

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG

CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA - CEEI

COORDENAÇÃO DE PÓS-GRADUAÇÃO EM INFORMÁTICA - COPIN

IG IBERT BITTENCOURT SANTANA PINTO

**UM MODELO COMPUTACIONAL PARA A CONSTRUÇÃO
DE SISTEMAS EDUCACIONAIS ADAPTATIVOS E
SEMÂNTICOS**

ÁREA DE CONCENTRAÇÃO: CIÊNCIA DA COMPUTAÇÃO

LINHA DE PESQUISA: MODELOS COMPUTACIONAIS E COGNITIVOS

PROF. DR. EVANDRO DE BARROS COSTA

CAMPINA GRANDE, PB

AGOSTO/2009

Modelos e Ferramentas para a Construção de Sistemas Educacionais Adaptativos e Semânticos

Ig Ibert Bittencourt Santana Pinto

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Modelos Computacionais e Cognitivos

Prof. Evandro de Barros Costa, Dr (Orientador)

Campina Grande, Paraíba, Brasil


©Ig Ibert Bittencourt Santana Pinto, Agosto de 2009

**“MODELOS E FERRAMENTAS PARA A CONSTRUÇÃO DE SISTEMAS
EDUCACIONAIS ADAPTATIVOS E SEMÂNTICOS”**

IG IBERT BITTENCOURT SANTANA PINTO

TESE APROVADA COM DISTINÇÃO EM 14.08.2009


PROF. EVANDRO DE BARROS COSTA, D.Sc
Orientador


PROF. CLÁUDIO DE SOUZA BAPTISTA, Ph.D
Examinador


PROF^a MARIA DEL ROSARIO GIRARDI GUTIÉRREZ, Dr^a
Examinadora


PROF. SERGIO CRESPO COELHO DA SILVA PINTO, Dr.
Examinador

CAMPINA GRANDE – PB

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

P659m

Pinto, Ig Ibert Bittencourt Santana

Modelos d ferramentas para a construção de sistemas educacionais adaptativos e semânticos / Ig Ibert Bittencourt Santana Pinto.— Campina Grande, 2009.

220 f.: il.

Tese (Doutorado em Ciência da Computação) - Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.

Referências.

Orientador: Prof. Dr. Evandro de Barros Costa.

1. Inteligência Artificial – Sistemas Multiagentes 2. Web Semântica 3. Educação a Distância I. Título.

CDU 004.89(043)

Dedicatória

Costumo classificar todos os aspectos que estão relacionados à minha vida segundo três categorias, sendo elas a vida pessoal, profissional e espiritual. Isto faz com que eu possa buscar uma harmonia entre estes aspectos de forma pragmática. Desta forma, dedico esta tese aos três grandes mestres que pude identificar em cada uma destas áreas (que são meu pai Péricles Argolo Pinto - aspecto pessoal, meu orientador Evandro de Barros Costa - aspecto profissional e meu Sensei do Karatê Chicão - aspecto espiritual).

*Não teria sido mais fácil que ele tivesse me ensinado,
desde o princípio, a respiração correta?*

*“Um grande mestre”, respondeu-me, “tem que ser
ao mesmo tempo um grande educador,
pois para nós esses atributos são inseparáveis.*

*Se o aprendizado tivesse sido iniciado com os exercícios respiratórios,
jamais se convenceria da sua influência decisiva.*

*Era preciso que você naufragasse nos próprios fracassos
para aceitar o colete salva-vidas que ele lhe lançou. Cria-me ...*

Ele lê nas nossas almas muito mais do que estamos dispostos a admitir.”

Professor Komachiya, no livro “A Arte Cavalheiresca do Arqueiro Zen”

Busco meu equilíbrio cortejando a insanidade.

Agradecimentos

Primeiramente a Deus, por estar sempre me dando oportunidades que permitam um contínuo aprendizado.

Ao meu orientador e amigo, Evandro Costa, que confiou no meu trabalho e proporcionou novas oportunidades de exploração do conhecimento e da vida acadêmica, apoiando as minhas incursões às diversas searas do conhecimento.

À minha amada noiva, Kika, que está sempre ao meu lado, buscando me compreender nos momentos de falta e tentar entender (ou pelo menos me ouvir) as minhas deliberações excêntricas.

Aos meus pais (Péricles Argolo Pinto e Ula Sinai) e irmãos (Ives Samir, Igor Lucas e Ibsen Mateus) que sempre me apóiam a todo momento, motivando-me a romper novos desafios. Um agradecimento especial ao meu irmão *nerd* Ibsen Mateus que tem também trilhado este caminho acadêmico com maestria.

Aos membros do Grupo de Otimização da Web - GrOW (Lucas, Marlos, Pedro, Diego, Douglas, Társis Toledo, Heitor, Rafael, Henrique Pacheco e Társis Marinho) que foram fundamentais no desenvolvimento desta tese. Agradeço também aos outros membros do GrOW que auxiliaram também no final da tese, Kalyne Nóbrega, Marcos Hilário, Olavo e Djalma.

Ao Sensei Francisco “Chicão”, por permitir o meu desenvolvimento espiritual através dos treinamentos de Karatê.

Ao meu amigo Alan que está sempre me apoiando desde o meu mestrado.

Aos meus grandes amigos Rafael, Bruno e Petrus que também acompanharam todo este processo.

À sempre calma e alegre Aninha por simplesmente ser o anjo da copin e a resolvedora de praticamente todos os problemas da pós. Agradeço também ao Professor Bernardo Lula por apoiar e aprovar as minhas solicitações de recursos.

Ao amigo que eu fiz durante o doutorado, Seiji Isotani.

Aos pesquisadores que tive oportunidade de trabalhar e interagir no doutorado e que ajudaram a engrandecer o meu trabalho, Edílson Ferneda, Sérgio Crespo, Julita Vassileva, Jim Greer, Daniel Krause, Nicola Henze, Darina Dicheva, Seiji Isotani e Riichiro Mizoguchi.

Aos membros da banca avaliadora, Rosário Girardi, Sérgio Crespo e Cláudio Baptista, por terem dado excelentes sugestões para o melhoramento do trabalho.

Aos professores do Instituto de Computação da UFAL (Willy, Patrick, Leandro Dias, Leandro Sales, Rodrigo Paes e Henrique Pacca) por me ajudarem nas prévias, com vários questionamentos enriquecedores.

À doida da secretária da casa dos meus pais, Cristina, que está sempre de bem com a vida.

Aos meus cachorros Patron e Zenit Júnior, por terem me ouvido nas madrugadas de trabalho em minha casa.

Aos filósofos e/ou escritores Sócrates, Platão, Aristóteles, Bertrand Russell, William James, Osho, Eugen Herrigel, Kafka, Dan Millman, Gichin Funakoshi, Musashi, Paulo Coelho, Sun Tzu, Jostein Gaarder, Willian de Ockham e Émile Durkheim, que com seus livros e/ou pensamentos permitiram-me a exploração de novas perspectivas e formas de vê o mundo e a mim mesmo.

À Jesus Cristo, Sidarta, Dalai Lama, Allan Kardec, Confúcio, Lao-Tse, por terem transformado a humanidade e, conseqüentemente, a mim.

Aos fabricantes dos vinhos *Country Wine* e Quinta do Morgado, que tornaram várias das minhas madrugadas de trabalho mais leves.

Aos músicos, compositores e/ou bandas Legião Urbana, Evanescence, Cramberries, Racionais, Raul Seixas, Pink Floyd, Pearl Jam, Charlie Brown Jr., Gal Costa, Cássia Eller, Los Hermanos, Bob Marley, Coldplay, Natiruts, Tan Dun, Janis Joplin, Paulinho Moska e Lenine, por estarem presentes em praticamente todos os momentos de minha tese.

Aos roteiristas, diretores e produtores dos filmes 300, Herói, Mestre das Armas, Poder Alem da Vida e Oyama, por me darem força em momentos decisivos e importantes.

As empresas aéreas de baixo custo e albergues que me deram a oportunidade de conhecer lugares fantásticos.

Aos estranhos que interagi durante minhas viagens mostrando diferentes culturas e nos ensinando a nunca termos preconceitos.

Um agradecimento especial aos mesquinhos, hipócritas e com falta de visão que eu tive a oportunidade de conhecer durante o doutorado, mostrando-me como eu nunca devo ser.

Enfim, a todos que de alguma forma colaboraram para o desenvolvimento desta tese.

Resumo

A presente pesquisa está relacionada à concepção e construção de sistemas educacionais mediados por computador dotados de adaptatividade e baseados na Web semântica, integrando técnicas e tecnologias provenientes da Inteligência Artificial e Web Semântica. Nesse sentido, esta tese apresenta uma solução para viabilizar a construção efetiva dos sistemas em pauta, consistindo especificamente na integração de ontologias, agentes e serviços semânticos. Para tanto, foram estabelecidos os seguintes objetivos: 1) propor um modelo conceitual para construção de tais ambientes educacionais; 2) propor e implementar um modelo computacional baseado em ontologias, agentes e serviços semânticos e 3) propor e implementar um modelo de infraestrutura para interoperabilidade entre aplicações educacionais. Para avaliação da proposta foram definidos três estudos de casos, abrangendo os domínios de Matemática e Medicina. Os resultados alcançados mostraram-se adequados, tendo como contribuição a concepção e desenvolvimento de tais modelos, as quais correspondem: i) à proposição dos modelos para ambientes educacionais adaptativos e semânticos; ii) à garantia do desenvolvimento e manutenção de ambientes educacionais adaptativos e semânticos de forma mais fácil; iii) à possibilidade de produção em larga escala de tais ambientes e para as diversas áreas do conhecimento e iv) à possibilidade de permitir que instituições de ensino possam cooperar na produção de materiais educacionais e no acesso aos estudantes que necessitem deste tipo de tecnologia, tanto no Brasil quanto em outros países.

Palavras chave: Ambientes Educacionais baseados na Web Semântica, Adaptação, Sistemas Multiagentes

Abstract

This research examines the design and effective development of adaptive and semantic Web-based educational systems, integrating artificial intelligence techniques and semantic Web-based technologies. In this sense, this research presents a solution in order to ensure the effective construction of such educational systems, which means the integration of ontologies, agents and semantic Web services. As a consequence, the following goals are 1) to propose a conceptual model to build such educational environments; 2) to propose and to implement a computational model based on ontologies, agents, and semantic Web services; 3) to propose and to implement an interoperability model between educational applications. Moreover, results obtained from three case studies (within Math and Medicine domains) are provided to evaluate the feasibility of the thesis. The outcomes achieved were the conception and development of such models and the contributions are summarized as follows: i) the proposition of models to adaptive and semantic Web-based educational environments; ii) to ensure the development and maintenance of such educational systems in an easy way; iii) to provide an structure to large-scale production of such environments and to different domains and iv) to provide the capability for universities to cooperate in the production of learning resources and access to students who needs this kind of technology as in Brazil as in abroad. **Keywords:** Semantic Web-based Educational Systems, Adaptation, Ontologies, Multiagent Systems

Conteúdo

1	Introdução	1
1.1	Motivação e Contextualização da Pesquisa	1
1.2	Abordagem da Pesquisa	8
1.2.1	Questões de Pesquisa	8
1.2.2	Objetivos da Pesquisa	12
1.2.3	Aspectos Metodológicos	13
1.3	Sumário das Contribuições	15
1.4	Organização do Documento	16
2	Web Semântica em Ambientes Educacionais	17
2.1	Web Semântica	17
2.1.1	O papel das ontologias na Web Semântica	34
2.1.2	O papel dos SWS na Web Semântica	37
2.1.3	O papel dos Agentes na Web Semântica	41
2.2	O papel da Web Semântica nos Ambientes Educacionais	44
2.2.1	Evolução dos Ambientes Educacionais	45
2.2.2	Ambientes Educacionais baseados na Web Semântica	49
2.3	Conclusão	50
3	Trabalhos Relacionados	51
3.1	SWSOA	52
3.2	Ambiente com MAS e SOA	54
3.3	ACOA	57
3.4	Integrated Collaboration Framework	59

3.5	<i>SQU</i>	60
3.6	Semantic e-Learning Framework	61
3.7	Multitutor	62
3.8	EBW	63
3.9	Semantic Integration Framework	65
3.10	Framework baseado em Serviços	66
3.11	Tabelas Comparativas	67
3.12	Conclusão	71
4	Modelos Propostos	72
4.1	Modelo Conceitual - MASSAYO-RM	72
4.1.1	Papéis	76
4.1.2	Recursos Educacionais	78
4.1.3	Interface	78
4.1.4	Ambientes Educacionais baseados na Web Semântica	79
4.1.5	Ambiente Web Semântico	83
4.1.6	Análise do Modelo Conceitual	83
4.2	Modelo Computacional - MASSAYO-CM	86
4.2.1	Camada do Controlador	88
4.2.2	Camada do Framework	110
4.2.3	Camada da Aplicação	118
4.2.4	Camada da Apresentação	118
4.3	Modelo de Infra-estrutura para Interoperabilidade - MASSAYO-IM	119
4.3.1	Repositório de Ontologias	121
4.3.2	Servidor para Interoperabilidade	121
4.3.3	Ambientes Educacionais	122
4.4	Guia de Utilização - MASSAYO-P	124
4.5	Conclusão	127
5	Avaliação dos Modelos Propostos	128
5.1	Estudo de Caso 1 - Sistema Tutor Inteligente em Fração	128
5.1.1	Especificação	129

5.1.2	Modelagem	133
5.1.3	Implementação e testes	142
5.2	Estudo de Caso 2 - Ambiente de Educação a Distância em MMC	146
5.2.1	Especificação	147
5.2.2	Modelagem	150
5.2.3	Implementação e testes	159
5.3	Estudo de Caso 3 - Ambiente Colaborativo em Medicina	160
5.3.1	Implementação e Testes	160
5.4	Estudo de Caso 4 - Interoperabilidade	164
5.5	Discussão	167
5.5.1	MASSAYO-RM	167
5.5.2	MASSAYO-CM	168
5.5.3	MASSAYO-IM	171
5.5.4	Outras avaliações	172
5.6	Conclusão	174
6	Conclusões e Trabalhos Futuros	175
6.1	Contribuições	176
6.2	Trabalhos Futuros	178
6.2.1	MASSAYO-RM	178
6.2.2	MASSAYO-CM	178
6.2.3	MASSAYO-IM	179
6.2.4	Outros Trabalhos	179
A	Tecnologias Utilizadas	198
A.1	Inteligência Artificial	198
A.1.1	Jade	198
A.2	Web Semântica	207
A.2.1	OWL	207
A.2.2	Implementações de SWRL	209
A.2.3	Protégé	209
A.2.4	Modelo Mathema Revisitado	218

A.2.5	Adaptações e Integrações ao Modelo MATHEMA	222
-------	--	-----

Lista de Figuras

1.1	Integração entre sistemas de EaD e IAEd.	5
1.2	Exemplo de Motivação para Interoperabilidade.	11
2.1	Páginas Web clássica e seus relacionamentos.	18
2.2	Páginas descrevendo a primeira idéia da Web Semântica.	19
2.3	Tecnologias da Web Semântica para processar uma consulta médica.	20
2.4	Tecnologias da Web Semântica para a criação e avaliação de projetos de pesquisa.	21
2.5	Camadas da Web Semântica inicialmente proposta.	22
2.6	Camadas da Web Semântica Revisitada.	22
2.7	Ilustração de um Recurso um WWW.	23
2.8	Arquitetura do RDF.	27
2.9	Classes, propriedades e relacionamentos em RDF-S.	28
2.10	Exemplo de Grafo RDF.	29
2.11	Formalismos providos para raciocínio.	33
2.12	Plano de Conhecimento provido pelas Ontologias à Internet.	34
2.13	Compartilhamento e Integração entre Ontologias.	35
2.14	Mapeamento/Alinhamento entre Ontologias.	36
2.15	Modelo da interação entre cliente, servidor e registro UDDI.	38
2.16	Serviço Web + Web Semântica= Serviço Web Semântico.	40
2.17	Estrutura proposta de um serviço semântico em OWL-S.	40
2.18	Visão de um Sistema Multiagente.	43
2.19	Visão da Internet, podendo ser estendida e valorada com Ontologias.	44
2.20	Interação Abstrata em Sistemas Educacionais.	45

2.21	Interação em Sistemas Educacionais com descrição de um ambiente.	46
2.22	Abordagem para Ambientes de Educação a Distância Adaptativos.	48
2.23	Evolução dos AAAC.	48
3.1	Arquitetura proposta baseada em Agentes e Serviços.	53
3.2	Arquitetura proposta baseada em Agentes e Serviços.	55
3.3	Arquitetura multicamadas proposta baseada em Agentes e Componentes. .	58
3.4	Arquitetura integrando Agentes, Serviços Web Semânticos e Ontologias. . .	59
3.5	Arquitetura abordando os serviços Web da Universidade <i>Sultan Qaboos</i> . . .	60
3.6	Arquitetura de um Framework para ambientes de <i>e-learning</i> semânticos e inteligentes.	61
3.7	Arquitetura do Multitutor.	63
3.8	Arquitetura para garantir a interoperabilidade entre ambientes de gerencia- mento de conteúdo.	64
3.9	Arquitetura do Framework para Integração Semântica.	65
3.10	Arquitetura do Framework baseada em serviços e modelos em RDF e OWL. .	67
4.1	Modelo Conceitual para Ambientes Educacionais Adaptativos e Semânticos. .	76
4.2	Tipos de Adaptação de Interface.	79
4.3	Modelo Conceitual para o ambiente SQU.	84
4.4	Modelo Conceitual para o ambiente com MAS e SOA.	84
4.5	Modelo Conceitual para o ambiente <i>Integrated Collaborative Framework</i> . .	85
4.6	Modelo Conceitual para o ambiente EBW.	86
4.7	Arquitetura Multicamadas.	87
4.8	Ontologias e suas relações e regras de mapeamentos.	89
4.9	Ontologia do modelo do domínio - Visão multidimensional.	90
4.10	Modelagem da ontologia do domínio - Visão dos currículos.	91
4.11	Modelagem da interação estudante-sistema na Ontologia do Estudante. . . .	92
4.12	Objetivos de aprendizagem da Ontologia do estudante.	93
4.13	Parte da Modelagem da Ontologia do Modelo Pedagógico.	94
4.14	Parte da Modelagem da Ontologia de Páginas Web.	96
4.15	Taxonomia de Métodos de Adaptação Hipermissão representada em OWL-Lite. .	97

4.16	Técnicas de Adaptação descritas na ontologia de adaptação (a).	98
4.17	Técnicas de Adaptação descritas na ontologia de adaptação (b).	99
4.18	Uma parte da Ontologia de GAIA.	100
4.19	Uma parte da Ontologia do JADE.	104
4.20	Uma parte da Ontologia do ForBILEAgent.	105
4.21	Parte da Ontologia de Aplicação.	108
4.22	Parte da ontologia provendo informações sobre as ontologias alinhadas.	109
4.23	Arquitetura baseada em agentes da camada de Framework.	111
4.24	Diagrama de Classe dos agentes do framework.	113
4.25	Hierarquia com Classificação para SWS.	117
4.26	Arquitetura para a camada de aplicação.	118
4.27	Exemplo de Apresentação Padrão(a).	119
4.28	Exemplos de Apresentação Padrão (b e c).	119
4.29	Arquitetura para suportar interoperabilidade entre ambientes educacionais.	120
4.30	Diagrama de Sequência descrevendo o registro e conexão do MASSAYO.	123
4.31	MASSAYO-P: Guia de Utilização do MASSAYO.	124
5.1	Diagrama de casos de uso do FraW.	130
5.2	Diagrama de classes conceitual do FraW.	131
5.3	Modelo Conceitual do FraW, segundo MASSAYO-RM.	133
5.4	Arquitetura do FraW.	134
5.5	Diagrama Web do sistema FraW.	136
5.6	Ontologia descrevendo os currículos.	138
5.7	Parte da Ontologia descrevendo os recursos presentes em um currículo.	138
5.8	Parte da Ontologia descrevendo a interação estudante-sistema-estudante.	139
5.9	Parte da Ontologia com estrutura curricular.	140
5.10	Parte da Ontologia GAIA com os recursos gerados a partir das regras de mapeamento.	140
5.11	Parte da Ontologia de GAIA.	141
5.12	Parte da Ontologia ForBILEAgent.	142
5.13	Passos 1 e 2 de resolução de problemas pelo FraW.	144

5.14	Passos 3 e 4 de resolução de problemas pelo FraW.	144
5.15	Último passo de resolução de problemas pelo FraW.	145
5.16	Passos 1 e 2 de resolução de problemas no FraW pelo estudante.	145
5.17	Passos 3 e 4 de resolução de problemas no FraW pelo estudante.	146
5.18	Último passo de resolução de problemas no FraW pelo estudante.	146
5.19	Diagrama de Casos de Uso do Ambiente de EaD Adaptativo de MMC.	148
5.20	Diagrama de Classes Conceitual do Ambiente de EaD Adaptativo de MMC.	149
5.21	Arquitetura descrevendo as ferramentas presentes no Ambiente de EaD Adaptativo de MMC.	151
5.22	Diagrama Web do ambiente de Ead Adaptativo de MMC.	152
5.23	Parte da ontologia do dominio com os curriculos de MMC.	154
5.24	Parte da ontologia contendo uma unidade pedagogica com os seus recursos.	154
5.25	Parte da ontologia do estudante com informação cognitiva para um usuário.	155
5.26	Parte da ontologia pedagógica contendo o sequenciamento do sistema (a).	155
5.27	Parte da ontologia pedagógica contendo o sequenciamento do sistema (b).	156
5.28	Parte da ontologia de GAIA com a especificação de agentes.	156
5.29	Parte da ontologia de GAIA com a especificação do ambiente.	157
5.30	Parte da ontologia de GAIA com a especificação do ambiente.	157
5.31	Parte da ontologia ForBILEAgent descrevendo um agente tutor.	158
5.32	Parte da ontologia ForBILEAgent detalhando um comportamento executado por um agente tutor.	158
5.33	Fluxo de execução para recomendação de usuários.	160
5.34	Fluxo de Interações entre os agentes no ambiente de Medicina (b).	162
5.35	Tela do ambiente de Medicina com um recurso sendo mostrado.	162
5.36	Tela do ambiente de Medicina com descrição segundo a ontologia de páginas Web.	163
5.37	Instâncias da Ontologia de páginas Web para o ambiente de Medicina.	163
5.38	Ontologia de aplicação com instâncias do ambiente de FraW.	164
5.39	Ontologia de aplicação com instâncias do ambiente de MMC.	165
5.40	Diagrama de Sequência descrevendo solicitação de ajuda externa.	165
5.41	Diagrama de Sequência entre dois agentes interoperando.	166

5.42	Parte da Interação de agentes para descobrir ambiente para interoperar. . . .	167
5.43	Tela para adicionar ontologia (a) e Tela com ontologias cadastradas (b). . .	169
5.44	Tela para especificar passo.	169
5.45	Tela com vários passos para construção de assistente.	170
5.46	Tela com assistente criado.	170
5.47	Gráfico de desempenho de um agente percebendo o ambiente no MASSAYO.	172
5.48	Gráfico de desempenho do Sesame (esquerda) e do Protégé (direita).	174
A.1	Modelo de referência da FIPA para plataformas de agentes.	199
A.2	Arquitetura JADE.	200
A.3	Estendendo um agente JADE.	201
A.4	Estendendo um comportamento Jade.	202
A.5	Tela Inicial do Protégé.	210
A.6	Tela de Salvar do Protégé.	211
A.7	Editor de Classes e Propriedades do Protégé.	212
A.8	Tela do Protégé com especificação em OWL-DL.	212
A.9	Tela de Indivíduos do Protégé.	213
A.10	Gerar a classe com o "esquema" de toda a ontologia OWL.	215
A.11	Mapeamento Ontologia-Objeto através do Protégé.	215
A.12	Tela do Protégé com o Mapeamento Ontologia-Objeto.	216
A.13	Pacote com as classes do Mapeamento Ontologia-Objeto.	216
A.14	Modelo Mínimo do MATHEMA	218
A.15	Visão tridimensional do modelo do domínio no Mathema.	220
A.16	Estrutura pedagógica do conhecimento do domínio no Mathema.	221
A.17	Modelo de Ambiente de Aprendizagem baseado em agentes no Mathema. .	221
A.18	Arquitetura do Mathema.	221
A.19	Arquitetura de um Agente Tutor no Mathema: visão interna	222
A.20	Estrutura pedagógica do conhecimento do domínio adaptada	224
A.21	Arquitetura de um Agente Tutor adaptada	224
A.22	Arquitetura do Agente Mediador	225

Lista de Tabelas

3.1	Tabela Comparativa dos Trabalhos Relacionados - Web Semântica.	69
3.2	Tabela Comparativa dos Trabalhos Relacionados - Engenharia de Software.	70
3.3	Tabela Comparativa dos Trabalhos Relacionados - Inteligência Artificial.	71
4.1	Mapeamento entre as ontologias Domain e GAIA utilizando SWRL.	102
4.2	Mapeamento entre as ontologias GAIA, JADE e <i>ForBILEAgent</i> utilizando SWRL.	106
4.3	Regras de mapeamento entre as ontologias <i>ForBILEAgent</i> e <i>Application</i>	110
5.1	Dados do Domínio de Matemática - Fração	134
5.2	Dados do Módulo Pedagógico - FraW	135
A.1	Construtores de OWL	209

Lista de Códigos Anotados

5.1	Regra de mapeamento GAIA-JADE/ForBILEAgent no FraW.	141
A.1	Associando um comportamento a um agente.	203
A.2	Cadastro nas Páginas Amarelas.	203
A.3	Cadastro nas Páginas Amarelas.	204
A.4	Comportamento (<i>Behaviour</i>) <i>MessageReceiver</i>	205
A.5	Agente que envia mensagens.	207
A.6	Carregamento e Salvamento de um arquivo OWL.	214
A.7	Classe EMATHEMAFactory gerado através da API do Protégé-OWL. . . .	217
A.8	Recuperação de indivíduo da classe <i>PartiionDomain</i>	217
A.9	Recuperação de indivíduos através de SPARQL.	218

Capítulo 1

Introdução

Para onde quer que fores, vai todo, leva junto teu coração.

Confúcio

O domínio de estudo abordado na presente pesquisa é o da concepção e desenvolvimento de ambientes educacionais adaptativos mediados por computador sob a perspectiva da Web Semântica, situando-se na linha de pesquisa Modelos Computacionais e Cognitivos da Copin¹/UFCEG. Trata-se de um tema inerentemente interdisciplinar, envolvendo, neste trabalho, uma conjunção de aspectos relacionados a Inteligência Artificial, Web Semântica e Engenharia de Software, aplicados aos ambientes educacionais em pauta².

1.1 Motivação e Contextualização da Pesquisa

Uma das motivações gerais para a presente pesquisa vem da demanda crescente por ambientes educacionais mediados por computador, os quais têm sido desenvolvidos e utilizados para proporcionar, aos atores envolvidos, mais oportunidades de interação, visando proporcionar diferentes e efetivas situações de aprendizagem. Estes ambientes têm sido criados e explorados há mais de meio século e discutidos em diversas

¹Coordenação de Pós-Graduação em Informática.

²As imagens descritas que não possuem fonte foram produzidas pelo autor.

conferências especializadas. Entretanto, importantes lacunas, assim como novos requisitos, ainda são identificadas no contexto da pesquisa e desenvolvimento desses ambientes.

Em uma perspectiva mais ampla, pode-se constatar requisitos complexos, revelados em iniciativas que buscam caracterizar desafios para a área de Computação. Um dos exemplos desse tipo de trabalho pode ser observado no documento criado pela Sociedade Brasileira de Computação - SBC [de Leon F. de Carvalho et al., 2006], com o título “Grandes Desafios da Computação 2006 - 2016”. Dentre os tópicos que foram apresentados neste documento, destaca-se o de número 5, chamado “Acesso Participativo e Universal do Cidadão Brasileiro ao Conhecimento”. Neste desafio, discute-se a utilização de tecnologias para garantir que os cidadãos tenham acesso ao conhecimento de forma personalizada e participativa, dando destaque a problemas como a diversidade populacional e as diferentes necessidades de cada cidadão. Desta forma, as tecnologias devem levar em consideração a individualização e a adaptação para cada cidadão/usuário com vistas a garantir o acesso ao conhecimento de forma personalizada.

Este desafio busca dar direcionamentos para transpor estas barreiras através da produção de tecnologias que garantam um acesso universal e participativo do cidadão brasileiro ao conhecimento, onde cada cidadão não é um usuário passivo, pois participa ativamente da construção do seu próprio conhecimento. Assim, sugerem-se plataformas tecnológicas³ que permitam e motivem a participação dos usuários de forma ativa no processo de produção deste conhecimento. Nesse contexto, surge frequentemente a necessidade de comunicação entre tais plataformas.

Na perspectiva acima, surgem problemas tecnológicos para vencer os desafios envolvidos⁴. Dentre eles, destacam-se [de Leon F. de Carvalho et al., 2006]:

- Criação de Plataformas de *e-Learning* que permitam a integração e disponibilização de ferramentas de comunicação: um ambiente educacional é

³Uma plataforma, sob uma perspectiva da computação, equivale à uma tecnologia para definir a infra-estrutura a ser utilizada com um determinado fim. Por exemplo, o provimento de plataformas educacionais para comportar software para Web [Mercer, 2008], TV Digital [Pagani, 2003] e/ou Computação Móvel/Pervasiva [Adelstein et al., 2004].

⁴Não é interesse desta tese abordar problemas referentes a aspectos econômicos, sociais, políticos, entre outros.

utilizado por vários usuários com papéis diferenciados, fazendo com que haja a necessidade de disponibilização de várias ferramentas para garantir a comunicação tanto entre usuários com o mesmo papel quanto com diferentes papéis. Frisa-se ainda que tais plataformas devam ser extensíveis de forma a possibilitar a evolução de requisitos para garantir e respeitar a diversidade populacional;

- Infra-estrutura para prover serviços personalizados aos cidadãos: tais serviços providos aos cidadãos devem utilizar mecanismos que garantam a personalização e adaptação dos recursos educacionais. Além disso, deve haver uma infra-estrutura que proporcione a interoperabilidade dos mesmos para utilização em outros ambientes educacionais;
- Projeto e implementação de estruturas de ontologias que possibilitem interoperabilidade entre áreas do conhecimento e integração de pessoas: ambientes educacionais, apesar de poderem possuir uma estrutura genérica, não garantem a representação de diversas áreas do conhecimento. Logo, estas plataformas devem possibilitar a interoperabilidade entre ambientes educacionais de modo a proporcionar o reuso de conhecimento⁵ e maior integração de pessoas.

Outras comunidades científicas também discutiram desafios computacionais da(s) próxima(s) década(s), como por exemplo, a Americana e a Britânica. Tais discussões foram feitas em 2002⁶ e 2004⁷, respectivamente. Além disso, outros desafios com objetivos de abordar os desafios da computação na educação também foram discutidos e criados (e.g. [Des, 2008]).

Até aqui, percebe-se a necessidade de criar ambientes educacionais atendendo aos seguintes requisitos: i) que sejam extensíveis, ii) que garantam o acesso participativo e universal do cidadão, iii) que possuam uma estrutura que levem em consideração características de cada cidadão, tornando o sistema adaptável aos mesmos. Além disso,

⁵Entende-se por reuso de conhecimento, no contexto desta tese, o processo de adquirir um conhecimento, empacotá-lo para reuso e distribuí-lo [Markus, 2001].

⁶Desafio cujo nome é *Grand Research Challenges in Computing*, através do desafio *Provide a Teacher for Every Learner* [Aspray et al., 2002].

⁷Desafio cujo nome é *Grand Challenges in Computing Workshop* [Hoare and Milner, 2004], através do desafio *Memories for Life* [Fitzgibbon and Reiter, 2003].

tal ambiente educacional deve permitir iv) a integração de ferramentas de comunicação e v) interoperabilidade entre os ambientes educacionais, garantindo o reuso de conhecimento e a integração de atores.

De fato, a abordagem aos desafios e requisitos mencionados pode ser identificada em diferentes momentos e os aspectos de apoio à atividade educacional, que já ocorre há mais de meio século. Nesse tempo, diversas abordagens, atendendo a diferentes propósitos, têm sido exploradas, conduzindo à construção do que se passou a denominar ambientes educacionais mediados por computador [Costa, 1997]. Assim, duas grandes linhas de pesquisa em ambientes educacionais⁸ mediados por computador exploradas na comunidade da área e que estão diretamente relacionadas com as características necessárias para atender aos desafios mencionados são: i) ambientes de educação a distância (EaD)⁹, que visam a interação entre estudantes, tutores e professores, fazendo uso da tecnologia da informação para garantir esta comunicação [Silva, 2003] e ii) ambientes de inteligência artificial e educação¹⁰, que visam fazer uso de técnicas provenientes da inteligência artificial para assegurar interações personalizadas, além de efetivar a aprendizagem e a resolução de problemas [Self, 1995].

Por um lado, os ambientes de educação a distância possuem diversas características como ensino em múltiplos domínios (através da apresentação de materiais/objetos de aprendizagem), *feedback* simples e configurado *a priori*, suporte colaborativo, entre outros. Por outro lado, os ambientes de inteligência artificial e educação provêm atividades personalizadas tanto para usuários quanto para grupos de usuários, através da utilização de informações dos mesmos. Além disso, estes ambientes dão apoio em atividades de estudantes, tutores e professores, como resolução de problemas, acompanhamento de atividades, avaliação, entre outras.

Apesar das características estarem explicitamente declaradas, diversas limitações são comumente identificados nestes tipos de ambientes educacionais. De um lado, os ambientes clássicos de educação a distância falham na falta de controle de avaliação

⁸Há outras abordagens existentes, como Aprendizagem Móvel e/ou Pervasiva [Denk et al., 2007][Bayoumi, 2007][Loidl, 2006] e Aprendizagem para TV Digital [Lopez-Nores et al., 2006], entretanto tais abordagens não são focalizadas na tese em questão.

⁹Também conhecidos como e-Learning ou LMS - *Learning Management System*.

¹⁰Também conhecidos como AIED - *Artificial Intelligence in Education*.

personalizada e na falta de personalização/adaptação às características dos usuários presentes no ambiente, como estudantes, tutores, autores, monitores, entre outros [Brooks et al., 2006]. Do outro lado, os ambientes AIED, em sua grande maioria, têm o foco estabelecido na resolução de problemas de aprendizagem e nas atividades sendo trabalhadas de forma individualizada/personalizada ao modelo do aprendiz, fazendo com que tais características sejam de domínios restritos e que possuam um alto custo de desenvolvimento e manutenção [Murray, 2003][Manuel Rodrigues and Santos, 2005].

Desta forma, estas limitações, identificadas em ambientes de educação a distância tradicionais, fazem com que haja uma convergência natural entre os mesmos com ambientes de inteligência artificial e educação, criando assim os ambientes conhecidos como Ambientes de Educação a Distância Adaptativos ou *Adaptive e-Learning* [Garcia-Barrios, 2006], como mostrado na Figura 1.1.

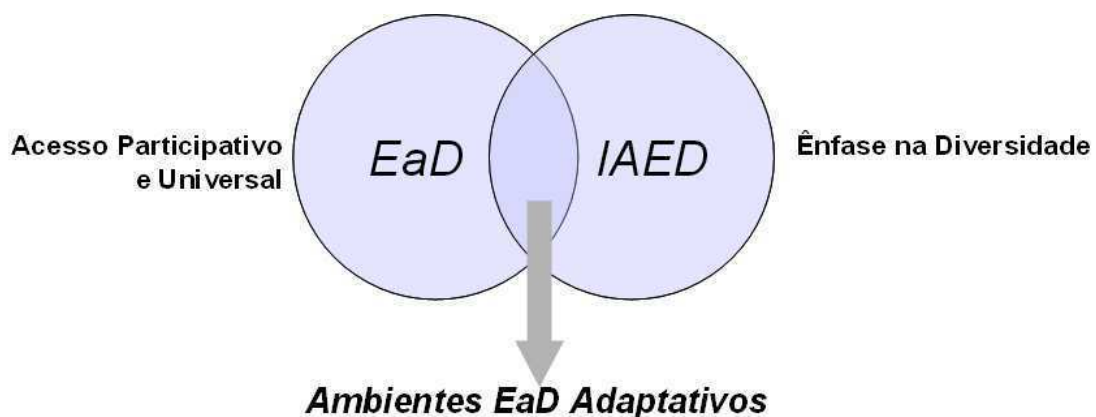


Figura 1.1: Integração entre sistemas de EaD e IAEd.

Entende-se por ambiente educacional adaptativo (AEA), os sistemas que alteram suas características para atender às necessidades dos usuários do sistema [Oppermann, 1994]. Em outras palavras, é a capacidade do ambiente mudar suas configurações, objetivando ajustar o ambiente com vistas a melhorar o aprendizado dos estudantes que estão interagindo com o ambiente. Com isso, o sistema faz a utilização de dados dos usuários e ferramentas para prover a adaptação através de técnicas provenientes da inteligência artificial.

Apesar de sistemas educacionais adaptativos serem foco de estudo de diversos grupos de pesquisa [Madmuc, 2008][AIRES, 1992][IIS, 2008][KRRRG, 2008][IS, 2008],

diversos desafios ainda existem que vão desde a definição de mecanismos de inteligência artificial, especificação, desenvolvimento, manutenção e evolução destes tipos de sistemas bem como questões de interoperabilidade entre aplicações, automatização de atividades e reuso de software e conhecimento [Brooks et al., 2006][Hauger and Köck, 2007][Garcia-Barrios, 2006][del Puerto Paule Ruiz et al., 2008][Murray, 2003][Manuel Rodrigues and Santos, 2005]. Além disso, tais desafios estão relacionados tanto com a necessidade de decomposição dos problemas quanto com a tentativa de representar informações de uma forma que máquinas possam entendê-las e manipulá-las [Bittencourt et al., 2008d].

Dentre as áreas que buscam abordar problemas como representação de conhecimento e processamento por máquinas, destaca-se a Web Semântica. A Web Semântica estende a Web clássica de forma a permitir a estruturação semântica de páginas Web, provendo assim suporte para que tanto agentes humanos quanto agentes artificiais possam entender o conteúdo descrito nas aplicações Web [Berners-Lee et al., 2001][Shadbolt et al., 2006]. Como resultado, a Web semântica provê um ambiente onde agentes de software podem navegar através de documentos Web e executar tarefas sofisticadas. Desta forma, a Web semântica oferece inúmeras melhorias no contexto de sistemas educacionais baseados na Web contribuindo para o aumento da qualidade da aprendizagem, criando assim os ambientes educacionais baseados na Web semântica¹¹ [Devedzic, 2004][Bittencourt et al., 2008b].

Um ambiente educacional semântico equivale a um ambiente que busca representar informações que podem ser utilizadas por máquinas para exibir recursos, bem como automatização, integração e reuso entre aplicações [Boley et al., 2001]. Entretanto, o foco aqui é dado para os ambientes adaptativos, onde a fusão dos ambientes educacionais adaptativos com os ambientes educacionais semânticos cria os chamados ambientes educacionais adaptativos e semânticos.

Com isso, ambientes educacionais adaptativos e semânticos buscam garantir a automatização, integração e reuso entre aplicações para prover mais adaptação (pelo provimento de mais informações/conhecimento) para os usuários e mais inteligência (pela disponibilização/acesso de mais mecanismos inteligentes de outros ambientes) para

¹¹Do Inglês: *Semantic Web-based Educational Systems*.

o sistema.

Além disso, outras características/requisitos que devem ser levados em consideração em ambientes educacionais adaptativos e semânticos estão relacionadas a seguir:

- (i) Disponibilização de diversas ferramentas interativas: a interação entre os diversos atores (estudantes, professores, tutores, autores, outros) envolvidos em um ambiente educacional é guiada através da utilização de ferramentas interativas, sejam elas síncronas ou assíncronas;
- (ii) Disponibilização de recursos inteligentes e de adaptação: a garantia de adaptação nos ambientes educacionais é feita através da utilização de mecanismos de processamento inteligentes. Além disso, existem diversas formas de prover adaptação para tais ambientes[Vasilyeva et al., 2008][Romero et al., 2009][Carmagnola and Cena, 2009][Giannoukos et al., 2008];
- (iii) Especificação e implementação de Agentes, Ontologias e Serviços Web Semânticos: diante dos requisitos citados referentes às tecnologias para provimento de automatização, integração e reúso entre aplicações, há a necessidade de especificação e implementação de tais tecnologias¹²;
- (iv) Necessidades de acesso constante (ambientes 24/7): ambientes baseados na Web, por natureza, possuem a necessidade de disponibilidade constante para acesso dos atores envolvidos nos ambientes educacionais;
- (v) Utilização de milhares ou milhões de usuários nas aplicações: ambientes educacionais de aprendizagem on-line têm sido utilizados com mais frequência nos últimos anos e a demanda tende a aumentar nos próximos anos. Desta forma, há a necessidade de garantir um ambiente com acesso e suporte para milhares ou milhões de usuários.
- (vi) Mudanças de requisitos: a evolução de requisitos nestes ambientes ocorre de forma natural. Além disso, a necessidade é ampliada a partir dos objetivos específicos de adaptação de cada ambiente e da diversidade identificada com relação às necessidades dos usuários de tais ambientes.

¹²Para maiores detalhes: Vide Questão de Pesquisa Q1.

A próxima seção aborda aspectos de pesquisa, envolvendo a construção de ambientes educacionais adaptativos e semânticos.

1.2 Abordagem da Pesquisa

Esta seção descreve a pesquisa da área de ambientes educacionais adaptativos e semânticos, fazendo uma análise dos problemas envolvidos na construção destes ambientes.

As subseções que seguem abordam as questões de pesquisa que são levadas em consideração nesta tese. Logo após, os objetivos de pesquisa são abordados buscando uma solução para cada uma das questões. Em seguida, o escopo e metodologia de pesquisa são abordados, respectivamente.

1.2.1 Questões de Pesquisa

Nesta tese, são focalizadas as quatro questões seguintes identificadas no processo de construção de ambientes educacionais adaptativos e semânticos. São elas:

Q1: Como conceber Ambientes Educacionais Adaptativos e Semânticos?

Ambientes construídos sob a perspectiva da Web semântica fazem uso de tecnologias inerentes a esta área objetivando, de alguma forma, a descrição semântica do domínio, de modo a garantir um processamento automático através de aplicativos. Entretanto, quando se faz uso de tecnologias provenientes da Web semântica, algumas características devem ser levadas em consideração:

- (i) Declaração explícita do conhecimento presente nas aplicações Web [Gómez-Perez and Corcho, 2002]: a Web semântica possui como base a declaração do conhecimento presente nas aplicações de forma explícita com o intuito de garantir um entendimento não só por humanos mas também por máquinas. Além disso, tal declaração deve fazer uso de linguagens que garantam consistência. Desta forma, ontologias têm sido utilizadas para garantir a declaração explícita do conhecimento de aplicações sob a perspectiva da Web semântica;

- (ii) Representar Informações para máquinas entenderem a fim de garantir automação, integração e reuso entre aplicações [Boley et al., 2001]: a representação das informações tem como objetivo garantir que máquinas possam fazer uso das mesmas sem necessariamente precisar de interferência humana. Com isso, agentes de software são entidades que buscam a automatização de atividades, além da comunicação entre diferentes aplicações de tal forma a garantir a integração e o reuso de conhecimento entre aplicações;
- (iii) Infra-estrutura para os agentes executarem, buscando a resolução de problemas complexos [Hendler, 2001]: agentes devem ser capazes de buscar, recuperar e manipular informação disponíveis na Web e que requerem a integração entre agentes para fazer garantir a resolução de problemas complexos através de atividades cooperativas de uma sociedade de agentes;
- (iv) Disponibilização de serviços que máquinas entendam e agentes inteligentes possam descobrir, compor e executar automaticamente [McIlraith et al., 2001]: apesar dos agentes garantirem a automatização de atividades, há a necessidade da disponibilização dos serviços que são utilizados, tanto pelos agentes de software quanto pelos usuários das aplicações. Desta forma, serviços Web semânticos têm sido utilizados através da descrição semântica dos serviços de modo a garantir a descoberta, composição e invocação automática dos mesmos.

