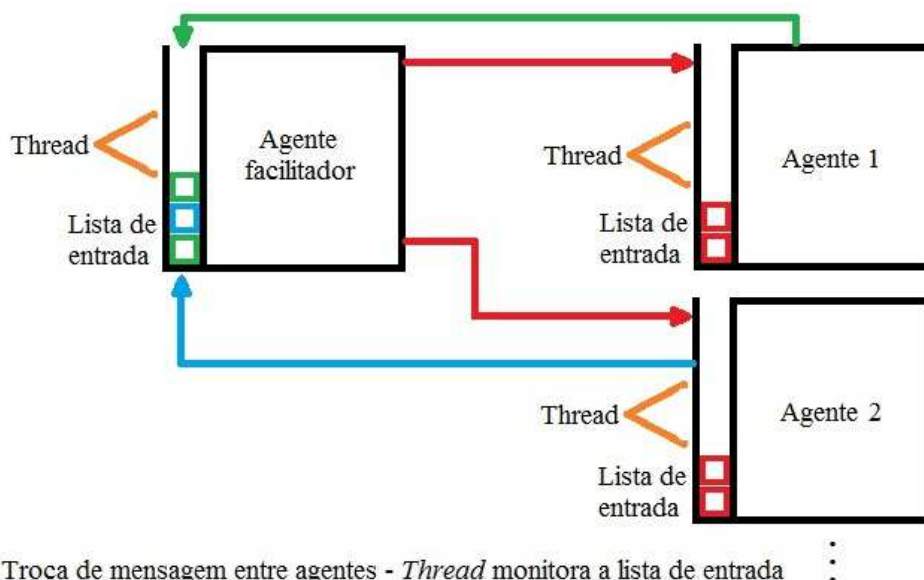


Figura 40. Funcionamento das *threads* no protocolo desenvolvido

Com base na Figura 40, pode-se observar que o agente remetente coloca a mensagem na fila de entrada do agente destinatário. Por sua vez, a *thread* implementada para monitorar a fila de entrada dos agentes fica responsável por detectar as mensagens contidas nesta fila. Contudo, ao perceber que há uma mensagem na fila de entrada do agente, a *thread* encaminha essa mensagem ao método responsável por ler esse “pacote” e realizar o seu processamento. Caso seja necessário algum retorno, essa mensagem é igualmente colocada na fila de entrada do agente destinatário (agente remetente da mensagem de origem).

Como a execução dessas mensagens ocorre de forma simultânea (concorrente), o agente facilitador não precisa ficar esperando que todos os agentes pedagógicos respondam suas mensagens para dar continuidade ao sistema. Com isso, esse mecanismo pode ser usado mesmo que existam diversos agentes pedagógicos inseridos no ambiente.

A próxima seção é responsável por descrever detalhadamente os cenários de interação montados para realizar a primeira homologação do mecanismo de comunicação desenvolvido. Para isto, foram montados casos de teste para testar partes do sistema, assim como o mecanismo de comunicação e apresenta os resultados obtidos.

### 3.8 Cenários de interação

Cenários de interação foram elaborados para homologar o mecanismo de comunicação desenvolvido. Inicialmente, foram montados cenários de interação para testar os erros levantados quando do estudo das necessidades do STI: comunicação simultânea dos agentes pedagógicos que, conseqüentemente apresentavam conteúdos diferentes ou repetidos ao estudante; e o empilhamento das mensagens (gerados pelos diversos estímulos provocados no ambiente). Por fim, cenários de interação também foram criados para validar a técnica desenvolvida para a comunicação dos agentes, além de cenários para validar o funcionamento de outras partes do sistema.

O cenário apresentado na Figura 41 demonstra que apenas um dos agentes interagiu com o estudante, pois apenas o agente Dimi exibiu uma mensagem ao estudante, informando que, se ele continuar pulando páginas pode ter problemas para compreender o conteúdo.

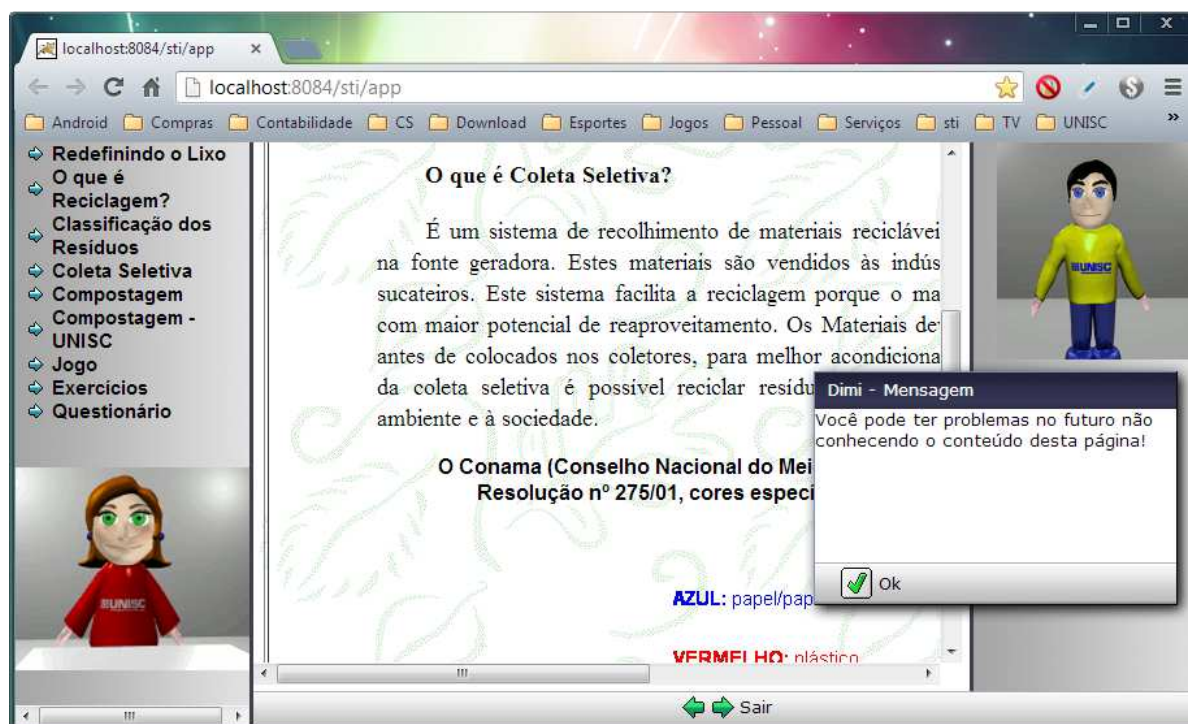


Figura 41. Cenário 01 – interação coordenada - Dimi

A Figura 42 ilustra a interação do estudante com a agente tutora. Pode-se observar que apenas Dóris realizou um pergunta ao usuário do sistema. Desta forma, em ambos os casos

(Figura 41 e Figura 42), os agentes interagiram de forma coordenada, um de cada vez. Portanto, esse tipo de interação resolveu os problemas resultantes de mensagens simultâneas.

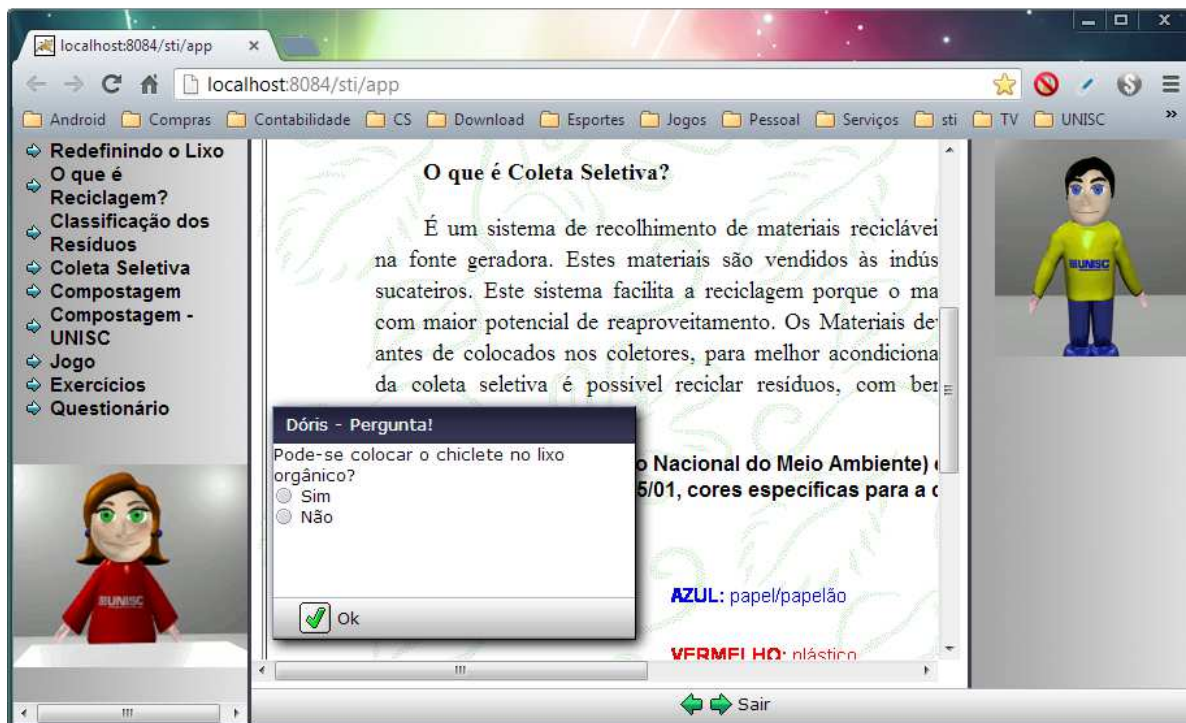


Figura 42. Cenário 01 – interação coordenada - Dóris

O problema de empilhamento de mensagens também foi destacado nas **Erro! Fonte de referência não encontrada.** Neste caso, as interações do estudante no sistema geram estímulos a serem tratados, entretanto, se diversos estímulos forem gerados, os mesmos ficam em uma fila até que sejam processados. Contudo, se o estudante não interagir com os agentes pedagógicos, todas as mensagens vão sendo empilhadas, o que pode causar a sensação de que nunca terminam (caso o estudante resolva iniciar a interação com os agentes pedagógicos após o empilhamento das mensagens).

Com base nisso, outro cenário de interação foi elaborado, onde diversos estímulos foram gerados no ambiente, sem que as interações com os agentes pedagógicos fossem sendo realizadas. É importante destacar que, o algoritmo implementado prevê o armazenamento do estímulo atual (interação que está em execução) e do último estímulo. Com isso, os estímulos intermediários são ignorados, justamente para evitar que várias mensagens entrem em uma fila de execução.



Para isto, vários cliques foram realizados no menu de opções do sistema e, em alguns casos, o ambiente não foi usado por um determinado período de tempo. Assim, estímulos resultantes de pulo de página; retrocesso da página de exercícios; pouco e muito tempo em uma página foram gerados e enviados para o agente facilitador, que é responsável por gerenciar essa sequência de ações executadas.

Contudo, o cenário ilustrado na Figura 43 é resultante do primeiro estímulo gerado no ambiente que, por sua vez, é apresentado quando o estudante permanece muito tempo em uma mesma página. Entretanto, neste cenário de interação, o mais importante era verificar se o sistema iria apresentar apenas dois casos de interação distintos com o estudante.

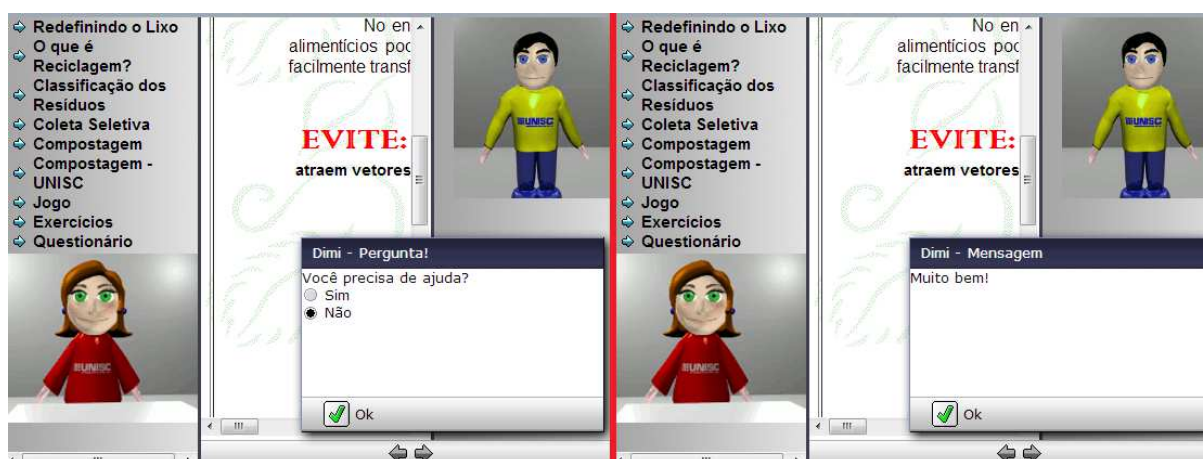


Figura 43. Cenário 02 – empilhamento de mensagens – estímulo 1

Desta forma, pode-se observar que a imagem do lado esquerdo exibe o agente Dimi realizando uma pergunta ao estudante, onde ele quer de saber se o mesmo precisa de ajuda. No entanto, a imagem da direita reporta a mensagem de incentivo ao estudante “Muito bem”, apresentada quando este estudante informa ao agente pedagógico que não precisa de ajuda. Essa primeira interação é encerrada no momento que o estudante pressiona o botão “ok”.

Após o encerramento da primeira interação com o estudante, o cenário de interação continua a ser executado, pois é necessário observar o comportamento do sistema em relação aos demais estímulos que gerados pelo estudante. Assim, a interação ilustrada na Figura 44 é apresentada como resultado do último estímulo gerado. Portanto, para que o algoritmo esteja correto, novas interações não podem ser apresentadas após a conclusão desta interação.

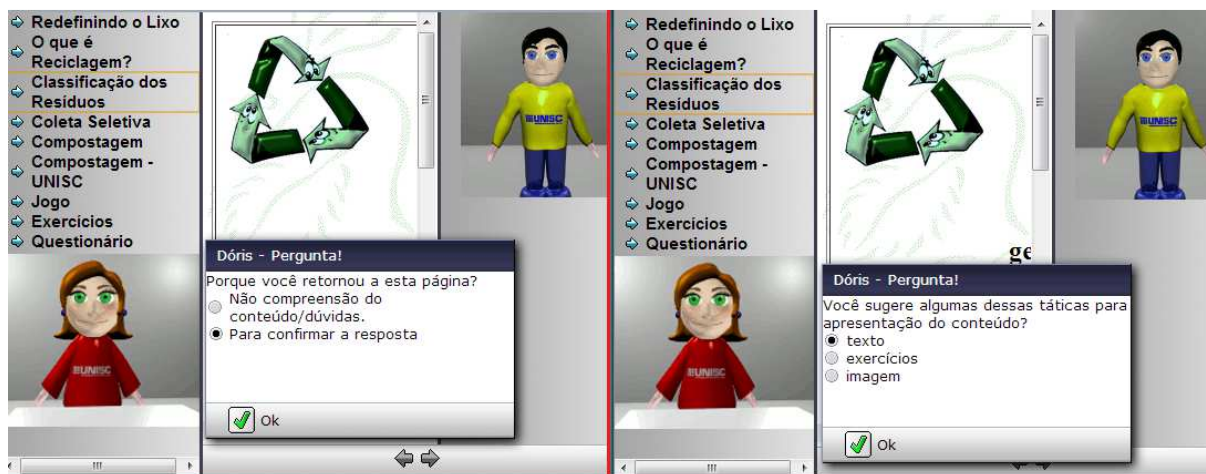


Figura 44. Cenário 02 – empilhamento de mensagens – estímulo 2

Ainda com base na Figura 44, pode-se observar que a imagem da esquerda apresenta uma pergunta realizada pela agente tutora, onde a mesma pergunta ao estudante porque ele retornou uma página. Portanto, pode-se perceber que a agente Dóris está tratando o estímulo referente ao retrocesso de página, que é diferente do primeiro estímulo gerado no ambiente (muito tempo em uma página).

Assim, ao marcar que o retorno se deu para confirmar uma resposta, a caixa de diálogo da direita é exibida, para que o estudante sugira se ele prefere visualizar o conteúdo em outra tática de ensino (imagem ou texto) ou se ele prefere ir direto aos exercícios. Ao optar pela opção “texto”, a imagem da Figura 45 é apresentada.



Figura 45. Cenário 02 – empilhamento de mensagens – fim da interação

Ainda com base nos dados apresentados na Figura 45, pode-se observar na imagem da esquerda, a caixa de diálogo de incentivo “Muito bem”. Ao pressionar o botão “ok”, a imagem do lado direito é exibida ao estudante, sem interação. Com isso, foi possível verificar que esse algoritmo funcionou conforme previsto, uma vez que, no máximo, dois estímulos diferentes foram tratados pelos agentes pedagógicos.