Diante dos requisitos supracitados, a abordagem utiliza uma convergência natural entre agentes, ontologias e serviços semânticos. Entretanto, há uma grande complexidade envolvida na concepção e desenvolvimento de ambientes educacionais adaptativos descritos semanticamente, com vistas a prover a automatização de atividades, integração e reuso de conhecimento.

Com isso, levanta-se a questão sobre como conceber ambientes educacionais adaptativos e semânticos.

Q2: Como garantir o reúso de conhecimento?

Da mesma forma que a questão anterior, o ambiente educacional pode fazer uso de informações para melhorar o processo de ensino através de informações já processadas em outros ambientes educacionais. Ou seja, o ambiente deve ser capaz de reusar conhecimento disponibilizado em outros ambientes para auxiliar aos atores de determinado ambiente.

Como consequência, faz-se a questão sobre como garantir o reúso de conhecimento.

Q3: Como construir um modelo baseado no reúso de software?

Ressalta-se ainda outro aspecto, porém não menos importante, que deve ser levado em consideração quando se constroem ambientes educacionais adaptativos e semânticos. Ambientes educacionais adaptativos e semânticos devem disponibilizar diversos recursos computacionais, bem como atender a diversos requisitos tanto funcionais quanto não funcionais citados anteriormente. Dentre os aspectos que devem ser levados em consideração, citam-se a disponibilização de ferramentas interativas, disponibilização de recursos inteligentes e de adaptação, especificação e implementação de agentes, ontologias e serviços Web semânticos, disponibilidade de ambiente 24/7, utilização de milhares ou milhões de usuários nas aplicações e mudanças de requisitos frequentes.

Q4: Como garantir interoperabilidade entre aplicações?

Um dos aspectos importantes em ambientes educacionais que se adaptam às características dos usuários é o provimento de informações personalizadas de acordo com o perfil de determinado usuário. Entretanto, nem sempre o ambiente educacional possui tais informações necessárias disponíveis para garantir esta personalização. Além disso, o ambiente pode ter a necessidade de resolver problemas complexos e que não são previstos no mesmo. Com isso, há um interesse crescente nestes sistemas em prover recursos educacionais, bem como componentes funcionais de modo a serem utilizados em diversas aplicações e com propostas e tarefas educacionais variadas. Tais sistemas necessitam frequentemente

interoperar, colaborar e gerar uma troca por conteúdo ou reusar funcionalidades, como mostrado na Figura 1.2.

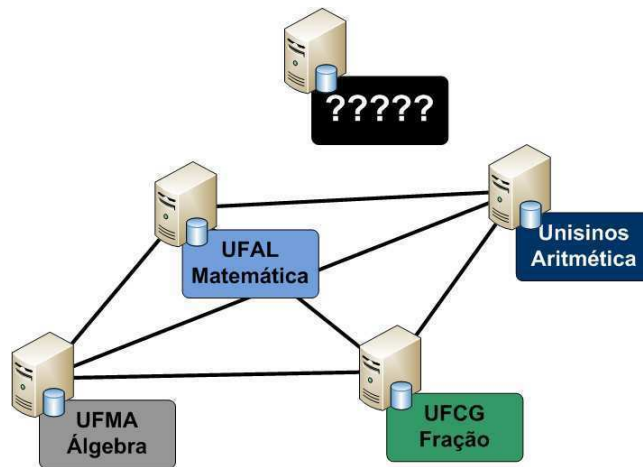


Figura 1.2: Exemplo de Motivação para Interoperabilidade.

Desta forma, a partir do momento que se constrói ambientes educacionais adaptativos sob a perspectiva da Web semântica, há a necessidade de garantir interoperabilidade entre as aplicações educacionais [Aroyo and Dicheva, 2004]. Entretanto, garantir interoperabilidade entre ambientes educacionais de forma automatizada equivale a descobrir ambientes capazes de interagir em determinado contexto, bem como identificar os protocolos de comunicação necessários para permitir a interação entre as aplicações [Dicheva and Dimitrova, 2008].

Levanta-se então a questão sobre como garantir interoperabilidade entre ambientes educacionais considerando que a aplicação é distribuída.

Diante disto, pode-se perceber que diversos requisitos precisam ser implementados, além da utilização de diversas tecnologias. Com isso, tanto a implementação dos requisitos quanto a mudança de requisitos nos ambientes demandam bastante tempo para serem providas.

Aborda-se então a necessidade de um modelo baseado no reúso e extensível para os ambientes educacionais adaptativos e semânticos.

Recentemente, poucas abordagens têm sido propostas para prover os requisitos descritos como necessários para conceber ambientes educacionais adaptativos e

semânticos, entretanto estes ambientes não possuem uma solução integradora que alcance as potencialidades providas pelas tecnologias da Web semântica.

1.2.2 Objetivos da Pesquisa

Após as questões de pesquisa abordadas na subseção anterior, apresentam-se aqui os objetivos de pesquisa da tese em questão.

O objetivo geral desta tese consiste na proposição de uma solução viável e efetiva para a construção de ambientes educacionais adaptativos e semânticos, através da integração de ontologias, agentes e serviços Web semânticos. Este objetivo pode ser remetido aos três objetivos específicos seguintes, focalizados em responder às questões anteriormente descritas.

O1. Propor um modelo conceitual para construção de tais ambientes: o modelo conceitual busca identificar outras entidades presentes em ambientes educacionais adaptativos e semânticos e com isso, relacionar tais entidades. Como consequência desta definição, busca-se ter controle sobre os aspectos relacionados ao ambiente educacional adaptativo e semântico. Desta forma, os objetivos específicos são:

O1.1. Estudar modelos conceituais existentes;

O1.2. Identificar as entidades e relações presentes nos ambientes educacionais adaptativos e semânticos;

O1.3. Avaliar o modelo conceitual proposto.

O2. Propor e implementar um modelo computacional baseado em ontologias, agentes e serviços semânticos: este objetivo está relacionado com duas questões Q2. e Q3., abordadas na subseção anterior. As questões abordam os problemas do reúso de software e de conhecimento entre aplicações educacionais. Logo, pretende-se propor e implementar um modelo computacional baseado em ontologias, agentes e serviços semânticos com vistas a garantir ambos os aspectos. Com isso, os objetivos específicos são:

O2.1 Estudar propostas de modelos baseados no reúso para ambientes educacionais adaptativos e semânticos;

- O2.2** Estudar propostas existentes em educação que implementam a integração de ontologias, agentes e serviços semânticos;
 - O2.3** Definir abordagens utilizadas para garantir o reuso de software e de conhecimento;
 - O2.4** Definir um modelo computacional que assegure o reuso de software e de conhecimento;
 - O2.5** Implementar e testar o modelo;
 - O2.6** Criar cenários para avaliar o modelo.
- O3.** Propor e implementar um modelo de infra-estrutura para interoperabilidade entre aplicações educacionais: este objetivo está relacionado com a necessidade de interoperar aplicações educacionais, como descrita na Questão 4, da subseção anterior. Com isso, os objetivos específicos identificados foram:
- O3.1** Estudar propostas de modelos que abordam a interoperabilidade entre aplicações educacionais;
 - O3.2** Definir um modelo de interoperabilidade que assegure interoperabilidade entre aplicações educacionais de forma automatizada;
 - O3.3** Implementar e testar modelo;
 - O3.4** Criar cenários para avaliar o modelo.

1.2.3 Aspectos Metodológicos

Esta tese propõe a concepção e o desenvolvimento de ambientes educacionais adaptativos semânticos através da proposição de modelos e ferramentas.

O ponto de partida para esta pesquisa foi uma reflexão sobre como resultados na área de Inteligência Artificial em Educação, por meio de uma abordagem centrada em agentes inteligentes, poderiam influenciar no avanço dos ambientes virtuais de aprendizagem em educação a distância. Nesse sentido, tomou-se por base o modelo de ambiente Mathema [Costa, 1997] como uma proposta adequada para construção de ambientes interativos de aprendizagem baseados em uma abordagem de sistemas multi-agentes. Além disso,

relacionado ao Mathema, teve-se em consideração o arcabouço *ForBILE* [Bittencourt, 2006][Bittencourt et al., 2007a], oferecendo uma plataforma para a construção de ambientes interativos de aprendizagem segundo o modelo Mathema revisitado. Posteriormente, analisou-se a literatura especializada em ambientes EaD [Brooks et al., 2006][Hauger and Köck, 2007],[del Puerto Paule Ruiz et al., 2008] baseados na Web, observando-se o que se pretende para uma nova geração de tais ambientes, extraíndo-se daí um novo conjunto de aspectos, entre os quais podem-se destacar alto custo de desenvolvimento, integração entre técnicas de inteligência artificial, escalabilidade, dificuldade de compartilhamento de materiais, extensibilidade, interoperabilidade entre aplicações e alto custo de manutenção [Brooks et al., 2006]. Desse modo, verificou-se a adequação de se propor uma abordagem que combina recursos de ontologias, de agentes inteligentes e de serviços Web semânticos, visando proporcionar ambientes que garantam a construção efetiva de ambientes educacionais adaptativos e semânticos, levando-se em consideração o reúso de conhecimento e software além de proporcionar interoperabilidade entre ambientes educacionais.

A abordagem combina os recursos ontológicos, de agentes de software e serviços Web semânticos para proporcionar ambientes que garantam automatização de processos e interoperabilidade entre aplicações educacionais. A idéia central da proposta é permitir que ambientes educacionais sejam concebidos, especificados e implementados através de um ferramental tecnológico que garanta a cobertura necessária para o cumprimento de cada requisito.

O problema de como conceber ambientes educacionais adaptativos e semânticos foi abordado através de uma análise do domínio, levando em consideração uma abordagem tanto *top-down* como *bottom-up*. Com esta análise, foi proposto um modelo conceitual que garantisse a identificação dos recursos presentes nestes tipos de ambientes educacionais.

Os problemas de reúso de conhecimento e de software foram abordados através da proposição de um modelo computacional onde as ontologias proporcionam o reúso de conhecimento e os serviços Web semânticos garantem o reúso de software. Além disso, os agentes são utilizados como o modelo de execução do ambiente. A avaliação do modelo computacional foi feita através da implementação de três estudos de caso, sendo

dois no domínio de matemática (Fração e MMC) e outro no domínio de Medicina.

O problema de garantir a interoperabilidade entre aplicações foi abordado através da proposição de um modelo de infra-estrutura para interoperabilidade de modo a proporcionar um mecanismo de comunicação entre os ambientes de forma automatizada. Para isto, foram feitos estudos na área de Web semântica e educação, onde uma arquitetura conceitual foi estendida, adaptada e implementada para garantir que este problema fosse resolvido. A avaliação do modelo foi feita através da garantia de comunicação entre os dois ambientes do domínio de matemática.

O trabalho possui um respaldo da comunidade científica através da publicação em periódicos e conferências internacionais bem como na cooperação com pesquisadores de renome da área [Bittencourt et al., 2008d][Isotani et al., 2009] [Calado et al., 2009][Bittencourt et al., 2009a][Bittencourt et al., 2008a][Abel et al., 2008] [Bittencourt et al., 2009b][Oliveira et al., 2009][Abel et al., 2010][Bittencourt et al., 2008b][Sibaldo et al., 2008][Isotani et al., 2008].

1.3 Sumário das Contribuições

As questões de pesquisa descritas foram abordadas, resultando na obtenção de um modelo conceitual, um modelo computacional e um modelo de infra-estrutura para interoperabilidade para a construção de ambientes educacionais adaptativos e semânticos. As contribuições originais associadas a esta tese consistem em: i) a proposição dos modelos para ambientes educacionais adaptativos e semânticos, proporcionando a especificação e modelagem destes ambientes nos moldes desta tese; ii) a garantia do desenvolvimento e manutenção de ambientes educacionais adaptativos e semânticos de forma mais fácil; iii) na possibilidade de produção em larga escala de tais ambientes e para as diversas áreas do conhecimento, permitindo o acesso do cidadão ao conhecimento e iv) na possibilidade de permitir que instituições de ensino possam cooperar na produção de materiais educacionais e no acesso aos estudantes que necessitem deste tipo de tecnologia tanto no Brasil quanto no exterior, possibilitando a participação e o acesso universal ao conhecimento.

1.4 Organização do Documento

O conteúdo seguinte deste documento consta de quatro capítulos (onde esta introdução equivale ao primeiro capítulo), sendo eles:

Capítulo 2 - Web Semântica em Ambientes Educacionais: neste capítulo são descritas características inerentes à Web semântica, bem como as potencialidades, vantagens e desafios desta área nos ambientes educacionais mediados por computador;

Capítulo 3 - Trabalhos Relacionados: este capítulo apresenta os trabalhos relacionados à tese em questão (descrevendo suas características, arquitetura, vantagens e desvantagens). Além disso, é feita uma comparação entre os trabalhos relacionados dando destaque aos requisitos atendidos de cada ambiente;

Capítulo 4 - Modelos Propostos: a modelagem da solução é apresentada, dando destaque a concepção e desenvolvimento dos modelos conceitual, computacional e de interoperabilidade, que foram nomeados como *MASSAYO*. Por fim, é apresentado um guia para construir aplicações baseadas em tais modelos;

Capítulo 5 - Avaliação dos Modelos Propostos: são apresentados três estudos de casos, descrevendo aspectos inerentes à utilização dos modelos apresentados na modelagem da solução, sendo eles: 1) Sistema Colaborativo em anatomia óssea do crânio; 2) Sistemas Tutor Inteligente em Fração e 3) Ambiente de Educação a Distância Adaptativo em MMC. Após a apresentação dos estudos de caso, uma avaliação é feita sobre os estudos de caso, descrevendo-se aspectos de usabilidade, reuso, desempenho, disponibilidade, interoperabilidade, descoberta de recursos, alinhamento de ontologias, entre outros.

Capítulo 6 - Conclusões e Trabalhos Futuros: a conclusão é apresentada, descrevendo como os objetivos foram alcançados, além dos resultados obtidos através de publicações e outros indicadores. Por fim, os Trabalhos futuros são apresentados.

Além dos capítulos supracitados, esta tese apresenta as referências utilizadas no trabalho e que nortearam a pesquisa. Por fim, é apresentado um apêndice que descreve as tecnologias utilizadas.

Capítulo 2

Web Semântica em Ambientes Educativos

A prática deve alicerçar-se sobre uma boa teoria.

Leonardo da Vinci

Este capítulo fornece uma visão geral dos conceitos que estão sendo utilizados nesta tese. Com isso, as tecnologias provenientes da Web Semântica, destacando o papel das ontologias, dos serviços semânticos e dos agentes de software são descritas. Em seguida, os ambientes educacionais são discutidos, dando destaque aos ambientes educacionais adaptativos e o papel da Web Semântica nestes ambientes.

2.1 Web Semântica

Tim Berners-Lee, na primeira conferência internacional sobre a WWW, em Genebra¹ em 1994 proferiu uma palestra sobre a necessidade de semântica na Web e segundo ele, a forma que os documentos Web estavam estruturados (através de nós e links) fazia com que apenas usuários humanos pudessem entender o significado contido neles [Berners-Lee, 1999]. Tal impossibilidade fazia com que máquinas não pudessem acessar

¹Conferência: *First International Conference on World Wide Web*

e obter significado dos documentos. A Figura 2.1 ilustra o relacionamento entre páginas sem semântica explícita.

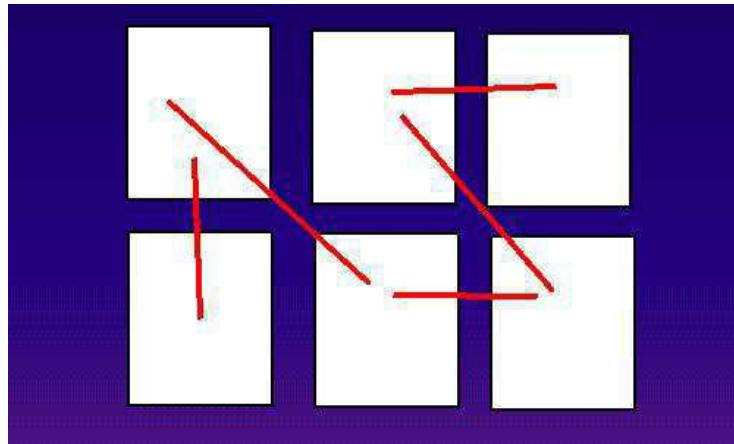


Figura 2.1: Páginas Web clássica e seus relacionamentos.

Fonte: <http://www.w3.org/Talks/WWW94Tim>

De forma mais detalhada, os documentos utilizam a linguagem HTML, que é uma linguagem que faz apresentação de hipertextos. Os hipertextos são documentos que possuem nós e links com outros documentos, permitindo assim a navegação entre os documentos. Entretanto, os links que existem para relacionar os documentos não tinham nenhum tipo de característica que diferenciava um do outro, de tal forma que não simplificava o processo de identificação de significado em um nó.

É importante frisar que esses problemas ficam mais evidentes nos dias de hoje, pois se calcula aproximadamente 28 bilhões de páginas Web indexadas². Esta quantidade de documentos torna praticamente impossível o acesso e a busca por informação de forma eficiente e consistente para os usuários, fazendo com que haja a necessidade de entidades de software buscarem informações e processar atividades para o homem. Para tanto, faz-se necessário introduzir a semântica na especificação de documentos Web em uma linguagem que possa ser processada por máquina e que permita que os links sejam criados com valor em seu relacionamento. Isto faz com que os documentos possuam seus recursos especificados em uma linguagem que possa ser acessada por máquina, permitindo a execução automática de atividades como compra de produtos personalizados, negociação por pacotes turísticos, agendamento de consultas, entre

²<http://www.worldwideWebsize.com/>

outras. A Figura 2.2 descreve os documentos e os relacionamentos com os recursos.

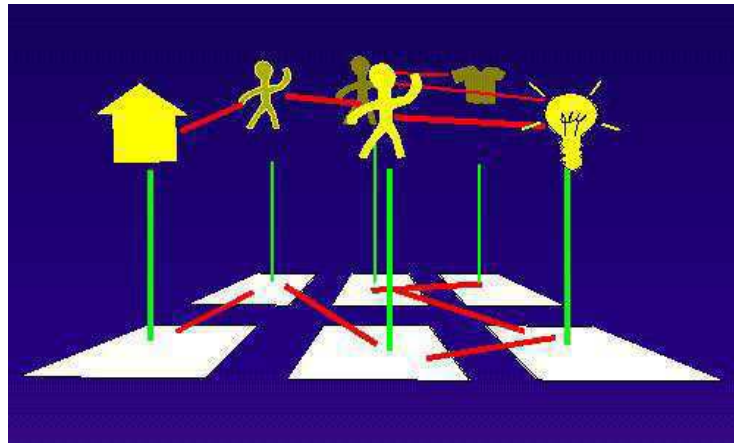


Figura 2.2: Páginas descrevendo a primeira idéia da Web Semântica.

Fonte: <http://www.w3.org/Talks/WWW94Tim>

Na perspectiva de solucionar o problema supracitado, foi proposta uma estrutura que descrevesse os recursos disponibilizados na Web de tal forma que pudessem ser identificados e diferenciados. Isso foi proposto por Tim Berners-Lee em seu livro intitulado: “*Weaving the Web*”, onde o termo Web Semântica apareceu pela primeira vez, em 1999. Já em 2001, o artigo publicado na revista *Scientific American*, marcou o início da pesquisa relacionada à Web Semântica. Este artigo, intitulado (e traduzido para o Português) “Web Semântica: Um novo formato de conteúdo para a Web que tem significado para computadores vai iniciar uma revolução de novas oportunidades”. Neste artigo, o autor aborda características da Web Semântica, propondo as camadas da Web Semântica e descrevendo como ela deve funcionar.

De forma mais detalhada, a Web Semântica busca utilizar recursos provenientes de áreas diferentes, tais como: Inteligência Artificial (agentes inteligentes e representação de conhecimento), Engenharia de Software (frameworks), Computação Distribuída (serviços Web), entre outras, para executar atividades na Web que antes só eram possíveis por agentes humanos. Objetivando ilustrar funcionalmente a Web semântica, seguem dois cenários. São eles:

Cenário 1: um cenário clássico, apresentado por Tim Berners-Lee em 2001, sobre a marcação de uma consulta médica. Nele, basicamente, Lucy precisava marcar uma consulta para a sua mãe e levá-la ao médico. Houve a necessidade de localizar

médicos qualificados e com o mesmo plano de saúde que a mãe de Lucy. Ela precisava estar disponível no horário da consulta e não poderia perder muito tempo com o deslocamento, ou seja, a consulta deveria ser em um local próximo à casa da sua mãe. Logo, Lucy precisaria configurar seu agente para que ele pudesse recuperar o tratamento médico, serviços disponíveis e consultórios próximos à casa de sua mãe e finalmente, marcar a consulta que melhor se encaixasse com as exigências de Lucy. A Figura 2.3 ilustra alguns dos recursos que podem ser utilizados em um cenário como este. Nela aborda-se a integração de ontologias, agentes inteligentes e serviços semânticos para alcançar o objetivo apresentado neste cenário.

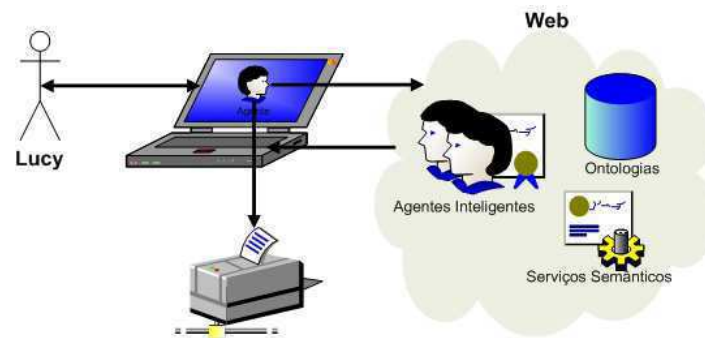


Figura 2.3: Tecnologias da Web Semântica para processar uma consulta médica.

Cenário 2: este cenário é sobre a construção e avaliação de projetos de pesquisa por instituições de fomento a pesquisa. O cenário é dividido em dois momentos:

- Criação do projeto de pesquisa: neste momento, o pesquisador pode criar o projeto de pesquisa e receber recomendações de pesquisadores que poderiam ser parceiros;
- Avaliação de projetos de pesquisa: as instituições de fomento poderiam interagir com outras instituições para avaliar tanto a confiabilidade de determinado pesquisador quanto a relevância e inovação de determinado projeto.

A Figura 2.4 apresenta um esquema ilustrando os recursos presentes no cenário 2, onde se observam diferentes instituições compartilhando ontologias e acessando serviços semânticos através de agentes inteligentes.

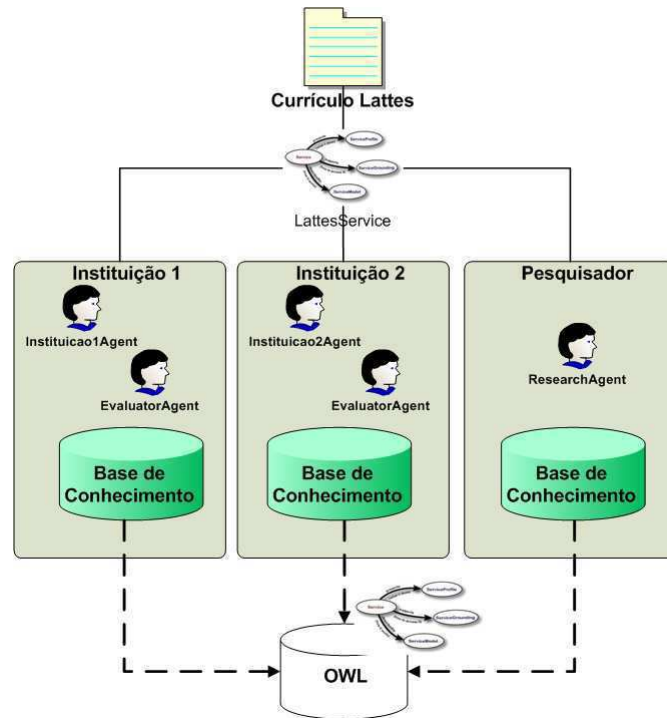


Figura 2.4: Tecnologias da Web Semântica para a criação e avaliação de projetos de pesquisa.

Fonte: [de Barros Costa et al., 2007]

A Web Semântica estende a Web clássica, provendo uma estrutura semântica para páginas Web, a qual permite que tanto agentes humanos quanto agentes de software possam entender o conteúdo presente em páginas Web. Dessa forma, a Web Semântica provê um ambiente onde agentes de software podem navegar através de páginas Web e executar tarefas sofisticadas. Em outras palavras, a Web Semântica é necessária para expressar informações de forma precisa, podendo tais informações serem interpretadas por máquinas e dessa forma permitirem que agentes de software possam processar, compartilhar, reusar, além de poder entender os termos que estão sendo descritos pelos dados [Devedzic, 2006].

Objetivando tornar a visão da Web Semântica mais factível em termos de implementação, Tim Berners-Lee propôs um modelo em camadas, conhecido como "bolo de noiva", descrevendo os recursos e linguagens para a Web Semântica, tal como mostrado na Figura 2.5

Após a publicação do artigo sobre Web Semântica na *Scientific American*, em 2001, a área tem se desenvolvido bastante, onde o tema tem evoluído rapidamente, tendo seus

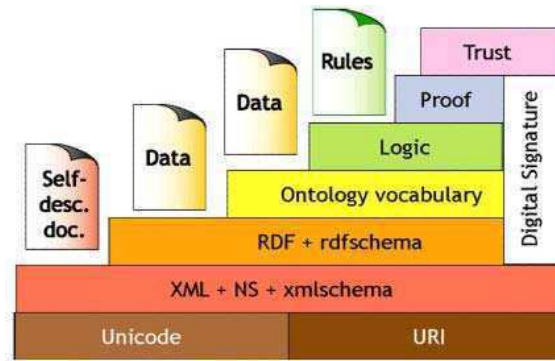


Figura 2.5: Camadas da Web Semântica inicialmente proposta.

Fonte: [Berners-Lee et al., 2001]

resultados divulgados em diversas conferências, periódicos, livros, grupos de pesquisa, entre outros tanto no Brasil quanto em outros países. Já houveram vários avanços desde então em busca do desenvolvimento das camadas propostas por Tim Berners-Lee [Dubost and Herman, 2008]. Entretanto, apesar de diversos trabalhos atuais ainda utilizarem esta perspectiva das camadas da Web Semântica, no ano de 2007 foi feita uma reavaliação das camadas propostas, descrevendo as tecnologias que foram recomendadas pelo W3C. A Figura 2.6 apresenta a nova perspectiva das camadas da Web Semântica.

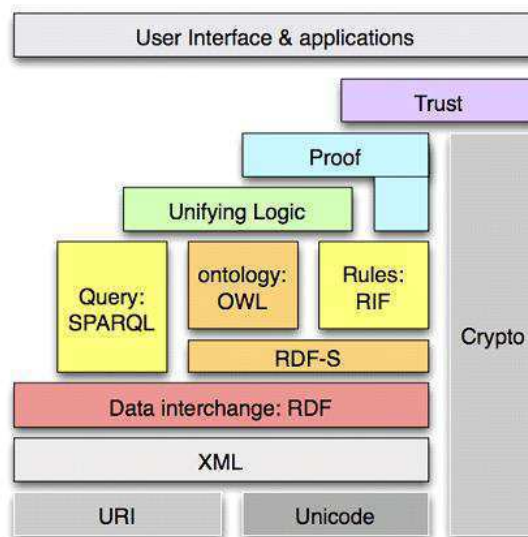


Figura 2.6: Camadas da Web Semântica Revisitada.

Fonte: <http://www.w3.org/DesignIssues/diagrams/sweb-stack/2006a.png>

As camadas e tecnologias exibidas na figura acima são descritas a seguir.

- *URI*: a proposição da Web foi da construção de uma comunidade global onde

qualquer parte poderia compartilhar informação com outras partes, isto é, a Web possui flexibilidade e capacidade para atribuir recursos a um determinado item de interesse. Essa capacidade é provida por um identificador de recurso uniforme, conhecido por *Uniform Resource Identifiers* (URI), provendo uma identificação comum através da Web. De um modo geral, a WWW é um espaço de informações onde os itens de interesse, referenciados como recursos, são identificados por um URI.

É importante ressaltar que o conceito de recurso não possui um escopo limitado, podendo equivaler a qualquer coisa que possa ser identificada por um URI, como uma imagem, vídeo, serviço, entre outros. Como exemplo, a Figura 2.7 apresenta um determinado recurso disponibilizado na Web, sendo representada por uma linguagem e identificada por um URI.

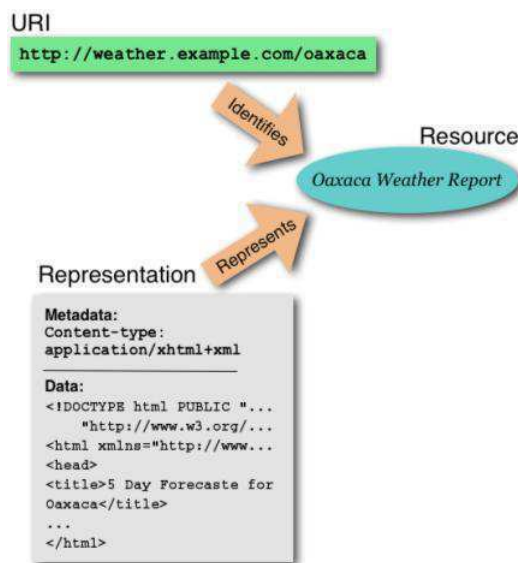


Figura 2.7: Ilustração de um Recurso um WWW.

Fonte: <http://www.w3.org/TR/Webarch/uri-res-rep.png>

Observa-se que a URI provê uma maneira simples e única de identificar recursos na Web, sendo caracterizado por:

- Uniformidade: utilização de recursos tanto no mesmo contexto quanto em contextos diferenciados;
- Recurso: qualquer coisa que pode ser identificada por um URI, como um

vídeo, imagem, serviço, documento, entre outros;

- Identificador: informação requerida para identificar e diferenciar um determinado recurso de qualquer outro.
- *Unicode*: o Unicode representa um padrão de codificação de caracteres, fornecendo um número único (pois computadores lidam com números) para cada caractere. Assim sendo, há a independência de plataforma (e.g. *Internet, Windows, Symbian*), de programa (e.g. *SQL Server, MSN*, programas *W3C*) e idioma (e.g. Português, Japonês, Inglês, Árabe);
- *XML*: XML é a segunda camada da Web Semântica. Apesar da Figura 2.6 não descrever o *Namespace* e o XML Esquema, os mesmos estão presentes nesta camada. É fundamental o entendimento tanto de espaços de nomes como de XML esquema para um melhor entendimento desta camada e das camadas superiores. XML³ é uma linguagem de marcação extensível que foi proposta com o objetivo de transportar e armazenar dados e de prover uma estrutura mais poderosa que o HTML. XML tem sido explorada pela comunidade desde o final da década de 80 do século passado e tendo sido recomendada pelo W3C em 1998. A partir de então, tanto a indústria quanto a academia têm utilizado esta linguagem no desenvolvimento de aplicações e de padrões para transporte e armazenamento de dados. Algumas características presentes no XML são:
 - XML é independente de plataforma: portar documentos XML é independente tanto do software que está sendo utilizado quanto do hardware. Isto é em função do XML ser armazenado no formato *plain text*;
 - XML simplifica o compartilhamento dos dados: as *tags* definidas pelos usuários fazem com que se defina uma estrutura hierárquica de um determinado domínio que possa ser compartilhada e reusada por outras aplicações;
 - XML é extensível: XML permite tanto a adição de novos elementos à estrutura de um documento quanto a criação de uma nova linguagem baseada

³Do inglês: *eXtensible Markup Language*

na estrutura XML (WSDL, RSS, RDF, entre outras);

- XML pode ser utilizada por diversos protocolos (como HTTP) e mecanismos (como URL) Web sem a necessidade de adição de qualquer requisito. De um modo geral, a linguagem XML foi desenvolvida já com a idéia da Web, fazendo que seja utilizada na Web naturalmente.

O XML oferece aos usuários uma descrição de dados semi-estruturados, de tal forma que ainda pode ser processado por aplicações computacionais. Para que tais aplicações possam processar documentos XML, é fundamental que haja uma estrutura que descreva os documentos XML e permita que tais documentos possam ser interoperados. Esta estrutura consiste de um esquema XML e de um espaço de nome, os quais são descritos como segue.

XML Esquema: O objetivo do XML esquema é descrever a estrutura de um documento XML, tendo sido criado com o objetivo de substituir o *Document Type Definition* (DTD). As razões pelas quais o XML esquema possui uma estrutura que torna o processo de utilização melhor que o DTD são:

- Linguagem base: esquema XML é uma estrutura baseada na própria estrutura de um documento XML, fazendo com que todos os benefícios providos por um documento XML sejam herdados pelo esquema;
 - Recursos disponíveis: tanto a linguagem como o suporte de processamento por máquina fazem com que esquemas XML possuam mais utilidades que os DTDs, além de possuírem mais recursos disponíveis;
 - Suportes: XML esquema oferece suporte tanto a tipos de dados quanto a *namespaces*;
 - Recomendação do W3C: proposto pela Microsoft, no ano de 2001, o XML esquema tornou-se recomendação do W3C .
- *RDF*: O *Resource Description Framework* (RDF) consiste de uma linguagem de representação de informação na Web, permitindo com que recursos possam ser acessados por máquinas. Em outras palavras, o objetivo do RDF é prover uma

representação minimalista do conhecimento da Web [Shadbolt et al., 2006]. Além disso, RDF possui uma estrutura que foi projetada para representação de informação na Web e integração de dados. Além disso, um documento XML é bastante sensível ao seu esquema, ou seja, quando o esquema muda, o documento XML torna-se inválido. Já em um Documento RDF, quando o esquema muda, os dados não são automaticamente rejeitados pelo *parser*. A seguir estão algumas características sobre a utilização de documento RDF:

- Escrito em XML: RDF é escrito em XML, fazendo com que as características/vantagens presentes em documentos XML sejam herdadas para documento RDF;
- Recursos: documentos RDF possuem o objetivo de descrever recursos na Web através da utilização de URI, então o RDF oferece suporte a tudo que pode ser especificado como um recurso. Ou seja, RDF descreve qualquer recurso presente na Web, como imagens, páginas Web, vídeos, pessoas e assim por diante;
- Significado: a modelagem feita para o RDF foi feita para que computadores entendessem e conseguissem extrair significado dos documentos, isto é, apesar do RDF ser baseado em XML e herdar características como facilidade de entendimento para usuários, ele não foi projetado para o entendimento humano;
- Processamento global: a forma que o RDF foi projetado faz com que os dados sejam processados fora do escopo de desenvolvimento de um determinado documento, fazendo com que o processamento por máquina seja para utilização global;
- Tripla: uma das principais características do RDF consiste na estrutura de um elemento base, ou seja, a estrutura de qualquer elemento em RDF é vista como uma tripla da forma <sujeito, predicado, objeto>;
- Recomendação W3C: o RDF é uma recomendação do W3C desde 1999.

O RDF é considerado um framework, sendo descrito como mostrado na Figura 2.8.

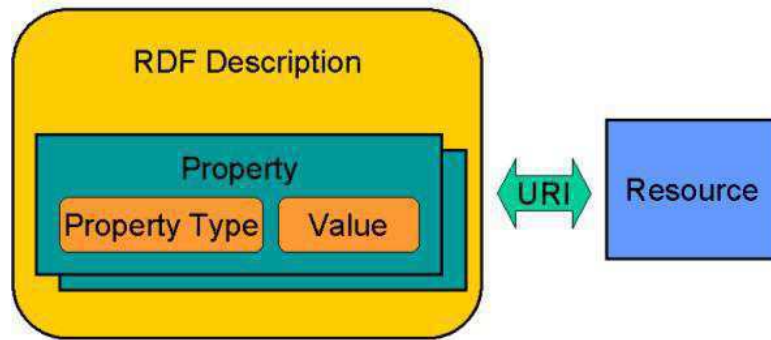


Figura 2.8: Arquitetura do RDF.

Fonte: <http://www.iaa.upf.es/~jblat/material/doctorat/rdf-model.gif>

- *RDF-S*⁴: o RDF Esquema é uma extensão semântica ao RDF e é responsável por prover um conjunto de recursos inter-relacionados. O objetivo do RDF-S é de proporcionar um conjunto básico de recursos que permita que documentos RDF sejam estruturados e validados, ou seja, como documentos RDF não possuem algum mecanismo para descrever suas propriedades bem como os relacionamentos presentes nos mesmos, o RDF-S tem o papel de disponibilizar um vocabulário capaz de descrever classes, propriedades e relacionamentos entre os recursos presentes em documentos RDF. Ressalta-se ainda que o RDF-S consiste em um formato RDF, fazendo com que herde todas as suas características (como o XML Esquema). É importante notar que qualquer documento RDF possui uma estrutura representada através de uma tripla. A vantagem desta tripla está em sua proposta geral e na capacidade de agentes de software poderem fazer transações sem conhecimento prévio de sua estrutura semântica. Desta forma, a própria semântica pode ser descoberta em tempo de execução. Esta característica permite que alterações, remoções e adições de documentos RDF possam ocorrer a todo momento e isto pode ser percebido em tempo real por agentes de software. A Figura 2.9 mostra um documento RDF-S e algumas de suas classes, propriedades e relacionamentos.

⁴Do Inglês *RDF Schema*.

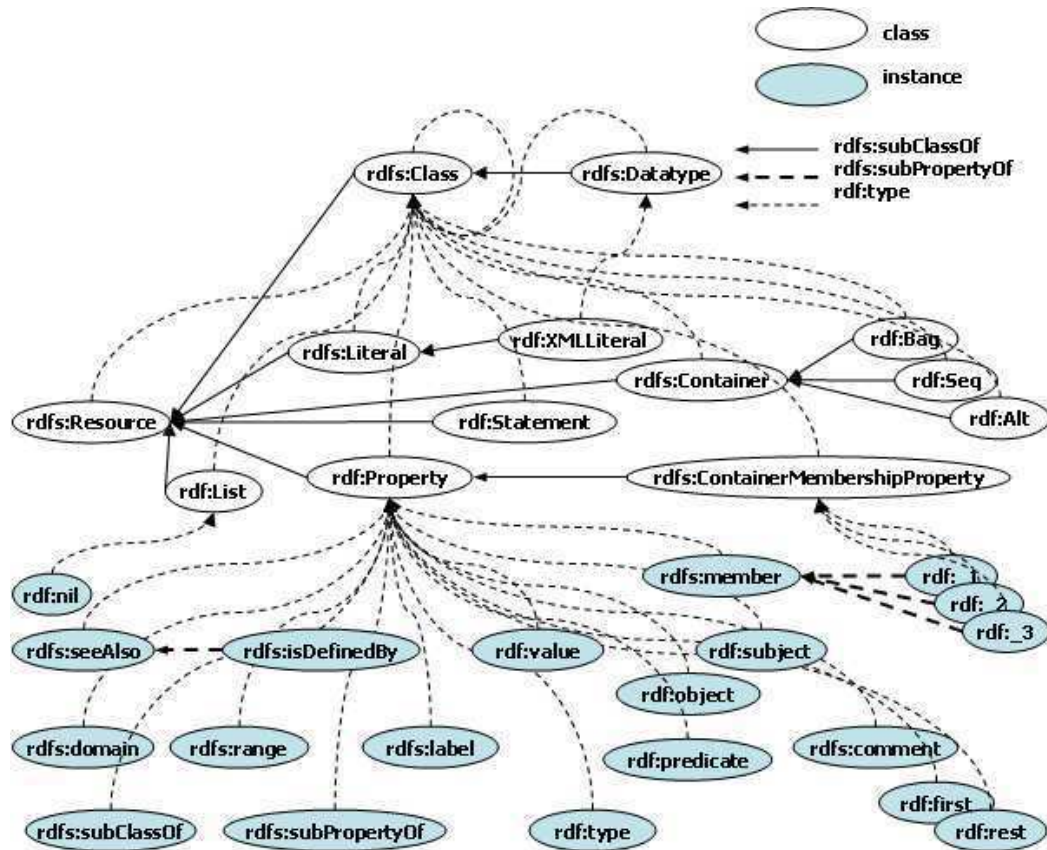


Figura 2.9: Classes, propriedades e relacionamentos em RDF-S.

Fonte: <http://www-kasm.nii.ac.jp/~koide/SWCLOS/Manual/FIGURES/RDFSHierarchy.png>

O RDF Esquema é útil para descrever a estrutura de um documento RDF. A limitação do RDF-S está na falta de suporte à inferência, além de possuir expressividade limitada, fazendo com que haja a necessidade de uma camada que permita inferência, bem como maior expressividade.

- *SPARQL*: é uma linguagem de consulta para repositório RDF. Através da linguagem de consulta SPARQL é possível extrair informações sobre valores de atributos, extrair subgrafos RDF e construir novos grafos RDF baseados nos resultados de consultas. Um grafo RDF é um conjunto de triplas. Cada tripla consiste de sujeito, predicado e objeto. Sujeitos são recursos ou “coisas” das quais se deseja tratar. Para exemplificar estas definições, considere a seguinte sentença: Ig Ibert Bittencourt participa do Grupo de Pesquisa GrOW. Esta sentença possui as seguintes partes:

- **Sujeito:** Grupo de Pesquisa GrOW;
- **Predicado:** participa;
- **Objeto:** Ig Ibert Bittencourt.