Após a execução dos dois primeiros cenários de interação apresentados anteriormente, novos cenários foram montados para validar o funcionamento das demais técnicas inseridas no ambiente. Tais testes são listados a seguir:

- Cenário 03 - Habilidades: Como ambos os agentes pedagógicos possuem conhecimento geral, eles tratam de assuntos relacionados ao uso do ambiente. Entretanto, além do conhecimento geral, a agente tutora também possui conhecimento específico, podendo tratar de assuntos relacionados à aula. Neste caso, cenários de teste foram executados para avaliar se cada agente atendeu apenas as habilidades que foram cadastradas para ele. Com isso, foi possível verificar que ambos os agentes interagem com o estudante, no entanto, sempre era a Dóris que fazia as perguntas referentes ao conteúdo da aula, atestando o correto funcionamento do sistema desenvolvido.
- Cenário 04 - Heurística: Como ambos os agentes pedagógicos interagem com o estudante no ambiente, é necessário que isso ocorra de forma organizada. Portanto, a heurística implementada pretende fazer com que os dois agentes possam interagir, evitando que apenas um deles fique se comunicando com o estudante. Através dos cenários de teste criados, foi possível observar que dificilmente um mesmo agente tratava duas interações seguidas com o estudante. Desta forma, pode-se concluir que a heurística usada funcionou conforme o esperado.
- Cenário 05 - Protocolo de comunicação/interação e *threads*: Também foi possível verificar que as interações dos agentes pedagógicos com o estudante ocorriam de forma completa, ou seja, ao ser gerado um estímulo, o agente logo inicia a interação com o estudante. Entretanto, se ao final da comunicação um novo estímulo fosse gerado, uma nova comunicação se iniciava normalmente. Em nenhum momento uma interação parou ou deixou de iniciar. Esse teste foi importante para verificar se todos os passos do protocolo seriam seguidos de forma correta, além disso, o uso de *threads*

faz com que diferentes processos sejam executados de forma simultânea/concorrente. Neste último caso, um controle inadequado poderia comprometer o funcionamento do protocolo de comunicação desenvolvido. Com isso, pode-se concluir que o mecanismo de comunicação suportou muito bem as *threads* e o protocolo de comunicação.

Outros cenários foram montados para que diferentes funcionalidades do STI pudessem ser homologadas, como por exemplo, os navegadores (browsers) e dispositivos que suportam as tecnologias usadas no ambiente. O ambiente foi testado e o seu funcionamento foi homologado para os seguintes *browsers*: *Google Chrome* (GOOGLE, 2012), *Safari* (SAFARI, 2012), *Netscape Navigator* (NETSCAPE, 2012), Navegador Opera (OPERA, 2012) e *Mozilla Firefox Web Browser* (MOZILLA, 2012).

Além disso, o ambiente também foi testado em *smartphones*, conforme ilustrado na Figura 46. O primeiro aparelho exibido na imagem executa o Sistema Operacional (SO) *iOS* (APPLE, 2013), enquanto o segundo dispositivo possui o SO *Android* (ANDROID, 2013). Ainda com base nos dados apresentados na Figura 46, pode-se perceber que o sistema foi executado corretamente em ambos os SOs. Entretanto, não foram usados aplicativos para execução do STI. Contudo, ambiente foi executado no *browser Safari* instalado em ambos os aparelhos.

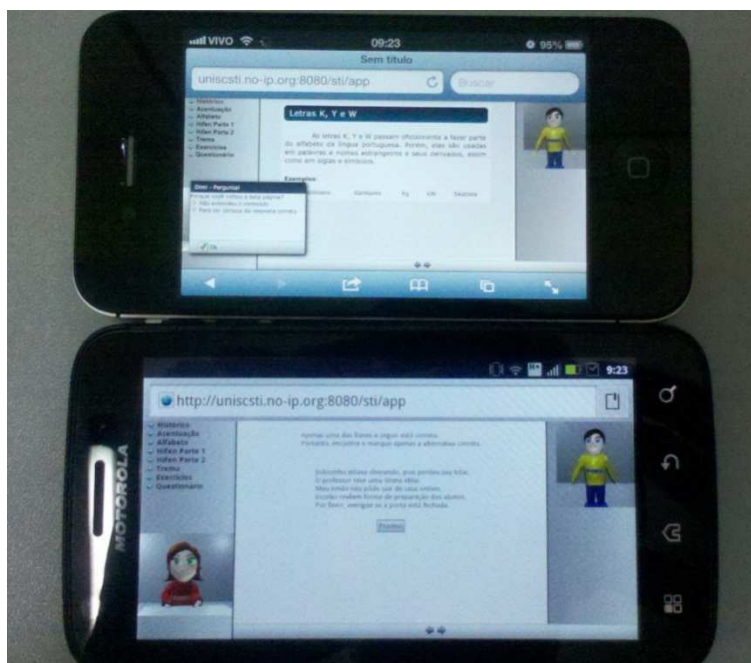


Figura 46. Cenário 06 - *Smartphones* executando o ambiente

### 3.9 Decisões tomadas pelos agentes

Alguns procedimentos foram realizados para que fosse possível analisar as decisões tomadas pelos agentes pedagógicos. Pela atuação de bolsistas no projeto de pesquisa no qual a dissertação está inserida, foi adicionado à base de conhecimento do ambiente uma nova aula, referente ao Novo Acordo Ortográfico da Língua Portuguesa. A partir disso, dez exercícios foram criados e incluídos no banco de dados para que os estudantes pudessem respondê-los.

Portanto, a Figura 47 ilustra o conteúdo apresentado ao estudante quando o mesmo opta por realizar esta aula. Assim, pode-se visualizar o menu de opções no canto esquerdo da imagem, onde as aulas estão distribuídas nos seguintes conteúdos: Histórico, Acentuação, Alfabeto, Hífen Parte 1, Hífen Parte 2, Trema, Exercícios e Questionário. Por outro lado, no centro da tela, pode-se observar que o histórico é apresentado como conteúdo principal da aula.

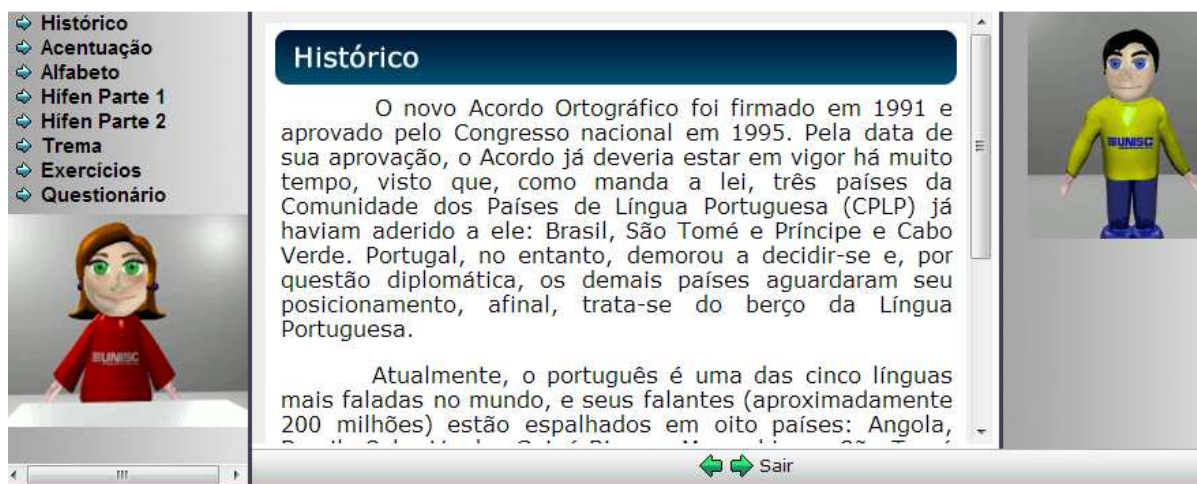


Figura 47. Aula sobre o novo acordo ortográfico – histórico

Já a Figura 48 ilustra o conteúdo que é apresentado quando a opção “Hífen parte 1” é selecionada pelo estudante. Desta forma, pode-se observar que apenas o conteúdo das aulas é alterado, conforme modelagem das páginas HTML adicionadas na base de conhecimento do sistema. Ainda é importante destacar que, o menu de opções com os seus respectivos *links* para as páginas HTML são cadastrados e exibidos de forma dinâmica. Desta forma, alterações no código-fonte do sistema não são necessárias para a inclusão das páginas.

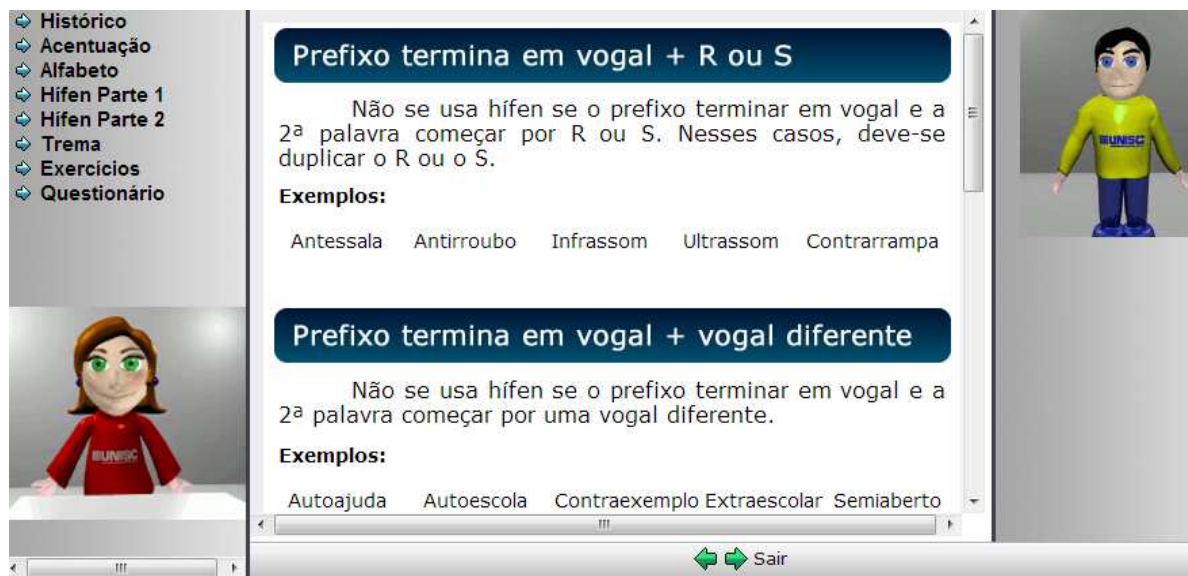


Figura 48. Aula sobre o novo acordo ortográfico – hífen parte 1

Por sua vez, a Figura 49 ilustra a página de exercícios, onde o estudante deve escolher a resposta correta dentre as cinco opções que lhe foram dadas. A cada pergunta respondida, o usuário deve pressionar o botão “pronto” para prosseguir até a próxima questão. No final dos dez exercícios o seu aproveitamento é exibido na tela. A lista com todos os dez exercícios e suas respectivas respostas está disponível no Anexo A.

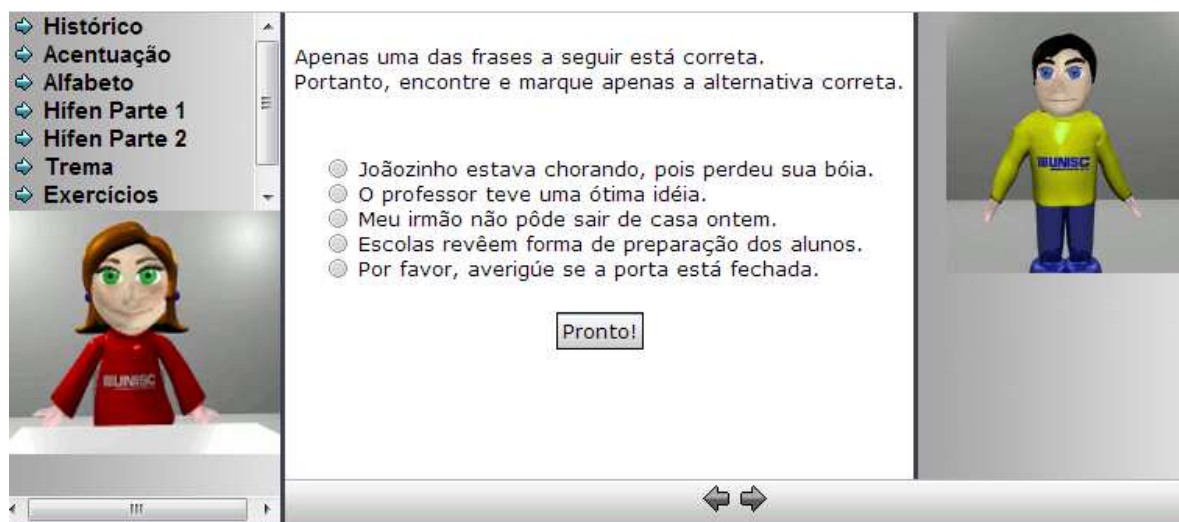


Figura 49. Aula sobre o novo acordo ortográfico – exercícios

Após a elaboração e inserção dos exercícios no ambiente foi necessária a criação de um questionário para que o ambiente pudesse ser avaliado, conforme descrito a seguir.

### 3.9.1 Questionário

O conceito de questionário foi criado e inserido no ambiente com o objetivo de avaliar o mecanismo de comunicação desenvolvido. Para isto, quatro novas tabelas foram incluídas no banco de dados do ambiente, conforme ilustrado na Figura 50. O modelo ER completo pode ser consultado no Anexo B.

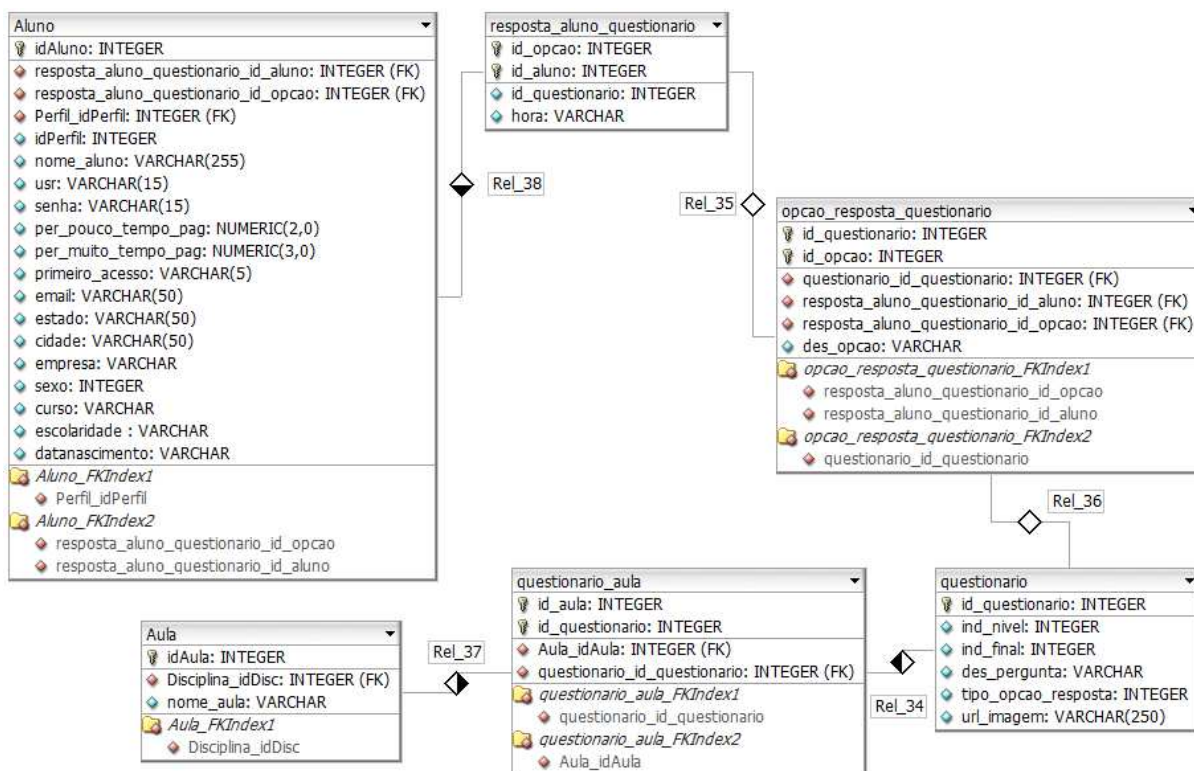


Figura 50. Modelo ER – Tabela Questionario

Ainda com base na ilustração da Figura 50, pode-se destacar as novas tabelas criadas no STI: “questionario\_aula”, “opcao\_resposta\_questionario” e “resposta\_aluno\_questionario” e “questionario”. Esta última tabela foi criada para armazenar a pergunta que deve ser feita ao estudante, enquanto a tabela “resposta\_aluno\_questionario” guarda a resposta deste estudante. Já a tabela “opcao\_resposta\_questionario” possui as opções de resposta para cada questão e, por fim, a tabela “questionário\_aula” indica a qual aula uma determinada questão pertence.

As páginas de exercícios e do questionário foram montadas de forma dinâmica no ambiente. Com isso, as informações apenas precisam estar corretas no banco de dados. A partir disto, o próprio sistema se encarrega de montar a página com as questões. As aulas que

não tiverem questionário cadastrado, não terão o respectivo item no menu de opções. Entretanto, o questionário apenas fica visível ao estudante no momento em que o mesmo responder todos os exercícios ou se a aula em questão não tiver exercícios.

O questionário pode ser visualizado no ANEXO C. A próxima seção descreve como o questionário foi usado para a avaliação do mecanismo de comunicação desenvolvido.

### **3.9.2 Avaliação**

Para homologar as decisões tomadas pelos agentes, uma avaliação foi agendada com uma turma de Lógica para Computação da UNISC. Este procedimento foi realizado na noite do dia 29 de novembro de 2012. Neste dia o “Termo de Consentimento Livre e Esclarecido”, disponível no ANEXO D, foi apresentado aos estudantes e o objetivo da avaliação foi explanado. Os estudantes foram esclarecidos de que a participação nesta avaliação não era obrigatória e eles poderiam desistir da participação a qualquer momento. Além disso, foi explicado que eles não seriam avaliados e sim seriam os avaliadores do ambiente (mais precisamente do mecanismo de comunicação desenvolvido).

Neste processo de avaliação, 11 (onze) estudantes concordaram em participar. A seguir, o ambiente e os agentes pedagógicos Dóris e Dimi foram apresentados aos estudantes. Cada estudante recebeu um usuário e uma senha para acesso ao ambiente e escolheu a aula referente ao Novo Acordo Ortográfico da Língua Portuguesa. Em seguida, os estudantes navegaram nas páginas e interagiram com os agentes. Após a compreensão do conteúdo, os estudantes responderam os 10 exercícios apresentados no Anexo A, referentes ao conteúdo apresentado pelo ambiente, e o questionário disponível no Anexo C, referente às questões elaboradas para identificar a atuação do mecanismo de comunicação desenvolvido.

É importante destacar que o principal objetivo desta avaliação foi validar de forma prática o mecanismo de comunicação desenvolvido. Portanto, a participação dos estudantes neste processo contribuiu no sentido de verificar se durante a interação com os agentes pedagógicos, incluindo o mecanismo de comunicação, perceberam a maneira coordenada e não mais simultânea das mensagens enviadas pelos agentes, bem como se as mensagens condiziam com as habilidades dos agentes.



Assim, a fotografia apresentada na Figura 51 demonstra os estudantes interagindo com o sistema durante a avaliação realizada em sala de aula.

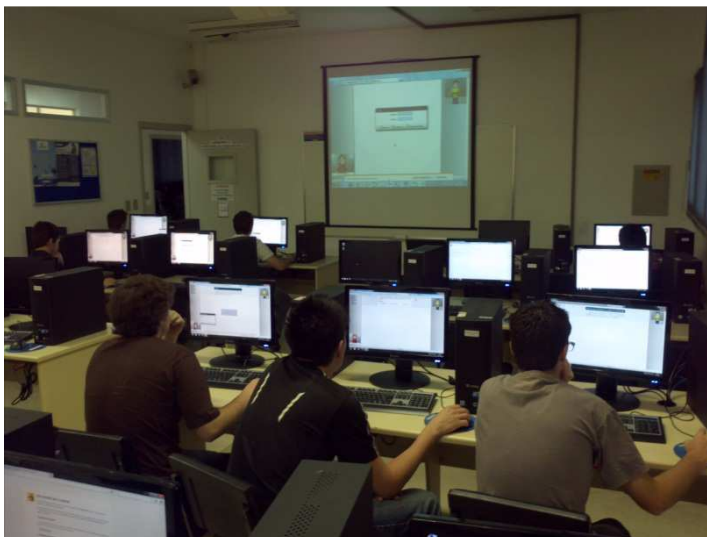


Figura 51. Alunos de Lógica para Computação da UNISC realizando a avaliação

Após a realização da avaliação, foi necessária a análise nas tabelas do banco de dados para obter o resultado do questionário. A partir desta análise, quatro gráficos foram gerados com as respostas dos estudantes. O primeiro gráfico está ilustrado na Figura 52.

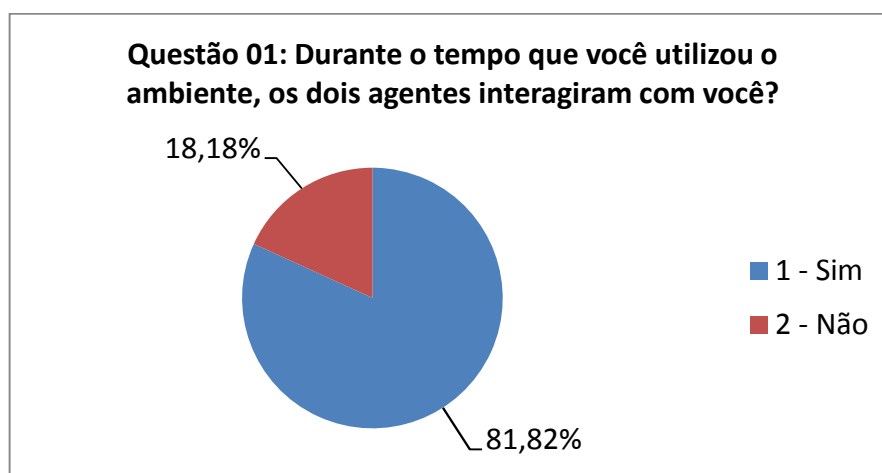


Figura 52. Gráfico com o resultado da questão 1 - questionário

Com base nos dados apresentados pela Figura 52, pode-se observar que a questão 01 teve como objetivo saber se os agentes pedagógicos, em algum momento, interagiram com o estudante. Quase 82% dos estudantes responderam que os dois agentes pedagógicos interagiram com ele.

A questão do gráfico ilustrado na Figura 53 foi elaborada com o objetivo de avaliar o mecanismo de comunicação desenvolvido, cujo um dos propósitos do mesmo era evitar a interação simultânea dos agentes pedagógicos. Pode-se observar que quase 82% dos estudantes responderam que os agentes não interagiram ao mesmo tempo com ele.

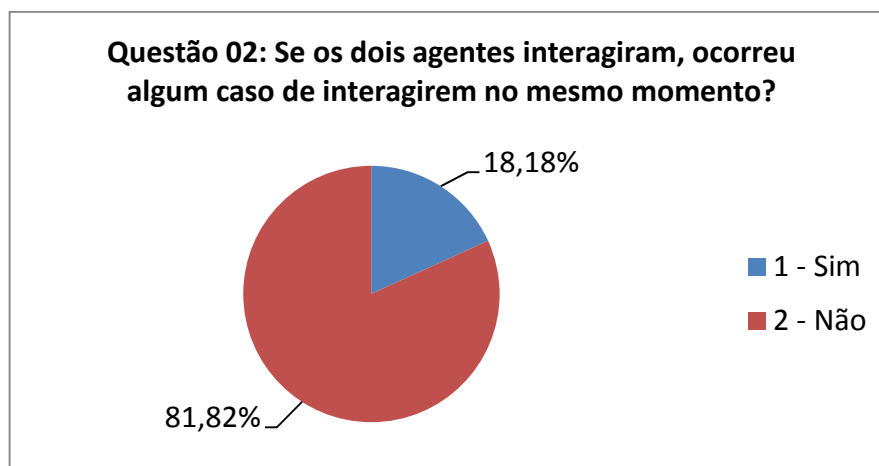


Figura 53. Gráfico com o resultado da questão 2 - questionário

Ainda com base nos dados apresentados pela Figura 52 e Figura 53, pode-se concluir que o mecanismo de comunicação obteve um bom desempenho na avaliação, uma vez que os alunos perceberam o seu correto funcionamento. Desta forma, o gráfico ilustrado na Figura 54 apenas comprova a afirmação anterior, uma vez que quase 91% dos estudantes concordam que a interação dos agentes ocorreu de forma coordenada (um de cada vez). Com isso, pode-se concluir que a heurística da roleta obteve um bom desempenho, fazendo com que ambos agentes interagissem com o estudante.

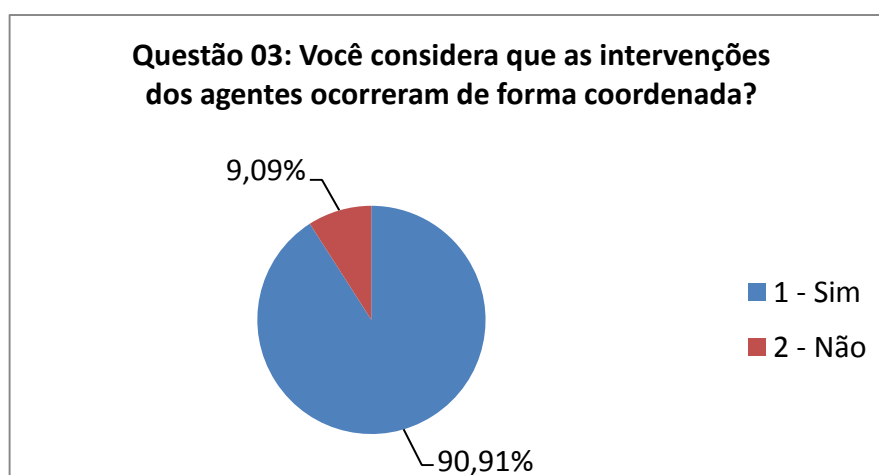


Figura 54. Gráfico com o resultado da questão 3 – questionário

É importante destacar que cada agente pedagógico possui suas funcionalidades e devem interagir com o estudante de acordo com essas características. Com isso, o agente tutor deve auxiliar e monitorar o estudante durante sua atuação no ambiente, enquanto o agente companheiro, por sua vez, deve apenas tratar questões gerais.

Portanto, ambos possuem uma base de conhecimento geral para auxiliar o estudante. Entretanto, o agente tutor também possui uma base de conhecimento específica (relacionada ao assunto da aula), onde ele pode inclusive, fazer perguntas ao estudante sobre o assunto estudado na aula.

Com base nisso, a questão que resultou no gráfico da Figura 55 foi elaborada. Essa pergunta final visa determinar se os agentes interagiram conforme suas características. Pode-se observar que quase 91% dos estudantes conseguiram distinguir as funcionalidades dos agentes pedagógicos, concordando que os agentes interagiram de acordo com o objetivo de cada um deles. Por sua vez, essa questão também foi criada para verificar se o conceito de habilidade estava sendo bem empregado pelo mecanismo de comunicação desenvolvido.

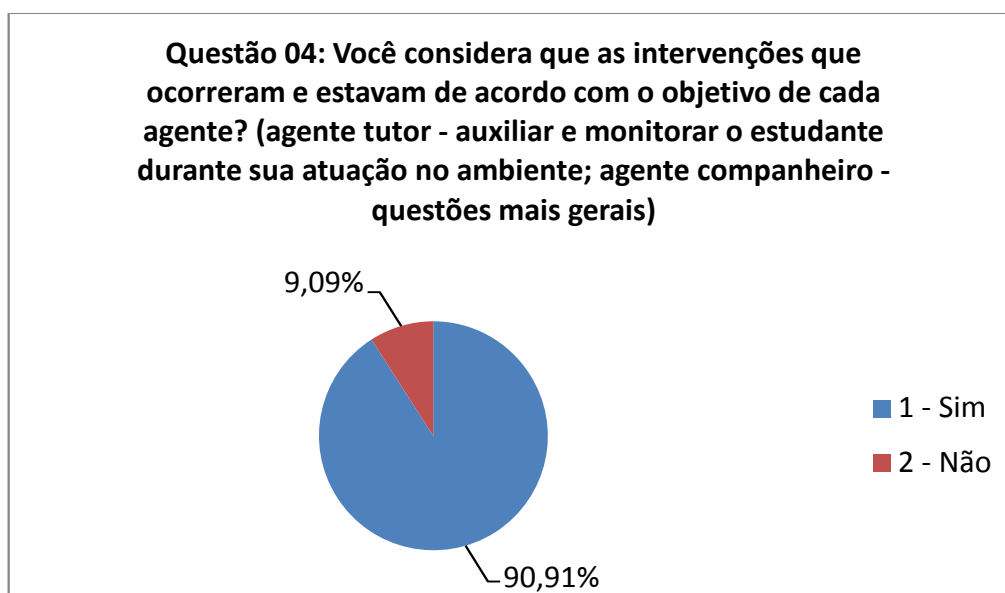


Figura 55. Gráfico com o resultado da questão 4 - questionário

Contudo, também foi possível observar outros aspectos relacionados à execução da avaliação, onde se verificou que todos os estudantes interagiram simultaneamente sobre o mesmo conteúdo, respondendo inclusive as questões relativas ao mecanismo de comunicação.

O objetivo desta avaliação não era avaliar os exercícios respondidos pelos estudantes. No entanto, é interessante observar esse desempenho para avaliar se o conteúdo apresentado para o estudante contribuiu para o seu aprendizado. Desta forma, o gráfico exibido pela Figura 56 apresenta esse desempenho. Os exercícios e as respostas corretas estão disponíveis no Anexo A. As questões de 1 a 10 estão representadas no eixo horizontal e a percentagem de acerto/erro no eixo vertical.

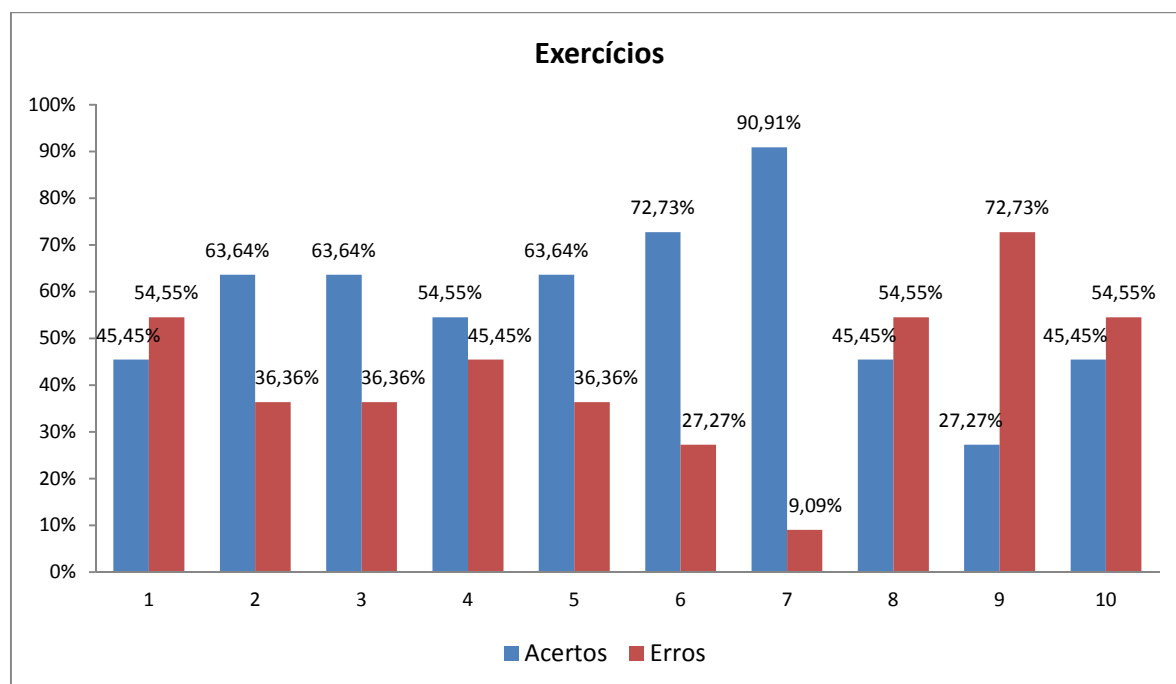


Figura 56. Gráfico com o percentual de acertos e erros - exercícios

Ainda com base nos dados da Figura 56, pode-se observar que os estudantes obtiveram um melhor desempenho no exercício 7, onde quase 92% dos estudantes acertaram o exercício. Entretanto, o pior desempenho dos estudantes foi no exercício 9, onde menos de 28% dos estudantes conseguiram acertar a questão.

No geral, o desempenho dos estudantes nos exercícios foi equilibrado, onde os índices de acertos e erros ficaram próximos na maioria das questões. Esse equilíbrio ainda pode ser observado no fato de apenas 60% dos exercícios possuírem mais acertos que erros. Contudo, pode-se concluir que os estudantes realmente tentaram acertar as questões propostas. No entanto, o fato dessa atividade não ser avaliativa pode ter prejudicado um número maior de acertos. Por outro lado, pode-se concluir que o ambiente cumpriu com o seu objetivo, que é auxiliar na aprendizagem do estudante.

Portanto, algumas mensagens e perguntas foram apresentadas pelos agentes durante a realização da atividade proposta. A Figura 57 ilustra uma dessas perguntas, onde a agente Dóris pergunta se o estudante já conhece o conteúdo da página ou se ele não está interessado. A tutora pode fazer essa pergunta quando um estudante pular de página, para averiguar o motivo deste pulo.