Em RDF, tal sentença seria descrita da seguinte forma, seguindo-se do grafo correspondente descrito na Figura 2.10:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:pp="http://www.grow.ic.ufal.br/ProjetoPesquisa#">
  <rdf:Description rdf:about="Grupo de Pesquisa GRoW">
<pp:participa>
  Ig Ibert Bittencourt
</pp:participa>
  </rdf:Description>
</rdf:RDF>
```



Figura 2.10: Exemplo de Grafo RDF.

Os resultados das consultas podem ser ordenados e restringidos de diferentes formas. Com isso, para responder a pergunta: *Qual(is) projeto(s) Ig Bittencourt participa?*, a consulta em SPARQL deveria ser construída como descrita abaixo:

```
SELECT ?ProjetoPesquisa WHERE
  {?ProjetoPesquisa :participa :Ig Ibert Bittencourt}
```

É importante ressaltar que em 2009 foi submetido para o W3C o *SPARQL Update*, que é uma versão estendida da linguagem SPARQL de modo a prover atualizações.

- *OWL*: *OWL* (Ontology Web Language) é a linguagem ontológica da Web. Ela permite expressividade de alto nível e inferência implícita. Além disso, há três dialetos diferentes de *OWL*:
 1. *OWL Lite*: representa uma linguagem muito simples. Portanto, *OWL Lite* é recomendada para a definição de hierarquias simples e/ou restrições;
 2. *OWL-DL*⁵: é mais expressiva que a *OWL Lite* e permite a verificação de satisfatibilidade de conceitos, classificação de hierarquias e inferência de tipos, pois a mesma corresponde a uma implementação da Lógica de Descrição [Baader et al., 2003];
 3. *OWL FULL*: representa a linguagem *OWL* mais expressiva. Esta linguagem deve ser utilizada quando a expressividade do conhecimento for mais importante que garantia de computabilidade.
- *RIF*⁶: o objetivo desta camada é prover um conjunto de regras que possa tanto aumentar a expressividade das ontologias e serem intercambiáveis na Web Semântica, ou seja, com *RIF* busca-se a interoperabilidade entre diversos formatos de regras. A linguagem de regras que é utilizada na Web Semântica é a *Semantic Web Rule Language* (*SWRL*).

SWRL é uma extensão dos axiomas *OWL* para incluir regras do tipo *Horn rules* [Ilex Sakharov, 2003], submetida pelos membros da W3C em 2004. *SWRL* combina sublinguagens de *OWL-DL* e *Lite* com algumas sublinguagens de *RWL*, adicionando uma forma simples de construir e manipular com regras do tipo *Horn* de uma maneira sintática e semanticamente coerente, tendo a finalidade de realçar a expressividade.

O principal axioma adicionado à *OWL-DL* são as cláusulas *Horn rules*, que são uma forma de uma implicação entre um antecedente (corpo) e o conseqüente (cabeça). O significado informal de uma regra pode ser lido como: sempre que as circunstâncias especificadas na expressão antecedente forem verdadeiras, então as circunstâncias especificadas no conseqüente devem também ser.

⁵DL significa *Description Logic* ou Lógica de Descrição

⁶Do Inglês *Rule Interchange Format*

Uma regra pode ser representada da forma abaixo, escrita em uma sintaxe informal compreensível pelo ser humano:

$$\textit{antecedente} \rightarrow \textit{consequente}.$$

O antecedente e o consequente de uma regra consistem sempre de zero ou mais cláusulas atômicas. As cláusulas atômicas podem ser da forma $C(x)$, $P(x, y)$, $Q(x, z)$, $\textit{sameAs}(x, y)$ ou $\textit{differentFrom}(x, y)$. Onde C é uma descrição no OWL-DL, podendo ser um nome de uma classe ou então uma combinação complexa de cláusulas booleanas, etc. P pode ser uma propriedade individual no OWL-DL, e Q pode ser uma propriedade OWL-DL avaliada sobre x , e z sendo qualquer uma a variável ou um valor dos dados de OWL. Ou seja, uma cláusula atômica $C(x)$ será verdadeira se x for uma instância da uma classe descrita por C , e uma cláusula atômica $P(x, y)$ (ou $Q(x, z)$) será verdadeira se x estiver relacionado a y (ou z) pela propriedade P (ou Q). A cláusula atômica $\textit{sameAs}(x, y)$ será verdadeira se x for considerado o mesmo objeto que y , e a cláusula atômica $\textit{differentFrom}(x, y)$ será verdadeira se x e y forem considerados instâncias diferentes [Li et al., 2005].

Considerando as regras citadas acima, torna-se fácil criar relacionamentos entre as propriedades, conforme exemplo abaixo:

$$\textit{mae}(?a, ?b) \wedge \textit{pai}(?b, ?c) \rightarrow \textit{avo}(?a, ?c).$$

Para representar variáveis em SWRL é usado o ponto de interrogação antes do identificador da variável, ainda, para a expressão acima, considere que \textit{pai} e $\textit{mãe}$ são propriedades no OWL-DL. Sendo assim, a regra acima afirma que a composição das propriedades \textit{pai} e $\textit{mãe}$ implicam na propriedade $\textit{avó}$. Em outras palavras, se Maria é mãe de João e João é pai de José, então isso implica em dizer que Maria é avó de José.

Usando SWRL, podem ser definidas algumas regras similares com a apresentada anteriormente para expressar relacionamentos entre classes e propriedades em alguma ontologia OWL, podendo deixar assim o significado de uma propriedade

mais claro. SWRL permite ainda que se possa inserir políticas de controle de acesso, na forma de regra, a partir de entidades definidas na própria ontologia.

Uma definição importante proposta pela W3C é a *XML Concrete Syntax*, ou *swrlx*. Essa proposta é uma combinação da *OWL XML Presentation Syntax* com a *RuleML XML Syntax* para expressar como as regras SWRL devem ser representadas em arquivos XML. SWRLX fornece uma série de vantagens: i) qualquer classe OWL pode ser usada como predicado em regras; ii) axiomas de regras e ontologias podem ser livremente misturados; iii) a interoperabilidade entre OWL e RuleML é simplificada [Horrocks et al., 2004].

A W3C também propôs um conjunto *Build-Ins* (conhecido por *swrlb*) para aumentar a expressividade e o poder de SWRL. Um Build-In é uma cláusula atômica e cada um possui uma função bem específica. Os *Build-Ins* são divididos em módulos, que tratam de problemas específicos, sendo esses formados por: *Build-Ins* para Comparação, *Build-Ins* Matemático, *Build-Ins* para Valores Booleanos, *Build-Ins* para Strings, *Build-Ins* para Datas, Tempo e Duração, *Build-Ins* URIs e *Build-Ins* para Listas, para maiores informações sobre esses módulos consulte [Horrocks et al., 2004].

- *Lógica Unificada*: esta camada propõe a unificação de lógicas para aumentar tanto a expressividade dos documentos descritos através das ontologias quanto garantir a capacidade de raciocínio de tal forma que seja decidível. Nesse sentido, diversas propostas provenientes de pesquisadores da Inteligência Artificial tem estendido (estendendo inclusive a proposta tradicional da OWL que usa Lógica Descritiva) e modificado as várias lógicas [Shadbolt et al., 2006]. Algumas destas propostas equivalem a Abordagens Probabilísticas [Calì et al., 2009], Lógica Fuzzy [Sanchez, 2006], Lógica Temporal [Huang and Stuckenschmidt, 2005] e Lógica Paraconsistente [Schaffert et al., 2005]. Nota-se que há várias abordagens nas quais proporcionam tanto capacidades de expressão quanto de raciocínio, que ultrapassam os próprios padrões da Web Semântica, como podem ser vistos na Figura 2.11;

- *Prova e Confiança*: A camada de prova executa as regras e verifica junto com os

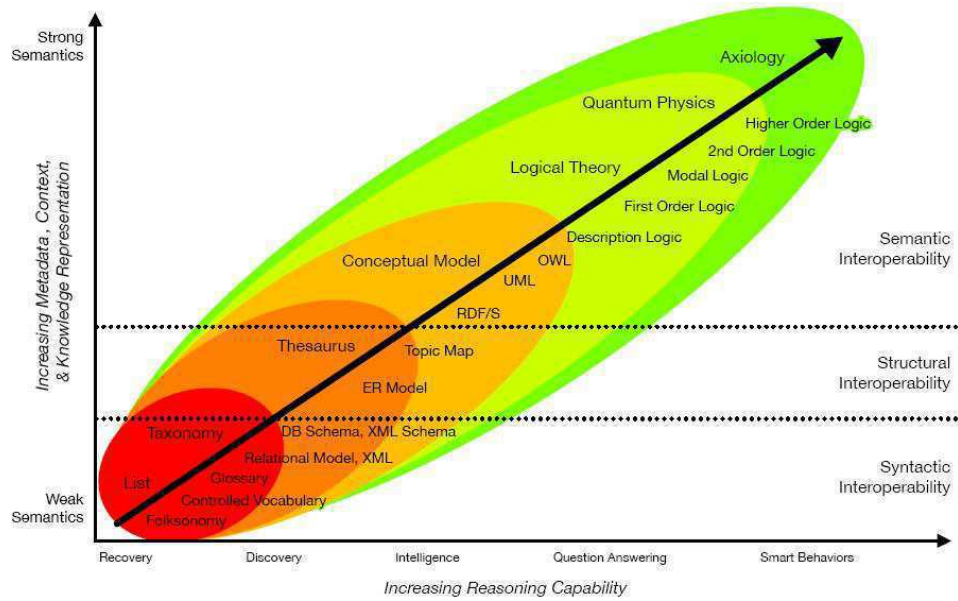


Figura 2.11: Formalismos providos para raciocínio.

Fonte extraída de [Davis, 2008].

mecanismos da camada de confiança se uma dada prova é confiável ou não [Koivunen and Muller, 2002]. Por um lado, a camada de prova pode explicar como uma determinada informação tem sido criada (quais fontes foram usadas), credibilidade, preservar a confiança (e.g. uma fonte está incorreta porque os dados recebidos foram de outra fonte incorreta), etc. Por outro lado, a camada de confiança pode garantir a confiança nos dados, nas fontes de dados, nas organizações e na própria Web. Além disso, outros aspectos que estão envolvidos com estas camadas são segurança, confiabilidade, privacidade, políticas (regras), autorizações e autenticações a sistemas, negociação, entre outras.

No entanto, essas camadas estão atualmente sendo pesquisadas e pequenas aplicações demonstrativas estão sendo construídas. De fato, ainda não há alguma proposta de implementação para as camadas de prova e/ou confiança;

- *Criptografia*: esta camada pode ser utilizada para verificar a autoridade de um documento;
- *Interface do Usuário e Aplicações*: uma das principais camadas da Web Semântica, sendo uma camada a permitir que tanto agentes de software quanto agentes humanos estejam habilitados a fazer uso de todas as potencialidades providas pela

Web Semântica e de seus formalismos de representação de conhecimento. Por um lado, ela permitirá que tanto os usuários técnicos (como engenheiros de software e do conhecimento) quanto leigos (como usuários finais de aplicações) possam fazer uso das aplicações dotadas da noção de semântica. Por outro lado, ela é importante para aplicações, pois permitirá que os mesmos processem as informações contidas nas ontologias e que também disponibilizem aplicações que verifiquem a semântica de documentos em múltiplos formatos. Além disso, aplicações que permitam que qualquer tipo de usuário possa usar ou até mesmo especificar semânticas de documentos de forma fácil.

2.1.1 O papel das ontologias na Web Semântica

A Web semântica possui como base a declaração do conhecimento presente nas aplicações de forma explícita com o intuito de garantir um entendimento não só por humanos, mas também por máquinas. Tal declaração é provida pelas ontologias. Desta forma, as ontologias possuem um papel fundamental para que a Web Semântica torne-se realmente pública, fazendo com que a Internet possua um plano de conhecimento explicitamente declarado, como mostrado na Figura 2.12.

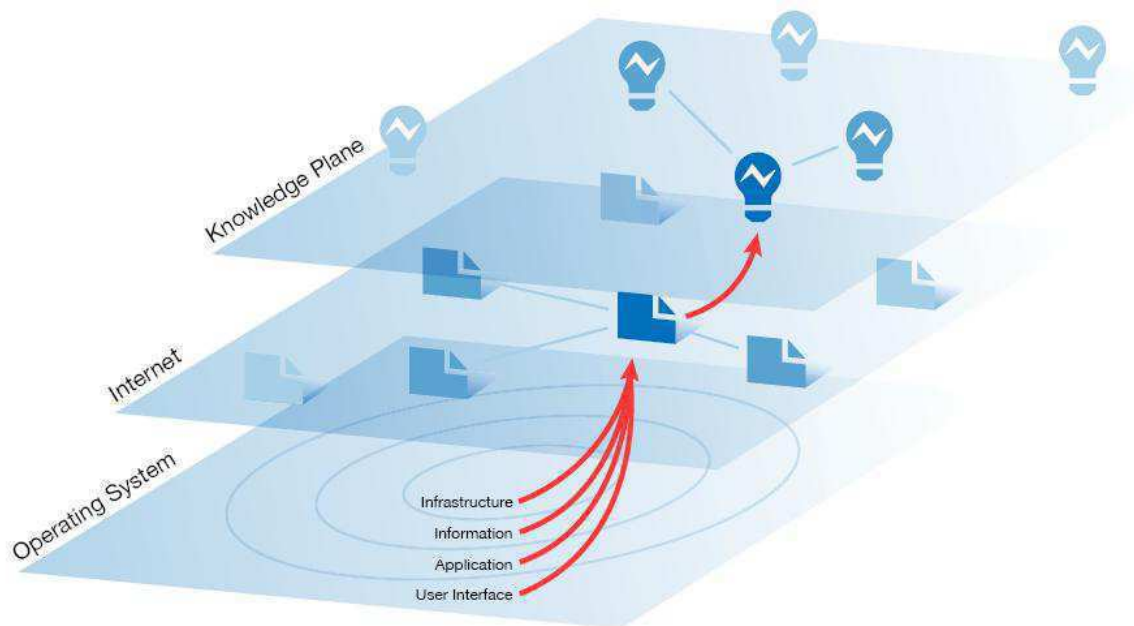


Figura 2.12: Plano de Conhecimento provido pelas Ontologias à Internet.

Fonte extraída de [Davis, 2008].

Do ponto de vista computacional, uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada [Gruber, 1993]. Para melhor entendimento, abaixo seguem detalhes sobre os termos citados nesta definição:

- Explícita: definições de conceitos, relações, restrições e axiomas;
- Formal: compreensível para agentes e sistemas;
- Conceitualização: modelo abstrato de uma área de conhecimento;
- Compartilhada: conhecimento consensual.

Pode-se observar com as características supracitadas sobre ontologias que o conhecimento é explicitado (como mostrado na Figura 2.12). Este conhecimento equivale à descrição de determinada área do conhecimento, garantindo um conhecimento consensual sobre esta área. A partir do momento em que há um consenso, há a possibilidade e a viabilidade de compartilhar tais ontologias e integrá-las a outras áreas de conhecimento (através de outras ontologias), como mostrado na Figura 2.13.

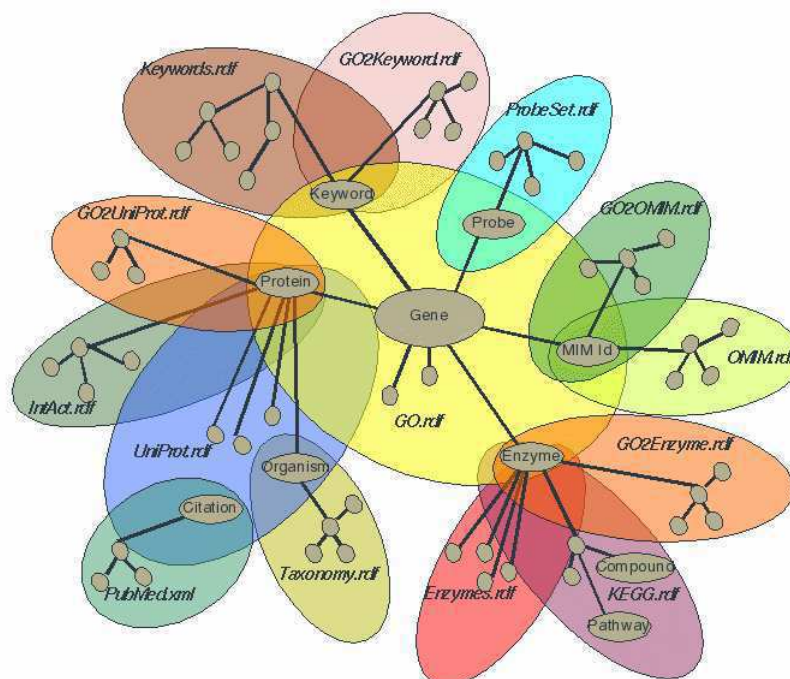


Figura 2.13: Compartilhamento e Integração entre Ontologias.

Fonte extraída de <http://www.w3.org/DesignIssues/diagrams/biopax/ontologies.png>.

Como visto na Figura 2.13, as ontologias são integradas e há uma ampliação do domínio, permitindo uma cobertura maior do conhecimento. Entretanto, para que estas ontologias possam ser integradas, há a necessidade de mapeá-las, permitindo assim que se faça equivalência de termos entre as ontologias. Por exemplo, na Figura 2.13 há duas ontologias (*GO.rdf* e *Enzymes.rdf*) que para haver a integração, foi necessário uma nova ontologia (*GO2Enzyme.rdf*) para fazer o mapeamento entre elas. Um exemplo mais detalhado sobre um mapeamento entre duas ontologias é provido na Figura 2.14.

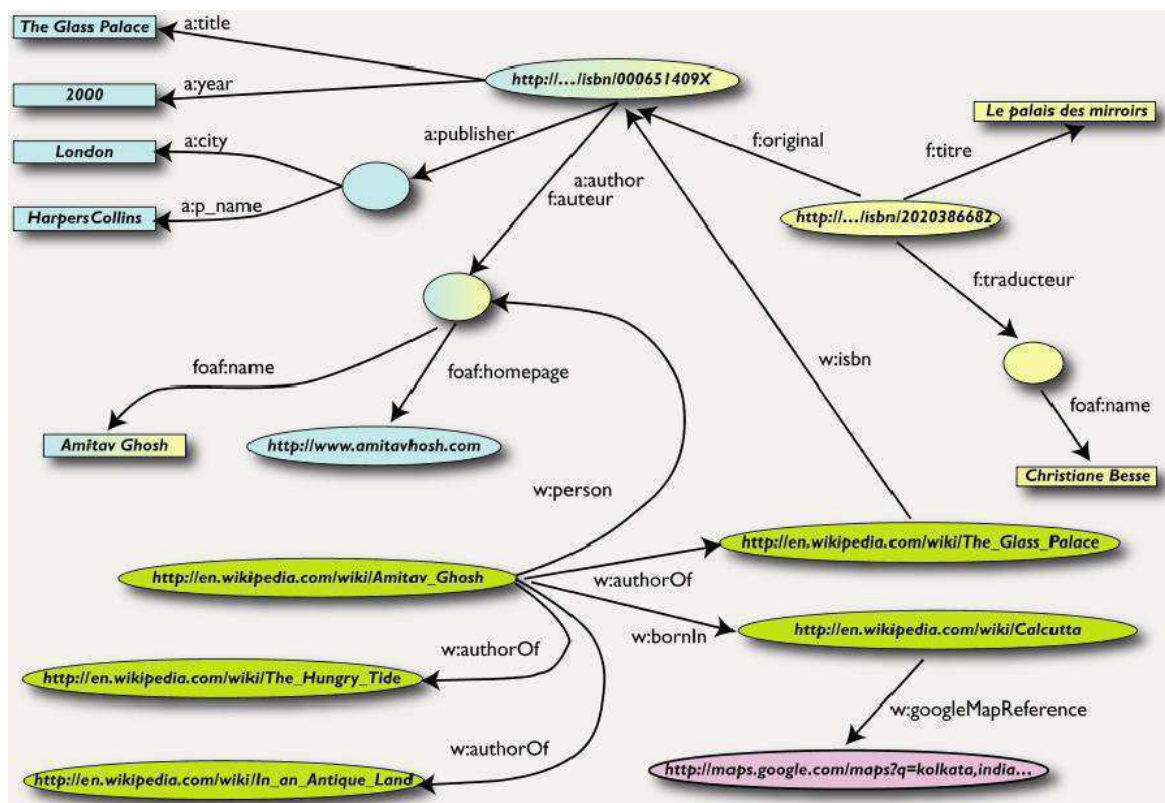


Figura 2.14: Mapeamento/Alinhamento entre Ontologias.

Fonte extraída de [Herman, 2008].

Além disso, estas ontologias têm o objetivo de serem expressas em uma linguagem que permita a compreensão por máquinas, provendo assim a automatização de atividades.

Apesar das ontologias serem a espinha dorsal da Web Semântica, elas correspondem apenas ao primeiro passo para o provimento de aplicações sob a perspectiva da Web Semântica e para a realização da Web Semântica pública. Assim, duas tecnologias que buscam tirar proveito das vantagens providas pela explicitação do conhecimento são os serviços Web semânticos e os agentes de software.

2.1.2 O papel dos SWS na Web Semântica

Esta subseção tem por objetivo descrever o papel dos serviços Web semânticos para a Web semântica, bem como sua importância para a mesma.

Arquitetura Orientada a Serviços

Antes de falar sobre serviço Web, faz-se uma breve apresentação sobre Arquitetura Orientada a Serviço⁷, pois é através da perspectiva de SOA que estes serviços são implementados.

O conceito de SOA, em geral, deriva de uma evolução da computação distribuída, na qual as lógicas de negócio são encapsuladas na forma de serviços de maneira modular, que podem ser invocados por clientes. As principais características providas pela Arquitetura Orientada a Serviços são [Singh and Huhns, 2005]:

1. Neutralidade de Implementação: serviços implementados em diferentes tecnologias podem se comunicar, visto que em SOA, o que importa é a interface do serviço;
2. Fraco acoplamento: em SOA, cada serviço deve ser entendido como um elemento independente;
3. Granularidade: os serviços devem ser tomados como elementos de baixa granularidade, pois deve-se ter uma visão mais abstrata quanto possível, ocultando detalhes como modelagens de ações e interações;
4. Configuração flexível: os sistemas devem suportar configurações de maneira flexível e dinâmica.

Dada algumas das características de SOA, uma das implementações largamente utilizada, tanto na indústria quanto na academia, são os Serviços Web⁸.

Serviço Web

Um serviço Web (ou *Web service*) é um componente de software que pode ser acessado pela Internet e usado remotamente. A comunicação dos serviços Web é construída sobre

⁷Do Inglês *SOA: Service Oriented Architecture*.

⁸Do Inglês *Web Services*.

tecnologias padrões da Internet, como XML, HTTP, e outros protocolos que suportam interoperabilidade. Usando protocolos padronizados, Web services permitem que desenvolvedores criem aplicações que são compatíveis com diferentes linguagens de programação, sistemas operacionais, plataformas de hardware e que são acessíveis de qualquer localização geográfica.

Como resultado, qualquer sistema capaz de efetuar comunicação utilizando os protocolos já bem estabelecidos da Internet, como os citados acima, pode comunicar-se com um serviço Web. A única informação que o provedor do serviço e o consumidor precisam compartilhar são as entradas, saídas e sua localização, disponibilizando meramente sua interface e abstraindo totalmente a maneira como foram implementadas.

Segundo a visão da IBM [Freger, 2001] a arquitetura dos serviços Web está baseada na interação dos seguintes papéis: provedor do serviço, registrador de serviços e consumidor de serviços. As interações envolvem a publicação, a busca e a invocação do serviço. Neste esquema, o provedor do serviço desenvolve o serviço e publica sua descrição em um registrador de serviços, o consumidor de serviço efetua busca nos registros de serviços para descobrir o serviço que atende às suas necessidades, através das descrições adicionadas pelos provedores de serviços. Após descobrir o serviço ideal, o consumidor utiliza a descrição para interagir diretamente com o consumidor e invocar o serviço desejado. Esta interação pode ser vista na Figura 2.15.

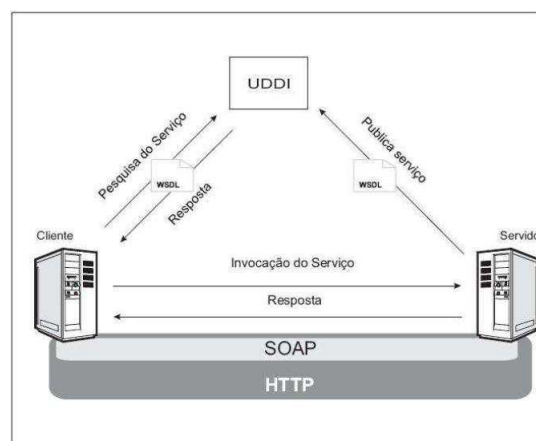


Figura 2.15: Modelo da interação entre cliente, servidor e registro UDDI.

Fonte: [de Barros Costa et al., 2007]

Diante do exposto, algumas tecnologias foram desenvolvidas como forma a permitir

tanto a publicação quanto a descoberta e invocação de serviços, como: registros UDDI (repositórios de descrições de serviços que permitem a busca), linguagem WSDL⁹ (linguagem para descrição de interfaces, protocolos de invocação e detalhes de distribuição de Web Services) e o protocolo de troca de mensagens SOAP¹⁰.

Por outro lado, sob a perspectiva da Web Semântica, há a necessidade de disponibilização de serviços de forma que máquinas entendam e agentes inteligentes possam descobrir, compor e executar automaticamente [McIlraith et al., 2001]. Entretanto, destaca-se que WSDL é uma linguagem baseada em XML (puramente sintática), não provendo uma especificação formal e semântica para serem utilizadas pelos agentes de software. Como consequência, não há como garantir que haverá automatização nas atividades de descoberta, composição e invocações dos serviços.

Por esta razão, foram propostos os Serviços Web Semânticos, que buscam garantir uma descrição semântica dos serviços Web através da utilização de ontologias. A próxima subseção descreve esta tecnologia.

Serviços Web Semânticos

Os Serviços Web Semânticos podem ser vistos como um ponto de intersecção entre a tecnologia dos Serviços Web e a Web semântica por prover a interoperabilidade semântica. Esta tecnologia herda características inerentes às duas abordagens, ou seja, por um lado provendo a interoperabilidade semântica encontrada na Web semântica, por outro lado, aliando a dinamicidade de recursos presentes nos serviços Web [Daconta et al., 2003] (Vide Figura 2.16).

Segundo [Payne and Lassila, 2004], os serviços Web semânticos podem ser definidos como um aprimoramento das descrições dos serviços Web, através da utilização de anotações semânticas, de forma que possa ser conseguido um alto grau de automação, tanto na descoberta de serviços quanto na composição, monitoramento e invocação de serviços Web.

Dentre as iniciativas para agregar semântica à atual tecnologia de serviços Web, destacou-se o OWL-S¹¹ Embora algumas vezes a OWL-S seja referida como uma

⁹Do Inglês *Web Service Description Language*

¹⁰Do Inglês *Simple Object Access Protocol*

¹¹Que significa *Ontology Web Language for Services*.

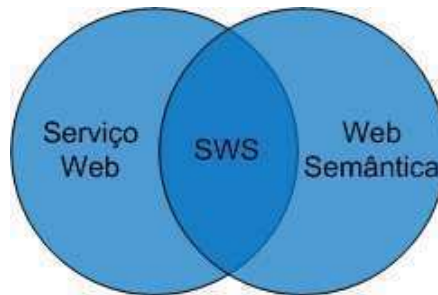


Figura 2.16: Serviço Web + Web Semântica= Serviço Web Semântico.

Fonte: [de Barros Costa et al., 2007]

linguagem porque provê um vocabulário padrão para representação de serviços, trata-se de uma ontologia de topo (*Upper Ontology*) lançada em meados de 2004 em continuação ao projeto DAML-S [Ankolekar et al. 2004]. Por ser uma recomendação da W3C, a OWL-S tem atraído o interesse de grande parte da comunidade científica relacionada a serviços Web semânticos e grande parte das ferramentas desenvolvidas tem como base a utilização dela.

A OWL-S foi desenvolvida tendo como objetivo permitir a criação de serviços Web semânticos capazes de atender aos requisitos esperados de um serviço semântico (ou seja, descoberta, composição e invocação automática) [McIlraith et al., 2001]. O modelo abstrato de um serviço Web semântico em OWL-S pode ser visto na Figura 2.17.

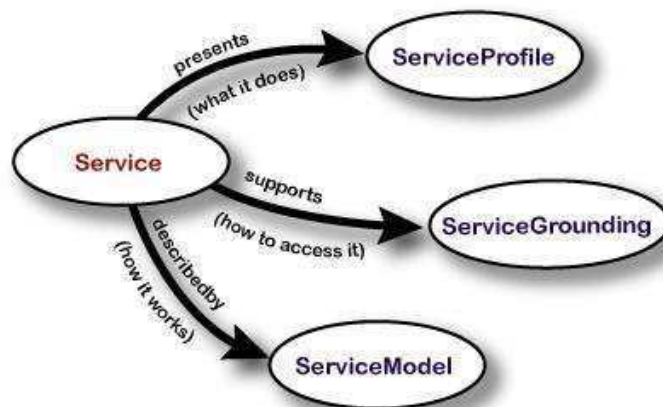


Figura 2.17: Estrutura proposta de um serviço semântico em OWL-S.

Fonte: <http://www.w3.org/Submission/OWL-S/Service-Ontology1.1.gif>

Como mostrado na Figura 2.17, a OWL-S foi desenvolvida tendo em mente três componentes básicos, sendo eles:

1. Um perfil (*Profile*) que apresenta as "intenções" do serviço, isto é, qual a funcionalidade do serviço, o que ele provê;
2. Um modelo do Serviço (*ServiceModel*) que apresenta um maior detalhamento do funcionamento do serviço;
3. Um *Grounding* que apresenta detalhes relacionados à maneira como o serviço pode ser invocado.

Apesar dos serviços proverem toda a infra-estrutura para garantir a descoberta, composição e invocação automatizada, ainda é necessária uma entidade de software autônoma que seja responsável por utilizar a infra-estrutura provida pelos SWS. Com isso, os agentes de software foram escolhidos como as entidades responsáveis por estas atividades. A subseção que segue descreve características inerentes aos agentes de software.

2.1.3 O papel dos Agentes na Web Semântica

Do ponto de vista da Inteligência Artificial, um agente é uma entidade que possui a capacidade de perceber um ambiente através de sensores, e agir no ambiente através de atuadores. Já do ponto de vista da Engenharia de Software, um agente é um componente de software com *threads* internas de execução (tanto pró-ativo quanto reativo) e que pode participar de protocolos de interação complexos e com estado.

Algumas características/propriedades que estão presentes nos agentes de software são:

- Situados: atuam em um ambiente;
- Interatividade: comunicam-se com outros agentes e com o ambiente;
- Autonomia: possuem a capacidade de agir sem intervenção externa;
- Pró-atividade: atuam não somente em resposta ao ambiente - são orientados a objetivos;
- Racionalidade: são capazes de selecionar suas ações baseadas em seus objetivos.

Entretanto, a construção de software robusto e de qualidade para domínios como ambientes educacionais adaptativos representa uma tarefa complexa. Neste contexto, a função do engenheiro de software é disponibilizar estruturas e técnicas que facilitem o gerenciamento desta complexidade. Assim sendo, para o desenvolvimento de sistemas complexos, há a necessidade de ações/medidas em prol do gerenciamento desta complexidade, como:

- **Decomposição:** recebe grandes problemas como entrada e os subdivide em pedaços menores, limitando o domínio de atuação e utilizando o princípio do “Dividir para conquistar”;
- **Abstração:** processo de definir um modelo simplificado do sistema que represente alguns dos detalhes e propriedades e omita outros, limitando o escopo de interesse em um determinado instante;
- **Organização:** processo de definir e manter os relacionamentos entre os componentes. Desta forma, permitindo uma visualização de um conjunto de pequenos componentes como uma unidade unificada.

Tendo apresentado visões e conceitos relacionados a sistemas complexos, torna-se visível que a maioria dos problemas considerados anteriormente requer ou envolve vários agentes para representar a natureza descentralizada do problema, múltiplas atividades de controle e/ou os interesses competitivos.

Logo, esta perspectiva é vista como um sistema multi-agentes, onde os agentes precisarão interagir entre si, almejando atingir seus objetivos particulares ou gerenciar dependências inatas de ambientes complexos. Estas interações podem variar de simples operações semânticas, que ocorrem através de interações cliente-servidor, para ricas interações sociais, onde encontramos a habilidade de cooperar, coordenar e negociar sobre um plano de ações.

Um sistema multi-agente geralmente atua em um contexto organizacional, definindo a natureza das interações, por exemplo: um agente pode ser uma pessoa trabalhando em um time, ou ainda ser um gerente em uma determinada organização. Logo, assim como os contextos são dinâmicos na vida real, os sistemas multi-agentes também apresentam

suporte a dinamicidade no estabelecimento de suas relações. Com isso, os agentes que atuam em um sistema multi-agentes geralmente possuem capacidade (além das capacidades citadas anteriormente) de:

- Sociabilidade: um agente tem que ter a capacidade de interação social com outros agentes;
- Cooperação¹²: um agente pode cooperar/ajudar para que outro agente alcance o seu objetivo.
- Colaboração: o agente pode colaborar em alguma atividade, coordenando tal interação inter-agentes.

Por fim, a Figura 2.18 aborda a visão de um sistema multi-agente, descrevendo suas relações e interações, além da visão que os agentes tem do ambiente. Desta forma, o paradigma orientado a agentes diz respeito a decompor os agentes em múltiplos componentes autônomos que podem atuar e relacionar-se em um caminho flexível, almejando atingir seu conjunto de objetivos. Além disso, modelar abstratamente uma parte de interesse do problema em agentes, interações e unidades organizacionais, e finalmente, desenvolver estruturas e mecanismos que frequentemente são usados para descrever e gerenciar a complexidade e mudanças ocorridas nos relacionamentos organizacionais existentes entre os agentes.

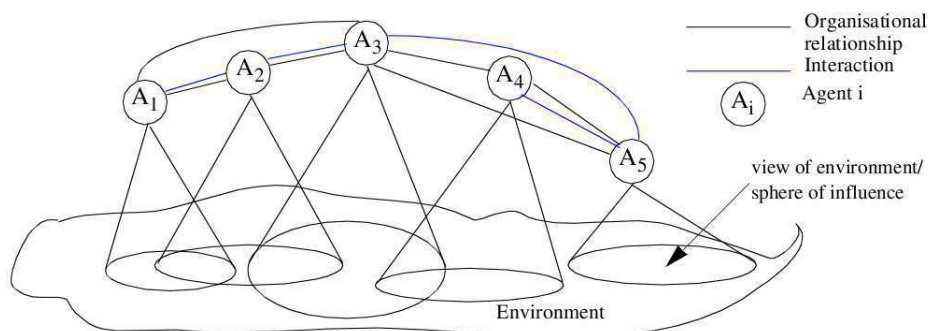


Figura 2.18: Visão de um Sistema Multiagente.

Fonte extraída de [Jennings and Wooldridge, 2000]

¹²Agentes cooperativos trabalham buscando alcançar objetivos comuns [Lesser, 1999].

Os agentes de software potencializam os recursos providos pela Web Semântica de modo a permitir interação entre diferentes entidades sem ter havido uma prévia especificação da interação entre os agentes. Além disso, a partir do momento que milhares (ou até milhões) de ontologias (Vide Figura 2.19) estiverem disponíveis na Web, os agentes podem fazer uso das mesmas em busca de serviços que possuam os requisitos necessários para a execução e cumprimento de determinada tarefa solicitada.

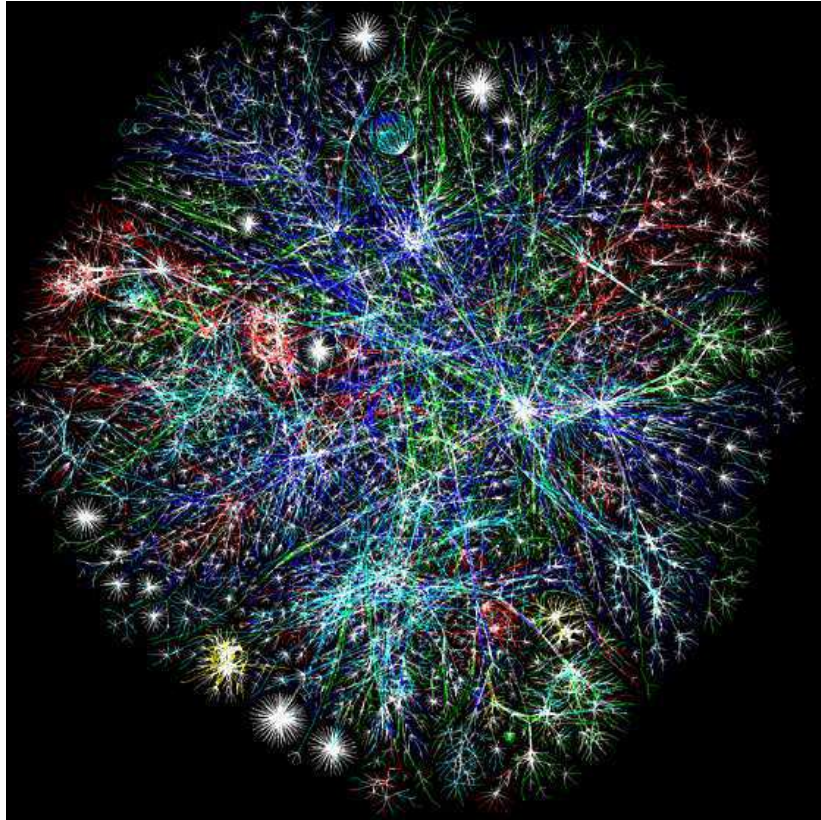


Figura 2.19: Visão da Internet, podendo ser estendida e valorada com Ontologias.

Fonte: <http://socialwhisper.files.wordpress.com/2008/12/semantic.png?w=265&h=265>

2.2 O papel da Web Semântica nos Ambientes Educacionais

Esta seção tem o objetivo de descrever características dos ambientes educacionais adaptativos, bem como o papel da Web Semântica nestes ambientes. Estes aspectos são descritos nas subseções que seguem.

2.2.1 Evolução dos Ambientes Educacionais

Basicamente, a maior parte da pesquisa envolvendo a concepção de ambientes educacionais pode ser vista como interação entre dois tipos de jogadores, a saber, o ambiente educacional e o usuário, como mostrado na Figura 2.20.

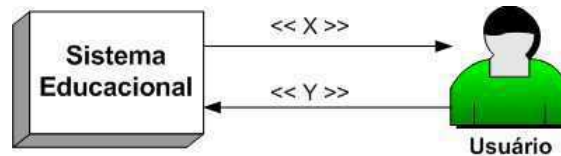


Figura 2.20: Interação Abstrata em Sistemas Educacionais.