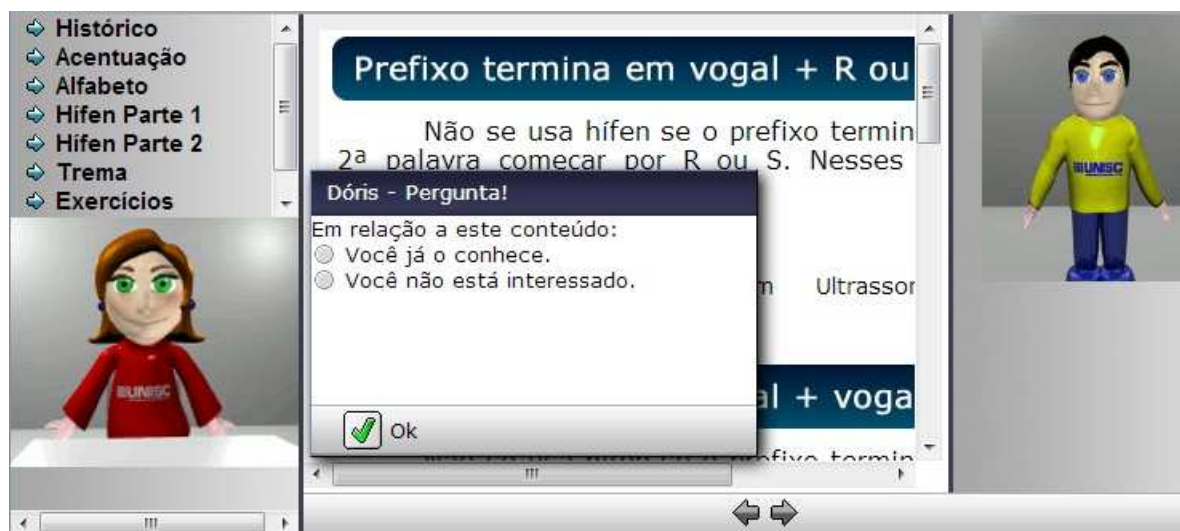


Figura 57. Pergunta para estudante que pulou de página

Ao retroceder uma página, o agente Dimi pode tentar ajudar o estudante, conforme a ilustração da Figura 58. Se o estudante estiver com dificuldades de compreender o conteúdo, Dimi pode sugerir ao estudante que ele revise os itens da aula.

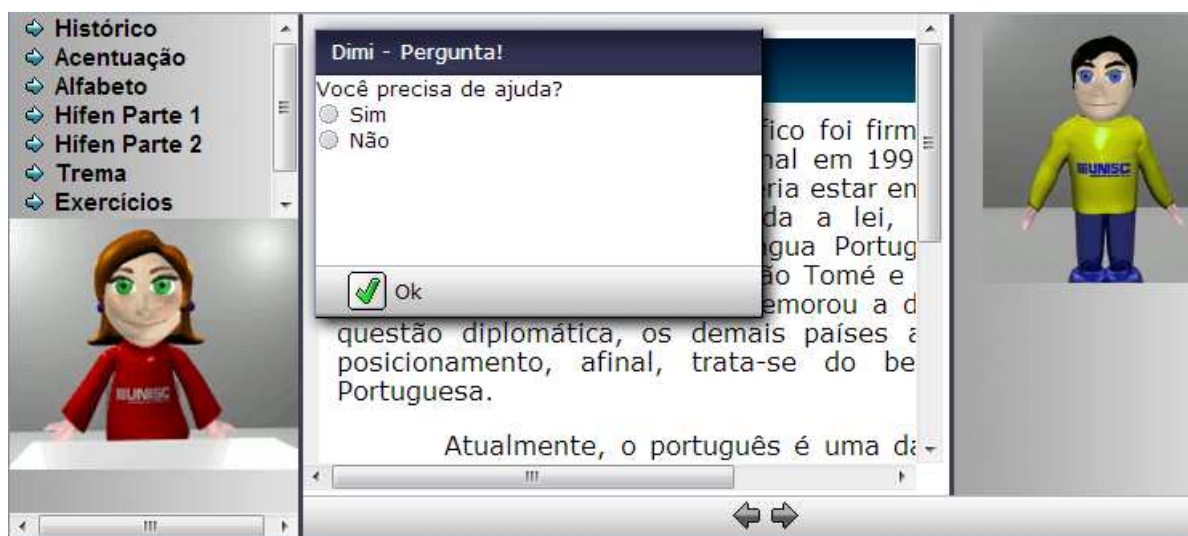


Figura 58. Pergunta para estudante que está com dúvida

Caso o estudante permaneça muito tempo sem interagir no ambiente, a agente Dóris pode fazer uma pergunta referente ao conteúdo de aula para o estudante, conforme ilustrado na Figura 59. Esta pergunta apenas pode ser feita pela agente tutora, já que apenas ela possui conhecimento específico.

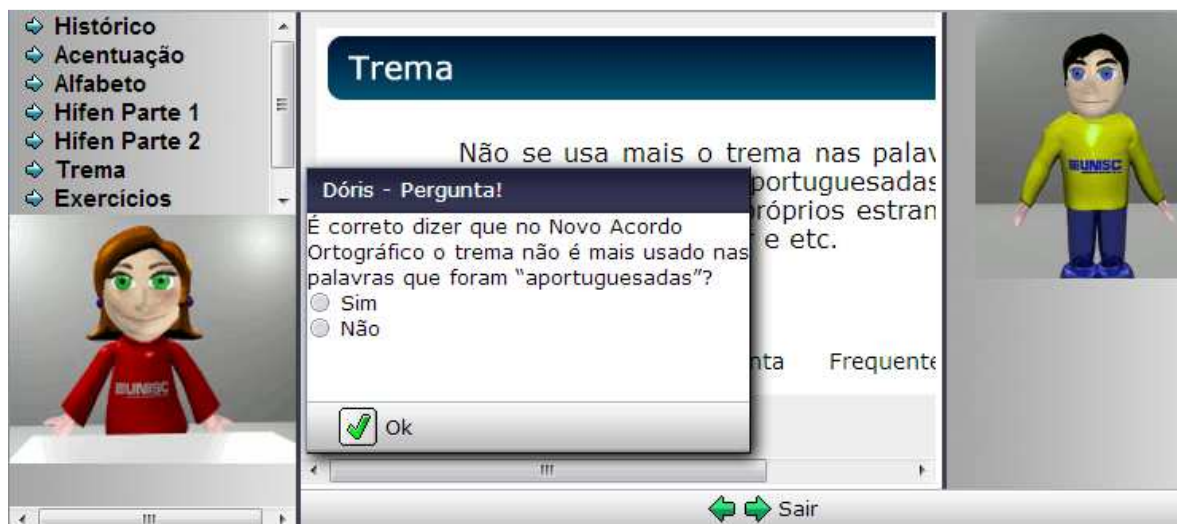


Figura 59. Pergunta específica realizada pela agente Dóris

Além das interações com os agentes pedagógicos, após a realização dos exercícios o estudante pode conferir seu aproveitamento e quais os exercícios que ele acertou e errou. Na imagem ilustrada na Figura 60 pode-se observar que o estudante teve um aproveitamento de apenas 40% na resolução dos exercícios. Entretanto, o estudante pode visualizar as questões que errou para complementar seu aprendizado.

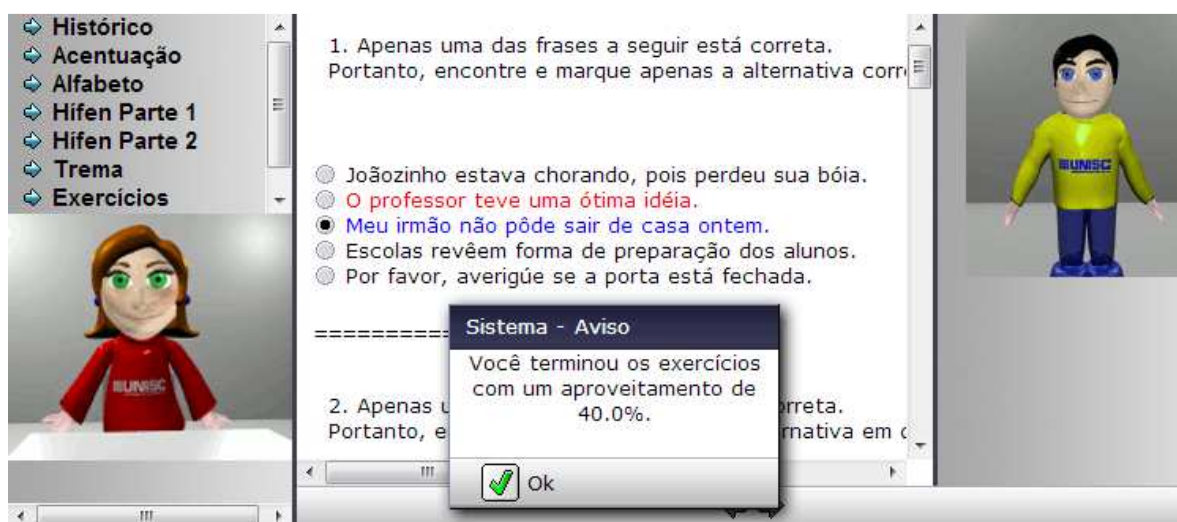


Figura 60. Aproveitamento do estudante nos exercícios

Entretanto, durante o desenvolvimento do mecanismo de comunicação, foi necessária a realização de uma série de modificações no sistema. Algumas das alterações realizadas estão descritas na próxima seção.

### 3.10 Outros desenvolvimentos

Como mencionado na Histórico de evolução do ambiente, do Fundamentação teórica, o STI usado como base do projeto teve seus primeiros trabalhos iniciados em meados de 1998. Contudo, diversos trabalhos foram e são realizados sobre este ambiente (SANTOS et al., 2001; SILVA, 2002; MERGEN e SCHREIBER, 2005; MAINIERI et al., 2005; FROZZA et al., 2007; 2009a; 2009b; 2011; SILVA et al., 2010; BORIN, 2010; KÜNZEL, 2010; KÜNZEL et al., 2011; KÜHLEIS, 2011). Sendo assim, diferentes pessoas (entre estudantes e professores) já realizaram alterações no STI e é inevitável que alguns erros fiquem remanescentes destas modificações.

Portanto, alguns desses problemas foram encontrados e corrigidos durante a realização deste trabalho de dissertação. Algumas melhorias na interface do ambiente foram realizadas e podem ser verificadas na Figura 61. Primeiramente foi executada a identificação dos agentes, para que o estudante pudesse saber com quem ele está interagindo. Em seguida, foi retirado o botão de fechar das caixas de diálogo, pois o estudante podia simplesmente fechar a janela de interação sem responder a pergunta.

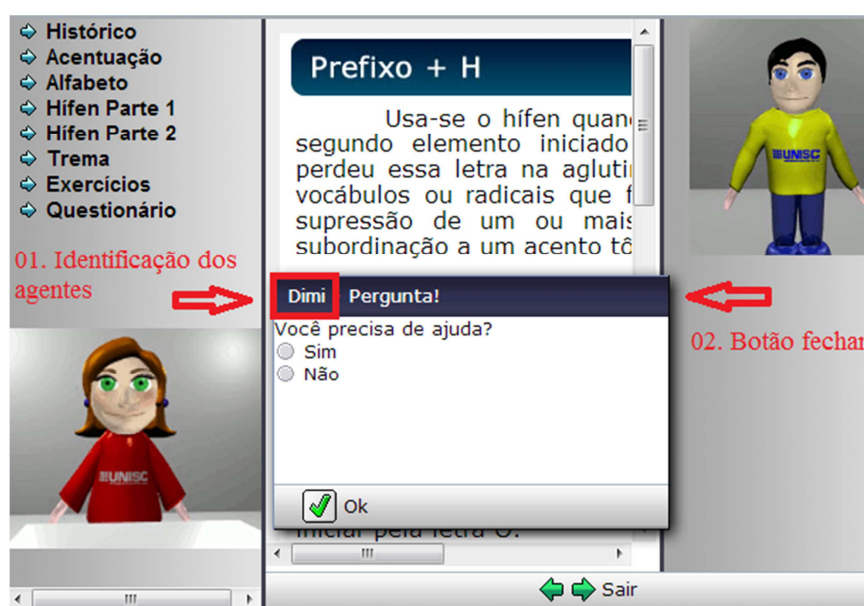


Figura 61. Identificação do agente e caixa de diálogo sem o botão fechar

Antes das alterações realizadas, também era possível pressionar o “ok” sem responder a pergunta do agente e, para piorar, essa resposta era considerada como correta, seguida da mensagem de incentivo “Muito bem”. Então, foi realizada uma modificação não permitindo que o estudante deixe de responder uma pergunta, conforme ilustrado na Figura 62.

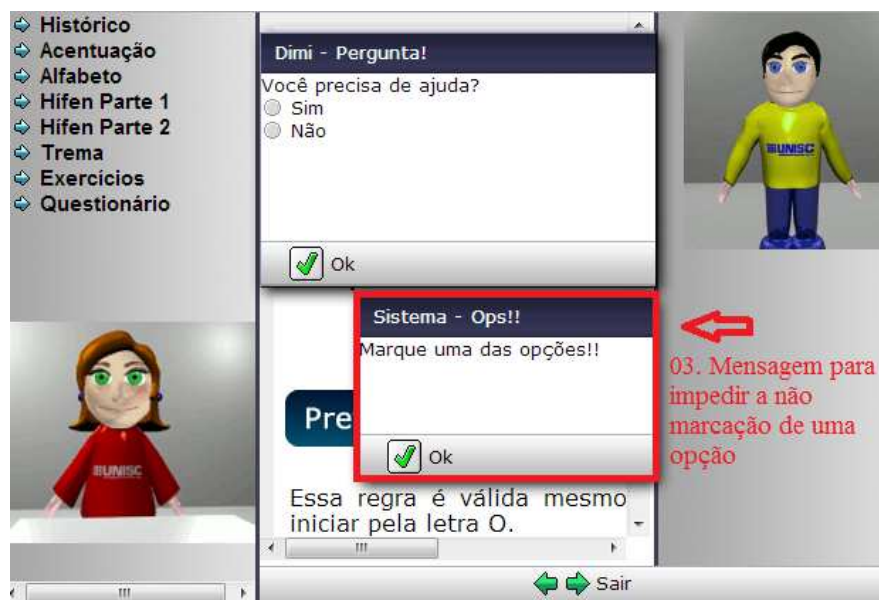


Figura 62. Mensagem para impedir que o estudante não responda uma pergunta

Também foi corrigido um problema ao não responder um exercício e pressionar o botão “pronto”. O problema causava uma exceção no sistema e, conseqüentemente, retorna para a tela de *logon* do sistema. A Figura 63 ilustra o erro que era apresentado ao estudante.

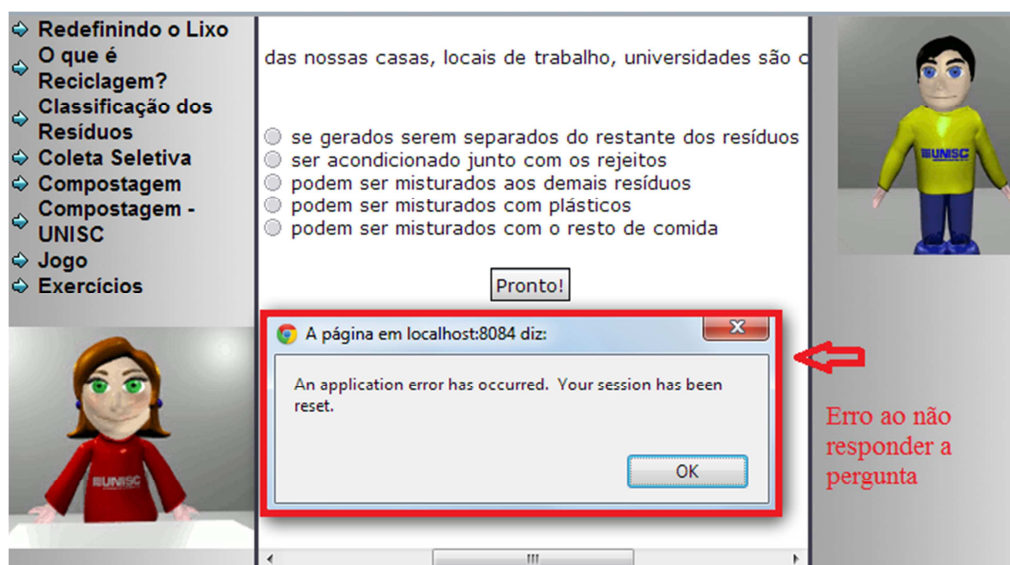


Figura 63. Erro ao não responder um exercício



No entanto, outras mensagens de erro semelhantes à apresentada na Figura 63 foram encontradas no sistema, algumas ao responder uma interação com um dos agentes, outras ao navegar no menu de opções, entre outras ações. Por este motivo, um mecanismo de registro foi criado no sistema. Este procedimento tinha como objetivo identificar o motivo dos erros mencionados. Como a mensagem é genérica, ficava difícil encontrar a origem do problema. Portanto, o arquivo “sti\_unisc\_log.log” localizado na pasta de instalação do *tomcat* possui os registros de algumas partes do sistema.

Além de armazenar as exceções de partes do sistema, o arquivo de registros também guarda informações referentes à comunicação entre os agentes. Desta forma, todas as interações do agente facilitador com os agentes pedagógicos ficam gravadas neste arquivo. Esse arquivo de registro foi bastante usado durante a implementação do mecanismo de comunicação.

O Quadro 7 apresenta um exemplo completo de uma comunicação entre os agentes, desde o momento que o agente facilitador pergunta aos agentes pedagógicos se eles possuem a habilidade de tratar o estímulo gerado no ambiente, até o momento que o agente escolhido avisa ao agente facilitador que a interação com o estudante terminou.

Com isso, pode-se observar que o estímulo a ser tratado é o estímulo de código 3 (pulo de página), especificado em várias mensagens do arquivo de registros, como por exemplo, a mensagem 1 (um). A agente tutora Dóris foi escolhida para interagir com o estudante (Agente 0 deve executar a habilidade) e as *threads* (*Thread: true*) estão sendo usadas para monitorar a fila de entrada dos agentes. Por fim, pode-se observar que o agente facilitador é avisado do fim da interação, conforme a mensagem 10: Enviou Tipo 4. Exibe comunicação. Interagindo: false.

Entretanto, o registro das comunicações é um recurso técnico. Neste caso, a mensagem do tipo 4 é uma mensagem que o agente facilitador recebe quando o agente pedagógico inicia ou encerra uma interação com o estudante. No caso deste exemplo, o agente Facilitador está sendo informado que a interação terminou (Interagindo: false).

Quadro 7 – Exemplo de registro da comunicação entre os agentes

**01:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Enviou Tipo 1. Qtd Msg Enviadas: 1. Estímulo: 3/0. Cód. Msg. Aleatória: 0/0. Interagindo: false. Thread: true

**02:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Enviou Tipo 1. Qtd Msg Enviadas: 2. Estímulo: 3/0. Cód. Msg. Aleatória: 0/0. Interagindo: false. Thread: true

**03:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Enviou Tipo 2. Agente: Dóris. Interações: 0.

**04:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Enviou Tipo 2. Agente: Dimi. Interações: 0.

**05:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Enviou Tipo 3. Agente 0 deve executar a habilidade. Interagindo: false.

**06:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Recebeu Tipo 3. Agente 0 vai executar a habilidade.

**07:** Sun Dec 09 16:12:52 BRST 2012 :: MSG.: 1. Recebeu Tipo 3. Agente 0 executou a habilidade.

**08:** Sun Dec 09 16:12:53 BRST 2012 :: MSG.: GerentePerguntas enviou Tipo 4. Adiciona comunicação <> null. Interagindo: true.

**09:** Sun Dec 09 16:12:58 BRST 2012 :: MSG.: GerentePerguntas enviou Tipo 4. Adiciona comunicação <> null. Interagindo: true.

**10:** Sun Dec 09 16:12:59 BRST 2012 :: MSG.: Enviou Tipo 4. Exibe comunicação. Interagindo: true.

Sun Dec 09 16:12:59 BRST 2012 :: MSG.: Enviou Tipo 4. Exibe comunicação. Interagindo: false.

Fonte: do autor.

Algumas melhorias também foram adicionadas ao sistema, como por exemplo, a exibição do aproveitamento do estudante nos exercícios. Esse aproveitamento é exibido quando o estudante termina os exercícios ou quando ele pressiona a alternativa “exercícios já realizados”. A Figura 64 ilustra uma mensagem com aproveitamento do estudante nos exercícios. Desta forma, pode-se perceber que o estudante obteve um aproveitamento de 60% nesses exercícios.

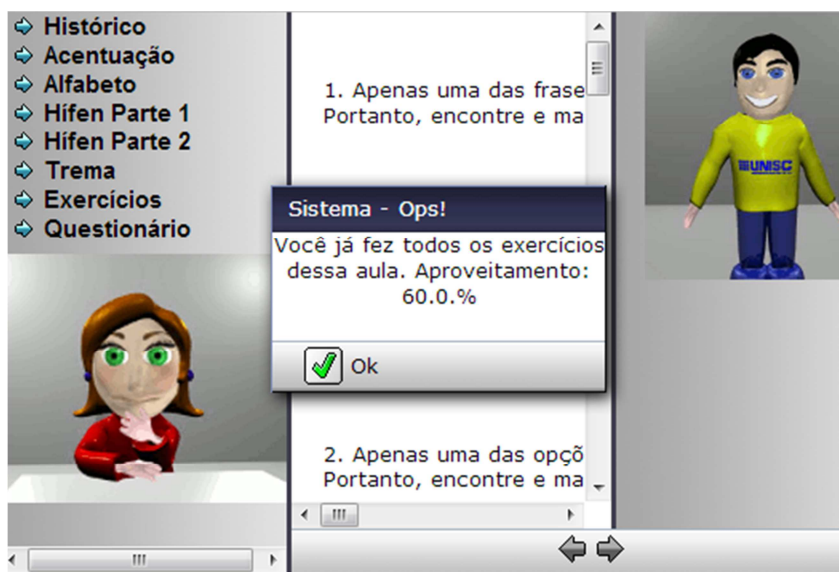


Figura 64. Mensagem de aproveitamento do estudante

Assim, como o percentual de acertos, as questões que o estudante acertou e errou também passaram a ser disponibilizadas ao usuário, no mesmo momento que a mensagem de aproveitamento. Portanto, ao pressionar o botão “ok”, o estudante pode analisar as questões que ele acertou e as que ele errou, conforme ilustrado na Figura 65. Em vermelho é exibida a questão que o estudante errou e a resposta correta em azul.

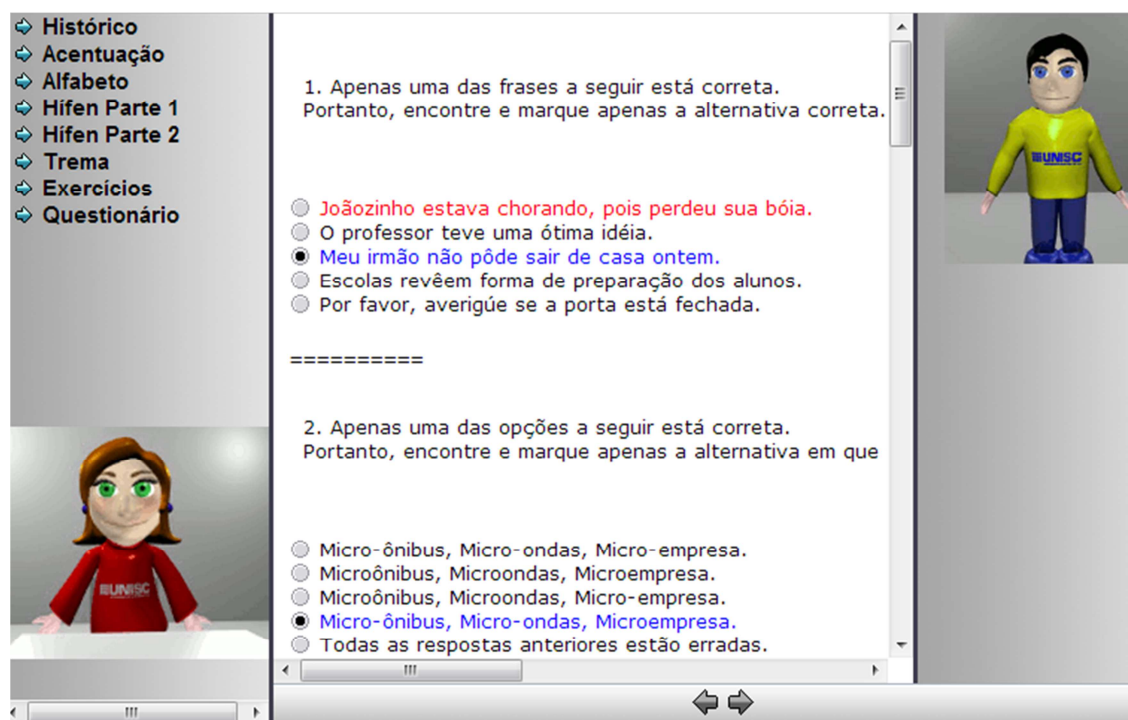


Figura 65. Gabarito do exercício

Talvez, o problema mais grave encontrado no sistema foi a inserção do agente Dimi, pois em muitos casos, ambos os agentes deixavam de interagir com o estudante. Através da análise realizada no sistema, verificou-se que muitas classes e métodos foram implementados inadequadamente e, os métodos da agente Dóris foram substituídos por métodos do agente Dimi.

Com isso, foi necessário encontrar uma versão antiga do sistema, com apenas a Dóris interagindo com o estudante para corrigir a implementação de ambos agentes. Desta forma, foi realizada a integração entre os sistemas até que a interação de ambos os agentes com o estudante ocorresse de forma correta.

A partir da correção mencionada anteriormente, outras melhorias internas foram realizadas, como por exemplo, a união de classes e métodos que originalmente estavam separados e possuíam as mesmas características. Essa separação fazia com que muitos dos métodos precisassem ser escritos em ambas as classes, o que causava repetição de trabalho e poderia causar erros ao atualizarem-se alguns desses métodos.

Pode-se citar o exemplo das classes agenteTutor.java e agenteCompanheiro.java. A solução, neste caso, foi juntar tais classes em apenas uma, renomeada para agente.java. Com isso, passou-se a apenas instanciar duas vezes o objeto da classe agente.java, uma vez para o agente tutor e outra vez para o agente companheiro. Desta forma, as características específicas de cada agente pedagógico apenas são setadas no momento de realizar a instância dos objetos.

### **3.11 Considerações**

Com base no desenvolvimento do mecanismo de comunicação entre os agentes do STI base e, através das validações realizadas, pode-se concluir que, todas as técnicas especificadas e desenvolvidas durante a realização deste trabalho, obtiveram sucesso em seus objetivos. Isto pode ser comprovado pela comunicação dos agentes pedagógicos contidos no ambiente, que passaram a se comunicar com o estudante de forma coordenada, sem mensagens simultâneas.

Este mecanismo de comunicação contou com uma série de fundamentações que foram implantadas no sistema: desenvolvimento do agente Facilitador para coordenar a

comunicação entre os agentes; arquitetura de troca de mensagens entre os agentes; protocolo e linguagem de comunicação baseados no padrão FIPA; linguagem de conteúdo com formato próprio de mensagens; heurística baseada no método da roleta para determinar o agente que deve iniciar a interação com o estudante, considerando o conceito de habilidades e o número de interações já realizadas pelos agentes pedagógicos; *threads* para gerenciar as mensagens que entram na fila dos agentes.

A validação do mecanismo de comunicação também foi uma etapa fundamental deste trabalho, tanto pelos cenários de interação elaborados, quanto pela avaliação prática realizada por alguns estudantes da UNISC.

Para finalizar, é importante destacar que, este trabalho sempre visou o funcionamento do sistema como um todo, o que trouxe várias outras contribuições para o STI, como por exemplo, a possibilidade dos estudantes poderem avaliar o ambiente, a exibição do percentual de aproveitamento do estudante nos exercícios, a exibição dos resultados destes exercícios, entre outras melhorias.

## 4 Conclusão

O mecanismo de comunicação modelado e desenvolvido está disponível para qualquer organização, tanto para a universidade quanto para outra instituição ou empresa. Com isso, o presente trabalho não possui objetivo comercial, apenas acadêmico. No entanto, não há nada que impeça alguém de usar as fundamentações e os resultados obtidos para a finalidade comercial, desde que mantenha os créditos ao autor.

O presente trabalho de dissertação de mestrado focou no estudo e no desenvolvimento de um mecanismo de comunicação entre agentes pedagógicos de um STI, com o objetivo de permitir a interação organizada/coordenada dos agentes pedagógicos com o estudante. Com isso, foi possível evitar a comunicação simultânea dos agentes pedagógicos com o estudante, uma vez que não há mais mensagens exibidas ao mesmo tempo para o estudante.

Para que o sucesso no presente trabalho pudesse ser alcançado, foi adicionado ao mecanismo de comunicação um protocolo de comunicação (interação) entre agentes. Antes disso, foi necessário realizar um estudo de protocolos de comunicação, linguagem de comunicação entre agentes e linguagem de conteúdo. Para isto, alguns trabalhos relacionados também foram pesquisados para a definição das tecnologias a serem usadas.

Desta forma, se optou pelo desenvolvimento de um protocolo baseado no *FIPA Contract Net Interaction Protocol (FIPA, 2002b)*. Outra parte importante do protocolo de comunicação é o formato das mensagens, responsáveis pela conversação entre os agentes propriamente dita. Desta forma, também se optou em usar um formato baseado nas performativas da FIPA-ACL. No entanto, como linguagem de conteúdo foi desenvolvida uma linguagem própria, pois não há troca externa de mensagens entre os agentes.

Antes de iniciar o desenvolvimento do mecanismo de comunicação, a fundamentação de habilidades foi criada para os agentes pedagógicos. Com isso, cada agente precisa possuir a habilidade para tratar um determinado estímulo gerado pelo estudante no ambiente. Este conceito foi elaborado para ser usado em conjunto com o protocolo de comunicação.

Além disso, também foi desenvolvido o agente facilitador que, por sua vez, é outro agente implementado no ambiente e possui a função de facilitar os processos de cooperação, coordenação e negociação. É esse agente que se comunica com os agentes pedagógicos, ou seja, o protocolo de comunicação desenvolvido visa à troca de mensagens entre os agentes pedagógicos e o agente facilitador. Esse novo agente visa auxiliar a tomada de decisão, ajudando a determinar qual dos agentes deve interagir com o estudante.

Para isto, o agente facilitador usa uma heurística desenvolvida para decidir quais dos agentes pedagógicos deve iniciar uma interação com o estudante. Desta forma, a heurística implementada é baseada no Método da roleta. Assim, esse método consiste em privilegiar o agente pedagógico que interagiu menos vezes com o estudante, dando mais oportunidades para que esse seja o próximo agente escolhido para iniciar a interação. Entretanto, é importante destacar que o método apenas é executado quando mais de um agente possuir a habilidade de tratar o estímulo gerado e recebido pelo agente facilitador.

A linguagem de programação Java foi usada no trabalho para o desenvolvimento das técnicas especificadas. Esta linguagem foi escolhida e se fez necessária pelo fato do STI também ter sido desenvolvido em Java. Como as implementações precisaram ser realizadas dentro do ambiente, o uso desta linguagem de programação se tornou indispensável. Desta mesma forma, o PostgreSQL foi usado para o armazenamento e gerenciamento da base de conhecimento do sistema.

Para a validação do mecanismo de comunicação desenvolvido, foram gerados cenários para que se pudesse homologar cada parte do mecanismo de comunicação. Estes cenários foram desenvolvidos para validar o funcionamento das **Erro! Fonte de referência não encontrada.**, das novas tecnologias agregadas ao sistema e, por fim, homologar o seu funcionamento em diferentes *browsers* e dispositivos móveis.

Com o objetivo de realizar uma avaliação externa e verificar as Decisões tomadas pelos agentes, foi elaborado um Questionário. Para isto, novas tabelas foram criadas no banco dados para que as questões relativas a esse Questionário pudessem ser cadastradas. Com isso, o sistema ficou passível de avaliação externa, uma vez que os estudantes podem usar esse Questionário para avaliar o ambiente ou partes dele.

Desta forma, um novo assunto também foi adicionado ao ambiente. Essa aula está relacionada ao “Novo Acordo Ortográfico da Língua Portuguesa”. Para isto, foram criadas as páginas HTML com o conteúdo das aulas, separadas por assunto. Além disso, um total de dez exercícios foi elaborado e inserido no banco de dados, assim como quatro questões do questionário.

A partir da conclusão de todas as etapas anteriores, o sistema ficou passível de ser usado e avaliado externamente. Com isso, uma avaliação foi realizada com uma turma de Lógica para Computação da UNISC. Neste caso, 11 (onze) estudantes concordaram em participar da atividade. A partir desta avaliação foi possível concluir que o mecanismo de comunicação funcionou conforme o esperado. Isto pode ser justificado pela avaliação realizada pelos estudantes, cujos resultados são listados a seguir:

- 81,82% dos estudantes avaliaram que, durante o tempo que utilizaram o ambiente, os dois agentes interagiram com eles.
- 81,82% dos estudantes afirmaram que em nenhum momento ocorreu algum caso dos agentes pedagógicos interagirem ao mesmo tempo.
- 90,91% dos estudantes consideraram que as intervenções dos agentes ocorreram de forma coordenada.
- 90,91% dos estudantes consideraram que as intervenções que ocorreram estavam de acordo com o objetivo de cada agente (agente tutor - auxiliar e monitorar o estudante durante sua atuação no ambiente, agente companheiro - questões mais gerais).

Portanto, pode-se concluir que o mecanismo de comunicação desenvolvido atendeu satisfatoriamente as necessidades relativas à comunicação dos agentes pedagógicos do STI. Isso pode ser justificado pelo sucesso conquistado através dos cenários de teste que foram criados e pelo retorno obtido da avaliação externa do ambiente.

Antes de finalizar, também é importante destacar que Outros desenvolvimentos foram realizados com a finalidade de melhorar, corrigir e agregar funcionalidades ao STI. Dentre eles, a correção mais significativa foi a reestruturação do Dimi (agente companheiro). No entanto, as melhorias foram desde uma simples identificação dos agentes (nome dos agentes nas caixas de diálogos) até problemas de implementação (como caixas de erros genéricas). Já



como inovação, pode-se destacar a exibição do aproveitamento do estudante na execução dos exercícios, exibição do gabarito das respostas e a criação do questionário.