De modo geral, o processo de interação ocorre seguindo um determinado contexto através de troca de mensagens ($\langle X \rangle$, $\langle Y \rangle$) entre os jogadores envolvidos. Entretanto, para que tal interação seja possível, os jogadores devem interagir seguindo seus conhecimentos e limitações tanto sobre o contexto abordado quanto sobre os próprios jogadores envolvidos, fazendo com que cada jogador possua um ambiente para acesso à informação/conhecimento objetivando maximizar a interação. Outro aspecto importante é que os jogadores que interagem, particularmente no lado dos usuários, podem variar consideravelmente. Dentre as variabilidades possíveis, alguns exemplos são estudantes, grupo de estudantes, tutores, autores, entre outros.

Estes ambientes educacionais passaram por grandes mudanças que vão desde aspectos pedagógicos (como ensino baseado em problemas, colaboração e modelos de avaliação através da interação) até os aspectos tecnológicos (como a inserção de técnicas provenientes da Inteligência Artificial para melhorar as interações e a utilização da Internet).

Sob uma perspectiva histórica, os ambientes de aprendizagem assistidos por computador remontam às décadas de 50 e 60 do século XX, com a chegada dos sistemas CAI (*Computer-Assisted Instruction*, idealização baseada na instrução programada). No final da década de 60 do século XX, opondo-se à proposta dos CAI, houve o surgimento dos Micromundos, valorizando e abordando uma proposta de aprendizagem através da ação¹³. Entretanto, a partir dos CAI, pode-se observar duas perspectivas evolutivas, sendo

¹³Outras abordagens relacionadas à aprendizagem, que surgiram, aproximadamente, na mesma época,

uma com a inserção inteligência artificial na educação e a outra com o surgimento da Web. A primeira deu início à área de Inteligência Artificial e Educação (IAEd), na década de 70 do século XX, dando origem aos ICAI (*Intelligent CAI*) ou STI¹⁴ (Sistemas Tutores Inteligentes), os quais foram concebidos para prover os estudantes com informações, tutoria personalizada [Centinia et al., 1995], além de mecanismos inteligentes para resolução automática de problemas [Self, 1995]. Além disso, constatou-se um movimento em favor de modelos computacionais de apoio à aprendizagem calcado na noção de cooperação, na década de 90. Nesse sentido, propõe-se que os STI tradicionais evoluam para os Ambientes Interativos/Inteligentes de Aprendizagem (AIA)¹⁵ ou ainda Sistemas Tutores Cooperativos [Costa, 1997]. Além disso, essa década também foi marcada pelo advento dos agentes de software [Plekhanova, 2002], provendo mais recursos aos ambientes interativos de aprendizagem. Com isso, a interação abstrata abordada na Figura anterior pode ser estendida como mostrada na Figura 2.21.

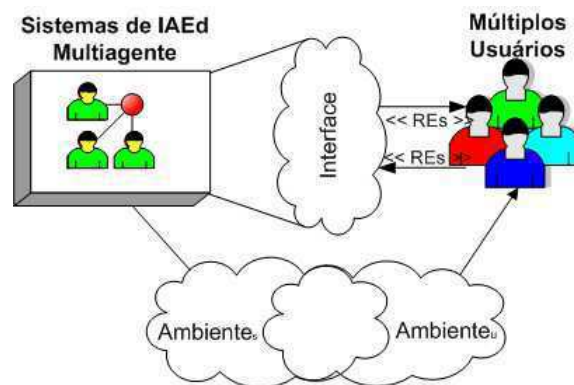


Figura 2.21: Interação em Sistemas Educacionais com descrição de um ambiente.

Por outro lado, através do desenvolvimento da Web, surgiram os ambientes de educação a distância (EaD)¹⁶, já na década de 90 do século XX. Estes ambientes visam a interação entre estudantes e tutores, fazendo uso da tecnologia da informação para garantir esta comunicação [Silva, 2003]. Os ambientes de educação a distância possuem referem-se a Simuladores e Jogos Educacionais.

¹⁴O foco no processo de aprendizagem leva em consideração 3 (três) perguntas básicas: i) o que ensinar?, ii) como ensinar? e iii) para quem ensinar?

¹⁵Do inglês: *Interactive/Intelligent Learning Environment* - ILE.

¹⁶Também conhecidos como e-Learning ou LMS - *Learning Management System*.

diversas características como ensino em múltiplos domínios (através da apresentação de materiais/objetos de aprendizagem), *feedback* simples e configurado a *priori*, suporte colaborativo, entre outros. Por outro lado, os ambientes de inteligência artificial e educação provêm atividades personalizadas tanto para usuários quanto para grupos de usuários, através da utilização de informações dos mesmos. Além disso, estes ambientes dão apoio em atividades de estudantes e tutores, como resolução de problemas, acompanhamento de atividades, avaliação, entre outras.

Apesar das características estarem explicitamente declaradas, diversas limitações são comumente identificadas nestes tipos de ambientes educacionais. De um lado, os ambientes clássicos de educação a distância falham na falta de controle de avaliação personalizada e na falta de personalização/adaptação às características dos usuários presentes no ambiente, como estudantes, tutores, autores, monitores, entre outros [Brooks et al., 2006]. Do outro lado, os ambientes AIED têm o foco estabelecido na resolução de problemas de aprendizagem e nas atividades sendo trabalhadas de forma individualizada/personalizada ao modelo do aprendiz, fazendo com que estas características sejam de domínios restritos e que possuam um alto custo de desenvolvimento e manutenção [Murray, 2003][Manuel Rodrigues and Santos, 2005].

Desta forma, estas limitações, identificadas em ambientes de educação a distância tradicionais, fizeram com que houvesse uma convergência natural entre os mesmos com ambientes de inteligência artificial e educação, isto entre o final da década passada e o início desta, criando assim os ambientes conhecidos como Ambientes de Educação a Distância Adaptativos¹⁷ ou *Adaptive e-Learning* (vide Figura 2.22) [Garcia-Barrios, 2006].

Com esta convergência, os sistemas de *e-learning* passam a ter características que antes eram exclusivas de sistemas AIED, como atividades personalizadas/individualizadas, resolução de problemas, *feedback* elaborado, entre outras [Self, 1995]. A construção destes tipos de sistemas está relacionada com diversos aspectos das áreas de engenharia de software, inteligência artificial e educação [Mizoguchi and Bourdeau, 2000][Brooks et al., 2006][Bittencourt et al., 2008b].

¹⁷Entende-se por ambiente educacional adaptativo (AEA), os sistemas que alteram suas características para atender às necessidades dos usuários do sistema [Oppermann, 1994]



Figura 2.22: Abordagem para Ambientes de Educação a Distância Adaptativos.

A Figura 2.23 apresenta um histórico evolutivo dos Ambientes de Aprendizagem Assistidos por Computador (AAAC) [Bittencourt et al., 2008b] como descrito nesta subseção.

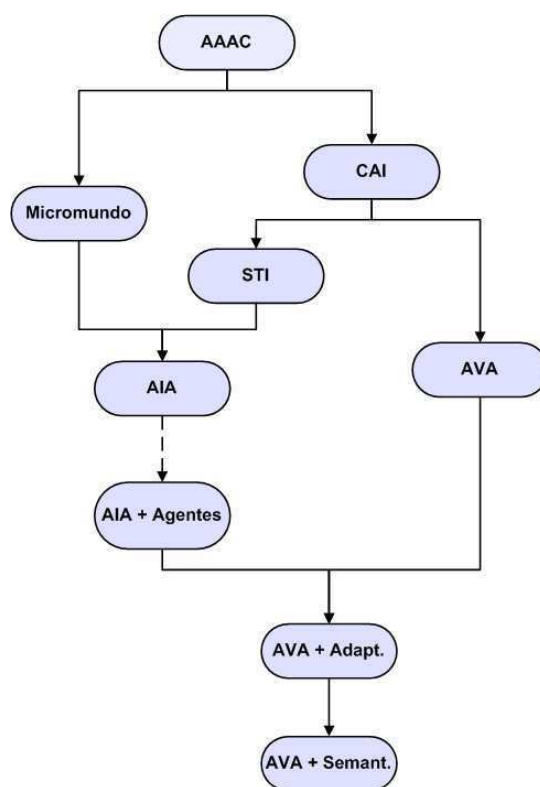


Figura 2.23: Evolução dos AAAC.

A subseção que segue descreve características dos ambientes educacionais baseados na Web semântica.

2.2.2 Ambientes Educacionais baseados na Web Semântica

Apesar dos ambientes educacionais adaptativos proporcionarem novas possibilidades de ensino, diversos outros problemas podem ser identificados com a convergência entre duas grandes áreas, como alto custo de desenvolvimento, complexidade em desenvolver algoritmos de inteligência artificial, integração entre técnicas de inteligência artificial, disponibilização de ferramentas interativas, dificuldade de compartilhamento de materiais, extensibilidade, interoperabilidade entre aplicações e alto custo de manutenção [Brooks et al., 2006].

De fato, é bastante evidente que a especificação, desenvolvimento e manutenção destes sistemas são bastante complexas e demanda bastante tempo. Desta forma, uma das melhores formas de gerenciar esta complexidade é através da decomposição dos problemas. Além disso, diversos dos problemas supracitados estão relacionados com a tentativa de representar as informações na Web de uma forma que computadores possam entendê-las e manipulá-las [Devedzic, 2004][Dicheva and Dimitrova, 2008][Bittencourt et al., 2009b]. Este campo de pesquisa é conhecido como Web semântica. Desta forma, abordam-se os ambientes conhecidos como ambientes educacionais adaptativos e semânticos.

Esta nova geração de ambientes surge com diversos propósitos como de garantir interoperabilidade entre as aplicações, reúso de conhecimento e software e automatização de processos/atividades. Estas características são marcadas pela área de Web Semântica. Tanto a comunidade de inteligência artificial e educação quanto a de educação a distância estão diante de uma nova geração de ambientes educacionais que possuem características de adaptatividade e semântica embutida, permitindo desta forma a interação entre agentes humanos e artificiais¹⁸.

Os citados ambientes proporcionam diversos ganhos que vão desde a construção dos ambientes, através da redução de tempo, risco e custo de desenvolvimento e manutenção dos sistemas até o aumento da produtividade e satisfação dos usuários através de interfaces inteligentes e adaptáveis, além de interoperabilidade entre as aplicações,

¹⁸É importante frisar que estes ambientes levam em consideração as características atuais da Web, conhecida com Web 2.0. Ou seja, a fusão entre a Web 2.0 e Web Semântica é conhecido como Web 3.0 ou do inglês *Social Semantic Web*.

permitindo o reúso automático de código e conhecimento, entre outros. Estes ambientes são conhecidos como Ambientes Educacionais Adaptativos e Semânticos¹⁹

Um ambiente educacional adaptativo e semântico busca a automatização, integração e reúso entre aplicações para prover mais adaptação para os usuários e mais inteligência para o sistema. Ou seja, estes ambientes fazem uso de tecnologias provenientes da Web semântica (e.g. ontologias e serviços Web semânticos) e da inteligência artificial (e.g. agentes de software) para atingir os seus objetivos.

Apesar de algumas iniciativas para prover uma arquitetura para ambientes educacionais adaptativos e semânticos terem sido propostas pela comunidade [Devedzic, 2003][Koper, 2001], elas não descrevem todos os papéis e/ou módulos presentes em tais ambientes. Além disso, a relação entre os módulos não é expressada.

2.3 Conclusão

Este capítulo descreveu o referencial teórico necessário para apoiar a pesquisa de tese em questão. Foram abordadas características das aplicações baseadas na Web Semântica, bem como sua potencialidade nos ambientes educacionais. Além disso, foram explicadas como a arquitetura orientada a serviço e os sistemas multi-agentes podem ser utilizados nos ambientes educacionais.

¹⁹Do inglês: *Semantic Web-based Educational Systems*.

Capítulo 3

Trabalhos Relacionados

*Não devemos evitar as dificuldades à medida
que procuramos nos desenvolver.
E sim desafiá-las de forma que possamos nutrir
a chama do crescimento e da criatividade dentro de nós.*
Daisaku Ikeda

Este capítulo fornece os principais trabalhos relacionados à proposta de tese aqui descrita.

De um modo geral, ambientes educacionais adaptativos e semânticos têm sido abordados na literatura, fornecendo evoluções significativas de resultados de pesquisas. A literatura tem revelado ambientes com objetivos específicos, orientados a domínio, como abordagens que usam serviços Web semânticos [Millard et al., 2006][Gomez-Perez et al., 2004][Costa et al., 2008] [Wang et al., 2004][Carmagnola et al., 2005], ou através da utilização de ontologias [Muñoz and de Oliveira, 2004][Wang and Chen, 2008][Azouaou and Desmoulins, 2006]. Além disso, pode-se observar vários consórcios entre grupo de pesquisa com o objetivo de investigar ambientes com características de adaptatividade e semântica [Kieslinger and Simon, 2005][LORNET, 2008][Boticario et al., 2007]. Outro aspecto considerado pela literatura é a proposição de modelos conceituais para construir tais sistemas [Devedzic, 2003][Koper, 2001][Mitrovic and Devedzic, 2004].

Em particular, a discussão aqui apresentada considera como relacionados a família dos sistemas educacionais com características de adaptatividade e semântica embutida, focalizando aspectos de engenharia de software, inteligência artificial e Web Semântica. Sob a perspectiva da engenharia de software, deseja-se investigar a família de sistemas que possibilitem o reúso de código e extensibilidade, além da garantia de interoperabilidade entre ambientes educacionais. Sob a perspectiva da inteligência artificial, avaliam-se ambientes que utilizem agentes de software e mecanismos inteligentes que proporcionem adaptatividade a tais ambientes educacionais. Avaliam-se ainda os ambientes que garantem a automatização dos processos. Por fim, porém não menos importante, pretende-se avaliar a utilização de recursos provenientes da Web Semântica, como a utilização de ontologias, reúso de conhecimento e serviços Web semânticos. Nessa perspectiva, descreve-se aqui um apanhado de trabalhos considerados relacionados à proposta desta tese.

As seções seguintes abordam os trabalhos relacionados. A explanação que é dada sobre estes trabalhos leva em consideração as características e arquitetura presentes em cada ambiente, objetivando facilitar o seu entendimento, além de suas vantagens e limitações. Após a exposição dos trabalhos relacionados, os mesmos são comparados através das características supracitadas.

3.1 SWSOA

Este trabalho tem por objetivo construir ambientes educacionais provendo adaptação e integração de diferentes metadados para recursos educacionais, sendo chamado de SWSOA¹. Este trabalho foi desenvolvido pelo Instituto de Mídia de Conhecimento, da Universidade Aberta².

A abordagem utilizada para prover a integração dos metadados e a adaptação dos recursos educacionais ocorre através de serviços Web semânticos. A arquitetura do sistema é dividida em camadas (vide Figura 3.1), dando suporte aos processos de aprendizagem através da composição e seleção de serviços Web semânticos [Dietze

¹Do Inglês: *Semantic Web Service Oriented Architecture*.

²<http://www.open.ac.uk/>

et al., 2007c].

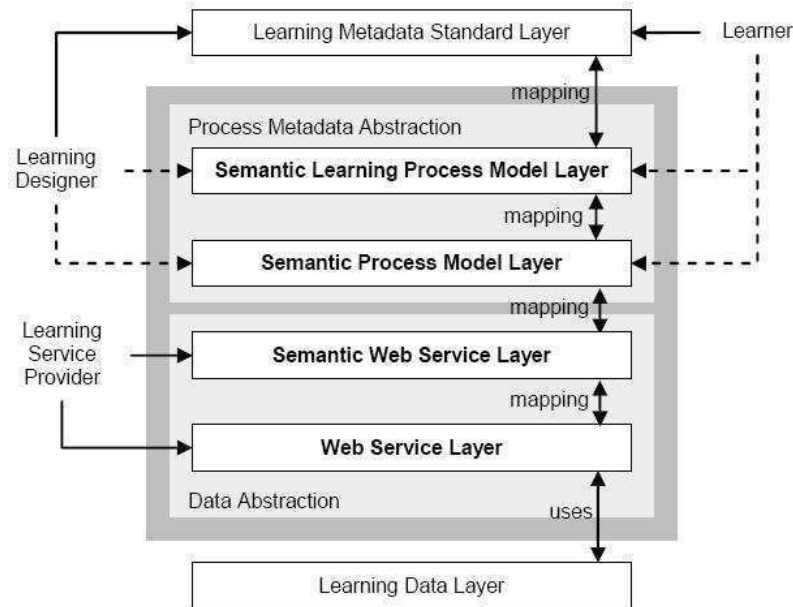


Figura 3.1: Arquitetura proposta baseada em Agentes e Serviços.

O princípio utilizado na arquitetura foi o uso de camadas com mapeamentos entre as mesmas. As abstrações presentes na arquitetura são descritas como seguem [Dietze et al., 2007a][Dietze et al., 2007b]:

- **Dados do Estudante:** esta camada equivale ao armazenamento dos dados do estudante;
- **Abstração dos dados:** todo o acesso aos dados dos estudantes para adaptação é feito através da utilização de serviços Web. A abstração dos dados possui duas camadas, sendo elas: i) Camada de serviço Web: um determinado serviço Web é selecionado para suprir determinado objetivo educacional e ii) Camada de serviço Web semântico: esta camada foi adicionada para garantir a seleção, composição e invocação dos serviços, através da descrição formal e explícita das capacidades de um determinado serviço;
- **Abstração do processo de metadados:** representa uma abstração dos metadados educacionais disponibilizados e dos processos dentro de um determinado domínio educacional. Esta abstração possui duas camadas, sendo elas: i) camada do modelo

de processo semântico: representação semântica de padrões de metadados educacionais, buscando garantir a interoperabilidade entre estes diferentes metadados e ii) camada de modelo de processo de aprendizagem semântico: esta camada é necessária para garantir o modelo de processo de aprendizagem para um domínio educacional específico e através da utilização de diferentes padrões de metadados. Por exemplo, esta camada suporta o mapeamento entre objetivos de aprendizagem e objetivos de negócio, dando assim suporte para tipos de processos organizacionais;

- Padrões de metadados do estudante: o estudante se comunica diretamente com esta camada através da utilização dos padrões de metadados.

É importante frisar que esta arquitetura faz o mapeamento desde a primeira camada até a última, como mostrado na Figura 3.1.

A vantagem do trabalho está na possibilidade de garantir a interoperabilidade entre padrões de metadados bem como na utilização de serviços Web semânticos para seleção, composição e invocação automática de serviços. Porém, toda a adaptação provida pelo ambiente limita-se à utilização dos metadados disponibilizados. Ou seja, nenhum mecanismo inteligente é provido para garantir a adaptação. Além disso, este ambiente não garante a interoperabilidade entre ambientes educacionais.

3.2 Ambiente com MAS e SOA

Este trabalho apresenta uma arquitetura para a construção de ambientes educacionais adaptativos e foi desenvolvido pelo centro de computação e sistemas de informação da Universidade de Athabasca³, Canadá.

Esta arquitetura se baseia numa abordagem multi-agentes e arquitetura orientada a serviços [Lin et al., 2006], permitindo a construção destes ambientes educacionais adaptativos. A proposta possui os seguintes tipos de agentes:

- Agentes Pessoais: os agentes pessoais são agentes de interface, de modo a garantir a comunicação entre os agentes humanos e os agentes de tarefa. Os agentes

³<http://www.athabascau.ca/>

peçoais presentes no ambiente são utilizados para apoiar dois tipos de usuários: i) instrutores: ajuda o instrutor em tarefas de geração, distribuição e manutenção de cursos *on-line*; ii) estudantes: são disponibilizados para apoiar o estudante ao longo do desenvolvimento de determinado curso;

- Agentes de Tarefa: estes agentes são i) provedores de serviços/execução de tarefas para os agentes pessoais (instrutor e estudante) e ii) clientes dos serviços Web disponibilizados, ou seja, executam tarefas específicas como prover serviços, conhecimento e recursos educacionais.

Além disso, esta arquitetura possui uma abordagem orientada a serviços, onde os serviços são para o gerenciamento de conhecimento e recursos educacionais. A arquitetura do ambiente pode ser vista na Figura 3.2.

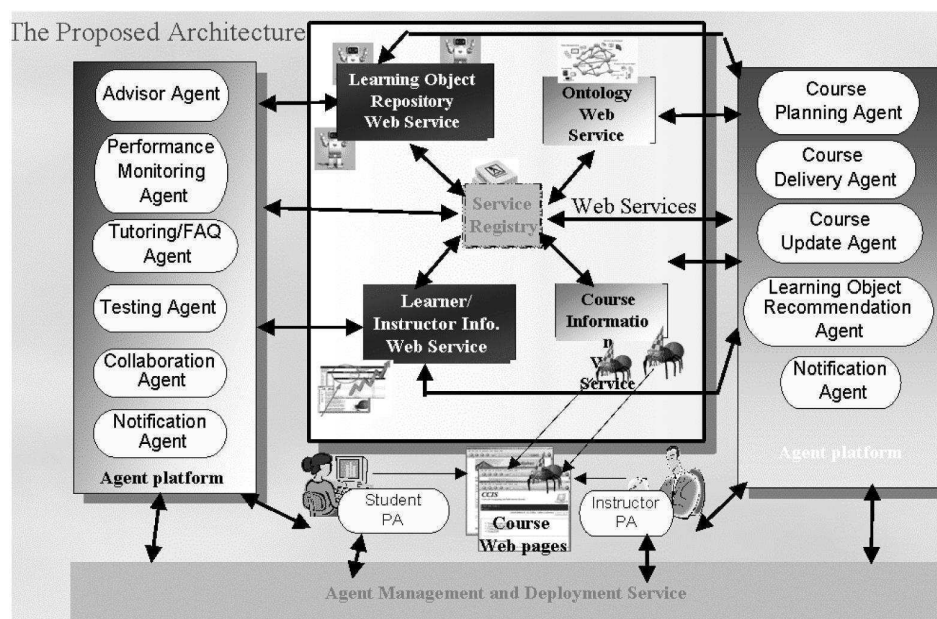


Figura 3.2: Arquitetura proposta baseada em Agentes e Serviços.

Pode-se ainda observar diversos agentes disponibilizados na plataforma de agentes. Estes agentes são utilizados para apoiar o agente pessoal do instrutor (sendo apoiado pelos agentes de planejamento de curso, distribuição do curso, atualização do curso, recomendação de objetos de aprendizagem e de notificação) e o agente pessoal do estudante (sendo apoiado pelos agentes de sugestão de objetos de aprendizagem,

monitoramento de performance, tutoramento, teste, colaboração e notificação) [Lin et al., 2005].

A vantagem da arquitetura está na possibilidade de adicionar novos serviços para serem utilizados pelos agentes e na descentralização dos objetos de aprendizagem disponibilizados, entretanto, diversos problemas podem ser identificados:

- Agentes pessoais: estes agentes fazem com que haja uma limitação na arquitetura, não permitindo a adição de novos papéis para o ambiente. Isto é, caso seja necessário adicionar um monitor para utilizar o sistema, há a necessidade de criar um agente pessoal para o mesmo, além de criar uma nova plataforma com as funcionalidades específicas do monitor;
- Agentes de tarefas: apesar destes agentes possuírem funcionalidades fundamentais para um ambiente educacional, nenhuma facilidade é disponibilizada para a criação destes agentes. Assim, toda funcionalidade provida pelo estudante é implementada necessariamente por um único agente. Isto faz com que, para toda nova funcionalidade necessária, por mais simples que seja, necessite da criação de um novo agente com a funcionalidade, fazendo que um agente seja criado sem a real necessidade [Jennings and Wooldridge, 2000];
- Tecnologias: algumas limitações referentes às tecnologias são: i) Serviços Web Semânticos: a utilização de serviços Web padrões, apesar de garantir a possibilidade de adicionar novas funcionalidades nos serviços, faz com que para toda nova funcionalidade provida, haja a necessidade de manutenção nos agentes para que possa fazer a invocação destes serviços; ii) interoperabilidade: apesar das tecnologias utilizadas propiciarem facilidades para a garantia de interoperabilidade, não há esta possibilidade de comunicação entre ambientes educacionais; iii) mecanismos inteligentes: nenhum mecanismo inteligente é provido para garantir a adaptação do ambiente educacional, isto é, caso um novo ambiente seja criado e pretenda-se fazer determinado tipo de adaptação, a equipe de desenvolvimento deve implementar tal mecanismo.

3.3 ACOA

Este trabalho aborda uma arquitetura para a construção de ambientes educacionais adaptativos baseados em componentes e agentes, chamada ACOA⁴. O mesmo foi desenvolvido pelo grupo de pesquisa de Ciência da Computação do Instituto Politécnico Nacional⁵, México.

Apesar do nome do ambiente endereçar apenas dois paradigmas, o mesmo faz uso de diversas tecnologias e paradigmas. A utilização de cada tecnologia depende do propósito de cada ferramenta. As ferramentas presentes são [Cruz et al., 2007]:

1. Gerenciamento de Cursos (SiDeC): corresponde a um sistema que disponibiliza ferramentas para auxiliar a autoria de conteúdo pelos tutores;
2. Biblioteca Digital para Serviços Web (DLWS): equivale a um conjunto de recursos educacionais que provê a administração dos serviços e acesso remoto aos serviços do repositório;
3. Recursos Educacionais (IRLCOO⁶): IRLCOO é um tipo de recurso educacional sendo caracterizado por considerar características como multimídia, interatividade e feedback;
4. Avaliação dos Estudantes: este sistema é baseado na análise do perfil do estudante que é construído durante a interação com o sistema de gerenciamento de cursos.

O sistema é dividido em camadas, destacadas abaixo [Cruz et al., 2006].

- Camada de Aplicação: representa a camada de comunicação entre os usuários com as ferramentas disponibilizadas no sistema. Além disso, esta camada garante a integração entre os recursos educacionais com a ferramenta de gerenciamento de cursos e a biblioteca digital para serviços Web;
- Camada de Agentes e Componentes: nesta camada são disponibilizados agentes e componentes para serem utilizados pelas camadas de aplicação. Com isso, os

⁴Do Inglês: *Agents & Components Oriented Architecture*

⁵<http://www.ipn.mx/>

⁶Do Inglês: *Intelligent Reusable Learning Components Object Oriented*

componentes são utilizados pelo SiDeC e pela DLWS, enquanto os agentes são utilizados pelo sistema de avaliação, através da análise do perfil de estudantes construído ao longo das interações com o sistema;

- Banco de Dados: esta camada possui o armazenamento tanto os recursos educacionais quanto os registros de interação dos estudantes com o sistema;
- Servidor: infra-estrutura necessária para fazer com que as outras camadas funcionem corretamente, ou seja, esta camada provê acesso aos protocolos de comunicação bem como aos frameworks utilizados pelos agentes e componentes descritos na arquitetura.

As camadas descritas acima podem ser vistas na Figura 3.3.

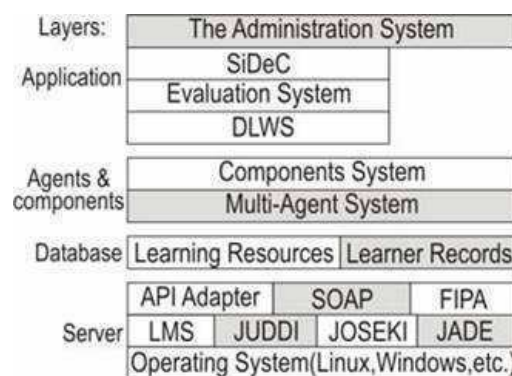


Figura 3.3: Arquitetura multicamadas proposta baseada em Agentes e Componentes.

As tecnologias baseadas na Web Semântica foram utilizadas apenas para a avaliação de estudantes. Ou seja, tais características são abordadas no lado servidor, pela ferramenta JOSEKI⁷.

A vantagem da proposta está no fato de utilizarem uma estrutura baseada em agentes para o sistema de avaliação. Entretanto, esta arquitetura não faz uso dos recursos provenientes da Web Semântica, buscando garantir maior adaptatividade e inteligência para o ambiente. Além disso, o ambiente não disponibiliza mecanismos inteligentes para prover mais facilidade na adaptação e a abordagem de agentes é utilizada exclusivamente para o sistema de avaliação. Destaca-se ainda que o ambiente não garante nenhum suporte à interoperabilidade.

⁷<http://www.joseki.org/>

3.4 Integrated Collaboration Framework

O trabalho abordado nesta seção equivale a integração de agentes de software, ontologias e serviços Web semânticos com o objetivo de colaboração na Web Semântica, chamado *Integrated Collaboration Framework*. O mesmo foi desenvolvido no instituto de pesquisa DERI⁸, da Universidade de Innsbruck⁹, na Austria.

O objetivo é garantir a colaboração de forma automatizada através da integração das tecnologias citadas anteriormente [Stollberg and Strang, 2005]. Esta arquitetura faz uso de agentes baseados em objetivos, possuindo serviços semânticos para colaboração, como mostrados na arquitetura na Figura 3.4.

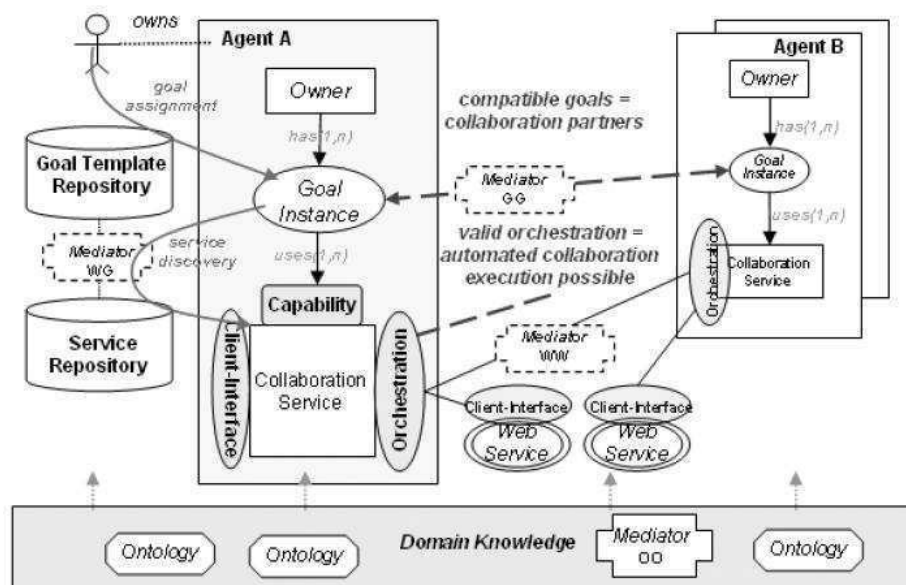


Figura 3.4: Arquitetura integrando Agentes, Serviços Web Semânticos e Ontologias.

Observa-se na Figura 3.4, que os agentes possuem arquitetura similar, mudando apenas os objetivos, e, a partir do momento que um agente pretende resolver determinado problema, o mesmo utiliza o serviço de colaboração para construir a possível execução de serviços. Desta forma, outros agentes são chamados para interagir com este agente de modo a resolver determinado problema.

Apesar da arquitetura abordar um modo eficiente de integrar agentes, ontologias e serviços, o mesmo não possui um foco educacional, consequentemente não provê

⁸Do Inglês: Digital Enterprise Research Institute.

⁹<http://www.uibk.ac.at/>

nenhuma facilidade para construir ambientes educacionais. Além disso, não está clara a garantia de interoperabilidade entre ambientes educacionais.

3.5 SQU

SQU¹⁰ *Web e-Learning Services* corresponde a um conjunto de serviços Web semânticos que são executados em um ambiente educacional. SQU equivale ao nome da universidade que desenvolveu os serviços Web Semânticos para serem utilizados em ambientes educacionais [Hatem et al., 2005b]. Estes serviços foram desenvolvidos no departamento de ciência da computação, do colégio de ciências, da Universidade SQU em parceria com o departamento de computação, da Universidade de Bradford¹¹, Reino Unido.

Os serviços semânticos utilizam os serviços providos pelo ambiente educacional WebCT[Hatem et al., 2005a] para oferta de cursos. Estes serviços estão integrados com o WebCT, além de outros serviços, como mostrado na Figura 3.5.

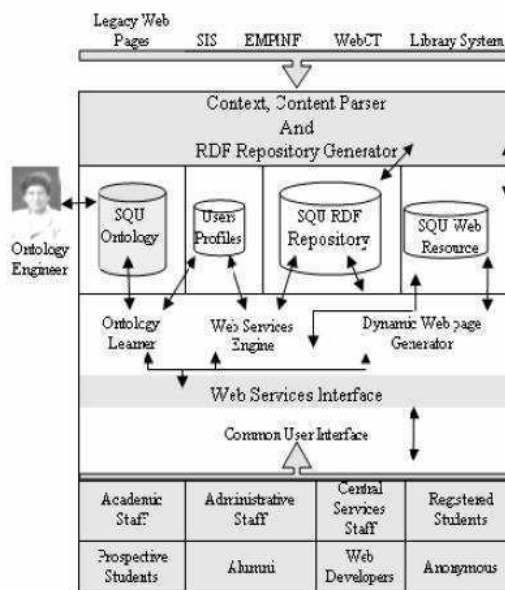