Alguns trabalhos futuros podem ser realizados para enriquecer o STI adotado como base da dissertação de mestrado, tanto no projeto como um todo, quanto no mecanismo de comunicação desenvolvido. Alguns dos trabalhos relacionados à comunicação dos agentes são listados a seguir:

- Tornar o cadastro das habilidades dinâmicas: atualmente, a atribuição de qual agente possui habilidade para tratar um determinado estímulo é realizada dentro do próprio código-fonte do sistema. Portanto, essas informações poderiam ser cadastradas no banco de dados e o próprio sistema se encarregaria de passar as habilidades para cada agente pedagógico.
- Tornar dinâmica as mensagens dos agentes pedagógicos: atualmente, as mensagens exibidas pelos agentes são informadas no código-fonte do projeto, ou seja, são fixas e para que possam ser alteradas precisam ser implementadas. Os exercícios e o questionário são cadastrados no banco de dados e o próprio sistema se encarrega de montar as questões e armazenar as respostas dos estudantes. Desta forma, as mensagens que são exibidas pelos agentes pedagógicos poderiam funcionar da mesma maneira.
- Mensagens no formato XML: atualmente, o sistema troca as mensagens dentro do próprio sistema, pois não há necessidade de comunicação externa (não há motivo para ficar lendo e escrevendo em arquivos sem que realmente haja esta necessidade). Entretanto, as mensagens poderiam ser armazenadas no formato XML para uma troca de informações com algum agente externo ao ambiente. Entretanto, neste caso, os sistemas envolvidos na comunicação deveriam possuir a mesma linguagem de conteúdo.

Por outro lado, também é possível que trabalhos futuros sejam realizados para agregar valor ao STI como um todo. Com base nisso, alguns dos trabalhos sugeridos são listados nos seguintes pontos:

- Criação de um aplicativo para acesso ao STI: atualmente, é necessário o uso de um *browser* para acessar o ambiente através de um dispositivo móvel, tanto rodando o SO

*Android* quanto o SO *iOS*. Portanto, o uso de um aplicativo eliminaria essa necessidade de um navegador para acessar o STI.

- Tornar os agentes pedagógicos dinâmicos: atualmente, existem dois agentes pedagógicos no sistema e eles são instanciados diretamente no código-fonte. Entretanto, os agentes pedagógicos poderiam ser cadastrados no banco de dados e instanciados de forma dinâmica, onde cada característica do agente também estaria armazenada neste banco de dados. Com isso, diferentes agentes poderiam ser cadastrados em cada aula, assim como a quantidade de agentes também poderia ser diferente. Por exemplo, em uma aula de matemática o agente pedagógico poderia ter o formato de uma calculadora.
- Reformular a tela principal do STI: atualmente, a tela principal apenas pode ser dividida em três partes, limitando o número de informações a serem exibidas. Por exemplo, seria muito complicado adicionar a figura de mais um agente no sistema, pois não seria possível fazer mais uma divisão na tela. Informação técnica: talvez usar *jPanel* ao invés de *Container*.
- Sistema para cadastro no banco de dados: atualmente, a grande maioria das informações é cadastrada diretamente no banco de dados. Portanto, é necessária a criação de algum sistema para auxiliar no cadastro das informações. Esse sistema poderia estar junto com o STI ou poderia ser um sistema externo, apenas para os cadastros desses dados.

## Referências

- ALMEIDA, H. O.; LOUREIRO, E.; VINÍCIUS, G.; PAES, R.; PERKUSICH, A.; COSTA, E. *Ambiente Integrado para o desenvolvimento de sistemas multi-agentes*. In: XVII Simpósio Brasileiro de Engenharia de Software - 10<sup>a</sup> Sessão de Ferramentas, 2003.
- ALMEIDA, H. O.; PERKUSICH, A.; COSTA, E. B.; PAES, R. B.; *COMPOR: a Methodology, a Component Model, a Component based Framework and Tools to Build Multiagent Systems*. In: CLEI Electronic Journal, 7(1), 2004.
- ANDERSON, J. R.; *Cognitive Psychology and Its Implications*. 7. ed. New York: Worth Publishing, 2010.
- ANDROID; *Android is the world's most popular mobile platform*. Disponível em: <http://www.android.com/>. Acessado em: janeiro de 2013.
- APPLE; *iPhone e IOS*. Disponível em: <http://www.apple.com/> e <http://www.apple.com/br/>. Acessado em: janeiro de 2013.
- AUER, K.; *Agents*. 1995. Disponível em: <http://www.biplane.com.au/kauer/>. Acessado: Janeiro de 2013.
- AZEVEDO, B. F. T.; TAVARES, O. de L.; *Um Ambiente Inteligente para Aprendizagem Colaborativa*, 2001. XII Simpósio Brasileiro de informática na educação – SBIE – UFES, 2001. Disponível em: <http://www.brie.org/pub/index.php/sbie/article/viewFile/144/130>. Acessado em: outubro de 2011.
- BARBOSA, T. L. R.; *Um sistema multiagente para monitoramento atmosférico*. 2005. Dissertação apresentada ao Curso de Mestrado Profissional em Rede de Computadores, Universidade Salvador – UNIFACS.
- BEHAR, P. A.; MEIRELLES, L. S.; MAZZOCATO, S. B.; SOUZA, L. B. de; SIQUEIRA, L. G.; *Avaliação de Ambientes Virtuais de Aprendizagem: O Caso do ROODA na UFRGS*. In: Revista Avances en Sistemas e Informática, v. 4, p. 81-100, Medellín, Bogotá, 2007.
- BELGRAVE, M.; *The Unified Agent Architecture. A White Paper*, 1995.
- BENECH, D.; DESPREATS, T.; RAYNAUD, U.; *A KQML-CORBA based Architecture for Intelligent Agents Communication in Cooperative Service and Network Management*. In: INTERNATIONAL CONFERENCE ON MANAGEMENT OF MULTIMEDIA NETWORKS AND SERVICES (MMNS'97). Proceedings, Montreal, Canada, 1997.

BENETTI, Jean Carlo. *O uso de Redes Bayesianas para identificar o estilo cognitivo de aprendizagem do aluno em um Sistema Tutor Inteligente*. 39 f. Projeto de Dissertação (Programa de Pós-Graduação em Sistemas e Processos Industriais – Mestrado) - Universidade de Santa Cruz do Sul, Santa Cruz do Sul. 2011.

BICA, F.; *Eletrotutor III – Uma Abordagem Multiagente para o ensino à Distância*. Dissertação de mestrado para a obtenção do grau de Mestre em Ciência da Computação, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 1999.

BITTENCOURT, G.; *Inteligência Artificial Distribuída*, Anais: I Workshop de Computação do ITA, 1998.

BORIN, M; FROZZA, R.; KIPPER, L.; SCHREIBER, J. N. C.; *Virtual Learning Environments with Emotional Pedagogical Agents for Training in Organizations*. In: International Conference on Industrial Engineering and Operations Management (ICIEOM), 2012, Guimarães. Proceedings of the XVIII International Conference on Industrial Engineering and Operations Management (ICIEOM). Baurú - SP: ABEPRO, 2012. v. único. p. ID57.1-ID57.10.

BORIN, P. M.; *O uso de ambientes virtuais de aprendizagem com agentes pedagógicos emocionais para capacitação em organizações*. Dissertação (Mestrado em Sistemas e Processos Industriais) - UNISC, Programa de Pós-Graduação em Sistemas e Processos Industriais, 2010.

BRADSHAW, J. M.; *Software Agents*. Chapter 1: An Introduction to Software Agents. MIT Press, Menlo Park, CA, USA, 1997.

CARBONELL, J. R.; *AI in CAI: an artificial intelligence approach to computer assisted instruction*. IEEE Transactions on Man Machine Systems, 1970.

CARVALHO, F. S.; *Integração entre Sistema Multi-Agentes e Sistemas de Banco de Dados Distribuídos*. Dissertação (Mestre em Engenharia Elétrica) – USP (Escola Politécnica da Universidade de São Paulo). São Paulo, 2008.

CASTELFRANCHI, C.; Guarantees for autonomy in cognitive agent architecture. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 56–70. Springer-Verlag: Heidelberg, Germany, 1995.

CASTELFRANCHI, C.; MICELLI, M.; CESTA, A.; *Dependence relations among autonomous agents*. In Dcai3, 1992.

CASTELFRANCHI, C.; *Modelling social action for AI agents*. Istituto di Psicologia del CNR - Unit of "AI, Cognitive Modelling & Interaction" v. Marx 15-00137, Roma, Itália, 1998.

CERVO, A. L.; BERVIAN, P. A; *Metodologia científica*. 4ª edição, São Paulo: Makron Books, 1996, 90 p.

CONTE, R.; CASTELFRANCHI, C.; *Mind is not enough: Precognitive bases of social interaction*. In: Proceedings of 1992 Symposium on Simulating Societies, Guildford, UK, 1992.

COSTA, A. L.; BITTENCOURT, G.; *Parla: A cooperation language for cognitive multi-agent systems*. 8<sup>th</sup> Portuguese Conference of Artificial Intelligence, Spring-Verlag, Springer, Lecture Notes Artificial Intelligence, EPIA, 1997.

CUTMORE, T. R. H., HINE, T. J., MABERLY, K. J., LANGFORD, N. M., HAWGOOD, G.; *Cognitive Ad Gender Factors Influencing Navigation in Virtual Environment*. International Journal Of Human - Computer Studies, 223-249, 2000.

DARWIN, C.; *On the origin of species*. The Pennsylvania State University, 1859. Disponível em: <http://www.hn.psu.edu/faculty/jmanis/darwin.htm>. Acessado em: janeiro de 2013.

DAVIS, J.; HYLANDS, C.; KIENHUIS, B.; LEE, E.; LIU, J.; OJUN-LIU, X.; MULIADI, L.; NEUENDORFFER, S.; TSAI, J.; VOGEL, B.; XIONG, Y.; *Heterogeneous Concurrent Modeling and Design in Java*. In Technical Memorandum UCB/ERL M01/12. EECS, University of California, Berkeley, 2001.

DECKER, K. S.; SYCARA, K.; *Intelligent Adaptive Information Agents*. In: Journal of Intelligent Information Systems, Kluwer Academic Publishers, Boston, 1997.

FARACO, R. Á.; GAUTHIER, F. Á. O.; *Uma Arquitetura de Agentes para Negociação dentro do Domínio do Comércio Eletrônico*. In: Dissertação de Mestrando do Curso de Pós-Graduação em Engenharia de Produção. Florianópolis: Universidade Federal de Santa Catarina (UFSC), 1998.

FERBER, J.; *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Great Britain: Addison Wesley, 1999.

FERNANDES, A. M. R.; *Inteligência Artificial: Noções gerais*. Florianópolis: Visual Books, 2003. 166 p.

FERNANDES, W. L.; *AGROREDE: Estudo exploratório acerca da implementação de uma rede virtual de aprendizagem colaborativa envolvendo as escolas agro técnicas federais*. Mestrado profissional em tecnologia da informação e comunicação na formação em educação a distância. Universidade Federal do Ceará (UFC) e Universidade Norte do Paraná (UNOPAR). Salvador, BA – Brasil, 2007.

FERREIRA, A. B.de H.; *O Novo Dicionário Aurélio da Língua Portuguesa*. Editora Positivo: 4<sup>a</sup>. Edição, 2009.

FERREIRA, G. V. V. M.; ALMEIDA, H. O.; PERKUSICH, A.; COSTA, E. B.; *Especificação e Implementação de Protocolos de Interação entre Agentes para a Plataforma COMPOR*. In: Revista de Ciência da Computação, InfoComp, Vol. 3, N<sup>o</sup> 2. Universidade Federal de Lavras (UFLA), Lavras (MG). ISSN 1807-4545, 2004. Disponível em: <http://www.lbd.dcc.ufmg.br/bdbcomp/servlet/Trabalho?id=5349>. Acessado em: janeiro de 2013.

FININ, T.; FRITZSON R.; MCKAY, D.; *A Language and Protocol to Support Intelligent Agent Interoperability*. Proceedings of the CE & CALS Washington '92 Conference, 1992.

FININ, T.; FRITZSON R.; MCKAY, D.; MCENTIRE, R.; *KQML - A Language and Protocol for Knowledge and Information Exchange*. Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence DAI94, 1994a.

FININ, T.; FRITZSON R.; MCKAY, D.; MCENTIRE, R.; *KQML as an Agent Communication Language*. Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), ACM Press, 1994b.

FININ, T.; LABROU, Y.; MAYFIELD, J.; *KQML as an agent communication language*. Software Agents, MIT Press, Cambridge, 1997.

FININ, T.; WEBER, J.; WIEDERHOLD, G.; GENESERETH, M.; FRITZSON, R.; MCKAY, D.; SHAPIRO, S.; MCGUIRE, J.; PELAVIN, R.; BECK, C.; *DRAFT Specification of the KQML Agent-Communication Language*. Plus example agent policies and architectures. The DARPA Knowledge Sharing Initiative External Interfaces Working Group, DRAFT, 1993.

FIPA.; *FIPA ACL Message Structure Specification*. FIPA Specification: SC00061G, 2002f. Disponível em: <http://www.fipa.org/specs/fipa00061/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Brokering Interaction Protocol Specification*. FIPA Specification: SC00033H, 2002d. Disponível em: <http://www.fipa.org/specs/fipa00033/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Communicative Act Library Specification*. FIPA Specification: SC00037J, 2002g. Disponível em: <http://www.fipa.org/specs/fipa00037/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Contract Net Interaction Protocol Specification*. FIPA Specification: SC00029H, 2002b. Disponível em: <http://www.fipa.org/specs/fipa00029/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Propose Interaction Protocol Specification*. FIPA Specification: SC00036H, 2002c. Disponível em: <http://www.fipa.org/specs/fipa00036/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Recruiting Interaction Protocol Specification*. FIPA Specification: SC00034H, 2002e. Disponível em: <http://www.fipa.org/specs/fipa00034/>. Acessado em: janeiro de 2013.

FIPA.; *FIPA Request Interaction Protocol Specification*. FIPA Specification: SC00026H, 2002a. Disponível em: <http://www.fipa.org/specs/fipa00026/>. Acessado em: janeiro de 2013.

FIPA.; *Foundation for Intelligent Physical Agents*. Disponível em: <http://www.fipa.org>. Acessado em: janeiro de 2013.

FIREFOX; *Mozilla Firefox Web Browser*. Acessado em: dezembro de 2012. Disponível em: <http://www.mozilla.org/en-US/firefox/fx/>.

FONSECA, J. M. M. R.; *Protocolos de Negociação com Coligações em Sistemas Multiagente - Uma aplicação à Gestão Distribuída de Recursos*. Dissertação apresentada para a obtenção do Grau de Doutor em Engenharia Eletrotécnica, Especialidade de Robótica e Manufatura Integrada. Universidade de Nova Lisboa, Faculdade de Ciências e Tecnologia, 2000.

FRANKLIN, S.; GRAESSER, A.; *Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents*, Proceedings of the Third International Workshop on Agent Theories, Springer-Verlag, 1996.

FREEMAN, R.; *What is an Intelligent Tutoring System?*. Published in *Intelligence*, 15-16, 2000.

FRIGO, L. B.; BITTENCOURT, G.; “*MathTutor: Uma ferramenta de apoio a aprendizagem*”. In: Anais do XXII Congresso da Sociedade Brasileira de Computação – X WEI, 2002.

FRIGO, L. B.; POZZEBON, E.; BITTENCOURT, G.; *O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes*. In: Anais do WCETE - World Congress on Engineering and Technology Education, São Paulo/SP, Brasil, 2004. Disponível em: <http://www.inf.ufsc.br/~l3c/artigos/frigo04a.pdf>. Acessado em: janeiro de 2013.

FROZZA, R., KONZEN, A., MAINIERI, A. G., SCHREIBER, J., MOLZ, K., TAUTZ, J., PEDÓ, R., DRESCH, j.; *Agentes tutor e companheiro em um ambiente educacional baseado em estilos cognitivos*. In: Workshop - Escola de Sistemas de Agentes para ambientes Colaborativos - WESAAC 2007. UCPEL, Pelotas, abril de 2007.

FROZZA, R., SILVA, A. A. K. da, SCHREIBER, J. N. C., LUX, B., MOLZ, K. W., KIPPER, L. M., BORIN, M. P., CARVALHO, A. B. de, BAIERLE, J. L., SAMPAIO, L.; *Agentes Pedagógicos Emocionais atuando em um Ambiente Virtual de Aprendizagem*. Centro Interdisciplinar de Tecnologia Educacional da Universidade Federal do Rio Grande do Sul (CINTED-UFRGS). RENOTE - Revista Novas Tecnologias na Educação, 2011.

FROZZA, R., SILVA, A. K. da, LUX, B., CRUZ, M. E. J. K. da, BORIN, M.; *Dóris 3D: Agente Pedagógico baseado em Emoções*. In: XX SBIE - Simpósio Brasileiro de Informática na Educação de 2009a.

FROZZA, R., SILVA, A. K. da, WAGNER, A. L., MOLZ, K. W., SCHREIBER, J., LUX, B., CRUZ, M. K. da, KIEHL, M. E; *Ambiente Educacional Aplicado ao Domínio da Geografia*. Biblioteca digital brasileira de computação (BDBComp), 2009b (2). Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wie/2009/021.pdf>. Acessado em: janeiro de 2013.

GALLIERS, J. R.; *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK, 1998.

GAMBOA, H., FRED A.; *Designing Intelligent Tutoring Systems: a Bayesian Approach*. 3rd International Conference on Enterprise Information Systems, ICEIS, 2001.

GANASCIA, J. G.; *Inteligência Artificial*. São Paulo. Editora Ática, 1993.

GARCIA, A. C. B.; SICHMAN, J. S.; *Agentes e sistemas Multiagentes*. In: *Sistemas Inteligentes: Fundamentos e Aplicações*, S. O. Rezende, Ed. Editora Manole Ltda., Barueri, São Paulo, Brasil, 2003.

GAVIDIA, J. J. Z.; ANDRADE, L. C. V. de.; *Sistemas Tutores Inteligentes*. 2003. Disponível em: <http://www.cos.ufrj.br/~ines/courses/cos740/leila/cos740/STImono.pdf>. Acessado em: janeiro de 2013.

GENESERETH, R. M.; KETCHPEL, S. P.; *A Software Agents*. In: *Communications of the ACM*, 1994.

GIL, A. C.; *Como elaborar projetos de pesquisa*. São Paulo: Atlas, 1991.

GILBERT, D.; MANNY, A.; *Intelligent agent strategic*. 1996. IBM Corporation, Research Triangle Park, NC USA.

GIMÉNEZ-LUGO, G. A.; JEUKENS, I.; FARIA, R.; HÚBNER, J. F.; *Um Enfoque para a Especificação e Execução Flexível de Protocolos em Sistemas Multiagentes*. In: *Revista Gestão & Conhecimento*, FATEC (Faculdade de Tecnologia), vol. 4, nº1, 2006: pp 10 – 19. Disponível em: <http://gc.facet.br/artigos/resumo.php?artigo=27>. Acessado em: janeiro de 2013.

GIRAFFA, L. M. M.; *Uma arquitetura de tutor utilizando estados mentais*. Universidade Federal do Rio Grande do Sul (UFRGS). Programa de Pós-Graduação em Computação (CPGCC), Porto Alegre, BR-RS. Tese de Doutorado, 1999.

GLUZ, J. C.; *Formalização da Comunicação de Conhecimentos Probabilísticos em Sistemas Multiagentes: uma abordagem baseada em lógica probabilística*. Tese de doutorado, programa de pós graduação em computação, Universidade Federal do Rio Grande do Sul (UFRGS), 2005.

GLUZ, J. C.; VICARI, R. M.; *Linguagens de Comunicação entre Agentes: Fundamentos Padrões e Perspectivas*. In: *III Jornada de Mini-Cursos de Inteligência Artificial - SBC2003*, 2003.

GOLDBERG, E. D.; *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional; 1 edition, 1989.

GOLEMAN, D.; *Emotional Intelligence*. New York: Bantam Books, 1995.

GONZÁLEZ, S. M.; TAMARIZ, A. R.; CARNEIRO, E. C.; ALMEIDA, J. S. de; *Agentes Inteligentes no Ambiente Virtual de Ensino de Lógica Halyen*. In: *Conferência IADIS Ibero-Americana WWW/Internet 2007*, Universidade Candido Mendes, Campos dos Goytacazes, RJ, Brasil. Disponível em: [http://www.iadis.net/dl/final\\_uploads/200713P068.pdf](http://www.iadis.net/dl/final_uploads/200713P068.pdf). Acessado em: janeiro de 2013.

GOOGLE; *Chrome web browser*. Disponível em: [www.google.com/chrome](http://www.google.com/chrome). Acessado em: dezembro de 2012.



GOULART, R. R. V.; GIRAFFA, L. M. M.; *Arquiteturas de Sistemas Tutores Inteligentes*. Technical Report Series. Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, 2011.

GREEN, S.; HURST, L.; NANGLE, B.; CUNNINGHAM, P.; SOMERS, F.; EVANS, R.; *Software Agents: A Review*. In: Technical report. Trinity Collega, Dublin, Ireland, 1997.

GUARDIA, R. B.; *Asesores Inteligentes para apoyar el Proceso de Enseñanza de Lenguajes de Programación*. Dissertação de mestrado em Ciências da Computação. ITESM (Instituto Tecnológico y de Estudios Superiores de Monterrey). Monterrey. México, 1997.

GÜRER, D.; *The Use of Distributed Agents in Intelligent Tutoring*. In: It's Workshop on Pedagogical Agents, 1998, San Antonio, Texas.

GYURJYAN, V.; ABBOTT, D.; HEYES, G.; JASTRZEMBSKI, E.; TIMMER, C.; WOLIN, E.; *FIPA agent based network distributed control system*. In: Proceedings of Computing in High Energy and Nuclear Physics. La Jolla, California, 2003. Disponível em: <http://www.mendeley.com/research/fipa-agent-based-network-distributed-control-system-1/>. Acessado em: janeiro de 2013.

HARASIM, L.; HILTZ, R.; TELES, L.; TUROFF, M.; *Redes de Aprendizagem: Um guia para o Ensino e Aprendizagem Online*. São Paulo, SENAC. 2005.

HAYES-ROTH, B.; *Adaptive Learning Guides*. In: IASTED Conference on Computers and Advanced Technology in Education, 2001.

HIRATA, I.; HÜBNER, J. F.; SICHMAN, J. S.; *Implementação de Protocolos de Interação no Ambiente SACI*. In: XIV Seminário de Computação, Blumenau. Anais, 2005. p. 103-113. Disponível em: <http://www.inf.furb.br/seminco/2005/artigos/106-vf.pdf>. Acessado em: janeiro de 2013.

HOLLAND, H. J.; *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. First edition: *The University of Massachusetts Institute of Technology*, 1975. First MIT Press edition, 1992.

HÜBNER, J. F.; SICHMAN, J. S.; *Saci Programming Guide*. USP - Universidade de São Paulo (USP), São Paulo, 2003.

HÜBNER, J. F.; SICHMAN, J. S.; *SACI: Uma ferramenta para a Implementação e Monitoração da Comunicação entre Agentes*. In: IBERAMIA/SBIA, Atibaia, São Paulo. Proceedings... São Carlos: ICMC/USP, 2000. Disponível em: <http://www.lti.pcs.usp.br/saci/doc/iberamia2000-saci.pdf>. Acessado em: janeiro de 2013.

IZARD, C.; *Emotion-cognition relationships and human development*. In: IZARD, C.; KAGAN, J.; ZAJONC, R.B. (Ed.). *Emotions, cognition, and behavior*. New York: Cambridge University Press, 1984.

JAQUES, P.; VICARI, R.; *PAT: Um agente pedagógico animado para interagir afetivamente com o aluno*. Revista Novas Tecnologias na Educação Maio/2005, vol. 3, no 1. Disponível em: <http://hdl.handle.net/10183/559>. Acessado em: janeiro de 2013.

JAQUES, P.A.; *Using an Animated Pedagogical Agent to Interact Affectively with the Student*. Tese de Doutorado em Ciência da Computação. Instituto de Informática, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2004.

JAVA; *Site oficial Java*. Disponível em: <http://www.java.com>. Acessado em: novembro de 2012.

JENNINGS, N. R.; *Agent-Oriented Software Engineering*. In: Proceedings of the 9<sup>th</sup> European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering, 1999.

JENNINGS, N. R.; *An agent-based approach for building complex software systems*. In: Communications of the ACM, 2001.

JENNINGS, N. R.; *Commitments and conventions: The foundation of coordination in multi-agent systems*. Knowledge Engineering, Review 8 (3), 223–250, 1993.

JENNINGS, N. R.; *Coordination Techniques for Distributed Artificial Intelligence*. In: O'HARE, G.M.P.; Jennings, N.R. (Eds.). Foundations of distributed artificial intelligence. New York: John Wiley & Sons, 1996, pp.187-210.

JENNINGS, N. R.; *On Agent-Base Software Engineering*. In: Artificial Intelligence, 277–296. Department of Electronics and Computer Science, University of Southampton, England, 2000. Disponível em: [www.econ.iastate.edu/tesfatsi/jennings.pdf](http://www.econ.iastate.edu/tesfatsi/jennings.pdf). Acessado em: janeiro de 2013.

JEON, H.; PETRIE, C.; CUTKOSKY, M. R.; *JATLite: A Java Agent Infrastructure with Message Routing*. In: Agent Technologies, Stanford Center for Design Research, 2000. Disponível em: <http://www-cdr.stanford.edu/~petrie/agents/jatlite.pdf>. Acessado em: janeiro de 2013.

JESS; *Jess, the Rule Engine for the Java™ Platform*. Disponível: <http://www.jessrules.com>. Acessado em: dezembro de 2011.

KAMPFF, A. J. C.; LIRA, A. F.; REITZ, D. S.; GOMES, F. J. L.; FONSECA, L. C. C.; MACHADO, N. C. F.; BERCHT, M.; *Relação entre o Perfil do Usuário e a Escolha do Perfil do Tutor*. In: Novas Tecnologias da Educação. V. 3 N° 1, Maio, 2005. Disponível em: <http://www.lume.ufrgs.br/bitstream/handle/10183/8010/000564671.pdf?sequence=1>, acessado em: janeiro de 2013.

KÜHLEIS, R., *CHATTERDÓRIS – Um agente pedagógico com interação em linguagem natural*. Monografia (graduação em Ciência da Computação) - Universidade de Santa Cruz do Sul, 2011.

KÜNZEL, M. F.; SILVA, A. K. da; FROZZA, R., BAGATINI, D., LUX; B.; *DIMI 3D - Agente Companheiro para um Ambiente Virtual de Aprendizagem*. 2. Ed. dos Anais do V Workshop-Escola de Sistemas de Agentes, seus Ambientes e aplicações - WESAAC 2011. Anais do evento disponível em: <http://www.das.ufsc.br/~jomi/pubs/2011/wesaac-v.pdf>. Acessado em: janeiro de 2013.

KÜNZEL, Michael F.; *Desenvolvimento de um Agente Companheiro para um Ambiente Virtual de Aprendizagem*. Curso de Ciência da Computação. Universidade de Santa Cruz do Sul (UNISC), 2010.

LABROU, Y.; FININ, T.; *A proposal for a new KQML specification*. UMB, Maryland, USA.

LACERDA, E. G. M.; CARVALHO, A. C.; *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*. Capítulo 03: Introdução a algoritmos genéticos. In: Porto Alegre, RS: Universidade do Rio Grande do Sul (UFRGS), 1999. p. 99–150. Disponível em: <http://www.dca.ufrn.br/~estefane/metaheuristicas/ag.pdf>. Acessado em: janeiro de 2013.

LAUREL, B.; *Interface Agents: Metaphors with Character*. In: Software Agents. Menlo Park, California: AAAI Press, 1997.

LEE, C. H. M.; CHENG, Y. W.; RAI, S., DEPICKERE, A.; *What Affect Student Cognitive Style In Development Of Hypermedia Learning System?* Computers & Education, 2004.

LEE, L C. H.; NWANA, S.; NDUMU D. T.; WILDE, P. D.; *The stability, scalability and performance of multi-agent systems*. In: BT Technol J Vol 16 No 3, 1998.

LENOX, T.; HAHN, S.; LEWIS, M.; PAYNE, T.; SYCARA, K.; *Task Characteristics and Intelligent Aiding*”, 2001. Disponível: <http://usl.sis.pitt.edu/ulab/pubs/SMC00LLHPS.html>, acessado em: novembro de 2011.

LESSER, V.; *Cooperative Multi-Agent Systems: A Personal View of the State of the Art*. In: IEEE Transactions on Knowledge and Data Engineering, Vol. 11, Nº 1, Janeiro/Fevereiro de 1999.

LESTER, J.; CONVERSE, S.; KAHLER, S.; BARLOW, T; STONE, B.; BHOGAL, R.; *The Persona Effect: Affective Impact of Animated Pedagogical Agents*. In: Proc. CHI '97 - Conference on Human Factors in Computing Systems, Atlanta, GA, 1997.

LIN, F.; NORRIE, D.; SHEN, W.; KREMER, R.; *A Schema-based Approach to Specifying Conversation Policies*. In Issues in Agent Communication, F. Dignum and M. Greaves(eds.). Lecture Notes in Artificial Intelligence. Springer, 1999.

LIU, J.; *Autonomous Agents and Multi-Agent Systems*. London: World Scientific, 2001.

LOINAZ, M. U.; *Sistemas inteligentes en el ámbito de la educación*. *Revista Iberoamericana de Inteligencia Artificial*, Valencia, n. 12, p. 5-12, 2001.

MAES, P.; *Agents that Reduce Work and Information Overload*. In: Communications of the ACM. New York, New York: ACM Press, 1994.

MAES, P.; *Intelligent Software: Programs That Can Act Independently Will Ease the Burdens that Computers Put on People*. IEEE Expert Systems, Dezembro de 1996.

MAINIERI, A. G.; FROZZA, R.; Schreiber, J.; Molz, K.; *Educational System based on Cognitive styles and/or Learning styles*. In: XXVII Annual Conference of the Cognitive Science Society. Stresa, julho de 2005.

MAMADOU, T. K.; SHIMAZU, A.; TATSUO, N.; *Software Agents and Quality of Services Issues in Distributed Systems*. Dissertation of Doctor Philosophy. Japan: Japan Advanced Institute of Science and Technology, 2000.

MARSELLA, S. C.; JOHNSON, W. L.; LABORE, C.; Interactive Pedagogical Drama”. In: International Conference on Autonomous Agents and Multiagents Systems, ACM Press, 2000.

MARTINS, J. G.; MARTINS, A.; MOÇO, S. S.; BARCIA, R. M.; A transformação do ensino através do uso da tecnologia na educação. In: XIX Congresso Nacional da Sociedade Brasileira de Computação (SBC), Rio de Janeiro, 1999.

MARTINS, J. G.; *Uma arquitetura baseada em agentes inteligentes para ambientes computacionais voltados à educação a distância*. Dissertação - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2002.

MERGEN, C. H. S., SCHREIBER, J. N. C.; *Agentes Pedagógicos Bayesianos*. In: SBIE 2005 - XII Simpósio Brasileiro de Informática na Educação, 2005, Juiz de Fora. Anais do SBIE, 2005.

MICHALEWICZ, Z.; *Genetic Algorithms + Data Structures = Evolution Programs*. Springer: Third, Revised and Extended Edition, 1998.

MITCHELL, M.; *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. A Bradford Book. The MIT Press. Cambridge, Massachusetts. London, England. Edition: Third Printing, 1998.

MOISSA H. E.; *Arquitetura de um Agente Identificador de Fatores Motivacionais e Afetivos em um Ambiente de Ensino e Aprendizagem*. In: Dissertação de mestrado para a obtenção do grau de Mestre em Ciência da Computação, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2001.

MORAN, J. M.; *Mudanças na Comunicação Pessoal: gerenciamento integrado da comunicação pessoal, social e tecnológica*. São Paulo. Editora: Paulinas, 1998.

MORAN, J. M.; *O que é um bom curso a distância?* Boletim do Programa um Salto para o Futuro – Integração das Tecnologias na Educação, TV Escola, 2002.

NETSCAPE; *Netscape ISP Homepage - Netscape Navigator*. Acessado em: dezembro de 2012. Disponível: <http://isp.netscape.com/>.

NICHOLSON, T.; *Optimization in industry, optimization techniques*. Longman Group Limited, v.1, 1971.

NORVIG, S.; RUSSELL, P.; *Inteligência Artificial*. 2ª Ed. Rio de Janeiro: Campus/Elsevier Editora Ltda, 2004.

NUNES, M. A. S. N.; DIHL, L. L.; FRAGA, L. M.; WOSZEZENKI, C. R.; OLIVEIRA, L.; FRANCISCO, D. J.; MACHADO, G. J. C.; NOGUEIRA, C. R. D.; NOTARGIACOMO, M. G. da; *Animated Pedagogical Agent in the Intelligent Virtual Teaching Environment*. Interactive Educational Multimedia, 2002.

NWANA, H. S.; *Software Agents: An Overview*. In: Knowledge Engineering Review, Cambridge University Press, 1996.

ODELL, J. J., PARUNAK, H. V. D., BAUER, B.; *Representing Agent Interaction Protocols In Uml*. Agent-Oriented Software Engineering. Berlin: Springer-Verlag, 2001. 121–140 P.

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

OLIVEIRA, J. R. F.; *O uso de algoritmos genéticos na decomposição morfológica de operadores invariantes em translação aplicados a imagens digitais*. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 1998.

OPERA; *Navegador Opera*. Disponível em: <http://www.opera.com/>. Acessado em dezembro de 2012.

PATIL, R.; FIKES, R. E.; PATEL-SCHNEIDER, P. F.; MCKAY, D.; FININ, T.; GRUBER, T.; NECHES, R.; *The DARPA Knowledge Sharing Effort: Progress Report*. 1992.

PEREIRA, A. S.; *Um Estudo de Aplicações de Ensino na Internet Orientada a Agentes*. Porto Alegre: UFRGS, 1997. (Trabalho Individual).

PEREIRA, A. T. C.; *AVA - Ambientes Virtuais de Aprendizagem em Diferentes Contextos*. Rio de Janeiro: Editora Ciência Moderna, 2007.

PERKINS, D.; *La Escuela Inteligente: Del Adiestramiento De La Memoria A La Edu Cacion De La Mente*. Editora: Gedisa, 1995.

PERUCIA, A. S., BERTHÊM, A. C., BERTSCHINGER, G. L., & MENEZES, R. R. C.; *Desenvolvimento de Jogos Eletrônicos: Teoria e Prática*. São Paulo: Novatec Editora Ltda (2005).