Figura 3.5: Arquitetura abordando os serviços Web da Universidade *Sultan Qaboos*.

Além disso, a arquitetura possui uma *engine* para a execução dos serviços, onde o

¹⁰Do Inglês: Sultan Qaboos University.

¹¹www.brad.ac.uk/

mesmo faz uso do perfil dos usuários dos ambientes. Entretanto, apesar do ambiente prover serviços Web semânticos, o mesmo não garante interoperabilidade entre ambientes educacionais, nem facilidades para a construção de tais ambientes levando em consideração aspectos de adaptação.

3.6 Semantic e-Learning Framework

Este trabalho propõe um framework para ambientes de *e-Learning* semânticos e inteligentes. O mesmo foi desenvolvido pelo Centro de Computação na Internet, da Universidade de Hull¹², Reino Unido.

O framework utiliza tecnologias da Web Semântica para a integração de conteúdos providos, modelo do estudante, entre outros [Huang et al., 2006]. Além da utilização de tecnologias provenientes da Web Semântica, o framework faz uso de agentes pessoais com o objetivo de prover informações para os usuários através da distribuição de serviços para aprendizagem personalizada. Além disso, a informação provida é baseada no perfil dos usuários. A Figura 3.6 ilustra a arquitetura do framework.

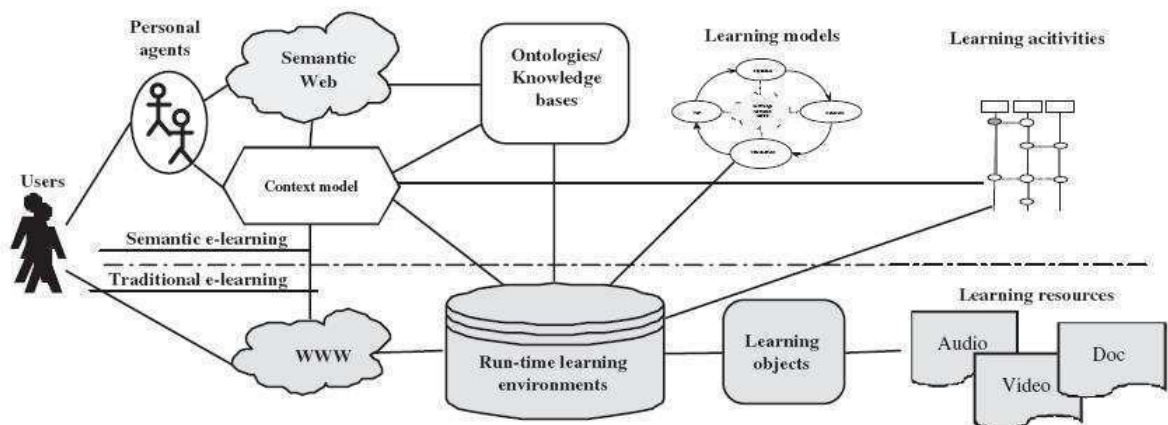


Figura 3.6: Arquitetura de um Framework para ambientes de *e-learning* semânticos e inteligentes.

Observa-se na arquitetura que duas perspectivas são mostradas, uma descrevendo uma abordagem de ambientes de *e-learning* tradicionais e outra semântica. Na perspectiva

¹²www.hull.ac.uk/

tradicional, há o ambiente educacional e os objetos de aprendizagem disponibilizados na Web. Já na perspectiva da Web semântica, há os agentes pessoais, além de bases de conhecimento/ontologias, modelos de aprendizagem e atividades de aprendizagem. Além disso, os agentes e ontologias fazem uso dos recursos tradicionais oferecidos pelos ambientes de *e-Learning*.

Apesar da arquitetura abordar utilização de tecnologias da Web Semântica nos ambientes de *e-Learning*, o mesmo não faz uso de serviços Web Semânticos, conseqüentemente, não garante a descoberta, composição e invocação de serviços, e também não garante interoperabilidade entre ambientes educacionais.

3.7 Multitutor

Multitutor é um ambiente para prover ambientes educacionais inteligentes através da utilização de tecnologias da Web Semântica e agentes de software. O ambiente foi desenvolvido pela Escola de Administração de Negócios¹³, da Universidade de Belgrado¹⁴, Sérvia.

A pesquisa desenvolvida pela universidade objetivou abordar a integração das tecnologias da Web Semântica com os sistemas inteligentes de aprendizagem e propõe os sistemas de gerenciamento de aprendizagem inteligentes. Como resultado, o sistema construído é o Multitutor, que permite que professores construam sistemas tutores para cada tipo do curso [Simic et al., 2004]. O ambiente possui os módulos tradicionais presentes em um sistema tutor inteligente, adicionando o módulo de autoria para professores especificarem características particulares de cursos. Além disso, o tutor possui um módulo especialista tendo regras como base de conhecimento. A Figura 3.7 aborda a arquitetura do Multitutor.

¹³Do Inglês: *FON - School of Business Administration*.

¹⁴www.bg.ac.yu/en_index.php

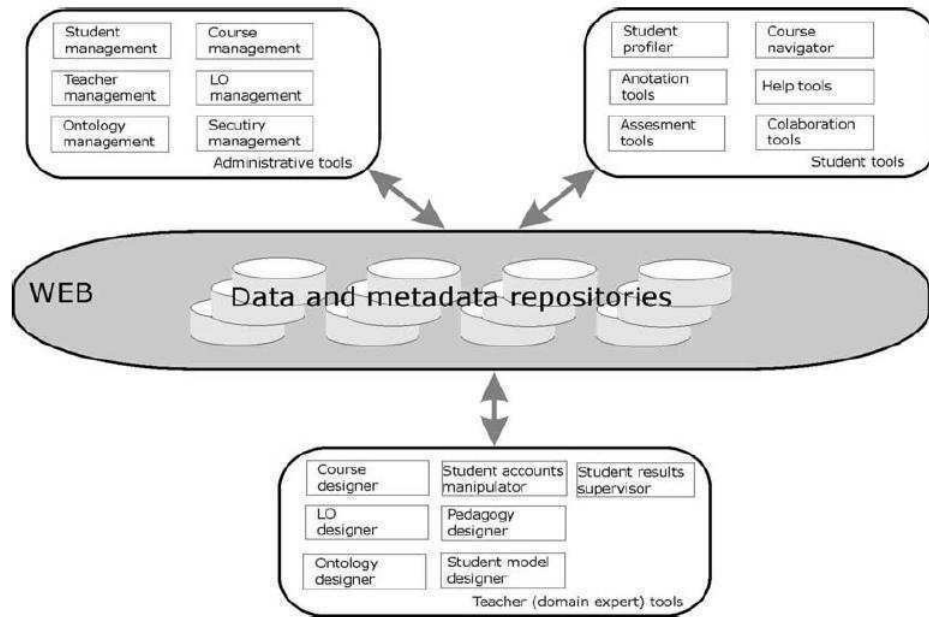


Figura 3.7: Arquitetura do Multitutor.

Apesar de prover o módulo de autoria para adicionar novos tutores baseados em cursos, o mesmo não é um framework, não permitindo que novas funcionalidades sejam adicionadas ao tutor, nem mesmo mudar o mecanismo de inferência utilizado pelo mesmo.

3.8 EBW

Este trabalho propõe a utilização de tecnologias provenientes da Web Semântica para a construção de sistemas de aprendizagem adaptativos. Ele foi desenvolvido pelo Departamento de Ciência da Computação, da Universidade Federal de Santa Catarina¹⁵, Brasil.

Este ambiente propõe um conjunto de ferramentas para garantir a interoperabilidade entre ambientes de gerenciamento de conteúdo [Neto, 2006a][Neto, 2006b], como mostrado na Figura 3.8.

Observa-se que o curso gerado é baseado em ontologias do domínio, pedagógico e do estudante e é configurado/criado através de uma ferramenta de edição de ontologias e de geração de cursos [Neto and Gauthier, 2006]. Entretanto, o sistema aborda uma

¹⁵<http://www.ufsc.br/>

perspectiva tradicional de utilização de ontologias e não oferece nenhum mecanismo de adaptação para o estudante. Ressalta-se ainda que a interoperabilidade entre os ambientes é através do curso que foi gerado. Ou seja, nenhuma interação entre os ambientes ocorre, não garantindo necessariamente o reúso de conhecimento. Além disso, todo o processo de construção e uso do sistema é manual, não garantindo efetivamente os requisitos propostos pela Web Semântica¹⁶.

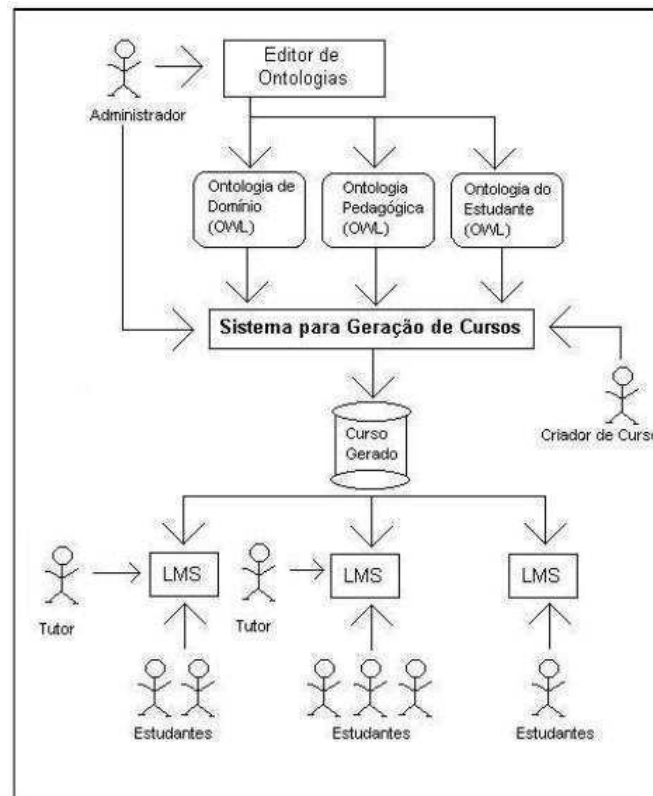


Figura 3.8: Arquitetura para garantir a interoperabilidade entre ambientes de gerenciamento de conteúdo.

¹⁶Para maiores detalhes, vide Seção 2.1

3.9 Semantic Integration Framework

Este trabalho é um framework para integração de serviços educacionais, chamado *Semantic Integration Framework*. Ele foi desenvolvido pelo Centro de Computação na Internet, da Universidade de Hull¹⁷, Reino Unido.

Este trabalho aborda como garantir interoperabilidade de serviços educacionais através da utilização de framework baseado em ontologias [Ishaya, 2005]. O framework é dividido em três camadas (vide Figura 3.9).

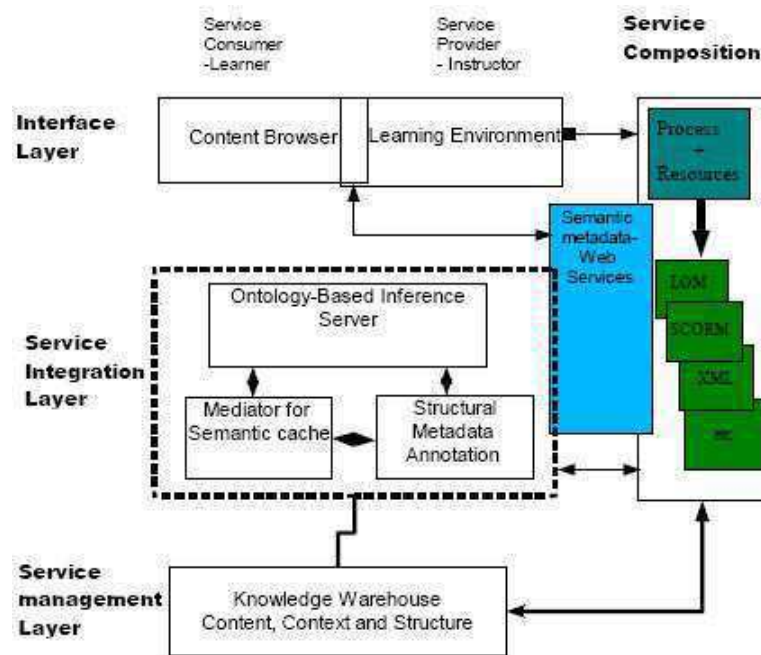


Figura 3.9: Arquitetura do Framework para Integração Semântica.

Ressalta-se que o mesmo provê uma estrutura para integração de plataformas de serviços que oferecem um suporte centrado no aprendiz. As camadas são divididas como segue:

1. Camada de Interface: esta camada é utilizada para garantir a comunicação/interação dos serviços com os estudantes;
2. Camada de Integração de Serviços: esta camada é responsável pela integração de serviços através da descrição em ontologias e da utilização de um servidor para

¹⁷www.hull.ac.uk/

fazer a verificação de consistência. Esta combinação possibilita a descoberta, composição e invocação de serviços;

3. Camada de Gerenciamento: com esta camada pode-se fazer uso de objetos de aprendizagem definidos através do uso de diferentes padrões de metadados. Com isso, os serviços baseados em uma ontologia são utilizados para acessar os diferentes objetos de aprendizagem.

Apesar da utilização de ontologias e serviços Web com o objetivo de garantir a interoperabilidade de objetos de aprendizagem e da possibilidade de descoberta, composição e invocação de serviços, o mesmo não disponibiliza serviços para facilitar a adaptação do sistema. Além disso, não é garantida a interoperabilidade entre ambientes educacionais e não são disponibilizados agentes para a automatização das atividades/processos do ambiente.

3.10 Framework baseado em Serviços

Este trabalho aborda um framework para a construção de ambientes de *e-Learning* através da utilização de tecnologias da Web Semântica, chamado de *Web-based Services*. O Framework foi desenvolvido pelo Departamento de Matemática da Faculdade de Ciências da Universidade Ain Shams¹⁸, Egito, em parceria com o Departamento de Ciência da Computação e Engenharia da Faculdade de engenharia da Universidade do Qatar¹⁹, Qatar.

A arquitetura do Framework é baseada em serviços e faz uso de modelos baseados em RDF e OWL [Ghaleb et al., 2006a]. Além disso, os serviços são definidos com funcionalidades voltadas para os estudantes e instrutores, como mostrados na Figura 3.10. Os serviços providos para o estudantes são com foco no conteúdo ou na avaliação [Ghaleb et al., 2006b].

¹⁸<http://net.shams.edu.eg/>

¹⁹<http://www.qu.edu.qa/>

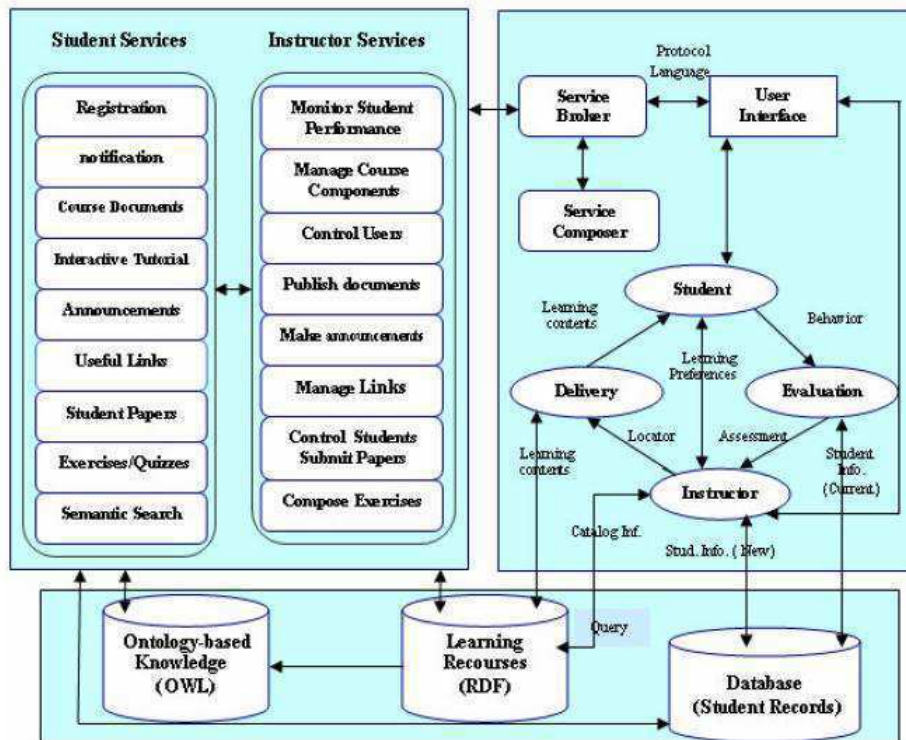


Figura 3.10: Arquitetura do Framework baseada em serviços e modelos em RDF e OWL.

Apesar da arquitetura prover diversos serviços para o estudante, além dos serviços para o instrutor que permitem a criação de cursos de *e-Learning*, a arquitetura não disponibilizar mecanismos inteligentes para adaptação nem garante interoperabilidade entre ambientes educacionais. Além disso, a arquitetura utiliza uma perspectiva tradicional de serviços Web, não permitindo a composição, descoberta e invocação dinâmica de serviços.

3.11 Tabelas Comparativas

Esta seção aborda um estudo comparativo entre as propostas citadas anteriormente. O estudo comparativo está separado na avaliação dos aspectos relacionados à Web semântica (vide Tabela 3.1), engenharia de software (vide Tabela 3.2) e inteligência artificial (vide Tabela 3.3).

A Tabela 3.1 descreve uma comparação dos trabalhos relacionados através da utilização das tecnologias provenientes da Web Semântica (como Ontologias e Serviços

Web Semânticos) e do provimento de uma estrutura que garanta o reúso de conhecimento. Desta forma, os principais trabalhos relacionados à proposta de tese em questão, quando leva-se em consideração a perspectiva da Web Semântica, são:

- Ambiente com MAS e SOA: este ambiente possui uma arquitetura multiagente e orientada a serviço, permitindo adicionar novos serviços para serem utilizados pelos agentes. Entretanto, o ambiente faz uso de serviços Web tradicionais²⁰, fazendo com que a descoberta, composição e invocação automatizada sejam complicadas;
- SQU: conjunto de serviços Web semânticos que são executados em um ambiente educacional (WebCT). Apesar da disponibilização de serviços, o reúso de conhecimento garantido através da interação entre ambientes educacionais fica comprometido, pois o mesmo não possui nenhum mecanismo de interoperabilidade entre os ambientes educacionais;
- *Integrated Collaborative Framework*: apesar deste framework utilizar ontologias e serviços Web semânticos, o mesmo não tem um foco diretamente voltado para educação, mas para colaboração de um modo geral. Desta forma, diversos aspectos com enfoque educacional não são devidamente contemplados, sendo muito custoso construir um ambiente educacional adaptativo e semântico;
- *Semantic Integration Framework*: framework para integração de serviços educacionais, buscando garantir interoperabilidade entre estes serviços. Entretanto, o ambiente é totalmente voltado para este tipo de integração, deixando de lado aspectos importantes como diferentes mecanismos de adaptação e até mesmo o reúso de conhecimento entre outras aplicações educacionais.

²⁰Serviços Web tradicionais são serviços que não possuem semântica embutida através de ontologias

Tabela 3.1: Tabela Comparativa dos Trabalhos Relacionados - Web Semântica.

Ambientes	Web Semântica		
	Ontologias	Reúso de Conhecimento	Serviços Web Semânticos
-			
<i>SWSOA</i>	x		x
<i>Ambiente com MAS e SOA</i>	x	x	x ¹
<i>ACOA</i>	x ¹		
<i>Integrated Collab. Fram.</i>	x	x	x
<i>SQU</i>	x	x	x
<i>Sem. e-Learning Fram.</i>	x	x	
<i>Multitutor</i>	x	x	
<i>EBW</i>	x	x	
<i>Sem. Integration Fram.</i>	x	x	x
<i>Fram. baseado em Serviços</i>	x		x

¹ Com restrições

A tabela 3.2 aborda características dos ambientes educacionais sob uma perspectiva da engenharia de software. São avaliadas características como o reúso de software, bem como o provimento de uma estrutura que garanta extensibilidade e interoperabilidade entre ambientes educacionais. Desta forma, os ambientes mais similares foram:

- Ambiente com MAS e SOA: este ambiente provê extensibilidade através da perspectiva de agentes e serviços providos no ambiente. Desta forma, quando se pretende adicionar um novo requisito, implementa-se um agente ou serviço e define-se como o mesmo será utilizado dentro do ambiente. Entretanto, ambientes educacionais gerados por ele não são capazes de se comunicar, pois não há a implementação de um mecanismo padrão de comunicação;
- EBW: este ambiente proporciona um sistema tutor para os ambientes poderem interagir. Entretanto, todo o reúso do ambiente, ocorre através do STI especificado anteriormente bem como a interoperabilidade é apenas entre os cursos que foram gerados e os ambientes fazem uso da mesma base. Desta forma, conceitualmente, não há interoperabilidade entre estes ambientes, mas sim, o compartilhamento entre a base de conhecimento.

Tabela 3.2: Tabela Comparativa dos Trabalhos Relacionados - Engenharia de Software.

Ambientes	Engenharia de Software		
	Extensível	Reúso de Software	Interoperabilidade entre Ambientes
-			
<i>SWSOA</i>		X	
<i>Ambiente com MAS e SOA</i>	X	X	
<i>ACOA</i>		X ¹	
<i>Integrated Collab. Fram.</i>		X	X
<i>SQU</i>	X	X	
<i>Sem. e-Learning Fram.</i>		X	
<i>Multitutor</i>			
<i>EBW</i>	X	X	X
<i>Sem. Integration Fram.</i>	X	X	
<i>Fram. baseado em Serviços</i>	X	X	

¹ Com restrições

Abaixo segue a Tabela 3.3 abordando ambientes educacionais sob uma perspectiva da Inteligência Artificial, sendo avaliados aspectos de automatização de processos/atividades, agentes de software e mecanismos inteligentes para adaptação. Desta forma, o trabalho mais similar ao proposto neste trabalho foi o Ambiente com MAS e SOA, onde o mesmo provê a utilização dos agentes de software para comunicação tanto com os papéis representados pelos estudantes quanto por professores/tutores. Entretanto, o mesmo não disponibiliza mecanismos inteligentes (nem mesmo implementados previamente em serviços) para prover adaptação aos ambientes educacionais. Além disso, a automatização é restrita, pois utilizam serviços tradicionais.

Tabela 3.3: Tabela Comparativa dos Trabalhos Relacionados - Inteligência Artificial.

Ambientes	Inteligência Artificial		
	Agentes de Software	Mecanismos Inteligentes	Automatização
-			
<i>SWSOA</i>			X
<i>Ambiente com MAS e SOA</i>	X		X ¹
<i>ACOA</i>	X ¹		
<i>Integrated Collab. Fram.</i>	X		X
<i>SQU</i>			X
<i>Sem. e-Learning Fram.</i>	X		
<i>Multitutor</i>	X		X
<i>EBW</i>		X	X
<i>Sem. Integration Fram.</i>			
<i>Fram. baseado em Serviços</i>			

¹ Com restrições

3.12 Conclusão

Este capítulo descreveu características sobre os trabalhos relacionados à esta tese. Foram identificadas as limitações de cada trabalho, além de ser feita uma análise comparativa através da identificação e do cumprimento de cada requisito previsto para a construção efetiva de ambientes educacionais adaptativos e semânticos. Ficou evidenciado que nenhum ambiente proporciona a integração de ontologias, agentes de software e serviços Web semânticos para a construção dos ambientes educacionais discutidos na tese.

Capítulo 4

Modelos Propostos

Quando tiver que andar, ande.

Quando tiver que sentar, sente.

Nunca hesite.

Sensei Francisco Araújo, “Chicão”

Neste capítulo descreve-se a proposta desta tese, enfocando os modelos conceitual (MASSAYO-RM¹), computacional (MASSAYO-CM²) e de infra-estrutura para interoperabilidade (MASSAYO-IM³). Além disso, apresenta-se um guia para auxiliar na utilização dos modelos. Este nome dado à solução proposta é chamado de MASSAYO. Este nome foi escolhido como uma homenagem ao antigo nome da cidade de Maceió, Terra Natal do propositor desta tese.

4.1 Modelo Conceitual - MASSAYO-RM

Como descrito no capítulo 2, a Web Semântica estende a Web clássica, provendo uma estrutura semântica do conteúdo presente nas páginas Web com o objetivo de dar suporte, no entendimento destes conteúdos, tanto para os agentes humanos quanto para os agentes artificiais. Desta forma, a Web semântica provê um ambiente onde agentes de

¹RM de *Referential Model*.

²CM de *Computational Model*.

³IM de *Interoperability Model*.

software autônomos podem navegar através de documentos Web e executar tarefas sofisticadas.

Sob uma perspectiva educacional, a comunidade de Inteligência Artificial e Educação (IAEd) está particularmente interessada nesta área com o intuito de evoluir os sistemas educacionais nesta direção, criando os chamados ambientes educacionais adaptativos e semânticos. De fato, o principal objetivo destes ambientes é fazer uso de recursos disponibilizados na Web através de padrões de tecnologia, buscando alcançar a AQQQ⁴: Aprendizagem a Qualquer momento, em Qualquer lugar e para Qualquer pessoa [Bittencourt et al., 2008d].

De acordo com [Anderson and Whitelock, 2004], a Web Semântica Educacional é baseada em três princípios fundamentais. O primeiro trata da capacidade de armazenamento e recuperação de informações efetivas, o segundo, da capacidade dos agentes de software autônomos para aumentar o aprendizado e informações recuperadas pelos humanos e o terceiro princípio é a capacidade da Internet para suporte, extensão e expansão das capacidades de comunicação dos humanos através de múltiplos formatos além dos limites de tempo e espaço. Além disso, outras características, descrevendo aspectos inerentes a qualquer ambiente sob a perspectiva da Web semântica, são abordadas como seguem:

- (i) Declaração explícita do conhecimento presente nas aplicações Web [Gómez-Perez and Corcho, 2002]: a Web semântica possui como base a declaração do conhecimento presente nas aplicações de forma explícita com o intuito de garantir um entendimento não só por humanos mas também por máquinas. Esta declaração deve fazer uso de linguagens que garantam consistência. Desta forma, ontologias têm sido utilizadas para garantir a declaração explícita do conhecimento de aplicações sob a perspectiva da Web semântica;
- (ii) Representar Informações para máquinas entenderem como garantir automação, integração e reuso entre aplicações [Boley et al., 2001]: a representação das informações tem como objetivo garantir que máquinas possam fazer uso das mesmas sem necessariamente precisar de interferência humana. Neste sentido,

⁴Do Inglês: AAAL: Anytime, Anywhere, Anybody Learning [Bittencourt et al., 2008d]

agentes de software podem ser vistos como entidades que buscam a automatização de atividades, além da comunicação entre diferentes aplicações com vistas a garantir a integração e o reúso de conhecimento entre aplicações;

- (iii) Infra-estrutura para os agentes rodarem buscando a resolução de problemas complexos [Hendler, 2001]: agentes devem ser capazes de buscar, recuperar e manipular informações disponíveis na Web, e que requerem a integração entre agentes para garantir a resolução de problemas complexos através de atividades cooperativas em uma sociedade de agentes;
- (iv) Disponibilização de serviços de forma que máquinas entendam e agentes inteligentes possam descobrir, compor e executar automaticamente [McIlraith et al., 2001]: apesar dos agentes garantirem a automatização de atividades, há a necessidade da disponibilização dos serviços que são utilizados tanto pelos agentes de software quanto pelos usuários das aplicações. Desta forma, serviços Web semânticos têm sido utilizados através da descrição semântica dos serviços de modo a garantir a descoberta, composição e invocação automatizada dos mesmos.

Diante dos requisitos supracitados, tanto da Web semântica quanto educacionais, percebe-se uma convergência natural de determinadas tecnologias, sendo elas:

1. Ontologias: informações na Web são geralmente representadas em uma linguagem para entendimento humano. Entretanto, para que os computadores entendam o significado presente nestes conteúdos, há a necessidade de representar as informações de uma forma que possa ser interpretada tanto sintaticamente quanto semanticamente. Estas representações ajudam no processo de análise, extração e integração de informações na Web, tornando mais fácil a criação de bases de conhecimento sólidas para que serviços inteligentes possam utilizá-los para dar suporte às necessidades dos usuários. Atualmente, pesquisas em ontologias têm sido consideradas como um dos pontos chaves para prover informação que sejam entendidas por computadores [Mizoguchi, 2004]. De acordo com [Devedzic, 2006], o uso de ontologias e o advento dos serviços inteligentes (para desenvolvimento de conteúdos Web, filtragem Web, máquinas de busca inteligentes, processamento

automatizado, entre outras) estão transformando a Web de informações na Web do conhecimento.

2. Padrões Educacionais: o uso de padrões é fundamental aos recursos educacionais para atividades de descrição, desenvolvimento, transporte, acesso, anotação, combinação e qualificação. Tantos os padrões Web semânticos (e.g., RDF, OWL, SWRL) como os recursos educacionais (e.g., IEEE, IMS) são usados no desenvolvimento de ambientes educacionais adaptativos e semânticos [Gasevic et al., 2004][Dietze et al., 2007a];
3. Serviços Web Semânticos: SWS provêm uma quantidade de diferentes atividades para transformar uma coleção de informações estáticas em informações de forma distribuída através do uso de tecnologias provenientes da Web semântica. Com isso, SWS faz com que os conteúdos sejam processáveis e interpretáveis por máquinas;
4. Agentes Inteligentes: eles são entidades de software autônomas nas quais provêm diversos tipos de suportes (e.g. resolução de problemas, buscas inteligentes, modelagem do usuário) através da interação com outros jogadores de acordo com os seus papéis.
5. Ferramentas Interativas: tais ferramentas são módulos que são utilizados para garantir a interação entre os agentes inteligentes com os diversos agentes humanos. As ferramentas interativas podem ser tanto síncronas quanto assíncronas, além de poderem ser utilizadas com diferentes propósitos.

Desta forma, a Figura 4.1 descreve o modelo conceitual para ambientes educacionais adaptativos e semânticos.

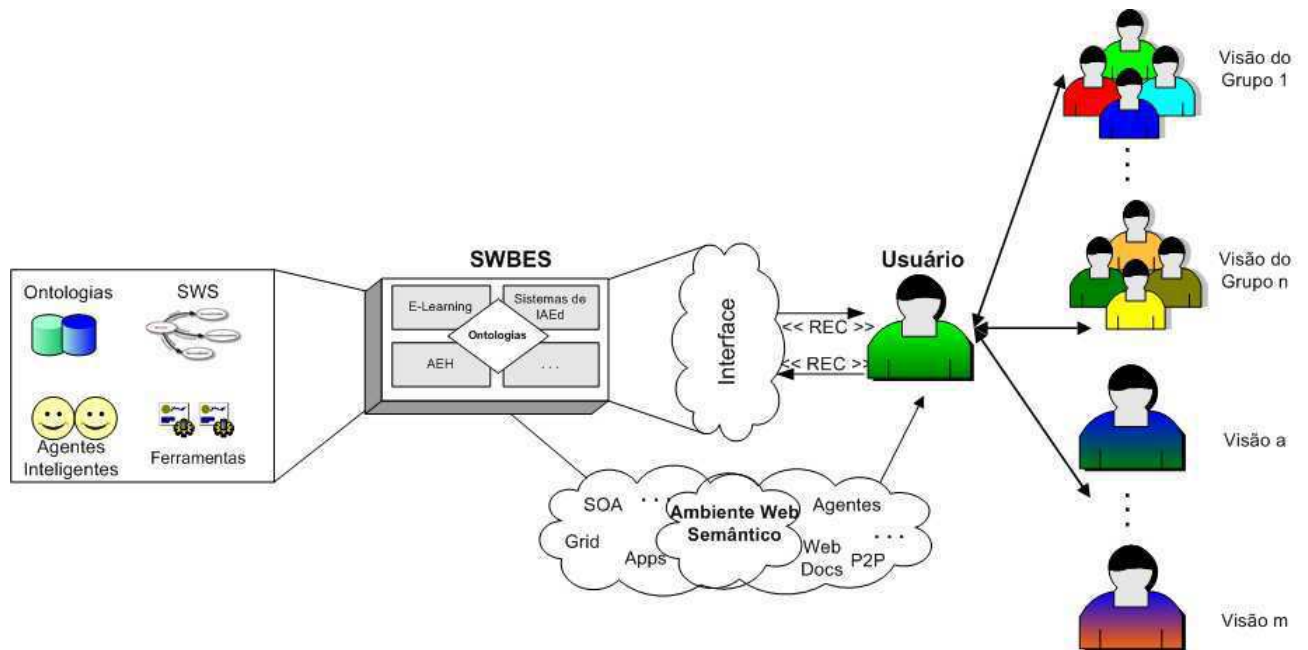


Figura 4.1: Modelo Conceitual para Ambientes Educacionais Adaptativos e Semânticos.

Cada elemento presente no modelo é discutido nas subseções que seguem.

4.1.1 Papéis

Diversas atividades educacionais estão envolvidas com os AEAS, como ensino, aprendizagem, cooperação, colaboração, autoria, entre outras. Estas atividades são distribuídas de acordo com papéis de cada jogador. Estes papéis são discutidos abaixo.

- Professor: responsável por monitorar as interações dos estudantes (resolução de problemas, avaliação, etc), configurar estratégias de aprendizagem, definir tipos de recursos educacionais a serem utilizados, entre outras [Kosba et al., 2007];
- Aprendiz: o principal interesse do aprendiz é interagir com o sistema baseado em conteúdos educacionais adaptáveis e personalizados com o intuito de melhorar seus conhecimentos e completar seus objetivos de aprendizagem. Estes jogadores podem ser agentes artificiais que interagem com o sistema na função de aprendizes. Exemplos destes tipos de agentes são os agentes companheiros [Uresti and du Boulay, 2004], co-aprendizes, agentes inteligentes pessoais [Abel et al., 2006], entre outros;

- Autor: responsável por estruturar o conteúdo educacional. As atividades de autoria podem ser divididas em i) conteúdos educacionais, ii) processos instrucionais e iii) adaptação e personalização [Aroyo and Dicheva, 2004];
- Tutor: aquele que os estudantes em ambientes de educação a distância através do desenvolvimento de atividades. Ele tem um papel muito importante nos ambientes de EaD, pois está diretamente comprometido com a formação do aluno, proporcionando-lhe a participação com vistas a gerar conhecimento de forma coletiva [Maggio, 2001];
- Grupo: diversas aplicações levam em consideração o processo de aprendizagem através de grupos. Eles estão interessados em aprendizagem colaborativa, interagindo com outros estudantes com o objetivo de alcançar objetivos pessoais, compartilhamento cognitivo, meta-cognição, motivação (tanto cognitiva quanto afetiva) em interagir com outros aprendizes, dentre outros;
- Desenvolvedor: eles são responsáveis pelo desenvolvimento e adição de novas funcionalidades nos ambientes educacionais. Além disso, esforço similar tem sido feito para construir ontologias com o objetivo de especificar tanto metodologias quanto scripts para distribuir a aplicação [Bittencourt et al., 2007b];
- Auditor: responsável por agrupar dados sobre a efetividade do sistema de um modo geral, ferramentas específicas e também sobre outros jogadores (levando em consideração seus papéis) [Santos et al., 2003]. Além disso, o agente responsável pela avaliação da efetividade de um componente específico pode ser tanto um agente artificial (agente de suporte ou controlador) como um agente humano (tutor, estudante, pais, etc);
- Administrador: responsável pelo gerenciamento do sistema como um todo. Além do mais, os mesmos cobrem todo o ciclo de vida da administração dos ambientes educacionais desde o registro (de estudantes, autores, agentes, serviços, etc) e seguindo até o fim do processo de ensino aprendizagem. Estes jogadores podem também ser agentes artificiais, como um agente controlador, no qual permite que um outro agente entre na sociedade.

4.1.2 Recursos Educacionais

Recursos educacionais são representados por objetos de aprendizagem em um domínio específico, podendo ser descritos como exemplo, problema, contra-exemplo, unidades de atividades, entre outros. Frisa-se ainda que toda interação estudante-sistema-estudante ocorre através dos recursos educacionais.

O objetivo destes objetos de aprendizagem é disponibilizá-los de tal forma que facilitem o reuso de conteúdo educacional, podendo os mesmos serem combinados com outros objetos de aprendizagem [Devedzic, 2006]. Estes objetos utilizam padrões para descrição dos mesmos de modo a garantir um comum entendimento, permitindo assim a reutilização dos conteúdos educacionais.

4.1.3 Interface

A Interface representa a comunicação entre os jogadores com um papel específico e o sistema educacional. Além disso, o contexto e o tipo de recurso educacional são importantes aspectos que necessitam ser considerados quando se define a tecnologia a ser utilizada pelo ambiente. Outro aspecto importante é que a interface pode ser adaptada de várias formas, levando-se em consideração o contexto de estudo, as características dos usuários, o tipo de dispositivo presente, dentre outras. A ilustra as diversas formas de adaptar a interface.