PETERS, O.; *Didática do ensino a distância: experiências e estágio da discussão numa visão internacional*. São Leopoldo, RS. Editora: Unisinos, 2003.

PICARD, R.; *Affective Computing*. Cambridge: MIT Press, 1997.

POPULAIRE, P.; BOISSER O.; SICHMAN J. S.; *Description et Implementation de Protocoles de Communication en Univers Multi-Agents*. Premières journées francophones IAD & SMA, Toulouse, 1993.

POSTGRESQL; *Sistema gerenciador de banco de dados objeto-relacional de código aberto*. Comunidade internacional PostgreSQL disponível em: [www.postgresql.org](http://www.postgresql.org). Comunidade brasileira PostgreSQL disponível em: [www.postgresql.org.br](http://www.postgresql.org.br), com documentação disponível em: <http://www.postgresql.org.br/docs>. Acessado em: novembro de 2012.

RANK, S.; *Toward Reusable Roleplayers Using an Appraisal-Based Architecture*. Applied Artificial Intelligence, 2005.

RAU, P-L. P.; CHOONG, Y-Y. & SALVENDY, G.; *A Cross Cultural Study On Knowledge Representation And Structure In Human Computer Interfaces*. International Journal Of Industrial Ergonomics; 117-129, 2004.

REEVES, C.R.; *Modern Heuristic Techniques for Combinatorial Problems*. John Wiley & Sons. Inc. NewYork, NY, 1993.

REIS, L. P.; *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico (Coordination in Multi-Agent Systems: Applications in University Management and Robotic Soccer)*. PhD in Electrical and Computers Engineering (Informatics/Artificial Intelligence/Robotics), Faculdade de Engenharia da Universidade do Porto (FEUP), Porto, Portugal, Julho de 2003.

REIS, L. P.; LAU, N.; OLIVEIRA, E. C.; *Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents*. Em Markus Hannenbauer, Jan Wendler and Enrico Pagello editores, Balancing Reactivity and Social Deliberation in Multi-Agent System – From RoboCup to Real-World Applications, Springer's Lecture Notes in Artificial Intelligence, Vol. 2103, pp. 175-197, Berlin, Portugal, 2001.

REZENDE, S. O. (Coord.); EVSUKOFF, A. G.; EBECKEN, N. F. F.; et al.; *Sistemas inteligentes: Fundamentos e aplicações*. Barueri, São Paulo: Manole, 2003.

ROSENSCHEIN, J. S.; GENESERETH, M. R.; *Deals among rational agents*. In: Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85), pages 91–99, Los Angeles, CA, 1985.

RUSSELL, S. J.; NORVIG, P.; *Artificial Intelllignce: a modern approach*. New Jersey: Prentice-Hall Inc., 1995.

RUSSELL, S.; NORVIG P.; *Artificial Intelligence: A Modern Approach*. Prentice Hall; 3ª. Edição, 2009.

SACI. *Simple Agent Communication Infrastucture*. Disponível: <http://www.lti.pcs.usp.br/saci> e acessado em: dezembro de 2011.

SAFARI; *Safari web browser*. Disponível em: <http://www.apple.com/safari/>. Acessado em: dezembro de 2012.

SANTOS, C. T. dos, FROZZA, R., DAHMER, A., GASPARY, L. P.; *Dóris – Um Agente de Acompanhamento Pedagógico em Sistemas Tutores Inteligentes*. In: XII Simpósio Brasileiro de Informática na Educação – SBIE – UFES, 2001.

SCHWAMBACH, M. M.; *Uma metodologia para desenvolvimento de sistemas baseados em agentes e objetos*. Dissertação de Mestrado em Informática - Departamento de Informática, Universidade Federal do Espírito Santo (UFES), Vitória, 2004.

SEARLE, John R.; *Expression and meaning - Studies in the Theory of Speech*. Acts. New York: Cambridge University Press, 1979.

SEARLE, John R.; *Os Actos de Fala - Um Ensaio de Filosofia da Linguagem*. Traduzido por: Carlos Vogt. Coimbra: Livraria Almedina, 1981. Tradução de: *Speech Acts - An Essay in the Philosophy of Language*, 1981.

SELF, J.; *Computational Mathematics: the missing link in Intelligent Tutoring Systems research?* NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES, 1993.

SELF, J. *The defining characteristics of intelligent tutoring systems research: ITSs care, precisely*. In: *International Journal of Artificial Intelligence in Education*, 1999.

SICHMAN, J. S.; DEMAZEAU, Y.; BOISSER, O.; *When can knowledge-based systems be called agents?* Anais do IX Seminário Brasileiro de Inteligência Artificial, Rio de Janeiro, Brasil, pp: 172-185, ISSN 0104-6500, 1992.

SICHMAN, J. S.; *Du Raisonnement Social Chez LES agents: Une Approach Fondée sur la Théorie de la Dépendence*. Tese de Doutorado, Institut National Polytechnique de Grenoble, Grenoble, France, 1995.

SILVA, A. K.; Frozza, R; Borin, M. P.; Molz, K.; Schreiber, J.; Lux, B.; Carvalho, A. B.; Baierle, J.; Kipper, L.; *O Método Clínico de Piaget como forma de Avaliação da Aprendizagem em um Sistema Tutor Inteligente com Agente Pedagógico*. XXI Brazilian Symposium on Computer in Education (SBIE). João Pessoa/PB, 2010. Disponível em: [http://www.ccae.ufpb.br/sbie2010/anais//WAvalia\\_files/78642\\_1.pdf](http://www.ccae.ufpb.br/sbie2010/anais//WAvalia_files/78642_1.pdf). Acessado em: janeiro de 2013.

SILVA, A. P. C.; *Aplicações de Sistemas Tutores Inteligentes na Educação a Distância: Possibilidades e Limites*. Relatório de pesquisa: TCC5056. Universidade Católica de Brasília (UCB), 2006.

SILVA, J. M. C. da; HECK G.; *Utilizando o questionário MSLQ para identificação de estados afetivos em um ambiente virtual de aprendizagem*. Itajaí, 2008.

SILVA, L. F. R. da; *DIMI : um agente selecionador de estratégias de ensino para sistemas tutores inteligentes*. Monografia (graduação) - Universidade de Santa Cruz do Sul, 2002.

SILVEIRA, R. A.; *Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino: JADE Java Agent framework for Distance learning Environments*. In: Tese de doutorado, Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, 2001. Disponível em: <http://hdl.handle.net/10183/17931>. Acessado em: janeiro de 2013.

SPERB, R. M.; *Agentes inteligentes difusos: uma ferramenta híbrida para exploração de processos espaciais em zonas costeiras*. Tese - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC), Florianópolis, 2002.

STERNBERG, R. J.; *Psicologia cognitive*. 4. ed. Porto Alegre: Artmed, 2008.

TANOMARU, J.; *Motivação, fundamentos e aplicações de algoritmos genéticos*. Anais do II Congresso Brasileiro de Redes Neurais, 1995.

THIRY, M.; *Uma arquitetura baseada em agentes para suporte ao ensino à distância*. Tese - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina (UFSC), Florianópolis, 1999.

TORREÃO, P. G. B. C.; *Project Management Knowledge Learning Environment: Ambiente Inteligente de Aprendizado para Educação em Gerenciamento de Projetos*. Dissertação de Mestrado. Universidade Federal de Pernambuco (UFPE), 2005.

TORSUN, I.S; *Foundations of Intelligent Knowledge-based systems*. London: Academic Press, 1995.

TOWNS, S. G.; VOERMAN, J. L.; CALLAWAY, C. B.; LESTER, J. C.; *Coherent gestures, locomotion, and speech in life-like pedagogical agents*. In IUI '98: Proceedings of the 3rd international conference on Intelligent user interfaces, New York, NY, USA. ACM Press, 1998.

VICARI, R. M., SANTOS, E. R., PRIMO, T. T., ROSSI, L. H. L.; *Objetos de Aprendizagem Baseados em Agentes*. Projeto OBAA - Relatório Técnico RT-OBAA-04 - Grupo Agentes e Ontologias. In: FAURGS – Fundação de Apoio da Universidade Federal do Rio Grande do Sul, 2009.

VILLAREAL, G. R. R., GIRAFFA, M. L.; *Arquiteturas de Sistemas Tutores Inteligentes*. Porto Alegre. Faculdade de Informática – PUCRS. Published by the Campus Global – FACIN – PUCRS, 2001.

VYGOTSKY, L.S.; *Thought and Language*. Cambridge. MA: MIT Press, 1962.

WEISS, G.; *Multiagent systems: A modern introduction to distributed artificial intelligence*, MIT PRESS, pp. 25-77, 1999.

WHITE, J. E.; *Telescript technology: The foundation for the electronic marketplace*. White paper, General Magic, Inc., 2465 Latham Street, Mountain View, CA 94040, 1994.

WOOLDRIDGE, M. J.; JENNINGS, N. R.; *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review, 1995.

WOOLDRIDGE, M.; *An Introduction to Multiagent Systems*. Department of Computer Science at the University of Liverpool, UK. 2. Ed., Editora: John Wiley & Sons, 2009.

WOOLF, B. P.; *Building intelligent interactive tutors*. Elsevier, 2007.

YU, E.; *Agent-oriented modelling: software versus world*. In: Proceedings of the Second International Workshop (AOSE-2301), Lecture Notes in Artificial Intelligence, Vol. 2424. Springer-Verlag, 2001.



ZAMBONELLI, F.; PARUNAK, H. V. D.; *Towards a Paradigm Change in Computer Science and Software Engineering: a Synthesis*. The Knowledge Engineering Review, 2004. Disponível: [http://www.agentgroup.unimore.it/didattica/cas/L1/Zambonelli\\_Revolution.pdf](http://www.agentgroup.unimore.it/didattica/cas/L1/Zambonelli_Revolution.pdf). Acessado em: janeiro de 2013.

## **Anexo A: Exercícios sobre o novo acordo ortográfico**

A seguir são apresentadas as questões elaboradas e inseridas no ambiente referente ao assunto Novo Acordo Ortográfico da Língua Portuguesa. Contudo, a resposta correta de cada questão está marcada com um “x”.

1. Apenas uma das frases a seguir está correta. Portanto, encontre e marque apenas a alternativa correta.

- Joãozinho estava chorando, pois perdeu sua bóia.
- O professor teve uma ótima idéia.
- Meu irmão não pôde sair de casa ontem.
- Escolas revêem forma de preparação dos alunos.
- Por favor, averigúe se a porta está fechada.

2. Apenas uma das opções a seguir está correta. Portanto, encontre e marque apenas a alternativa em que todas as palavras estão de acordo com o Novo Acordo Ortográfico da Língua Portuguesa.

- Micro-ônibus, Micro-ondas, Micro-empresa.
- Microônibus, Microondas, Microempresa.
- Microônibus, Microondas, Micro-empresa.
- Micro-ônibus, Micro-ondas, Microempresa.
- Todas as respostas anteriores estão erradas.

3. Apenas uma das opções a seguir está correta. Portanto, encontre e marque apenas a afirmação que está de acordo com o Novo Acordo Ortográfico da Língua Portuguesa.

- Não se usa o hífen com o prefixo CO, exceto as palavras em que o segundo elemento iniciar pela letra O.
- Apenas Brasil e Portugal aderiram ao novo acordo.
- O hífen foi abolido após os prefixos HIPER, INTER, NUPER e SUPER (sem exceção).

( ) Usa-se hífen se o prefixo terminar em vogal e a 2ª palavra começar por uma vogal diferente.

(x) Usa-se trema em nomes próprios estrangeiros.

4. Apenas uma das opções a seguir está correta. Portanto, encontre e marque apenas a alternativa em que todas as palavras estão de acordo com o Novo Acordo Ortográfico da Língua Portuguesa.

(x) Girassol, Pontapé, Antevéspera, Vice-presidente.

( ) Gira-sol, Ponta-pé, Ante-véspera, Vice-presidente.

( ) Gira sol, Ponta pé, Ante véspera, Vice presidente.

( ) Girassol, Pontapé, Antevéspera, Vicepresidente.

( ) Gira-sol, Pontapé, Ante-véspera, Vice-presidente.

5. Apenas uma das frases a seguir está correta. Portanto, encontre e marque apenas a alternativa correta.

( ) Quem é o seu co-orientador de mestrado?

(x) Semana passada Grüner fez cinquenta e dois anos.

( ) Empresários prevêem o crescimento da exportação da carne brasileira.

( ) Você pode por suas compras na mesa.

( ) A universidade convocou uma assembleia geral para discutir o início do ano letivo.

6. “Não, não sou sistemático. Mas há coisas que não dá para agüentar.” (Fonte: Quem agüenta? Por Ivan Angelo - Folha on-line). Questão: No trecho anterior, quais são as palavras acentuadas (acento agudo ou trema) que perderam o acento em função do acordo?.

( ) Sistemático, há, agüentar.

( ) Dá, há, agüenta.

(x) Agüentar, agüenta.

7. No trecho: "... fosse no dia seguinte, fosse dali a 60 anos, ter na ponta da língua uma frase de efeito com que impressionar a platéia". (Fonte: Palavras de adeus, por Eustáquio Gomes). Questão: a palavra 'platéia', no trecho acima, sofreu mudança na ortografia. Assinale a alternativa a seguir em que todas as palavras passaram pela mesma mudança:

- Café, pele, alem, ortografia.
- Tem, pode, jiboia, voce.
- Ideia, assembleia, heroico, paranoico.
- Atlantico, panico, assembleia, lampada.

8. Leia as afirmações a seguir e assinale a alternativa que NÃO contém erro de acentuação gráfica: "Pesquisas científicas detectaram vários danos causados pelo consumo de álcool." (Texto adaptado de Veja, 06/12/2006). Vejamos alguns:

- Diminuição da densidade mineral óssea.
- Danos ao fígado e aos neuronios.
- Baixa nos níveis de estrogênio nas meninas e de testosterona nos meninos.
- Baixa capacidade para por os estudos em dia.

9. No trecho: "Meu/minha querido(a) colega, estou entrando em parafuso: o microondas lá de casa queimou, meu filho está gordo e precisa de uma re-educação alimentar, minha empregada está com enjôo, e eu acho que estou tendo uma crise de paranóia. Preciso conversar contigo. Te espero na ante-sala ao meiodia. Amélia.". Questão: Quantos erros de ortografia você encontrou no recado escrito por seu colega? Assinale a alternativa correta:.

- 3.
- 4.
- 5.
- 6.
- 7.

10. Assinale a alternativa que contém as palavras corretamente formadas:

- Bem-vindo, pan-americana, sub-base, protomártir.
- Pré-histórico, mal-estar, mini-saia, prematuro.
- Auto-afirmação, autocrítica, excombatente, neolatinas.
- Pós-graduação, afro-descendente, malmequer, sub-aéreo.
- Autocontrole, anti-corrosivo, grão-mestre, aero-espacial.





### **Anexo C: Questões do questionário**

A seguir são apresentadas as questões do questionário elaborado e inserido no STI. O objetivo deste questionário foi fazer com que o estudante pudesse avaliar o mecanismo de comunicação desenvolvido.

1. Durante o tempo que você utilizou o ambiente, os dois agentes interagiram com você?

- Sim.
- Não.

2. Se os dois agentes interagiram, ocorreu algum caso de interagirem no mesmo momento?

- Sim.
- Não.

3. Você considera que as intervenções dos agentes ocorreram de forma coordenada?

- Sim.
- Não.

4. Você considera que as intervenções que ocorreram estavam de acordo com o objetivo de cada agente? (agente tutor - auxiliar e monitorar o estudante durante sua atuação no ambiente; agente companheiro - questões mais gerais).

- Sim.
- Não.

## **Anexo D: Termo de Consentimento Livre e Esclarecido**

### **Termo de Consentimento Livre e Esclarecido**

Você está sendo convidado para participar da pesquisa **“Proposta de Mecanismo de Comunicação entre Agentes Pedagógicos em um Ambiente Virtual de Aprendizagem”**.

Sua participação não é obrigatória. A qualquer momento você pode desistir de participar e retirar seu consentimento. Sua recusa não trará prejuízo em sua relação com o pesquisador ou com a instituição (Programa de Pós-Graduação em Sistemas e Processos Industriais – Mestrado. Universidade de Santa Cruz do Sul - UNISC).

O objetivo desta pesquisa é realizar a validação do mecanismo de comunicação entre os agentes pedagógicos Dóris e Dimi em um ambiente virtual de ensino-aprendizagem.

Sua participação nesta pesquisa consistirá em utilizar o ambiente e interagir com o conteúdo disponibilizado, sobre o domínio *Nova Ortografia da Língua Portuguesa*, e com os agentes pedagógicos *Dóris e Dimi*.

As informações obtidas através dessa pesquisa serão descritas de forma a assegurar o sigilo sobre sua identificação.

Você receberá uma cópia deste termo onde consta o telefone e o endereço institucional do pesquisador principal, podendo tirar suas dúvidas sobre o projeto e sua participação, agora ou a qualquer momento.

Declaro que entendi os objetivos, riscos e benefícios de minha participação na pesquisa e concordo em participar.

---

Geovane Griesang  
Aluno do Programa de Pós-Graduação em Sistemas e Processos Industriais da UNISC  
Responsável pela pesquisa  
Fone: (51) 3717.7632

---

Sujeito da pesquisa (nome e assinatura)

Validação realizada em: 29 de novembro de 2012 nos laboratórios de Informática da UNISC.