Outro ponto importante é que a Interface dá suporte para diferentes papéis, provendo interfaces especializadas de acordo com o papel, como por exemplo, ferramentas de autoria, ferramentas para dispositivos móveis, ambientes de desenvolvimento etc.

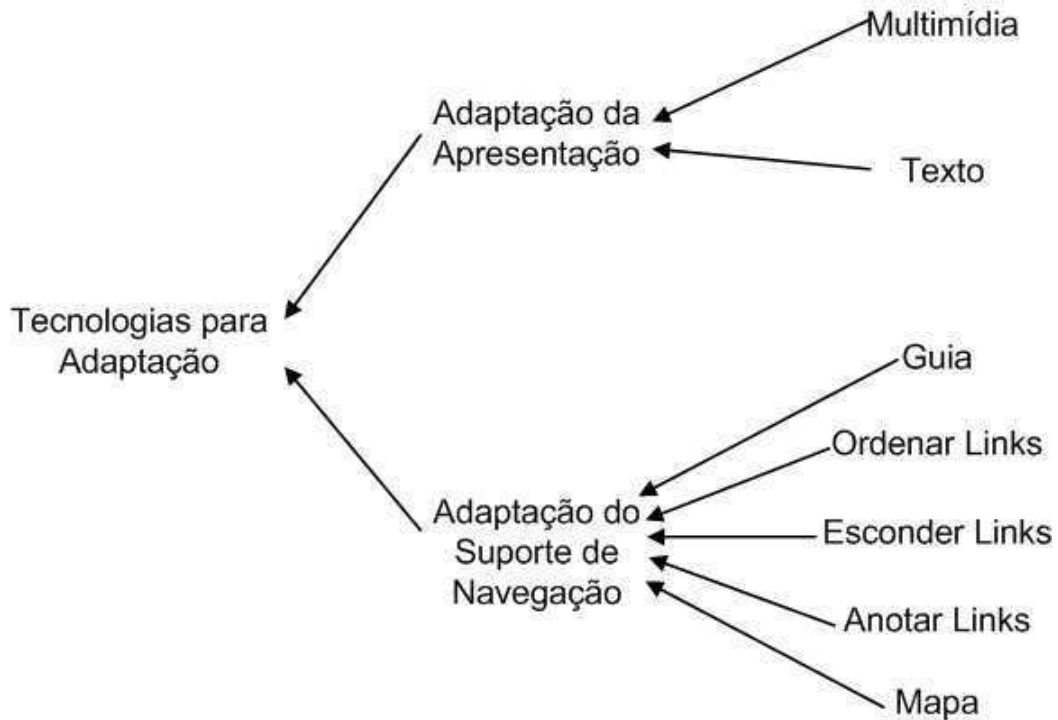


Figura 4.2: Tipos de Adaptação de Interface.

Fonte extraída e traduzida de [Brusilovsky, 1996].

4.1.4 Ambientes Educacionais baseados na Web Semântica

Os ambientes dão suporte aos usuários para guiar e ajudar os mesmos a alcançar os seus objetivos de aprendizagem. Assim, o ambiente representa uma nova geração de sistemas educacionais baseados na Web, onde há o objetivo de prover diversas melhorias na qualidade dos serviços através do uso de tecnologias provenientes da Web semântica. De fato, os recursos da Web semântica podem ser usados em diversos ambientes educacionais, como por exemplo, sistemas tutores inteligentes [Self, 1995], sistemas de gerenciamento de conteúdo [Abazi-Bexheti, 2008], ambientes virtuais de aprendizagem [Ivanova and Smrikarov, 2004], sistemas educacionais adaptativos [Benyon and Murray, 1993], sistemas de aprendizagem pervasiva [Loidl, 2006], sistemas educacionais para TV Digital [Lopez-Nores et al., 2006], entre outros⁵. Os componentes presentes em um ambiente deste tipo são descritos na subseções que seguem.

⁵Diversos exemplos sobre ambientes educacionais baseados na Web semântica são encontrados na série do Workshop SWEL (Semantic Web Technologies for e-Learning)

Ontologias

As Ontologias buscam fazer uma descrição dos dados presentes em determinado universo e permitir a interação entre os dados disponibilizados através de diferentes formatos [Shadbolt et al., 2006]. Elas têm sido endereçadas pela comunidade como um importante requisito para garantir interoperabilidade entre ambientes educacionais. No contexto de Web semântica e educação, as ontologias têm sido aplicadas para resolver problemas complexos, como representação de conhecimento em sistemas inteligentes, compartilhamento de conhecimento e reúso entre aplicações, anotação e busca de objetos de aprendizagem, personalização de conteúdos educacionais, além de muitos outros desafios [Bittencourt et al., 2008d].

Destaca-se ainda que notáveis resultados no desenvolvimento e uso das ontologias para aumentar a experiência da aprendizagem baseada na Web (incluindo aspectos tanto tecnológicos quanto teóricos) são enfatizados pela comunidade e reportado em [Dicheva and Dichev, 2007][Mizoguchi et al., 2007]. De um lado, os aspectos tecnológicos representam o uso das ontologias para permitir a navegação, compartilhamento e reúso de conteúdos educacionais que são possíveis localizar em diferentes repositórios, auxiliando na interoperabilidade e integração de diferentes aplicações educacionais. Do outro lado, os aspectos teóricos representam o uso de ontologias para dar suporte ao fluxo de conhecimento desde a teoria até a prática através de metodologias que permitem a criação, descoberta, compartilhamento e apresentação eficiente de conhecimento para serem utilizadas durante a autoria de conteúdos educacionais e de aprendizagem. É fundamental para qualquer aplicação educacional inteligente na área de Web semântica possuir o conhecimento compartilhável e bem definido para o desenvolvimento de uma aprendizagem efetiva. Os tipos de ontologias que podem ser usadas nestes sistemas são:

1. Ontologias Educacionais: responsáveis pela especificação e validação do conhecimento referente à uma característica específica do sistema educacional, como ontologias do modelo do estudante, modelo pedagógico, modelo de adaptação, tarefas, entre outras;
2. Ontologias para Interação: estas possuem o conhecimento sobre os componentes de um sistema educacional, como ontologias para especificar agentes de software,

serviços Web semânticos, ferramentas educacionais, entre outros;

3. Ontologias de Aplicação: responsáveis pela descrição dos ambientes educacionais baseados na Web semântica com o objetivo de garantir a comunicação e a publicação dos sistemas educacionais na Web. Como resultado, os ambientes educacionais podem interoperar entre si (através de seus agentes de software).

Ferramentas

Um ambiente educacional pode oferecer uma quantidade considerável de ferramentas, e diferentes formas de comunicação e sofisticação ao longo das interações. Observa-se que um sistema educacional deve prover ferramentas de acordo com o tipo de usuário, o papel do usuário e o dispositivo (celulares, PDAs, navegadores, etc) que está sendo utilizado. Os tipos de ferramentas que podem ser desenvolvidos ou reusados pelo ambiente educacional são:

1. Ferramentas Educacionais: estas possuem um papel importante no processo de aprendizagem de cada estudante. Ou seja, quanto mais sofisticação e interação os ambientes educacionais proverem nas ferramentas (tanto para comunicação síncrona quanto assíncrona), maior será a capacidade do ambiente de treinar e guiar os estudantes em sessões de aprendizagem. Algumas ferramentas educacionais tradicionais que podem ser citadas são fórum, chat, lista de discussão, entre outras. As ferramentas da Web 2.0 [Rollett et al., 2007] também são exemplos que podem ser descritos como uma ferramenta educacional. Desta forma, a integração destas ferramentas com tecnologias da Web semântica proporciona novas possibilidades para os sistemas através da abordagem da Web semântica social [Dicheva and Dimitrova, 2008]. Além disso, diversas ferramentas podem ser providas de acordo com o papel do usuário, por exemplo, ferramentas de autoria, ferramentas de auxílio ao tutor, ferramentas de auditoria, entre outras;
2. Ferramentas de Suporte: algumas são necessárias para fazer com que a Web semântica pública (como descrita na Seção 2.1 do Capítulo 2) torne-se realidade. Estas ferramentas têm o intuito de prover características de semi-automatização para agentes inteligentes, como por exemplo, ferramentas de

mapeamento/alinhamento de ontologias, anotação de ontologias, Web semântica social, visualização de ontologias, engenharia de ontologias, entre outras;

3. Ferramentas Administrativas: estas são importantes para configurar o ambiente com relação aos papéis, usuários, autenticação, entre outras atividades.

Serviços Web Semânticos

Serviços Web Semânticos (SWS) são utilizados para prover aos usuários, agentes e ferramentas, acesso (orientado a serviços) às características de acordo com o interesse de cada uma delas. A pesquisa conduzida pela comunidade da Web semântica com relação aos serviços Web semânticos afirma que os mesmos são uma excelente opção para a integração de atividades através da criação, descoberta, composição e invocação automática de SWS [Brambilla et al., 2007][Al-Masri and Mahmoud, 2007]. Os tipos de SWS que podem ser disponibilizados são:

1. SWS Educacionais: eles disponibilizam características para garantir que as funcionalidades dos sistemas educacionais sejam cumpridas. Em outras palavras, os SWS implementam funcionalidades no contexto educacional, como a integração de conteúdos, descoberta de recursos educacionais, avaliação e comparação entre aplicações educacionais, personalização de conteúdos educacionais e interfaces, colaboração, recomendação, resolução de tarefas, diagnóstico, entre outras;
2. SWS de Suporte: diversas atividades nestes ambientes não necessariamente são atividades com propósito educacional, logo, podem ser disponibilizados serviços Web semânticos presentes nestes ambientes educacionais, como serviços de conversão, busca, mineração, entre outros. Por esta razão, SWS de suporte são importantes tanto para os ambientes educacionais quanto outros tipos de ambientes.

Agentes Inteligentes

Os agentes possuem um papel fundamental nos sistemas educacionais baseados na Web semântica. Eles são responsáveis por garantir a automatização das atividades, evolução dinâmica dos requisitos, dentre outras atividades. Os tipos de agentes presentes são descritos abaixo:

1. Agentes Tutores (ou Pedagógicos): estes agentes buscam garantir a interação entre os sistemas (através de seus requisitos educacionais) e os usuários (principalmente estudantes) de acordo com seus papéis. Como exemplos, eles podem ser usados para o tutoramento, descoberta de novo conhecimento sobre um determinado usuário, transporte de mensagens com conteúdos educacionais entre aplicações, personalização do conteúdo de diferentes usuários, entre outros;
2. Agentes de Suporte: eles dão suporte aos outros agentes para assegurar que os mesmos alcancem os objetivos em cada atividade educacional. Exemplos destes agentes são agentes de suporte para mineração na Web;
3. Agentes Controladores: estes agentes são importantes para administrar as atividades, como por exemplo, verificação de desempenho dos agentes e serviços Web semânticos presentes no ambiente, identificação de agentes não confiáveis, identificação de mudanças de comportamentos e/ou características dos agentes, entre outras.

4.1.5 Ambiente Web Semântico

Este ambiente Web semântico está disponibilizado para ser utilizado tanto pelos usuários quanto pelos ambientes educacionais. Por um lado, através deste ambiente, os usuários podem navegar, descobrir, selecionar e invocar os recursos disponibilizados na Web. Por outro lado, os ambientes educacionais podem fazer uso de uma infra-estrutura tecnológica para automaticamente executar tarefas e recuperar recursos disponíveis na Web.

4.1.6 Análise do Modelo Conceitual

Esta subseção tem por objetivo fazer uma avaliação da abrangência do modelo conceitual baseado nos trabalhos relacionados. Sendo assim, abaixo são descritas de forma geral as características de cada trabalho relacionado e então aborda-se como o modelo conceitual trata cada aspecto presente nos trabalhos.

- O trabalho SWSOA equivale a uma arquitetura de software orientada a serviços

Web semânticos. Todas as ferramentas identificadas são ferramentas educacionais (para projeto instrucional e objetos de aprendizagem). Além disso, todos os serviços educacionais semânticos são educacionais com o objetivo de abstrair o acesso a metadados educacionais, permitindo a integração dos mesmos. Por fim, o papel presente equivale ao de estudante. Além disso, o framework baseado em serviços e SQU (vide Figura 4.3) possuem as mesmas tecnologias que o SWSOA, mudando apenas os papéis, ou seja, o SWSOA possui apenas o papel do estudante.

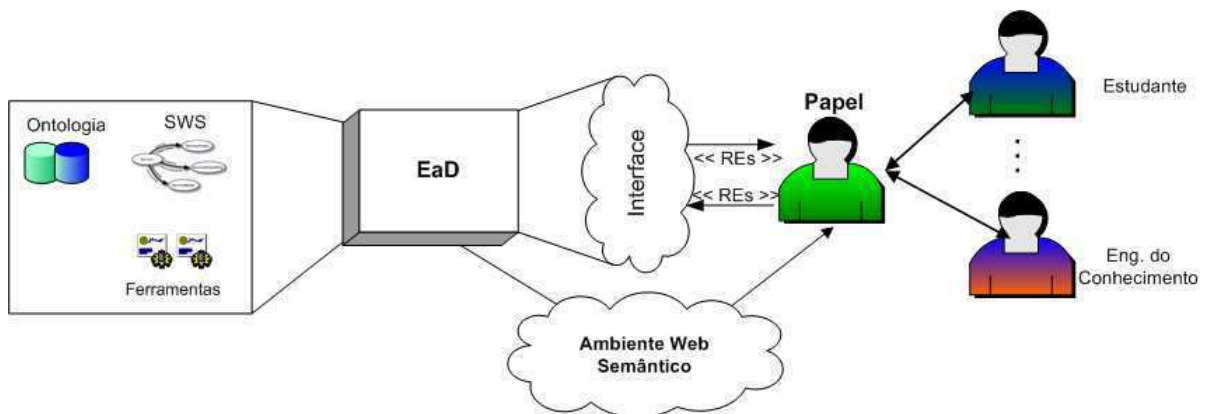


Figura 4.3: Modelo Conceitual para o ambiente SQU.

- O ambiente com MAS e SOA (vide Figura 4.4) aborda as tecnologias de agentes e serviços Web tradicionais. Os papéis presentes são os de estudante e tutor. Este ambiente possui diversas ferramentas que são tanto educacionais quanto de suporte. Já os agentes de software são agentes de suporte e agentes tutores.

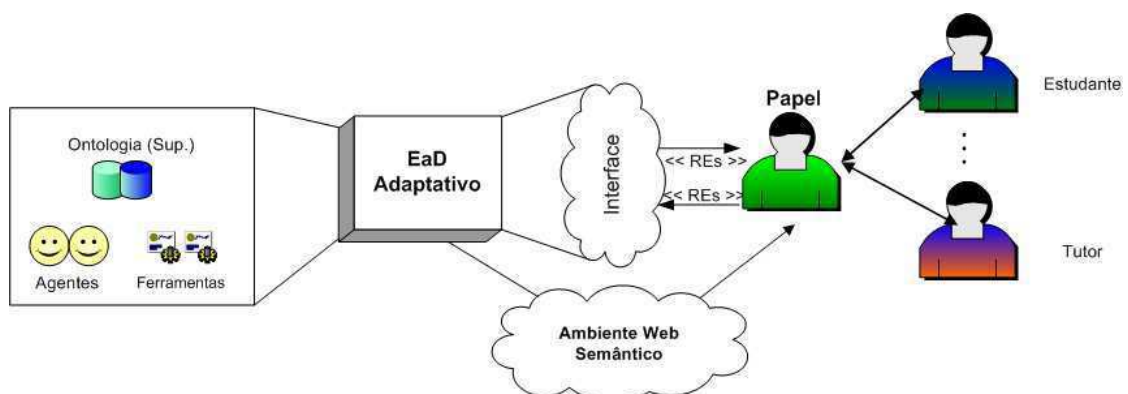


Figura 4.4: Modelo Conceitual para o ambiente com MAS e SOA.

- Os trabalhos *Integrated Collaborative Framework* (vide Figura 4.5), *Semantic e-Learning Framework* possuem as mesmas tecnologias, as quais são baseadas em agentes de software e ontologias. Todos os agentes presentes são agentes pedagógicos.

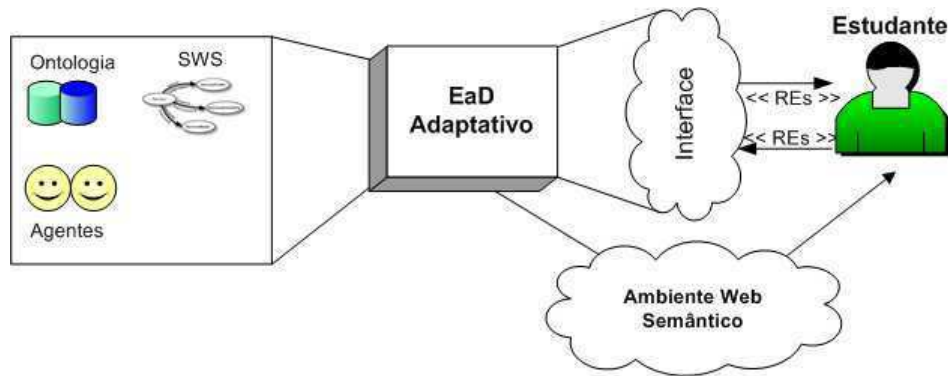


Figura 4.5: Modelo Conceitual para o ambiente *Integrated Collaborative Framework*.

- O ambiente EBW (vide Figura 4.6) utiliza ontologias e ferramentas de autoria para especificar o ambiente baseado em ontologias. Este ambiente é voltado para os papéis de estudante, autor e professor e utiliza agentes pedagógicos para garantir a interoperabilidade entre ambientes educacionais. Os ambientes Multitutor e ACOA possuem as mesmas tecnologias diferenciando os papéis presentes e o tipo de ambiente educacional. Ou seja, o Multitutor é um sistema tutor inteligente com o papel do estudante e o ACOA é um ambiente educacional adaptativo e semântico com os papéis de estudante e tutor. Além disso, o trabalho ACOA diferencia do EBW e do Multitutor pela quantidade de ferramentas presentes, sendo categorizadas como ferramentas educacionais e administrativa.

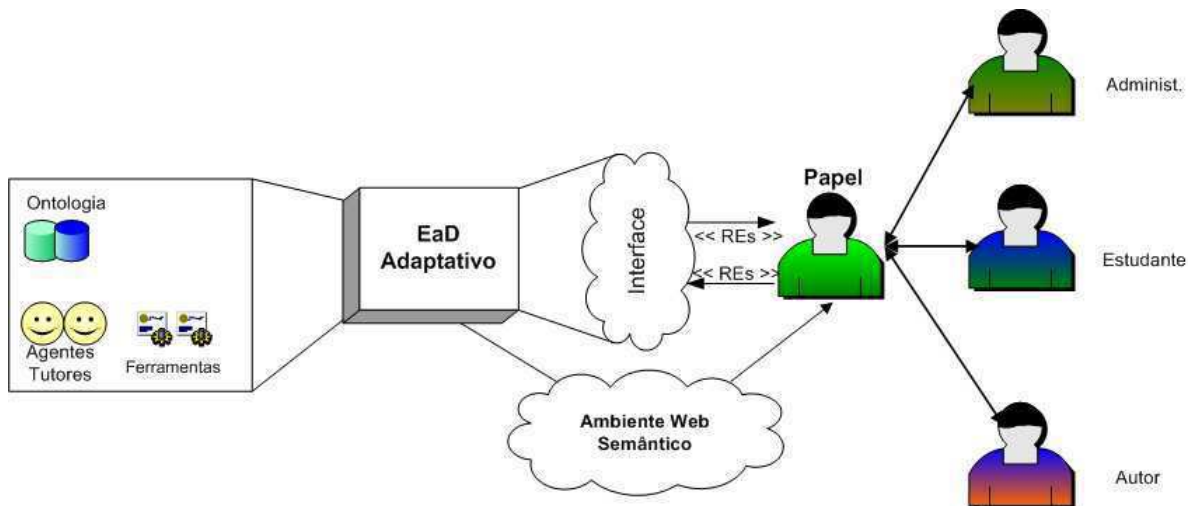


Figura 4.6: Modelo Conceitual para o ambiente EBW.

Com a análise dos trabalhos relacionados baseada no modelo conceitual do MASSAYO, pode-se perceber que o modelo possui a necessária abrangência para descrever ambientes educacionais.

4.2 Modelo Computacional - MASSAYO-CM

Esta seção descreve um modelo computacional (plataforma) para construção de ambientes educacionais adaptativos e semânticos baseado no modelo conceitual anteriormente apresentado. A arquitetura é dividida em camadas, tal como apresentado na Figura 4.7.

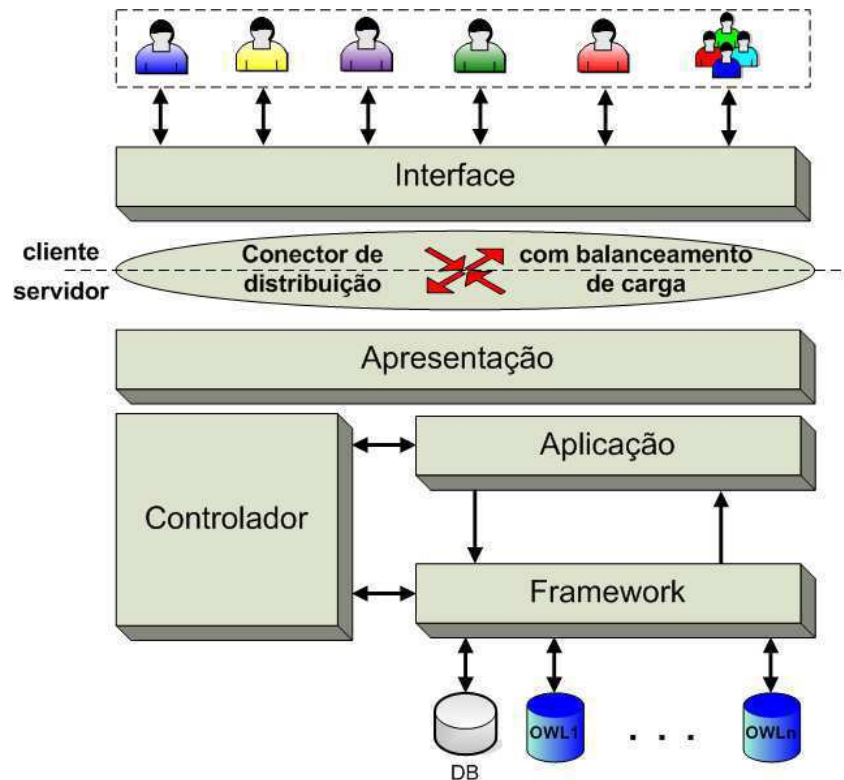


Figura 4.7: Arquitetura Multicamadas.

A arquitetura foi desenvolvida através de uma abordagem multicamadas [Bittencourt et al., 2008c], fazendo uso do padrão de projeto MVC⁶[Kazman and Klein, 2009]. MVC é um padrão de arquitetura de software que diminui a complexidade das aplicações e separa as responsabilidades em Modelo, Visão e Controle, sendo bastante utilizado em aplicações Web. As camadas são descritas como segue:

- Framework: esta camada é mantida pelos desenvolvedores, podendo estes adicionar novas funcionalidades, bem como alterar ou remover uma funcionalidade. Este framework faz uso de tecnologias provenientes da Web semântica (ontologias e serviços Web semânticos) e de uma abordagem de sistemas multi-agentes para prover ambientes educacionais adaptativos e semânticos;
- Aplicação: esta camada representa as aplicações com as quais o usuário pode interagir, sendo definidas ferramentas para interação tanto síncrona quanto assíncrona, bem como para os diversos tipos de usuários. As ferramentas que estão

⁶Modelo-Visão-Controle ou *Model-View-Control*.

disponíveis nesta camada são: fórum, *chat*, *wiki*, enquete, biblioteca digital, calendário, atividades, *blogs*, *login*, RSS, notícias, notificação, comunidades, gerenciador de atividades e visualizador de recursos;

- Controlador: representa a configuração necessária para garantir a utilização das regras de negócio do modelo, sendo refletidas na camada de apresentação. Esta camada faz uso de ontologias para definir os agentes presentes nos ambientes, esquemas de adaptação, estilos de ensino, entre outras. As ontologias presentes no modelo, bem como a relação entre elas, são mostradas na Figura 4.8;
- Apresentação: responsável por gerar o esquema de apresentação a ser mostrado na interface do usuário. Como exemplo, cita-se a utilização dos *portlets*, encapsulando as ferramentas disponibilizadas para interação com o usuário;
- Interface: responsável por mostrar os recursos para o usuário poder interagir com as aplicações/framework;
- Conector de distribuição com balanceamento de carga: responsável por intermediar a comunicação entre clientes e servidores através da implementação da distribuição e do balanceamento de carga disponível entre os servidores.

4.2.1 Camada do Controlador

O controlador representa a configuração necessária para garantir a utilização das regras de negócio do modelo computacional, sendo refletidas na camada de apresentação. Desta forma, esta camada faz uso de ontologias para definir os agentes presentes nos ambientes, os esquemas de adaptação, os estilos de ensino, entre outras. Além disso, diversas ontologias, com diferentes propósitos, foram criadas tornando complexo o entendimento de cada uma delas. Outro importante aspecto é que foram utilizadas regras para garantir consistência e mapear o conhecimento entre as ontologias, proporcionando assim uma redução no tempo de desenvolvimento/especificação das instâncias das ontologias.

A Figura 4.8 apresenta as ontologias e regras, além da relação entre as mesmas, podendo-se identificar quatro grupos de ontologias, sendo eles.

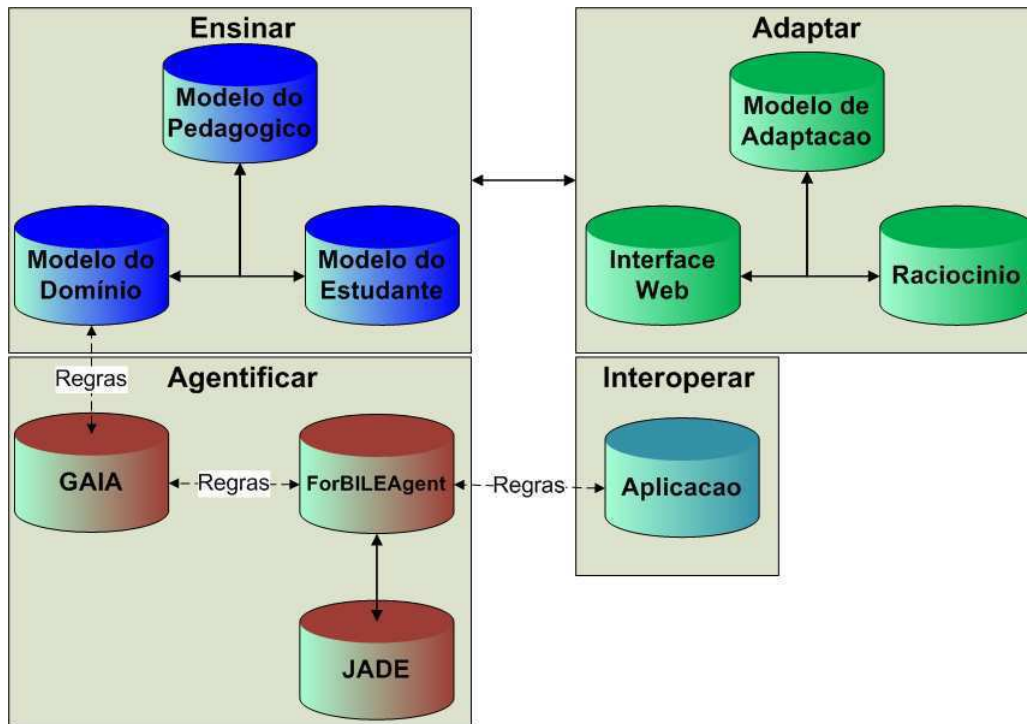


Figura 4.8: Ontologias e suas relações e regras de mapeamentos.

1. Ensinar: estas ontologias são ontologias educacionais com vistas a descrever características inerentes do objeto de estudo;
2. Adaptar: ontologias de adaptação são utilizadas para especificar os conceitos e relações necessárias para garantir a adaptação do ambiente;
3. Agentificar: estas ontologias são classificadas como ontologias de interação, tendo o objetivo de especificar os agentes presentes no ambiente educacional, permitindo assim interações dinâmicas entre os papéis e o sistema;
4. Interoperar: objetivando permitir a interoperabilidade entre ambientes educacionais, há a necessidade de utilizar uma ontologia de aplicação de modo a descrever os recursos disponibilizados no ambiente. Isto possibilita a interoperabilidade entre ambientes educacionais.

Este modelo possui ontologias educacionais (ensino e adaptação), ontologias de interação (agentificação) e ontologia de aplicação. É importante ressaltar também que as

ontologias deste trabalho foram criadas utilizando-se OWL-DL⁷ e as regras através de SWRL (vide Apêndice 2). Desta forma, descreve-se a seguir características inerentes às ontologias descritas na Figura 4.8.

Modelo do Domínio

O modelo do domínio, tal como visto em um STI, descreve informações sobre a questão *o que ensinar*. As características do modelo do domínio são providas através de uma visão multidimensional do conhecimento, baseado no modelo Mathema revisitado (vide Seção A.2.4). A ontologia do modelo do domínio é mostrada na Figura 4.9 e na Figura 4.10.

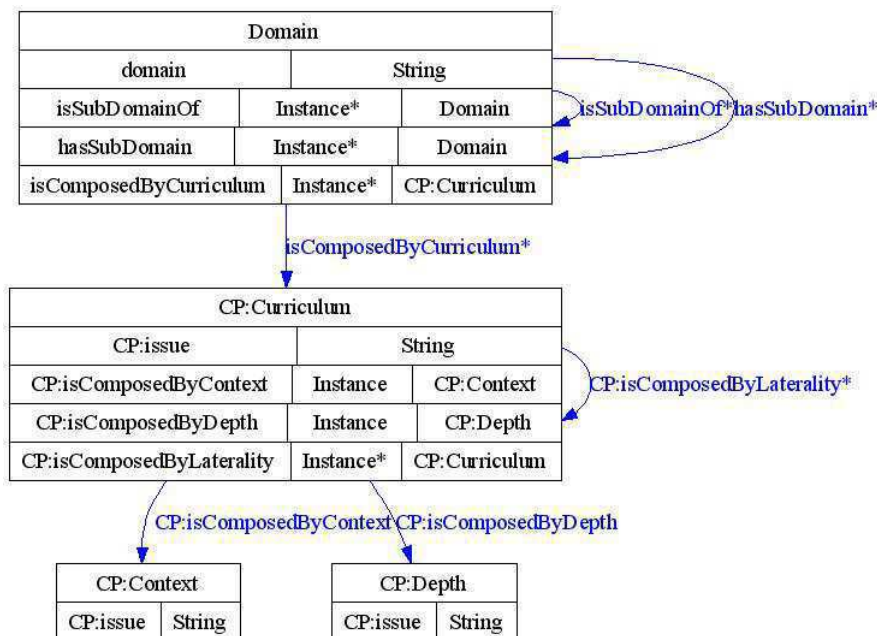


Figura 4.9: Ontologia do modelo do domínio - Visão multidimensional.

Algumas classes presentes nas ontologias são:

- *Domain*: detalha sobre os domínios nos quais pretende-se ensinar;
- *Resource:Resource*: representa o reuso de recursos educacionais dentro dos ambientes. Estes recursos educacionais podem ser exemplos, contra-exemplos,

⁷OWL-DL foi utilizada por ser a linguagem padrão de ontologias para a Web Semântica, padronizada pelo W3C.

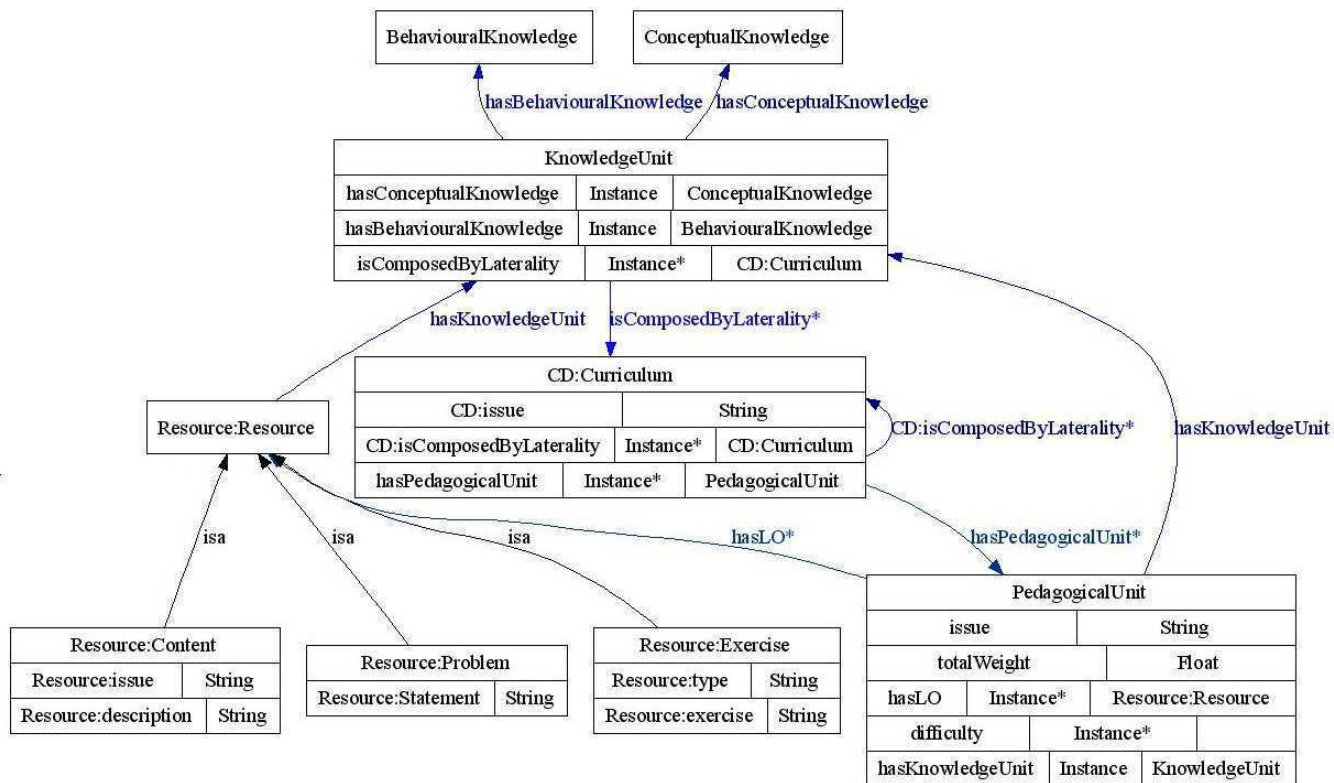


Figura 4.10: Modelagem da ontologia do domínio - Visão dos currículos.

problemas, conteúdo, avaliação e termos. A ontologia pode ser estendida de acordo com a necessidade dos ambientes educacionais;

- *LO:Problem*: define informações sobre problemas. Os problemas disponibilizados são de múltipla escolha, relacionar colunas, preencher lacunas e Verdadeiro ou Falso;
- *CD:Curriculum*: foi definida segundo o modelo Mathema, que faz o mapeamento das partições de um domínio em uma estrutura curricular.

Modelo do Estudante

A construção do modelo do estudante foi feita através da avaliação dos trabalhos [Chen and Mizoguchi, 2004][Chepegin et al., 2004]. Estes trabalhos foram utilizados por ser uma das primeiras e principais referências sobre modelos do estudante da literatura, mostrando-se ainda bastante consistentes.

O modelo do estudante é responsável pelo conhecimento sobre a questão *para quem*

ensinar, ou seja, esse módulo conterà informações sobre o estudante ou um grupo de estudantes que está sendo ensinado. Dentre as informações relevantes ao modelo do estudante, têm-se:

- **Informações Estáticas:** informações que não mudam de acordo com a interação estudante-sistema-estudante. Alguns exemplos de informações estáticas são, nome, telefone, endereço, *login*, senha, características de personalidade, estilo de aprendizagem, entre outros. Algumas destas informações foram baseadas no IMS LIP [IMS, 2007];

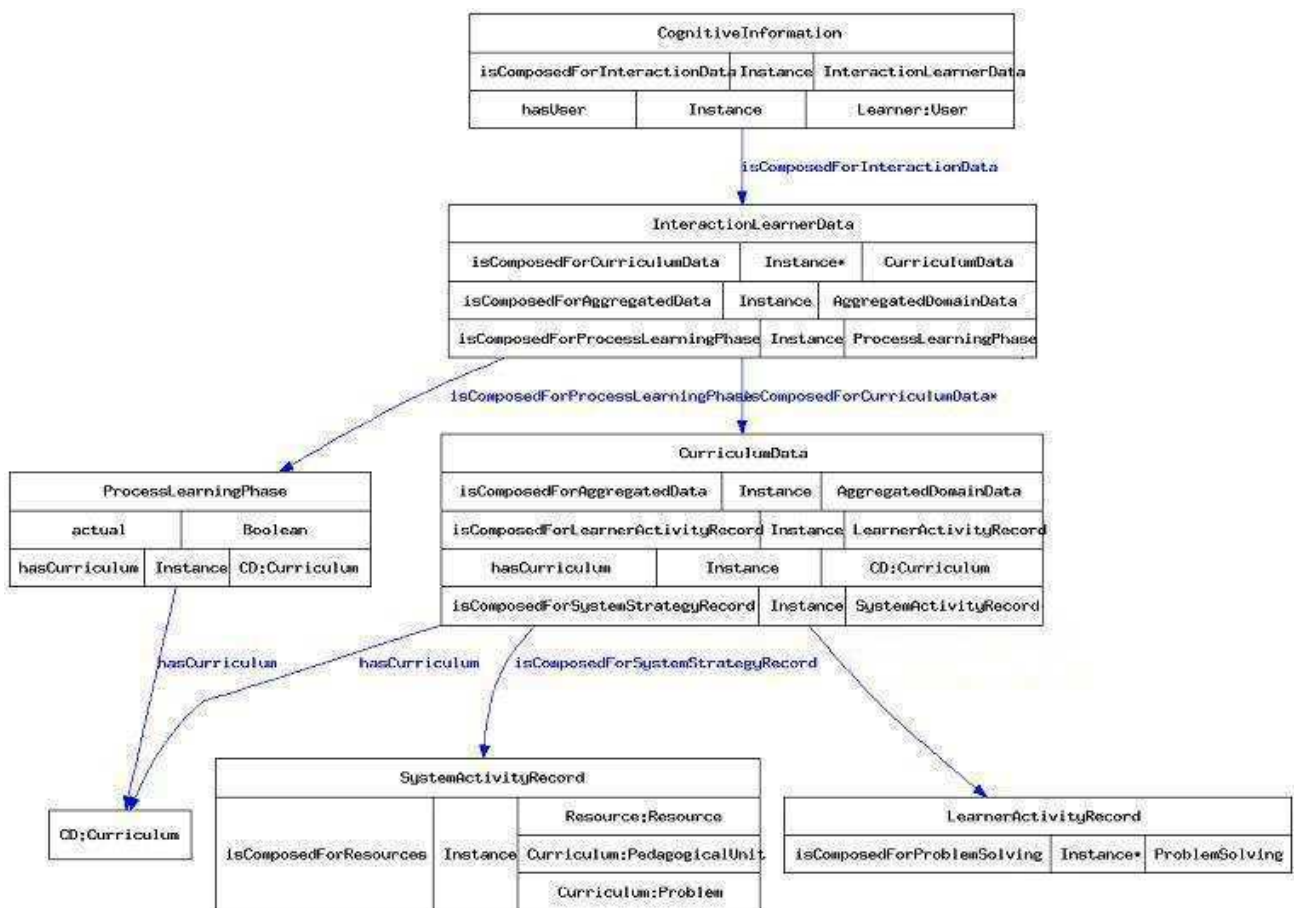


Figura 4.11: Modelagem da interação estudante-sistema na Ontologia do Estudante.

- **Informações Dinâmicas:** informações que mudam durante a interação estudante-sistema-estudante. Normalmente estas informações estão relacionadas com informações do domínio, como os objetivos de aprendizagem de cada estudante, nível de conhecimento do estudante, conteúdos já estudados, respostas

dos problemas, dentre outras. A Figura 4.11 mostra as características de interação entre o estudante e o sistema. Todas as informações sobre as ações do sistema e do estudante são registradas através das classes *SystemStrategyRecord* e *LearnerActivityRecord*. Além disso, as informações contidas na classe *LearnerActivityRecord* são baseadas na estrutura curricular (definida na ontologia do modelo do domínio).

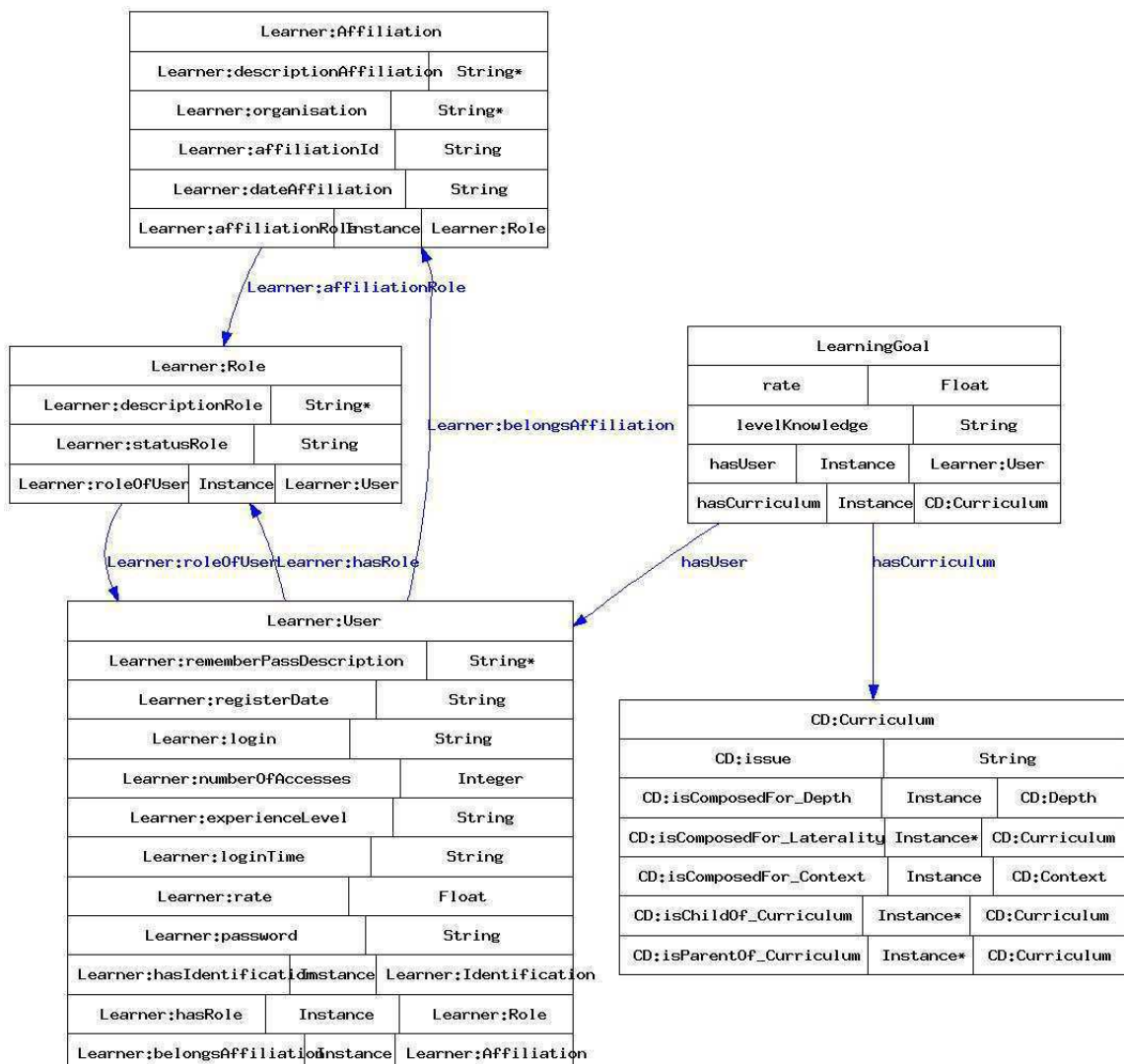


Figura 4.12: Objetivos de aprendizagem da Ontologia do estudante.

Outros aspectos importantes da ontologia do estudante são os objetivos de aprendizagem que devem ser definidos de acordo com o currículo especificado no modelo de conhecimento sobre determinado domínio, ou seja, a partir do momento que o

modelo do domínio é definido, o estudante está habilitado a descrever quais são os seus objetivos de aprendizagem de acordo com cada currículo, como mostrado na Figura 4.12.

Modelo Pedagógico

O modelo pedagógico (vide Figura 4.13) é responsável pelo conhecimento sobre a questão *como ensinar*, ou seja, como uma interação pode ser conduzida. Normalmente, o modelo realiza esta interação através de um planejamento instrucional, considerando-se aspectos cognitivos sobre os estudantes. O modelo pedagógico foi construído baseado nos trabalhos [Kumar et al., 2004][du Boulay and Luckin, 2001], pois são dois dos principais trabalhos sobre estratégias pedagógicas da literatura.

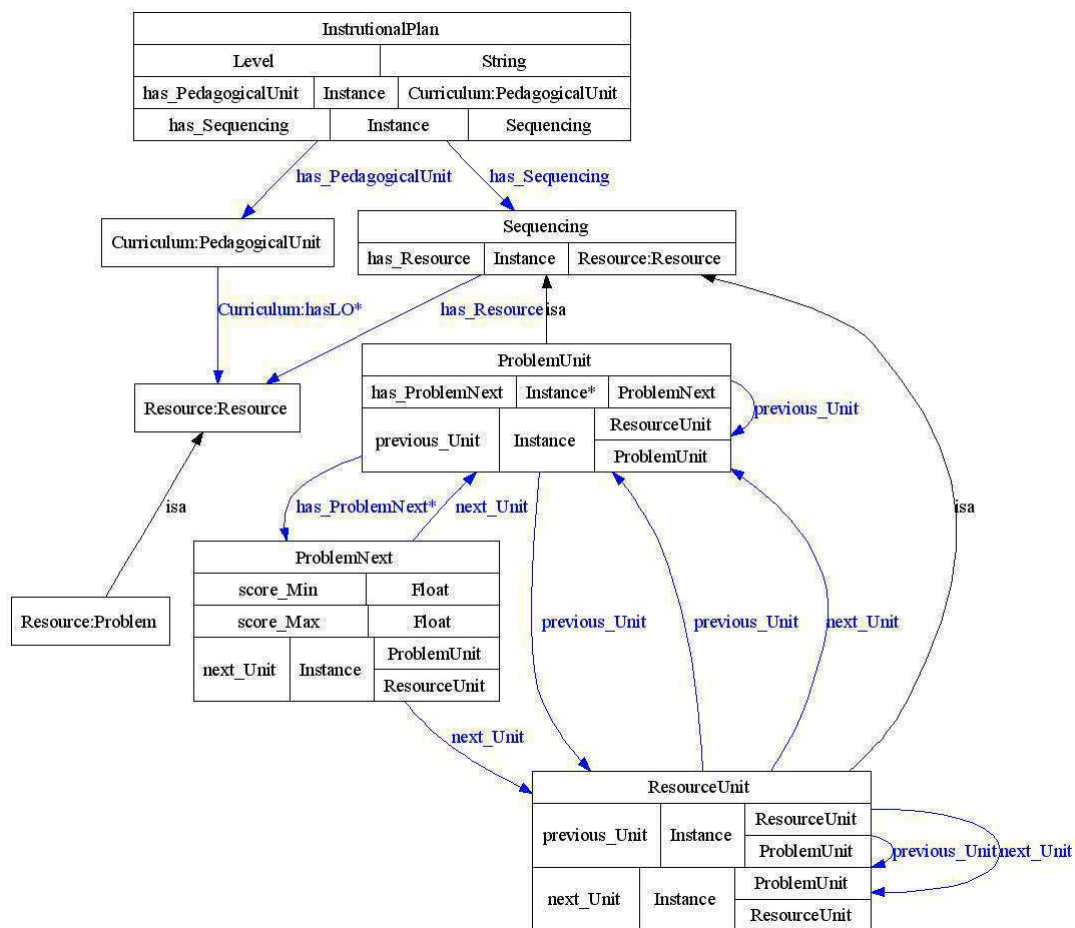


Figura 4.13: Parte da Modelagem da Ontologia do Modelo Pedagógico.

Ressalta-se ainda que o plano instrucional faz uso de estratégias pedagógicas que correspondem à forma que o estudante ou um grupo de estudantes são conduzidos ao

longo do processo de ensino-aprendizagem.

Algumas classes desta ontologia são descritas abaixo:

- *InstructionalPlan*: representa o plano instrucional que é definido para um determinado currículo. Este plano define uma sequência de visualização de recursos educacionais pelo estudante, podendo ser previamente especificada ou construída dinamicamente;
- *Sequencing*: define exatamente o sequenciamento presente no plano instrucional, onde o mesmo possui recursos educacionais que apontam para o próximo recurso a ser mostrado e o recurso anterior. Desta forma, o plano instrucional só necessita apontar para o primeiro recurso. Além disso, o sequenciamento define duas unidades, sendo elas:
 - *ResourceUnit*: representa qualquer recurso educacional que não necessita de uma avaliação para a tomada de decisão sobre o próximo recurso. Ou seja, recursos como conteúdo, exemplo, conceitos, etc, são descritos segundo esta classe;
 - *ProblemUnit*: uma unidade define um recurso educacional *Resource:Problem* para adicionar ao sequenciamento do plano instrucional. Entretanto, esta unidade foi criada por haver a necessidade de definir qual será o próximo recurso de acordo com o grau de acerto do estudante.

Interface Web

Esta ontologia tem o objetivo de descrever os aspectos referentes ao projeto de interface bem como o projeto navegacional. Com isso, os recursos presentes em uma interface Web⁸ foram descritos em uma ontologia, onde os principais recursos são mostrados na Figura 4.14.

A ontologia possui informações do estudante, relacionando-as com os tipos de páginas que o mesmo tem acesso. Os recursos que podem ser especificados na interface

⁸Não foi objetivado na tese a descrição de interfaces para outras plataformas, como dispositivos móveis ou TV Digital.

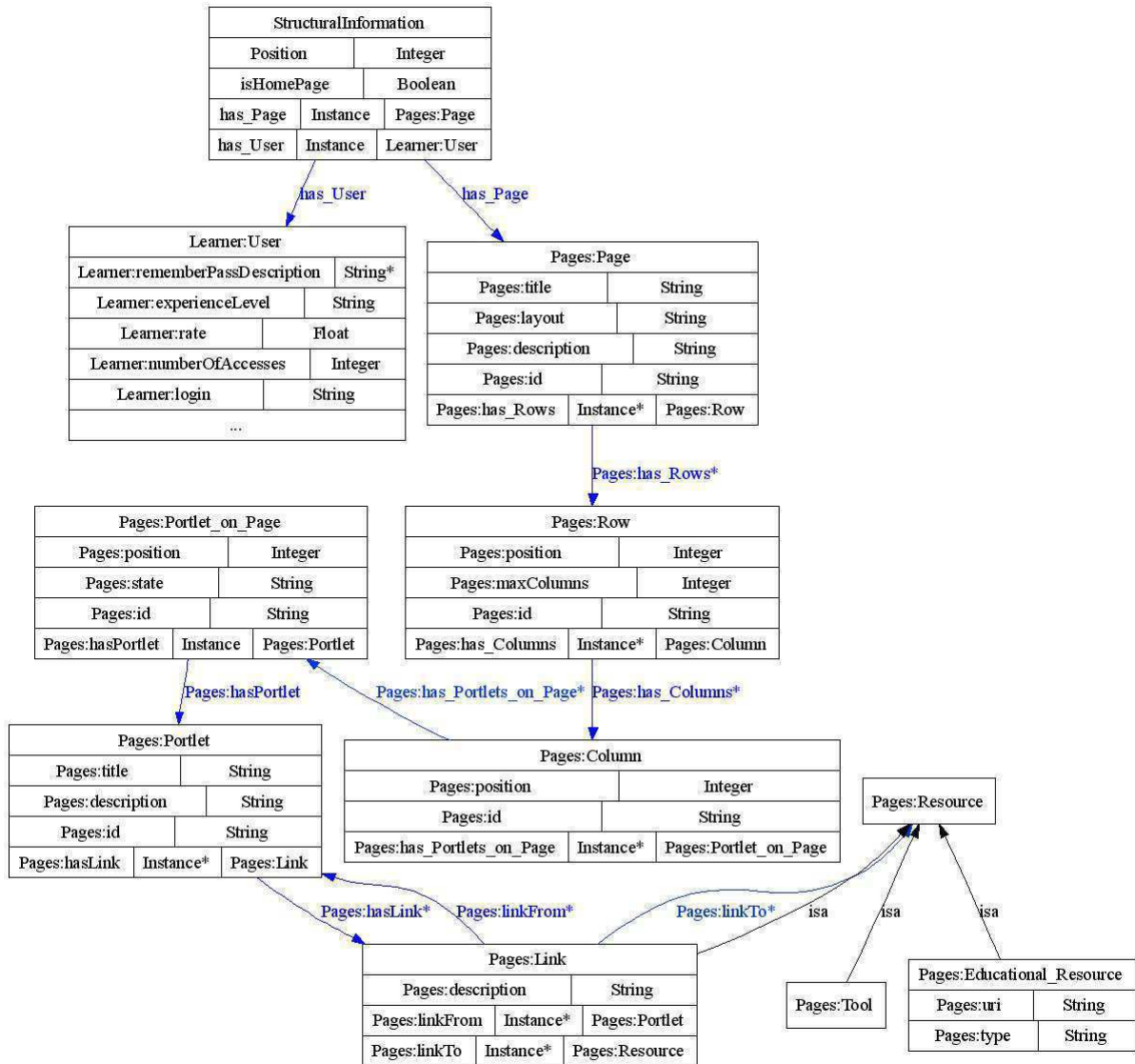


Figura 4.14: Parte da Modelagem da Ontologia de Páginas Web.

de uma aplicação Web (e.g. linhas, colunas, links, *portlets*, ferramentas) são descritos na ontologia.

Modelo de Adaptação e Raciocínio

O modelo de adaptação e raciocínio descreve o que deve ser adaptado e como, ou seja, aqui são descritas técnicas e métodos de adaptação hipermídia e as abordagens tecnológicas para a realização da adaptação. Estas ontologias são também integradas com a ontologia de interface Web.

Os métodos de adaptação em [Brusilovsky, 1996], descritos na Figura 4.2 e

representados nesta ontologia, como mostrada na Figura 4.15.

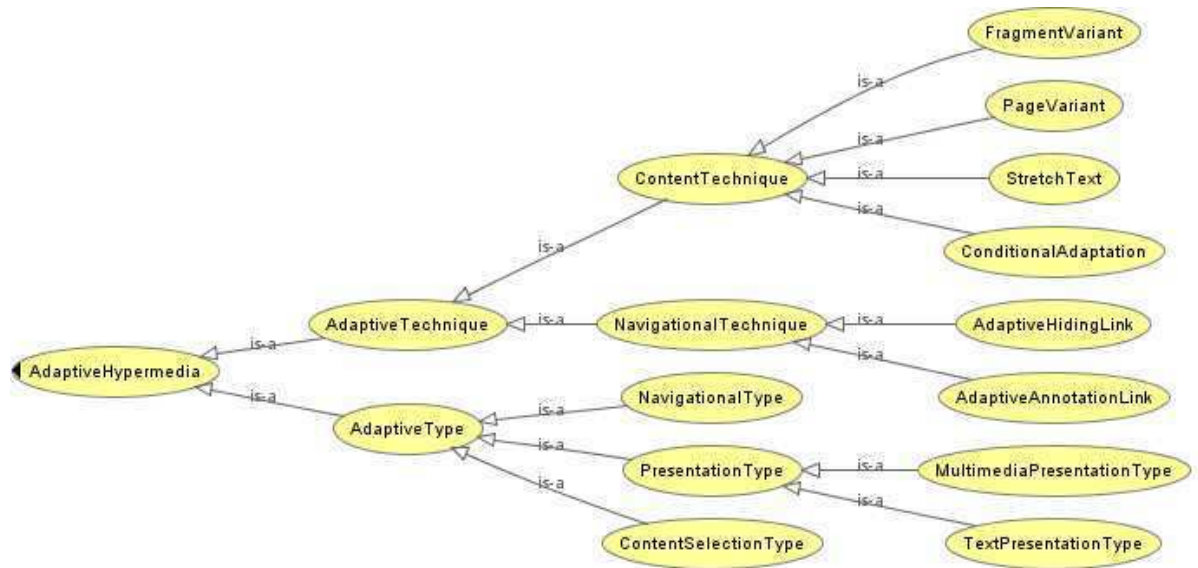


Figura 4.15: Taxonomia de Métodos de Adaptação Hipermédia representada em OWL-Lite.

As técnicas de adaptação foram divididas em *ContentTechnique* e *NavigationalTechnique*. No primeiro grupo, objetiva-se adaptar o conteúdo que será exibido (e.g. assunto, exercícios, exemplos). No segundo grupo é adaptada a forma de navegação. Entre as técnicas que adaptam o conteúdo, estão apresentadas nas Figuras 4.16 e 4.17.

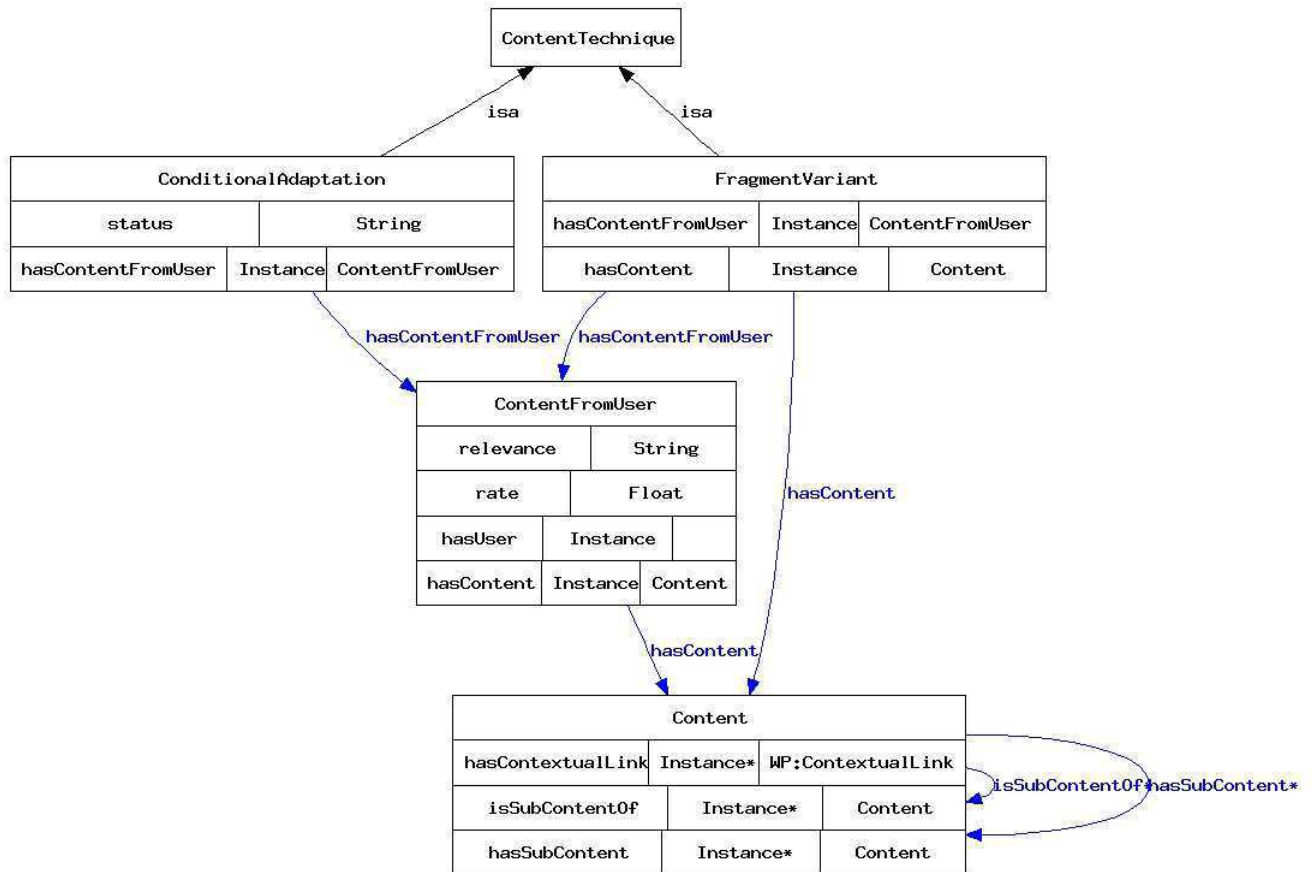


Figura 4.16: Técnicas de Adaptação descritas na ontologia de adaptação (a).

Os conceitos presentes na ontologia são descritos como seguem:

- **Conteúdo**: representa o que será adaptado. Um conteúdo corresponde a um objeto de aprendizagem modelado no domínio e pode ser formado por outros conteúdos (*subcontent*). Além disso, possui uma determinada relevância para cada usuário (*ContentFromUser*);
- **ConditionalAdaptation**: determina, através da propriedade *status*, se um conteúdo será ou não apresentado ao estudante;
- **FragmentVariant**: decide entre dois ou mais conteúdos correspondentes;
- **PageVariant**: de forma semelhante à técnica anterior, esta decide dentre um conjunto de páginas com conteúdos similares, qual será exibida. Isso é feito de acordo com a relevância de cada página (*PageFromUser*);

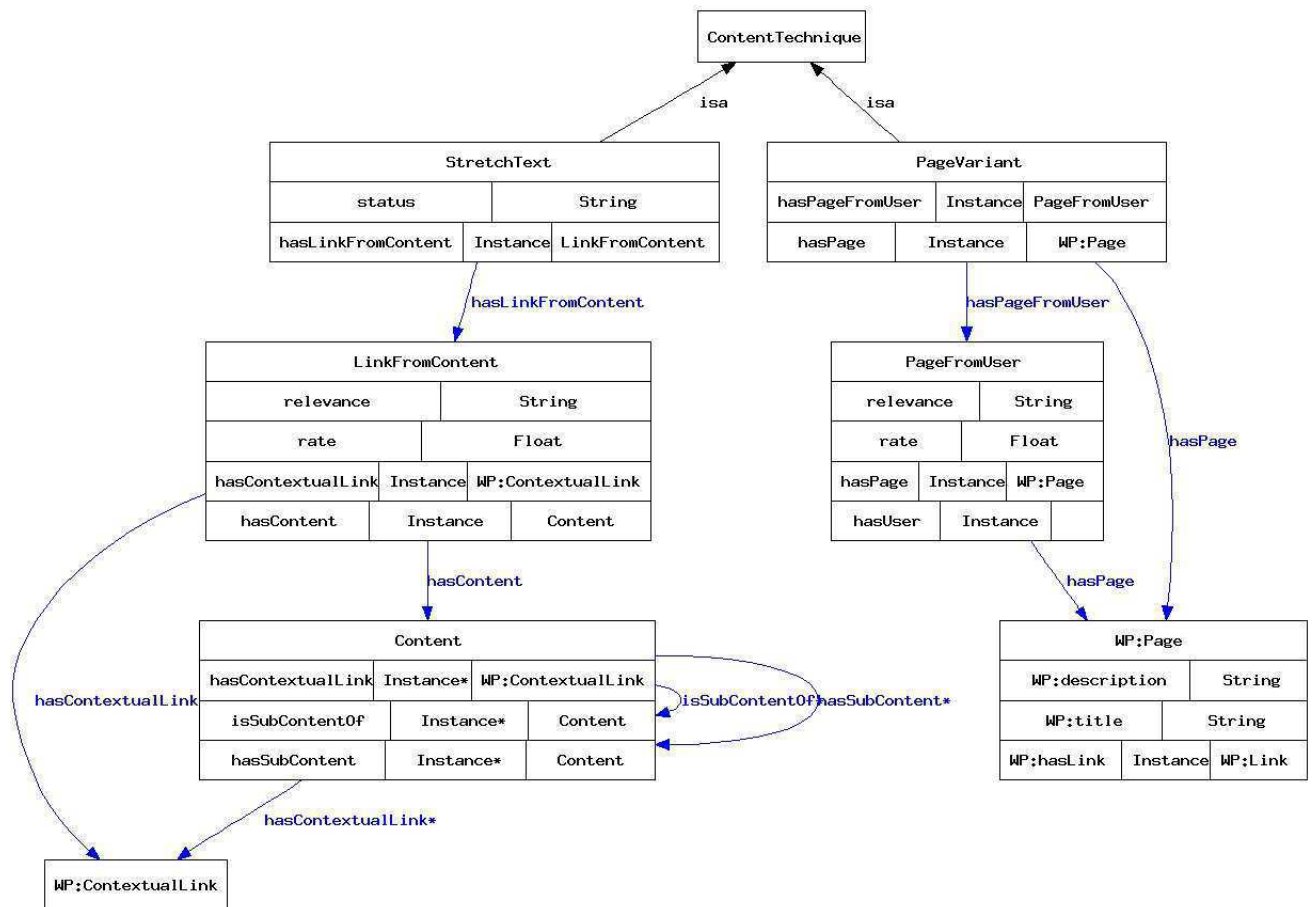


Figura 4.17: Técnicas de Adaptação descritas na ontologia de adaptação (b).

- *StretchText*: esta técnica cria no conteúdo um link para cada outro conteúdo que o compõe. Assim como na *ConditionalAdaptation*, será decidido se um conteúdo aparecerá ou não.

GAIA

Esta ontologia (vide Figura 4.18) foi criada baseada na metodologia para sistemas multi-agentes GAIA⁹, a qual provê suporte para as fases de análise e projeto de um sistema multi-agentes.

Algumas classes que estão presentes nesta ontologia seguem abaixo:

- *Environment*: sistemas multi-agentes estão inseridos em um ambiente. Desta forma, esta classe equivale ao ponto inicial para a modelagem dos SMAs. Portanto, cada

⁹Para mais detalhes sobre esta metodologia, vide [Zambonelli et al., 2003].

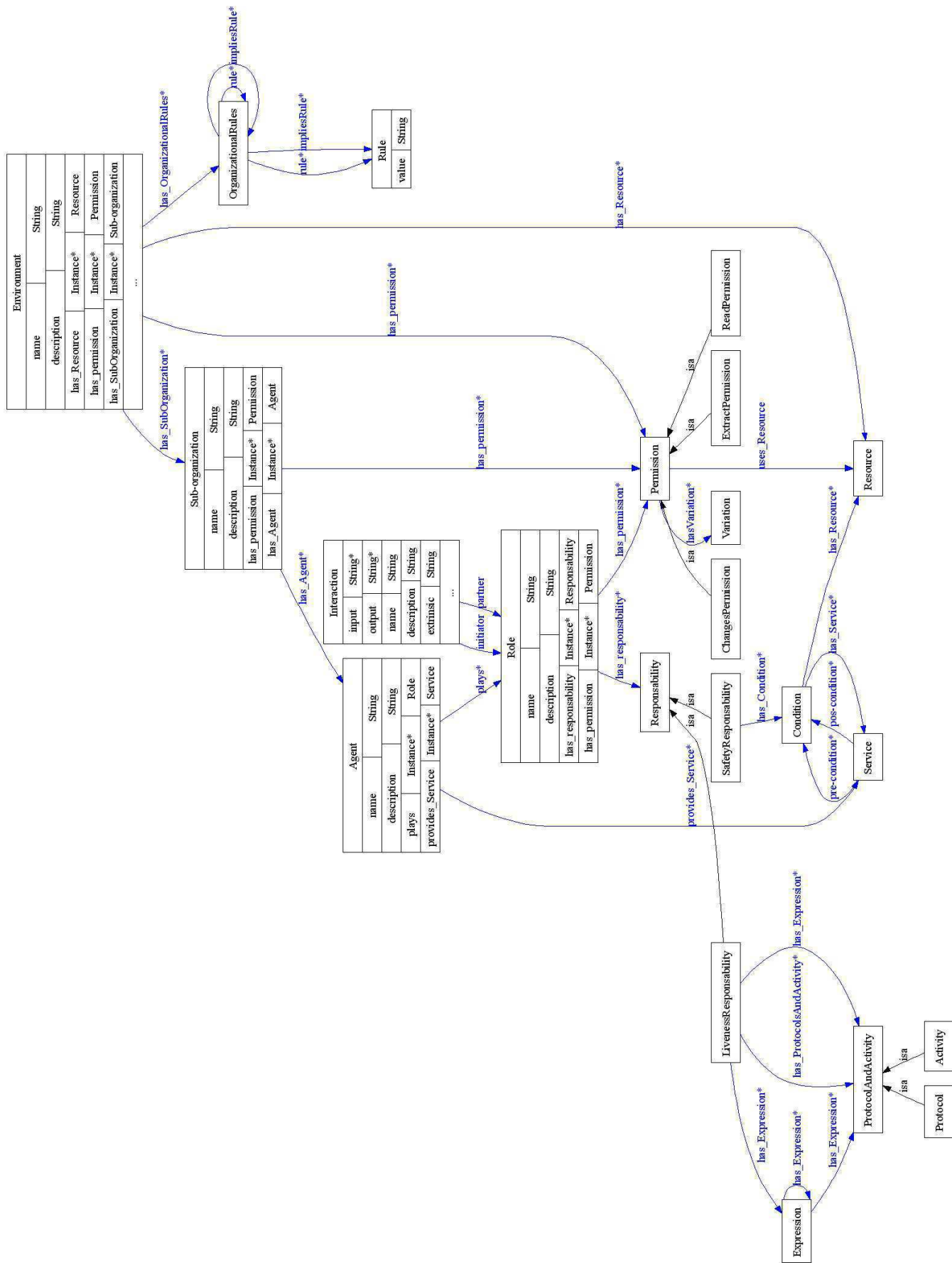


Figura 4.18: Uma parte da Ontologia de GAIA.

ambiente possui um conjunto de recursos (*resources*) que são utilizados pelos agentes e pelas permissões de acesso (*permissions (read, write)*). Além disso, outras classes importantes que estão relacionadas à classe *Environment* são as sub-organizações (*Sub-Organization*) e regras organizacionais (*OrganizationalRules*), que são descritas a seguir.

- *OrganizationalRules*: o objetivo desta classe é especificar as restrições nas quais os agentes devem aceitar durante seu ciclo de vida. Esta classe é muito importante para a criação de sistemas onde os agentes estão habilitados a entrar nos ambientes/organizações. Por esta razão, o sistema deve ter regras de interação onde todo novo agente deve levar em consideração quando o mesmo for inserido no ambiente;
 - *Sub-Organization*: esta classe é responsável pela divisão dos SMA em organizações com sub-objetivos, objetivando desta forma a modularização, reduzindo a complexidade e melhorando a capacidade de gerenciamento. Como resultado, cada organização possui um conjunto de agentes e permissões.
- *Agent*: definido como uma entidade que representa uma lista de papéis (*roles*). De fato, cada agente tem um conjunto de responsabilidades e permissões no ambiente ou sub-organização de modo a garantir que as interações entre os agentes ocorram de forma correta. Um agente também possui um conjunto de serviços, como descritos abaixo.
 - *Service*: representa uma capacidade do agente ou um comportamento executado pelo mesmo. Por esta razão, a interação entre os agentes ocorre através da identificação/descoberta dos serviços providos por cada agente. Desta forma, cada serviço possui parâmetros de entrada (*input*) e saída (*output*), bem como pré e pós-condições (*pre* e *post-conditions*, respectivamente). Estas propriedades são importantes para garantir que a execução do serviço ocorra de forma correta.

Regra Modelo de Domínio - GAIA

As regras de mapeamento do modelo do domínio para a ontologia de GAIA são bastante simples e só há necessidade de uma regra. Esta regra tem o objetivo de mapear um currículo definido na ontologia de domínio para um agente na ontologia de GAIA, ou seja, para cada currículo descrito no modelo de domínio, há a necessidade de criar um agente de software.

Devido à abordagem utilizada pela metodologia GAIA para construção de sistemas multi-agentes, todos os currículos do modelo do domínio são mapeados em um recurso de um papel em GAIA chamado Tutor. Desta forma, quando o mapeamento de GAIA para a ontologia *ForBILEAgent*, os agentes tutores são descritos, ou seja, para cada recurso de um papel tutor descrito na ontologia de GAIA, é gerado um agente tutor na ontologia de *ForBILEAgent*. Apesar de ser necessária apenas uma regra de mapeamento entre a ontologia do domínio e GAIA, outras três regras são necessárias para garantir o correto e completo mapeamento. As quatro regras são descritas na Tabela 4.1 e explicadas após.

Tabela 4.1: Mapeamento entre as ontologias Domain e GAIA utilizando SWRL.

Conceito Origem	Conceito Destino	Regra SWRL
Domain:Curriculum	GAIA:Resource	CurriculumToResource
GAIA:Protocol	GAIA:Interaction	ProtocolToInteraction
GAIA:LivenessResponsability	GAIA:Service	LivenessResponsabilityInput
GAIA:LivenessResponsability	JADGAIA:Service	LivenessResponsabilityOutput

A explicação das regras segue abaixo.

swrl-CurriculumToResource: responsável por mapear um currículo em um recurso de um papel de GAIA;

swrl-ProtocolToInteraction: responsável por mapear o protocolo descrito para o papel tutor em interações. Este mapeamento é necessário para garantir que o mapeamento para *JADE:ACLMessage* ocorra de forma correta e com isso garanta a comunicação entre agentes tutores;

swrl-LivenessResponsabilityInput: cria uma relação entre a *livenessResponsability* em GAIA com um serviço, também de GAIA. O nome do serviço é o mesmo nome da *livenessResponsability*. Com isso, os protocolos são mapeados em parâmetros de entrada do serviço;

swrl-LivenessResponsabilityOutput: esta regra é bastante similar à regra anterior, mudando apenas o mapeamento dos protocolos para os parâmetros de saída do serviço.

Esta estrutura de mapeamento e geração de agentes foi baseada no Modelo Mathema revisitado, como descrito na Seção A.2.4.

JADE

Esta seção descreve características referentes à ontologia do JADE¹⁰. Desta forma, o objetivo desta ontologia é especificar os agentes criados pelo framework JADE. Com isso, objetiva-se reduzir o tempo de modelagem e implementação através de um ambiente que implemente os agentes baseados nas ontologias propostas (vide Figura 4.19).

Uma breve descrição das classes presentes nesta ontologia é mostrada como segue:

- ***AgentPlatform***: representa um local onde os agentes são executados. Com isso, uma plataforma tem um conjunto de sub-organizações (*sub-organizations* - conceito do GAIA) que é mapeado como *containers*. Além disso, a plataforma é responsável pelo ciclo de vida dos agentes presentes em uma determinada organização;
- ***JADEAgent***: responsável pela especificação dos agentes (possuindo um identificador de agentes - *AID*) onde executa seus serviços (*services*) de acordo com os comportamentos (*behaviors*) de modo a alcançar seus objetivos. Além disso, a interação entre os agentes é provida pela troca de mensagens utilizando mensagens ACL (*ACLMessages*). Detalhes sobre as mensagens ACL e os comportamentos são providos abaixo:

¹⁰Para maiores detalhes sobre o JADE, vide A.1.1

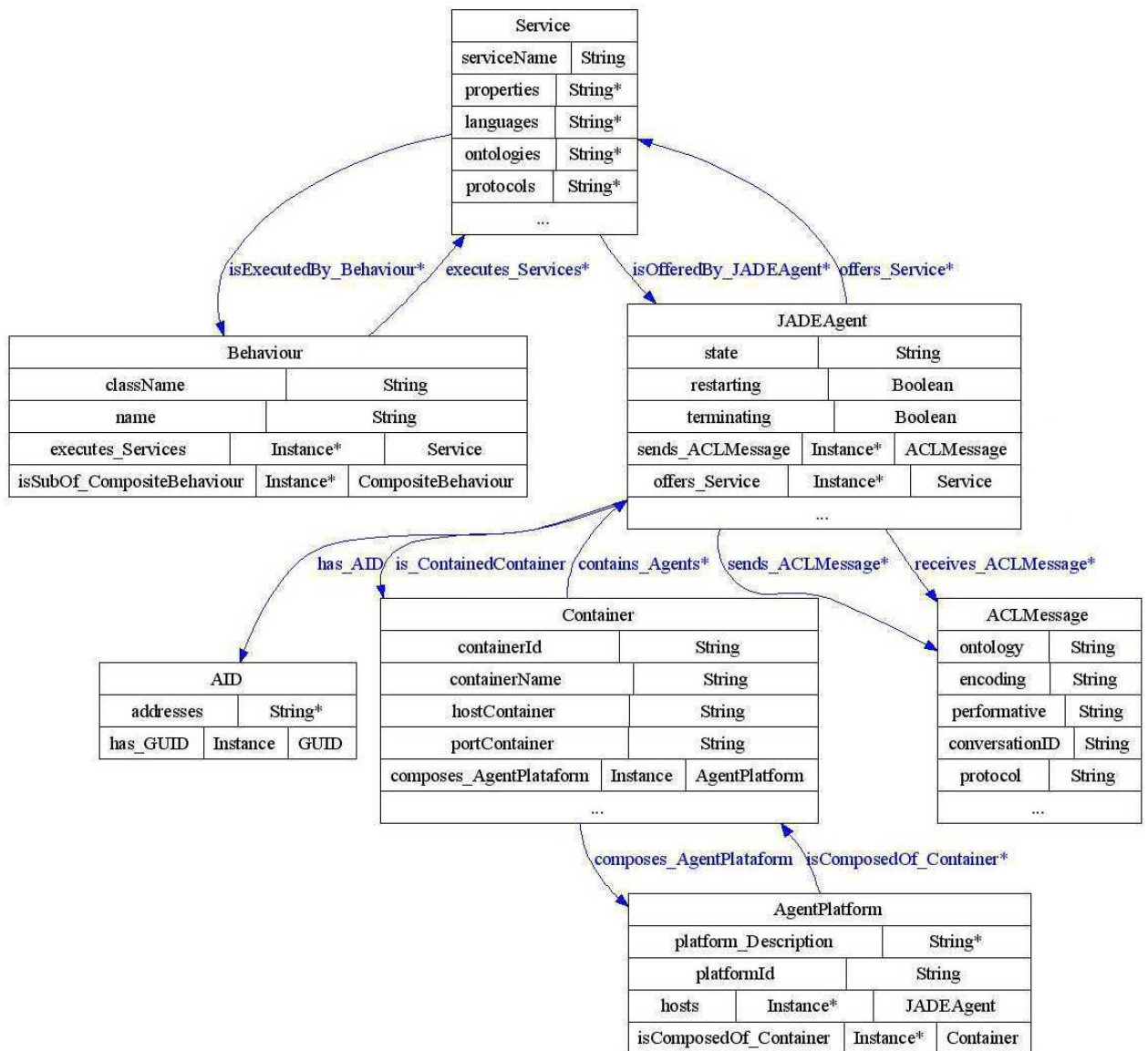


Figura 4.19: Uma parte da Ontologia do JADE.

- *ACLMessage*: as mensagens são usadas para garantir a interação entre agentes e provêm uma padronização definida pela FIPA¹¹. Estas mensagens representam todos os protocolos de interação entre os agentes e como os mesmos devem reagir de acordo com uma determinada requisição ou informação recebida;
- *Behaviour*: o comportamento representa as tarefas que podem ser executadas pelos agentes, sendo utilizados de acordo com os objetivos dos agentes.

¹¹Foundation for Intelligent Physical Agents [FIPA, 2008].

ForBILEAgent

Apesar da ontologia descrevendo/especificando as características presentes no JADE, o mesmo prevê apenas a infra-estrutura para a execução de agentes desenvolvidos em *Java*. Com isso, há a necessidade de estendê-lo com vistas a garantir um rápido desenvolvimento de aplicações educacionais baseadas em agentes. Verifica-se que a extensão ocorreu através da criação de uma estrutura padronizada para criação de agentes no MASSAYO. Esta estrutura possui uma ontologia que estende a ontologia do Jade (vide Figura 4.20), chamada *ForBILEAgent*.

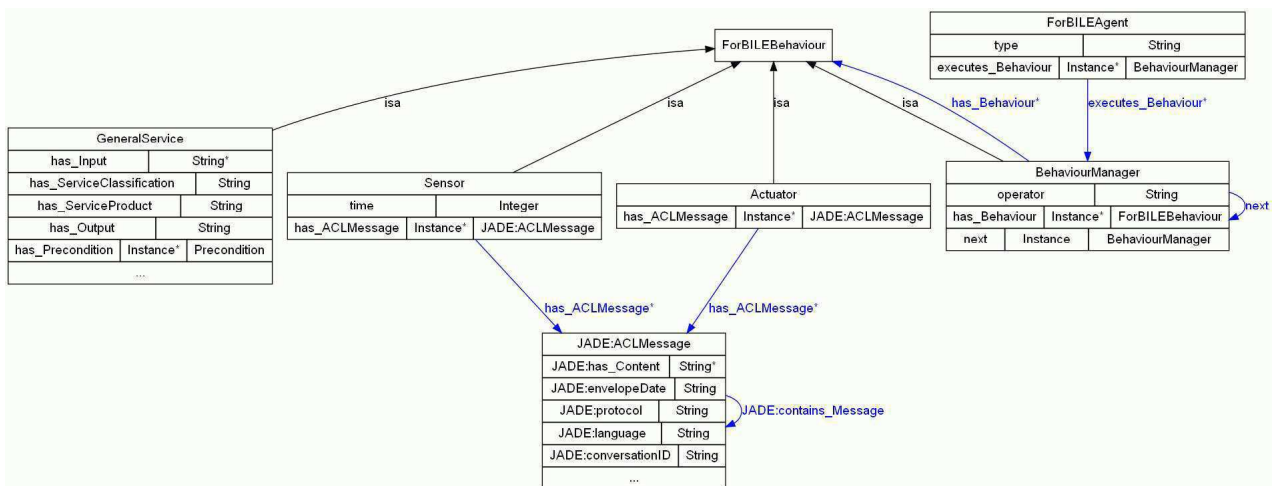


Figura 4.20: Uma parte da Ontologia do ForBILEAgent.

Após a especificação das ontologias do GAIA, JADE e *ForBILEAgent*, é necessário prover um mecanismo para conectar/mapear os conceitos presentes nestas ontologias. Por esta razão, foram descritas regras em SWRL, como seguem nas subseções.

Regras GAIA - ForBILEAgent

SWRL foi utilizado neste contexto para garantir o mapeamento de instâncias entre ontologias. A Tabela 4.2 aborda os conceitos mapeados através do uso das regras SWRL.

Tabela 4.2: Mapeamento entre as ontologias GAIA, JADE e *ForBILEAgent* utilizando SWRL.

Conceito Origem	Conceito Destino	Regra SWRL
gaia:Agent	JADE:JADEAgent	AgentToJADEAgent
gaia:Service	JADE:Service	ServiceToService
gaia:Interaction	JADE:ACLMessage	InteractionToACLMessage
gaia:Activity	JADE:Behavior	ActivityToBehavior
gaia:LivenessResponsability	gaia:Service	LivenessResponsabilityToService
gaia:LivenessResponsability	gaia:Service	LivenessResponsabilityToOutput
gaia:Protocol	gaia:Interaction	ProtocolToInteraction
JADE:Agent	JADE:ForBILEAgent	AgentToForBILEAgent
gaia:SubOrganization	JADE:Container	SubOrganizationToContainer
GAIA:Resource	ForBILEAgent:TutorAgent	ResourceToTutorAgent
GAIA:Activity	ForBILEAgent:GeneralService	ActivityToGeneralService
GAIA:Interaction	ForBILEAgent:ForBILESensor	InteractionToForBILEAgent
GAIA:Interaction	ForBILEAgent:ForBILEActuator	InteractionToForBILEActuator
GAIA:Expression	ForBILEAgent:BehaviourManager	ExpressionToBehaviourManager

Algumas das regras são descritas como seguem:

swrl-AgentToJADEAgent: nesta regra, o agente definido em GAIA é mapeado para o mesmo agente em JADE. Para isto, é necessário que o sistema crie uma *GUID* e a propriedade *localName*, sendo também equivalente em GAIA. Já a *GUID* é definida para a propriedade de identificação do agente - *AID*.

swrl-ServiceToService: com esta regra, um serviço GAIA é mapeado em um serviço do JADE. O foco desta regra é garantir que todos os agentes que foram mapeados em JADE possuam seus respectivos serviços.

swrl-InteractionToACLMessage: relaciona as interações descritas em GAIA com as mensagens ACL pertencentes a JADE. Basicamente, o iniciador da interação (*initiator*) em GAIA é mapeado para a propriedade ACL do agente que envia a

mensagem *sender*. Já o parceiro (*partner*) da interação em GAIA é mapeado para o receptor (*receiver*) da mensagem em JADE.

swrl-ActivityToBehavior: as atividades de um papel em GAIA são mapeadas para os comportamentos do JADE *behavior*. Finalmente, o serviço do JADE é configurado como representando uma execução de determinado comportamento.

swrl-LivenessResponsabilityToService: cria uma relação entre a *livenessResponsability* em GAIA com um serviço, também de GAIA. O nome do serviço é o mesmo nome da *livenessResponsability*. Com isso, os protocolos são mapeados em parâmetros de entrada do serviço.

swrl-LivenessResponsabilityToOutput: esta regra é bastante similar à regra anterior, mudando apenas o mapeamento dos protocolos para os parâmetros de saída do serviço.

Aplicação

Para garantir a descoberta e interoperabilidade entre ambientes educacionais, há a necessidade de um repositório onde os ambientes devem prover uma instância da ontologia que descreve o ambiente educacional. Em outras palavras, cada ambiente é descrito através de uma ontologia de aplicação e então os mesmos têm uma referência da instância desta ontologia no repositório. Esta ontologia presente no repositório pode ser dividida da seguinte forma [Bittencourt et al., 2009c]:

1. Informações do Ambiente: a ontologia de aplicação tem as informações necessárias para que os ambientes possam se registrar e então serem descobertos. Além disso, mais detalhes são providos sobre esta parte da ontologia (vide Figura 4.21), como mostrado abaixo:
 - *Environment*: representa as informações gerais sobre o ambiente educacional (por exemplo, o nome do ambiente, local na rede, etc) registrado. De fato, esta é a principal classe, pois a mesma identifica as entidades e serviços disponíveis para interoperar com outros ambientes, ou seja, esta classe possui

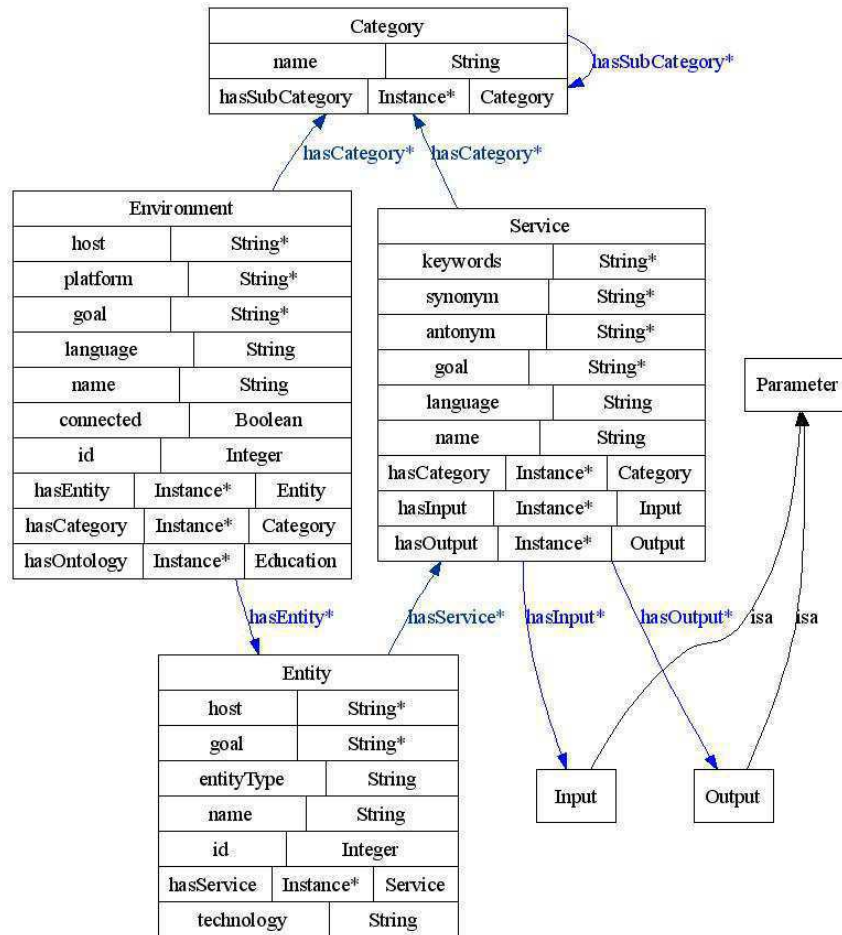


Figura 4.21: Parte da Ontologia de Aplicação.

relacionamento com todas as classes que possuem as informações sobre os tipos de interação que podem ser feitas entre os ambientes educacionais;

- *Category*: representa o contexto referente ao ambiente educacional, por exemplo, medicina, matemática, entre outros;
- *Entity*: responsáveis pela execução dos serviços/funcionalidades providas pelo ambiente. Esta classe possui uma propriedade chamada *entityType* com o intuito de descrever a abordagem utilizada pela entidade, como por exemplo, baseada em agentes, e em componentes, orientada a serviços, etc;
- *Service*: as funcionalidades providas pelo ambiente são implementadas através de serviços. Desta forma, com a disponibilização dos serviços, os ambientes capazes de interagir poderão ser descobertos.

2. Informações sobre os Alinhamentos: estas informações representam uma parte da ontologia de aplicação, descrevendo informações sobre as ontologias de determinado ambiente que foram alinhadas com outras externas. Ou seja, o objetivo é identificar se uma ontologia de um ambiente está alinhada com outras ontologias. Como definida na Figura 4.22, a classe *Environment* tem informações gerais sobre o ambiente educacional. Entretanto, esta ontologia pode ser alinhada com outras ontologias pertencentes a domínios variados. Por esta razão, a classe *Education* descreve as ontologias do ambiente e a classe *Alignment* possui o URI da outra ontologia.

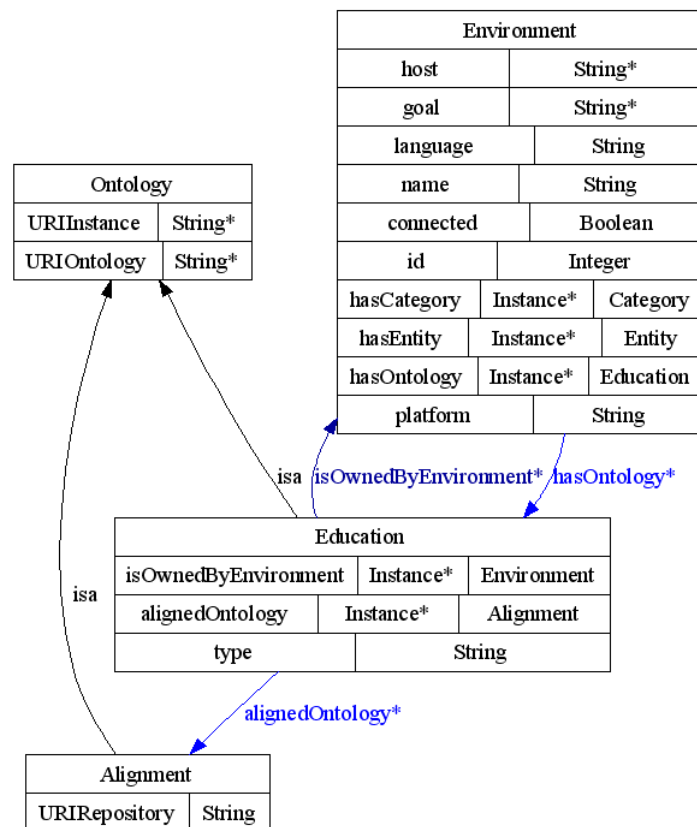


Figura 4.22: Parte da ontologia provendo informações sobre as ontologias alinhadas.

Regras ForBILEAgent - Aplicação

Para que a interoperabilidade seja automatizada, há a necessidade que ocorra o mapeamento entre as ontologias *ForBILEAgent* e *Application*. Como descrito anteriormente, a ontologia *ForBILEAgent* possui a descrição de todos os agentes

presentes nos ambientes educacionais bem como os serviços implementados por cada agente. Desta forma, o mapeamento permite que a descrição do ambiente educacional seja representada na ontologia de aplicação de forma automatizada.

Para que isto seja possível, três regras foram descritas (vide na Tabela 4.3).

Tabela 4.3: Regras de mapeamento entre as ontologias *ForBILEAgent* e *Application*.

Conceito Origem	Conceito Destino	Regra SWRL
ForBILEAgent:ForBILEAgent	Application:Entity	ForBILEAgentToEntity
ForBILEAgent:GeneralService	Application:EntityService	GeneralServiceToEntityService
ForBILEAgent:GeneralService	Application:Service	GeneralServiceToService

A explicação das regras segue abaixo.

swrl-ForBILEAgentToEntity: faz o mapeamento de um agente descrito na ontologia *ForBILEAgent* e para uma entidade na ontologia *Application*. É importante frisar que este mapeamento também já define que o tipo de entidade equivale a *Agent*;

swrl-GeneralServiceToService: responsável por mapear os serviço, descrevendo a categoria do serviço, bem como os parâmetros de entrada e saída;

swrl-GeneralServiceToEntityService: responsável por definir a relação existente entre as entidades e os serviços descritos nas regras anteriores.

4.2.2 Camada do Framework

Um framework baseado na Web semântica foi desenvolvido com o intuito de facilitar o desenvolvimento de sistemas educacionais multi-agentes, chamado MASSAYO-F¹². Desta forma, o mesmo busca assegurar um desenvolvimento rápido para a construção de tais sistemas, com o mínimo possível de modificação e/ou adição de código. Ele possui o intuito de prover uma aplicação com recursos de adaptação de acordo com as necessidades dos usuários, além de expandir as capacidades e conhecimento dos agentes tutores inteligentes [Bittencourt et al., 2007b]. As tecnologias usadas no desenvolvimento do framework foram Tomcat, Jade, Protégé, OWLSEditor, Mindswap e OWL-DL. A Figura 4.23 mostra a arquitetura proposta pelo framework.

¹²F de Framework.

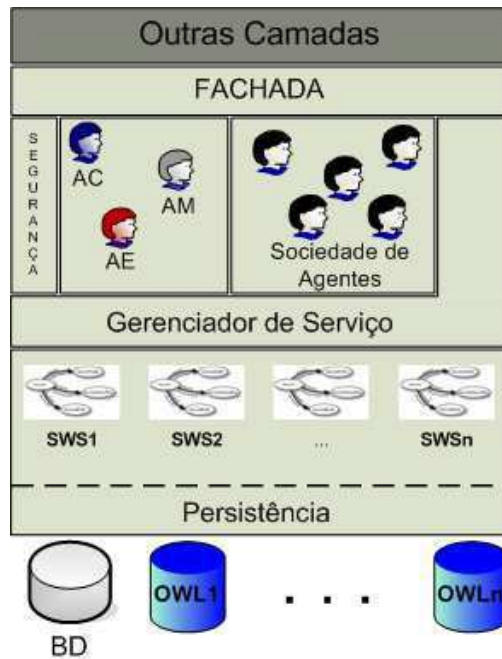


Figura 4.23: Arquitetura baseada em agentes da camada de Framework.

Detalhes sobre as entidades presentes na arquitetura são descritos como seguem:

- **Serviços de Persistência:** representam a implementação do mecanismo de persistência (criação, recuperação, atualização e deleção em ontologias) através da utilização do *Jena* e *ProtegeOWLAPI*;
- **Agentes:** os agentes são disponibilizados no ambiente com diversos objetivos diferenciados. Eles são compostos por Agente Controlador, Agente Mediador, Agente de Evolução e Sociedade de Agentes. Tais agentes são descritos como seguem:
 - *AM (Agente Mediador)*: possui duas funcionalidades: i) recomendar atores de acordo com as necessidades do requisitante e ii) garantir a interoperabilidade com outros ambientes educacionais;
 - *AC (Agente Controlador)*: possui três habilidades fundamentais, que são: i) inicializar os agentes, ii) adicionar, remover e atualizar a sociedade de agentes e iii) prover dados para a adição, remoção e atualização dos serviços de cada agente.

- AE (*Agente de Evolução*): possui duas habilidades fundamentais, sendo elas:
 - i) executar as regras para geração/mapeamento dos agentes do ambiente educacional e
 - ii) verificar se houve alguma alteração na ontologia *ForBILEAgent* para refletir as mudanças dinamicamente no ambiente;
- Sociedade de Agentes: representada por agentes heterogêneos composta pelos seguintes agentes:
 - i) agentes de suporte (*SupportAgents*): eles dão suporte aos outros agentes para assegurar que os mesmos alcancem os objetivos de cada atividade educacional. Exemplos destes agentes correspondem aos agentes que utilizam mecanismos de inferência para prover determinado tipo de adaptação no ambiente ou a resolução de problemas e
 - ii) agentes tutores autônomos (*ATAAgents*): eles representam os principais agentes presentes no ambiente e executam os requisitos necessários para ensinar um estudante em determinado subdomínio.
- Serviços Web Semânticos: têm as funcionalidades que são providas aos usuários, agentes e ferramentas presentes no ambiente educacional. Desta forma, os mesmos garantem a descoberta, composição e invocação dos serviços de forma automática;
- Gerenciador de Serviços: o objetivo do gerenciador de serviços é garantir a descoberta, composição e invocação de serviços automaticamente. Isto é feito através da utilização de algoritmos de casamento de padrões [Calado et al., 2009].

As subseções que seguem, descrevem de forma mais detalhada cada uma das entidades presentes no framework proposto.

Agentes

Esta subseção descreve os agentes presentes na arquitetura. Eles são compostos pelo Agente Controlador, Agente Mediador, Agente de Evolução e Sociedade de Agentes (Agentes Tutores Autônomos e Agentes de Suporte). A Figura 4.24 descreve o diagrama de classes dos agentes.

Como pode ser visto na Figura 4.24, a extensão do framework Jade ocorre pela classe *ForBILEAgent* estendendo a classe *Agent* do Jade. A Classe *ForBILEAgent* é uma

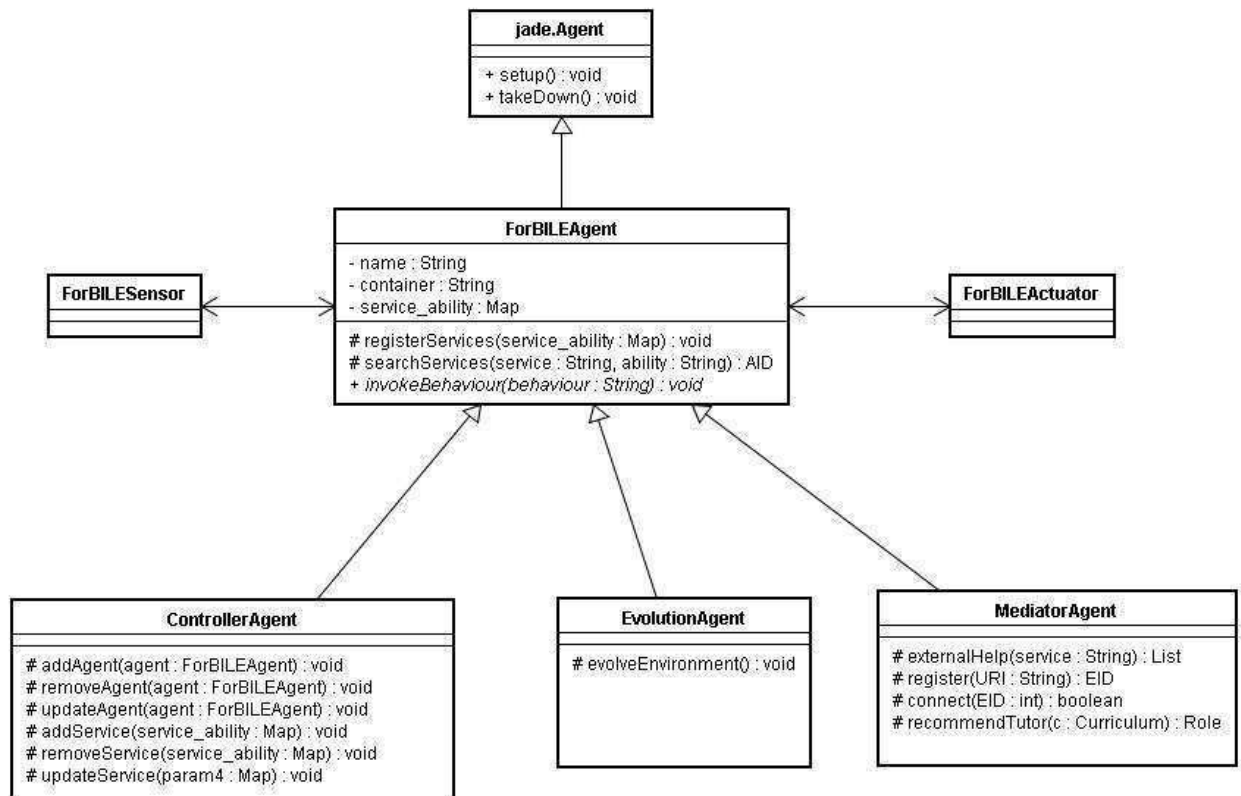


Figura 4.24: Diagrama de Classe dos agentes do framework.

classe concreta que implementa algumas das funcionalidades padrões inerentes a qualquer agente de software presente no MASSAYO. Algumas dessas funcionalidades são: a) registrar serviços do agente; b) sensor para perceber o ambiente; c) atuador para atuar no ambiente; d) busca de agentes presentes na sociedade, entre outras. Com isso, todos os outros agentes presentes na arquitetura estendem o *ForBILEAgent* e consequentemente herdam suas características. Por exemplo, todo o mecanismo de interação entre os agentes está implementado no sensor do *ForBILEAgent*, desta forma, os outros agentes não precisam especificar e implementar o mecanismo de interação no ambiente.

Com o intuito de discutir as funcionalidades e características de cada agente presente na arquitetura supra descrita, aborda-se abaixo cada um deles.

Agente Controlador Este agente é responsável por garantir que os agentes estejam coesos com o modelo descrito nas ontologias de agentes, através do controle/gerenciamento dos agentes e serviços dos mesmos. Além disso, todas as

informações sobre os agentes e suas funcionalidades são especificadas através das ontologias descritas na subseção 4.2.1.

Este agente possui três funcionalidades, sendo elas:

1. Iniciar Agentes: quando o ambiente é inicializado, há a necessidade de executar os agentes que estarão presentes.

Desta forma, cada agente requer uma tripla formada por $\langle \text{Agente}, \text{Serviço}, \text{Habilidade} \rangle$, onde:

Agente: agente que será inicializado;

Serviço: parâmetro que corresponde aos serviços presentes em cada agente do ambiente;

Habilidade: diz respeito a uma capacidade específica do serviço de um agente. Por exemplo, um agente poderia ter o serviço de *Tutorar* e a habilidade de tutorar *fração*;

2. Controlar Agentes: equivale à capacidade do agente controlador poder adicionar, remover e atualizar agentes presentes na sociedade. Isto garante a manutenção dinâmica dos agentes da sociedade;
3. Controlar Serviços: o agente controlador não faz o controle direto dos serviços presentes em cada agente. Entretanto, o mesmo é responsável por informar quando os serviços de cada agente foram alterados. Ou seja, o agente controlador informa se um serviço deve ser adicionado, removido ou atualizado. Desta forma, toda manutenção nos serviços presentes em cada agente pode ser feita dinamicamente.

Agente Mediador A complexidade envolvida com o gerenciamento das interações em sistemas multi-agentes foi um dos motivadores para a criação e utilização do agente mediador. Além disso, este agente exerce papel fundamental nos ambientes educacionais criados através do MASSAYO. Este agente possui duas funcionalidades fundamentais, sendo elas:

1. Recomendação: serve para recomendar atores, como por exemplo, agentes tutores autônomos (ATA) para um determinado estudante ou outro ATA. Esta

recomendação é possível através dos protocolos de comunicação especificados para os agentes e através da ontologia do domínio que possui informações sobre os currículos;

2. Interoperabilidade: oferece comunicação com outros ambientes educacionais com vistas a atingir determinado objetivo do ambiente educacional, como por exemplo, a recomendação de algum objeto de aprendizagem para o estudante que não estava disponível no ambiente¹³.

Agente de Evolução Tanto as necessidades de automatização do processo de geração dos agentes quanto as constantes mudanças no ambiente fizeram com que um agente de evolução fosse adicionada à arquitetura do framework. Desta forma, tal agente possui duas funcionalidades básicas, que são:

1. Execução de regras: tanto a ontologia do modelo do domínio quanto as ontologias de aplicação (GAIA, JADE e *ForBILEAgent*) possuem regras que devem ser executadas quando o ambiente for distribuído. Desta forma, este agente primeiramente executa as regras da ontologia de domínio com o objetivo de mapear os agentes tutores que serão gerados. Logo depois, o mesmo executa as regras de GAIA, JADE e *ForBILEAgent*, respectivamente. Esta segunda execução garante a especificação de todas as interações presentes entre os agentes;
2. Verificar alterações nas ontologias: há duas formas principais de manutenção no framework, que são através dos agentes ou serviços. Os serviços Web semânticos garantem a descoberta dinâmica e automática de novos serviços que foram adicionados. Entretanto, para que o ambiente garanta a manutenção dos agentes, há a necessidade de um agente que verifique de tempos em tempos se houve alterações nas ontologias. Caso haja alteração, o mesmo identifica quais foram e solicita ao *Agente Controlador* as mudanças necessárias no ambiente.

Sociedade de Agentes A sociedade de agentes foi criada com o intuito de decompor os problemas relacionados à resolução de problemas, diversos subdomínios envolvidos,

¹³Mais detalhes são apresentados na Subsecção 4.3

necessidade de personalização de acordo com um usuário ou grupo, entre outros. Por estas razões, uma sociedade de agentes heterogêneos foi criada com o objetivo de tutorar o estudante e de auxiliar os agentes em atividades específicas. Com isso, os tipos de agentes são:

- **Agentes Tutores Autônomos:** modelados baseados no modelo Mathema revisitado. Além disso, estes agentes são responsáveis por manipular as ontologias do domínio, estudante e pedagógico com vistas a interagir com o estudante em prol do seu desenvolvimento cognitivo. Desta forma, as funcionalidades presentes nestes agentes são i) tutoramento, ii) resolução de problemas, iii) avaliação e iv) diagnóstico;
- **Agentes de Suporte:** responsáveis por auxiliar todas as entidades de um ambiente educacional que necessitem de alguma funcionalidade adicional. Por exemplo, tais agentes são responsáveis por prover todo o mecanismo de adaptação do ambiente.

Serviços Web Semânticos

Os serviços Web semânticos são responsáveis por implementar os requisitos funcionais do modelo. Desta forma, os serviços proporcionam a implementação de diversas atividades educacionais e de forma distribuída na Web.

Com isso, os serviços, tanto educacionais quanto de suporte, foram divididos em categorias e sub-categorias, como mostrados na Figura 4.25, tendo sido implementados em torno 47 serviços Web semânticos, sendo os principais de raciocínio (aproximadamente 17) e educacionais (aproximadamente 22).

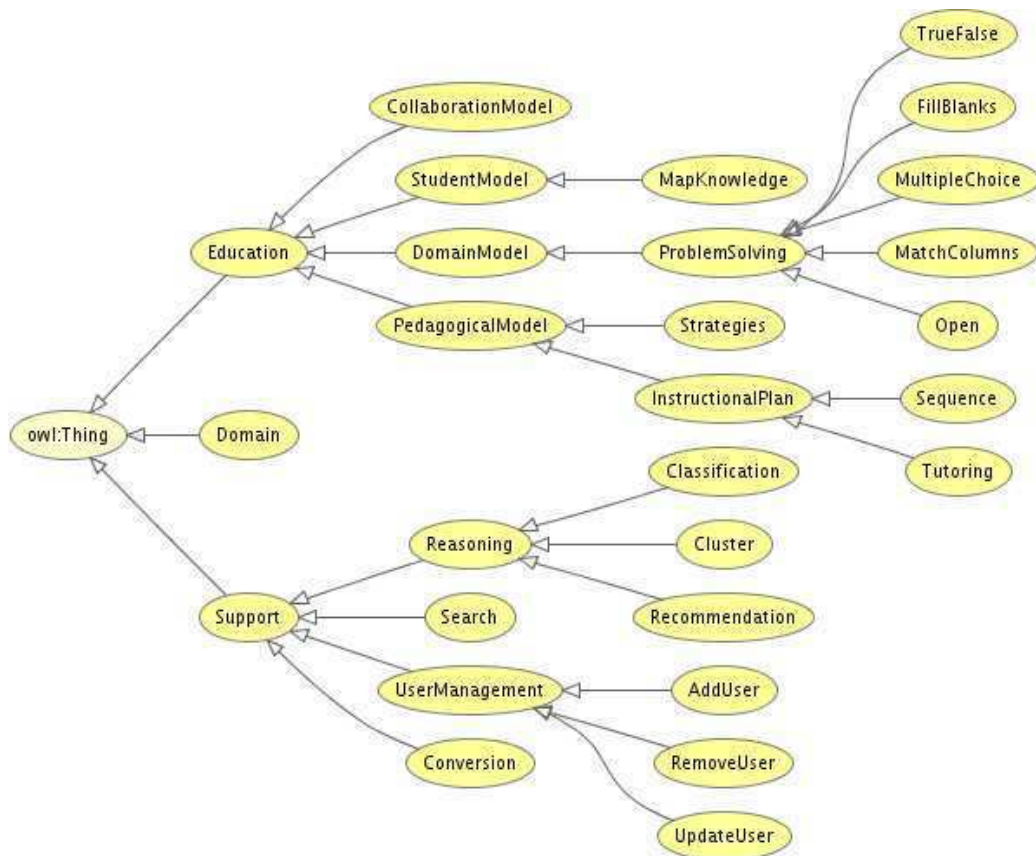


Figura 4.25: Hierarquia com Classificação para SWS.

Gerenciador de Serviço

O objetivo do gerenciador de serviço é garantir a descoberta, composição e invocação de serviços automaticamente. Isto é feito através da utilização de algoritmos de casamento de padrões [Calado et al., 2009].

A implementação do algoritmo foi feita usando, além da linguagem Java¹⁴, a API do *MindSwap* [Mindswap, 2009] para criar e manipular objetos que foram representados com serviços semânticos. Com isso, o algoritmo faz o uso de métricas de similaridade para avaliar a qualidade da solução proposta. Algumas características do algoritmo são:

- Combinação de serviços de forma direta e indireta;
- Representação de grafos diretos ponderados para o relacionamento de serviços;

¹⁴java.sun.com

- Algoritmo do menor caminho para localizar o melhor caminho no grafo ponderado, buscando aumentar a acurácia da composição.

4.2.3 Camada da Aplicação

Esta seção descreve as características e componentes disponibilizados, representando as aplicações que o usuário pode interagir, sendo definidas ferramentas para interação tanto síncrona quanto assíncronas, bem como para os diversos tipos de usuários. A Figura 4.26 descreve os componentes presentes na camada de aplicação.

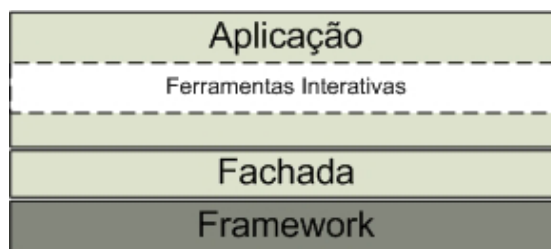


Figura 4.26: Arquitetura para a camada de aplicação.

Sendo assim, os ambientes educacionais necessitam de ferramentas que possibilitem a interação entre os usuários dos papéis. Entretanto, observa-se na literatura diversas ferramentas interativas maduras e robustas, motivando assim o reuso das mesmas. Como exemplo de ambientes que disponibilizam estas ferramentas, encapsuladas como componentes de software, há o SAKAI. Desta forma, ferramentas disponibilizadas pelo *Liferay* foram reusadas pelo MASSAYO, como: *fórum*, *scrap*, *blog*, *enquete*, *e-mail*, *comunidades*, *notícias*, *biblioteca digital*, dentre outras.

4.2.4 Camada da Apresentação

Esta camada é responsável por gerar a apresentação para que depois seja enviada para o cliente. No contexto desta tese, o cliente seria o próprio navegador sendo utilizado por um usuário com um papel específico.

As tecnologias utilizadas para apresentação do ambiente educacional foram TOMCAT e *Portlets* [Linwood and Minter, 2004]. Sendo assim, todo componente disponibilizado na camada de aplicação será encapsulado em um *portlet*. Além disso, o ambiente disponibiliza uma apresentação padrão, como mostrado na Figura 4.27.



Figura 4.27: Exemplo de Apresentação Padrão(a).

Outros exemplos de interface são vistos na Figura 4.28.

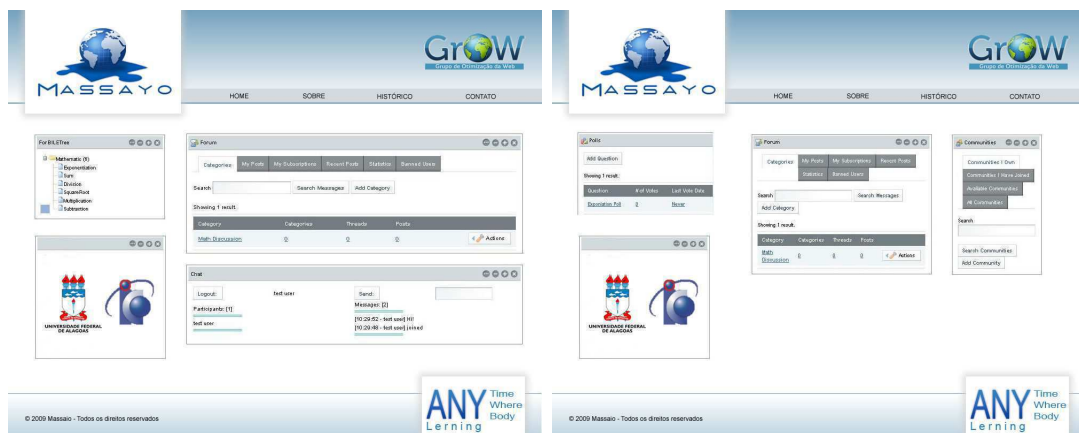


Figura 4.28: Exemplos de Apresentação Padrão (b e c).

4.3 Modelo de Infra-estrutura para Interoperabilidade - MASSAYO-IM

Esta seção descreve a arquitetura proposta com o intuito de garantir interoperabilidade entre ambientes Educacionais baseados na Web Semântica. A construção desta

arquitetura e a implementação de ambientes educacionais com vistas a permitir interoperabilidade é uma tarefa complexa e lenta [Dicheva and Aroyo, 2004b]. Por esta razão, a aplicação foi decomposta para reduzir a complexidade, permitindo a interação entre estes ambientes, tanto com a mesma infra-estrutura quanto com diferentes infra-estruturas. Com isso, esta arquitetura enfatiza dois aspectos. Primeiro, ela garante a integração de diversos ambientes educacionais baseados nas tecnologias da Web semântica. Segundo, ela provê um caminho para escalar os ambientes educacionais disponíveis através do uso de repositórios de ontologias e para estender os recursos de interoperabilidade através do uso de serviços Web semânticos. A Figura 4.29 apresenta a arquitetura proposta e seus componentes.

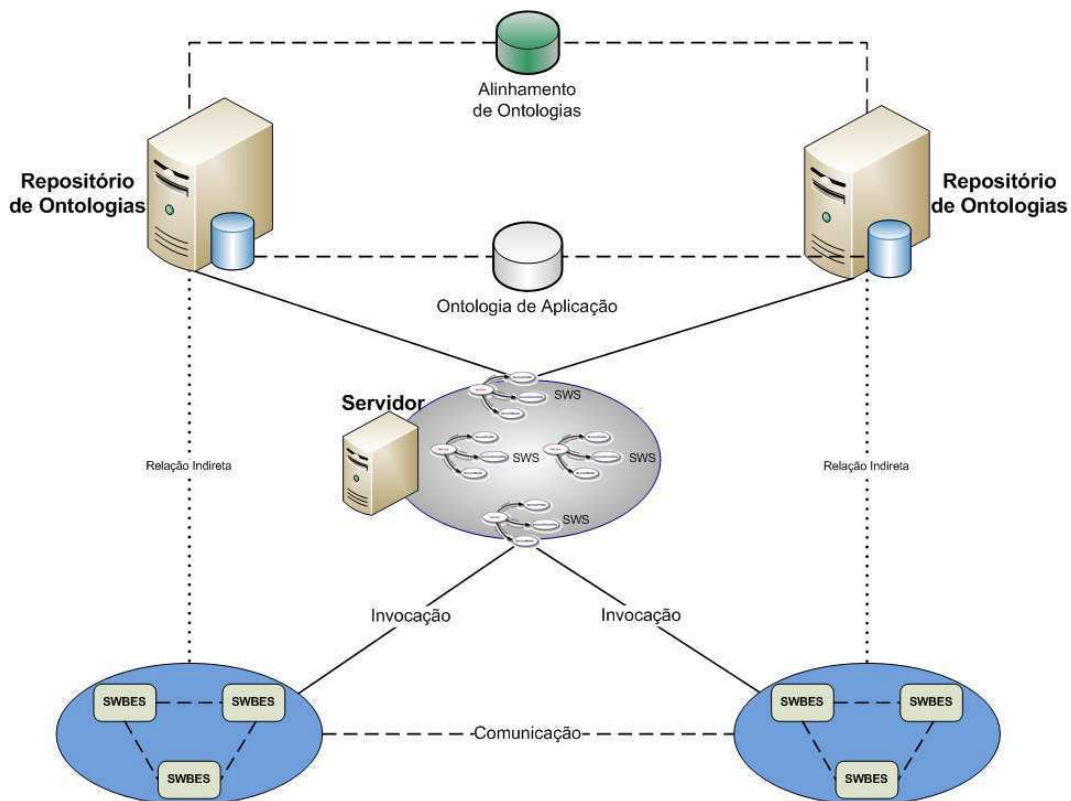


Figura 4.29: Arquitetura para suportar interoperabilidade entre ambientes educacionais.

A arquitetura representa uma extensão da arquitetura proposta em [Dicheva and Aroyo, 2004a]¹⁵. A idéia foi prover uma arquitetura de implementação descrevendo as tecnologias e como cada uma delas foram utilizadas. A descrição de cada componente

¹⁵A arquitetura proposta por [Dicheva and Aroyo, 2004a] é uma arquitetura conceitual. Além disso, o servidor de interoperabilidade através da utilização de serviços foi adicionado.

presente nesta arquitetura é provida nas subseções que seguem.

4.3.1 Repositório de Ontologias

O repositório de ontologias possui as instâncias da ontologia de aplicação descrita na Subseção 4.2.1, garantindo que os ambientes sejam registrados de forma corrente. Com isso, a partir do momento em que ambientes necessitem interagir com outros ambientes, os serviços farão uso do repositório de ontologias para descobrir tais ambientes.

4.3.2 Servidor para Interoperabilidade

O servidor para interoperabilidade é responsável por prover os serviços Web semânticos de modo a garantir tanto o registro do ambiente educacional bem como o algoritmo de descoberta capaz de comunicar com outros. Além disso, o Servidor tem como objetivo disponibilizar recursos que maximizem a capacidade do mesmo de descobrir novos ambientes. Os serviços presentes são descritos como seguem:

- *RegisterService*: o mesmo é responsável por registrar o seu ambiente educacional para ser usado no futuro, como histórico. Como resultado, os parâmetros de saída do serviço são identificados (*ID*) como ambiente educacional;
- *ConnectService*: este serviço permite que o ambiente conecte-se ao servidor. Como resultado, os parâmetros de saída do serviço é um *Boolean* informando se o registro foi feito bem sucedido ou não;
- *RequestService*: este serviço é usado pelo ambiente para solicitar ajuda externa aos ambientes educacionais. Em outras palavras, este serviço indica quais ambientes educacionais estão disponíveis para interoperar. Este serviço recebe o *nome* de um serviço como uma entrada (e.g. *Recommend*) e retorna a tripla $\langle \textit{entity}, \textit{protocol}, \textit{hostEntity}, \textit{service}, \textit{inputService}, \textit{outputService} \rangle$;
- *MatchingEnvironmentService*: este serviço é o mais complexo por que este algoritmo é responsável pela descoberta de ambientes disponíveis para interoperar e que atende aos requisitos do ambiente educacional solicitante. A complexidade

está na necessidade de avaliar diversos conceitos em busca do agente mais similar para a descoberta. Com isso, os parâmetros utilizados para verificar a similaridade são: *environmentId*, *serviceName*, *input*, *output* e *category*. Cada parâmetro tem um peso diferente, onde os mais importantes para o cálculo são *category* e *input*. Com isso, o cálculo é feito através da utilização da fórmula do cosseno [Baeza-Yates and Ribeiro-Neto, 2005].

4.3.3 Ambientes Educacionais

Os ambientes educacionais são entidades que podem ser registradas para interoperar. Apesar dos repositórios da ontologia manterem todas as instâncias descritas por um ambiente, a comunicação com os ambientes acontece através da utilização de serviços Web semânticos. Estes ambientes educacionais podem ser configurados e implementados de modo a permitir interoperabilidade de forma automática com outros ambientes. Isto é possível através dos componentes providos na arquitetura e o uso dos serviços Web semânticos.

Além do mais, os ambientes educacionais invocam serviços Web semânticos em três situações:

1. Primeiro Acesso: o sistema estará disponibilizado pela primeira vez;
2. Novo acesso: o sistema registrado estava *OFF-Line* e passou a estar disponibilizado para interoperabilidade. Tanto o novo acesso quanto o primeiro acesso possui um fluxo de execução bastante similar, mudando apenas de acordo com o tipo de acesso que é feito. Esta sequência é mostrada na Figura 4.30.

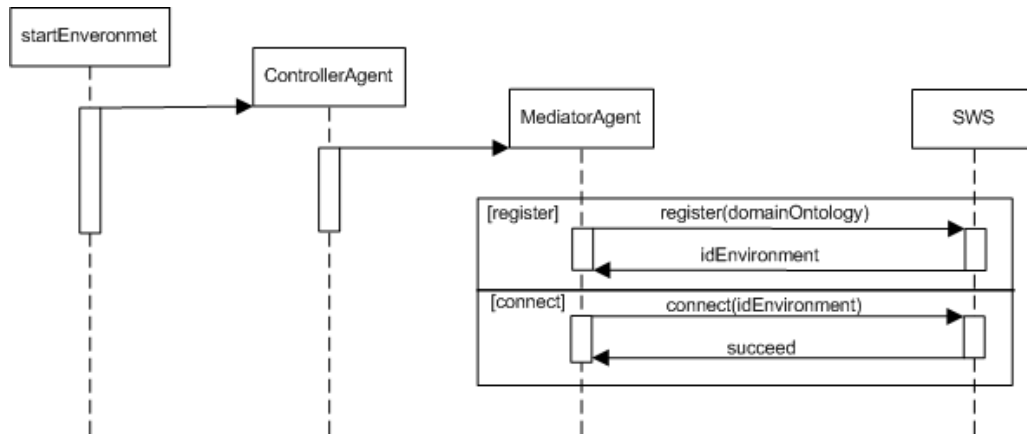


Figura 4.30: Diagrama de Sequência descrevendo o registro e conexão do MASSAYO.

Pode-se observar na Figura 4.30 que a invocação do serviço ocorre logo após a criação do agente mediador pelo agente controlador. Ou seja, quando o ambiente é iniciado, é feito o registro ou conexão do mesmo no servidor de interoperabilidade.

3. Solicitar Ajuda: o mesmo é usado quando o sistema identifica uma situação na qual é necessário interoperar com outros ambientes. O MASSAYO já possui 7 cenários (3 de recomendação e 4 de recursos educacionais) para garantir interoperabilidade entre os ambientes. Os cenários são descritos abaixo:

- (a) Recomendação de agente (humano ou artificial): nestes cenários, a solicitação de recomendação ocorrerá por um agente tutor inteligente quando não consegue atender a uma solicitação do estudante de resolução de problema;
- (b) Mais detalhes de recurso: este cenário ocorre quando o estudante pede mais informações sobre determinado recurso educacional e o sistema não possui o recurso no ambiente;
- (c) Novo recurso: da mesma forma que o cenário anterior, quando há a requisição de um novo recurso educacional inexistente no ambiente, a solicitação é feita externa ao ambiente.

Desta forma, o agente *ForBILEAgent* possui um mecanismo de comunicação automática quando não atinge determinado objetivo. A vantagem disto é que qualquer extensão feita no ambiente através da adição de algum agente não altera a característica de interoperabilidade do mesmo.

Já o agente mediador possui em seu comportamento um conjunto de regras que analisa se foi feita uma solicitação para comunicação externa. O serviço semântico de busca por ambiente para interoparar retorna uma sêxtupla $\langle agentName, protocol, agentHost, serviceName, inputs, outputs \rangle$. Estas informações são necessárias para garantir que o agente solicitante possa identificar o local que determinado agente se encontra (*agentName* e *agentHost*) e como se comunicar com o mesmo (*serviceName*, *inputs* e *outputs*).

4.4 Guia de Utilização - MASSAYO-P

Esta seção tem por objetivo prover um guia com passos para auxiliar na construção de ambientes educacionais adaptativos e semânticos, levando-se em consideração cada um dos modelos descritos anteriormente, o guia é dividido como segue¹⁶ (vide Figura 4.31):



Figura 4.31: MASSAYO-P: Guia de Utilização do MASSAYO.

Especificação: nesta fase o usuário irá especificar os requisitos e ao final deverá ser

¹⁶Não foi objetivado neste guia, e conseqüentemente nesta tese, descrever um processo de software para a construção de ambientes educacionais adaptativos e semânticos sob a perspectiva do MASSAYO. Este guia preocupa-se apenas com algumas atividades presentes na construção de qualquer tipo de software.

identificado o tipo de ambiente educacional a ser construído, os papéis (com suas funcionalidades) e os tipos de interações usuário-sistema presentes (que posteriormente serão representados através de ferramentas interativas). Pode-se observar que os agentes, serviços semânticos e ontologias não foram descritos. Isto ocorre pelo fato de que os papéis são responsáveis por identificar tanto os tipos de agentes humanos quanto artificiais. Já os serviços semânticos são identificados pelas funcionalidades presentes em cada papel. Por fim, as ontologias do ambiente são padrões e não há a necessidade de descrevê-las;

Modelagem: na fase de modelagem leva-se em consideração o projeto de interface, gráfico, conteúdo, navegacional e arquitetural¹⁷. Para o projeto gráfico, é disponibilizada uma interface padrão, entretanto o time de desenvolvimento está habilitado a fazer alterações no projeto gráfico do ambiente educacional. Já o projeto de interface, conteúdo e navegacional são descritos no ambiente através de ontologias. Por fim, no projeto arquitetural, é definida uma arquitetura padrão em camadas utilizando o padrão arquitetural MVC. Com isso, há a necessidade apenas de especificar os agentes de software que serão utilizados bem como os serviços semânticos que serão implementados e as ferramentas interativas que serão utilizadas (os agentes, serviços e ferramentas interativas são descritos baseados na fase de especificação, descrita anteriormente). Da mesma forma que os outros projetos, as entidades identificadas no projeto arquitetural são descritas através de ontologias. Com isso, os passos presentes de cada ontologia para modelar a aplicação são descritos como segue.

1. Ensinar: este passo equivale à modelagem do domínio através da criação das instâncias nos modelos educacionais descritos anteriormente. Os modelos são descritos abaixo.

- (a) Modelo do Domínio: nesta ontologia são descritas as características do domínio, especificando o domínio e partições do domínio, estrutura curricular e atividades pedagógicas, objetos de aprendizagem a serem utilizados (e.g. exemplos, problemas, conceitos) e descrição do

¹⁷Todas as fases descritas estão presentes na engenharia de software Web [Pressman and Lowe, 2009].

- conhecimento comportamental;
- (b) Modelo do Estudante: este modelo descreve as informações sobre o estudante que serão utilizadas para guiar o ensino de forma personalizada. Com isso, deve-se descrever os objetivos de aprendizagem do estudante para cada currículo e informações que serão utilizadas durante a interação estudante-sistema-estudante;
 - (c) Modelo do Pedagógico: no modelo pedagógico, descreve-se se o ensino será guiado ou livre. Caso a abordagem seja guiada, descrevem-se ainda a estratégia pedagógica, a abordagem a ser utilizada (estática ou dinâmica) e o plano instrucional.
2. Adaptar: a adaptação é a descrição da interface e de suas formas de organização visual (ontologia sobre páginas Web) bem como das adaptações de interface (ontologia do modelo de adaptação e raciocínio).
3. Agentificar: nesta fase os agentes são definidos e então criados dinamicamente pelo ambiente. Os passos são descritos abaixo.
- (a) Regras Modelo do Domínio - GAIA: esta fase equivale ao mapeamento do currículo do modelo do domínio para recursos presentes no papel do tutor na ontologia de GAIA;
 - (b) GAIA: com esta ontologia há a modelagem do sistema multi-agente presente no ambiente educacional;
 - (c) Regras GAIA - GAIA/JADE/ForBILEAgent: com a especificação da ontologia de GAIA, as regras de mapeamento para a ontologia de JADE e *ForBILEAgent* são executadas e deixando a modelagem dos agentes praticamente pronta;
 - (d) *ForBILEAgent*: nesta ontologia, toda a especificação está praticamente pronta depois da execução das regras de mapeamento advindas de GAIA. Desta forma, só há necessidade de especificar as classes *java* de execução dos agentes.

Implementação e testes: a partir do momento que a modelagem foi definida, precisa-se apenas implementar o que não foi reusado pelo MASSAYO. Desta forma, as

implementações que devem ser feitas são das ferramentas interativas que não estão previstas, bem como dos serviços Web semânticos não previstos;

Distribuição: esta fase equivale à distribuição da aplicação. A distribuição possui os seguintes passos:

1. criar o .war da aplicação educacional;
2. instalar o servidor *apache*;
3. colocar a aplicação (.war) no *apache* e rodar;
4. Regras *ForBILEAgent - Application*: as regras são executadas pelo agente de evolução assim que o mesmo é criado. Este mapeamento permite que o ambiente esteja preparado para interoperar.

É importante destacar que o modelo de interoperabilidade não precisa ter passos, pois a comunicação entre ambientes educacionais ocorre de forma automatizada e transparente para a equipe. Na seção 4.3 foram descritas as situações que necessitam de interoperabilidade, proporcionadas pelo MASSAYO.

4.5 Conclusão

Este capítulo abordou os modelos propostos na tese, descrevendo os aspectos presentes e decisões referentes à utilização de cada tecnologia. Além disso, foi descrito um guia para auxiliar o processo de construção de ambientes educacionais adaptativos e semânticos.

Capítulo 5

Avaliação dos Modelos Propostos

*Tempere a si mesmo com mil dias de prática
e refine-se com dez mil dias de treinamento.*

Miyamoto Musashi

Este capítulo tem o propósito de demonstrar a viabilidade dos modelos propostos neste trabalho, considerando-se a utilização do MASSAYO através do desenvolvimento de três estudos de caso: 1) um sistema tutor inteligente em fração; 2) um ambiente de educação a distância adaptativo em matemática; 3) Ambiente colaborativo em Medicina. Além disso, a avaliação de interoperabilidade é feita através dos estudos de caso em matemática, onde ao final da apresentação dos estudos de caso é feita uma discussão sobre o MASSAYO baseada nas implementações, ressaltando-se a adequação e viabilidade dos modelos propostos.

5.1 Estudo de Caso 1 - Sistema Tutor Inteligente em Fração

Esta seção apresenta um Sistema Tutor Inteligente, chamado FraW, para ajudar estudantes, nas séries iniciais do ensino fundamental a aprenderem operações de Fração mediante uma abordagem pedagógica de aprendizagem baseada em resolução de problemas. Este sistema está habilitado a resolver problemas propostos pelos estudantes,

inclusive explicando passo a passo a sua resolução. Além disso, o sistema pode avaliar as soluções apresentadas pelos estudantes, bem como oferecer ajuda nos erros que venham a ocorrer no processo de elaboração da solução [Sibaldo et al., 2008].

As subseções que seguem descrevem a modelagem do problema e da solução do FraW.

5.1.1 Especificação

O sistema possui os módulos tradicionais presentes em um sistema tutor inteligente, a saber, módulo do estudante, pedagógico e aprendiz. A interação do sistema com o estudante ocorre após o cadastramento básico do estudante e com isso o sistema inicia o processo de tutoramento, através do envio de recursos educacionais. As características presentes no ambiente são descritas a seguir:

- O processo de ensino ocorre através de resolução de problemas;
- A solução de um problema é mostrada para o estudante em passos;
- Os problemas podem ser de quatro tipos, sendo eles:
 1. Múltipla Escolha: questões que o estudante pode selecionar uma ou mais respostas corretas;
 2. Relacionar Colunas: questões que são definidas opções para os estudantes relacionarem uma coluna com a outra;
 3. V ou F: questões de verdadeiro ou falso;
 4. Abertas: nestas questões são passadas expressões matemáticas, como por exemplos $(1 + 3/4 * 7 - (3/4^2))$.
- Tanto o estudante pode enviar problemas para o sistema resolver quanto o inverso. Entretanto, o estudante solicita resolução apenas de problemas abertos;
- O estudante pode solicitar dicas durante a resolução de um problema;
- O sistema faz uma avaliação das respostas do estudante e mapeia o nível de conhecimento do mesmo;

- O estudante muda de nível (currículo) quando o mesmo atinge um certo percentual de aprendizagem;
- A interação entre o estudante e o sistema acaba quando o estudante atingiu o nível definido como necessário para aprender.
- O sistema possui um módulo de recomendação de estudantes para auxiliá-lo na resolução de um determinado problema quando o mesmo não está apto a resolver.

Desta forma, com as características fundamentais do Sistema FraW explicitados, abordam-se os casos de usos do sistema. Os casos de uso são requisitos de software (funcionais e/ou comportamentais) que indicam o que o sistema deve fazer/fornecer. Estes casos de uso definem as interações entre os atores e o sistema (maiores detalhes em [Alhir, 2003] e/ou [Larman, 2004]). O diagrama de casos de uso do FraW pode ser visto na Figura 5.1.

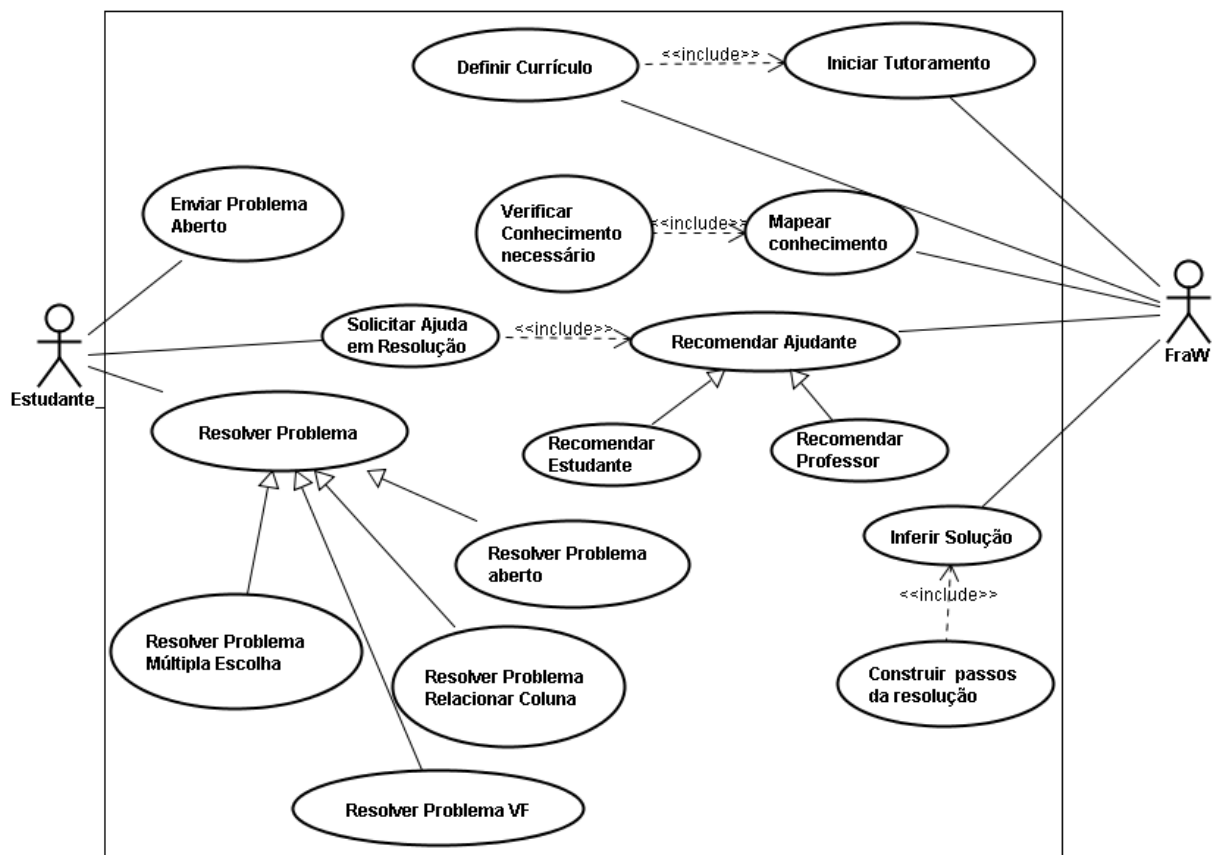


Figura 5.1: Diagrama de casos de uso do FraW.

A partir do momento em que os casos de uso foram definidos, passa-se então a construção do diagrama de classes conceitual. Um diagrama de classes conceitual representa os conceitos do domínio em estudo e é uma perspectiva destinada ao cliente [Larman, 2004]. Este diagrama é descrito na Figura 5.2.

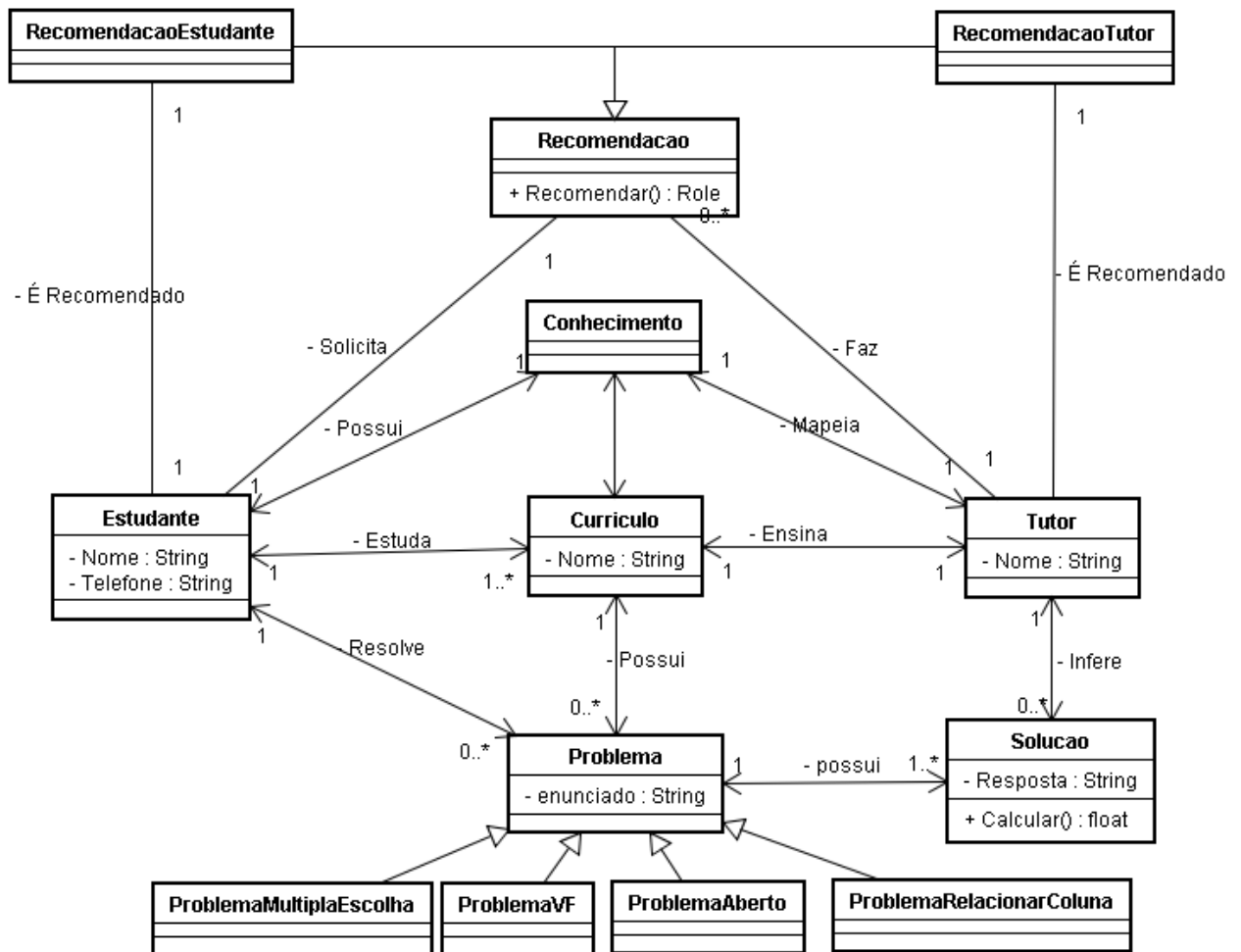


Figura 5.2: Diagrama de classes conceitual do FraW.

A partir do momento em que o diagrama de classes conceitual foi definido, pode-se identificar as ferramentas, serviços e entidades presentes no sistema, como descritas a seguir.

- Papel de Estudante: presente para a interação com o sistema equivale ao de estudante;
- Ferramenta de Especificação de Problemas: provê o estudante com uma ferramenta onde o mesmo pode especificar problemas de frações para que o sistema resolva;

- Ferramenta de Visualização de recursos: garante a visualização dos diversos recursos educacionais solicitados pelo estudante ou recomendados pelo sistema. Estes recursos podem ser conteúdo, exemplos, contra-exemplos, problemas (de múltipla escolha, relacionar colunas, verdadeiro ou falso e abertos). Além disso, tais recursos podem estar definidos em *pdf*, *html*, *doc* e *flash*;
- Serviços: o sistema também deve ser capaz de disponibilizar serviços, dentre os quais são citados os principais:
 - Gerenciamento do tutoramento: serviço responsável por gerenciar as atividades de tutoramento dos estudantes;
 - Resolução de Problemas: estes serviços representam atividades de resolução dos diversos tipos de problemas presentes no ambiente;
 - Explicação de Problema: a explicação do problema ocorre de acordo com o tipo de problema e com o percentual de acerto do estudante;
 - Mapeamento de nível: este serviço calcula o novo nível de conhecimento do estudante de acordo com a resolução de determinado problema;
 - Raciocínio baseado em Regras: este serviço infere que tipo de solução deve ser adotada para resolver um determinado problema;
 - Recomendação de Recursos Educacionais: a recomendação é feita com o intuito de disponibilizar para o estudante diversos recursos educacionais que auxiliem no processo de construção de seu conhecimento;
 - Recomendação de Estudante: este serviço verifica os estudantes que já possui um nível mais avançado para poder recomendá-lo;
 - Recomendação de Professor: este serviço verifica o professor de determinada área do conhecimento e que esteja disponível para ser recomendado.

É importante frisar que o MASSAYO já disponibiliza os serviços de mapeamento de nível, raciocínio baseado em regras, recomendação de recursos educacionais, ferramenta de visualização de recursos. Serviços e ferramentas não previstas no MASSAYO devem ser implementados. O modelo conceitual é mostrado na Figura 5.3

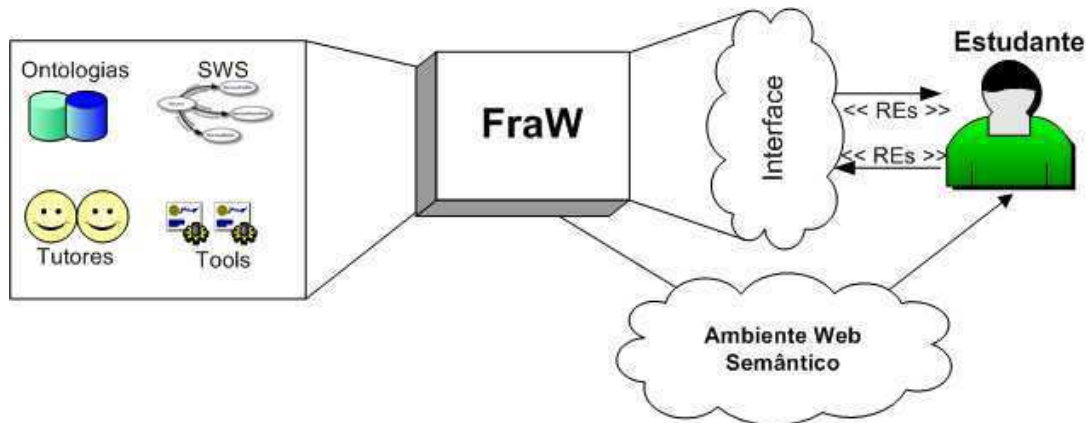


Figura 5.3: Modelo Conceitual do FraW, segundo MASSAYO-RM.

Observa-se que no modelo conceitual pode-se identificar os tipos de recursos e entidades presentes no ambiente educacional FraW, bem como os papéis que estão envolvidos no sistema.

5.1.2 Modelagem

Esta seção descreve como o sistema tutor inteligente pode ser construído, utilizando-se do MASSAYO. A partir do momento que a modelagem do problema foi definida, busca-se identificar as entidades que estão presentes em tal ambiente, seguindo a perspectiva do guia de utilização do MASSAYO descrita na Subseção 4.4, do Capítulo 4.

Desta forma, a modelagem está preocupada com aspectos arquiteturais, interface, navegacional e conteúdo. Cada aspecto é descrito nas subseções que seguem.

Projeto Arquitetural

Este projeto segue uma arquitetura tradicional de sistemas tutores inteligentes, com os módulos sendo descritos na Figura 5.4

As características de cada módulo da arquitetura são descritas como seguem [Sibaldo et al., 2008]:

- **Módulo do Domínio:** neste módulo estão as informações a respeito do que será passado para os alunos. No trabalho aqui apresentado o domínio corresponde a assuntos relacionados a frações com números inteiros. O FraW adota uma

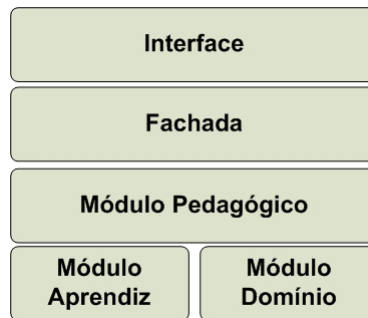


Figura 5.4: Arquitetura do FraW.

abordagem pedagógica de aprendizagem baseada em resolução de problemas, onde os problemas são expressões que podem conter as seguintes operações: soma, subtração, multiplicação, divisão, potenciação e radiciação. A tabela 5.1 descreve as informações presentes no módulo do domínio.

Tabela 5.1: Dados do Domínio de Matemática - Fração

Classe	Conteúdo
Domínio	Matemática
LearningResource(Content)	Introdução
Subdomínio	Fração
Currículo ₁	Operações
Unidade Pedagógica ₁	Geral
Problema ₁	Fácil
Problema ₂	Médio
Problema ₃	Difícil
...	...
Currículo ₁₁	Soma
Unidade Pedagógica ₁	Simples
Problema ₁	Fácil
...	...
Currículo ₁₄	Divisão
...	...
Pré-requisito ₁	Mínimo Múltiplo Comum
...	...

- **Módulo do Aprendiz:** neste módulo, encontram-se informações sobre o aluno, referindo-se ao seu aprendizado. Podem-se encontrar as questões por ele

resolvidas, os erros cometidos, os acertos, suas dificuldades em determinados tópicos, entre outras informações pessoais e estudantis. Este modelo caracteriza individualmente cada estudante no seu processo de aprendizado. Na Tabela 5.2 segue o conhecimento que deve ser definido com relação aos aspectos pedagógicos.

Tabela 5.2: Dados do Módulo Pedagógico - FraW

Classe	Conteúdo
<i>Strategy₁</i>	<i>Problem-based Learning</i>
<i>LearningGoal₁</i>	Currículo ₁ (Operações)
<i>levelKnowledge</i>	<i>Advanced</i>
<i>rate</i>	<i>0.80</i>
<i>LearningGoal₁₁</i>	Currículo ₁₁ (Soma)
<i>LevelKnowledge</i>	<i>Advanced</i>
<i>rate</i>	<i>0.80</i>
...	...
<i>LearningGoal₁₂</i>	Currículo ₁₂ (Subtração)
<i>LevelKnowledge</i>	<i>Advanced</i>
<i>rate</i>	<i>0.80</i>
...	...

- Módulo Pedagógico: este módulo faz o gerenciamento do sistema tutor inteligente, tendo a função de coordenar o funcionamento do sistema. Além disso, o mesmo contém estratégias e táticas de ensino, seleciona o conteúdo a ser apresentado ao aluno, monitora, critica o desempenho do aluno e fornece assistência quando solicitado. Quando uma questão for proposta para que o estudante a resolva, a interface recebe esta questão através do Módulo Pedagógico que por sua vez obtém do Módulo do Domínio. O Módulo Pedagógico comunica-se diretamente tanto com o Módulo do Domínio quanto com o Modelo do Aprendiz. Essa última comunicação ocorre, por exemplo, quando o usuário solicita a visualização do seu estado de aprendizagem. Desta forma, este módulo deve manter informações sobre a estratégia pedagógica adotada. Esta estratégia pedagógica adotada equivale a Aprendizagem Baseada em Problemas¹[Yeo et al., 2006]. A partir do momento que

¹Do Inglês, Problem-Based Learning

a estratégia pedagógica é definida, deve-se definir os pesos que fazem com que o estudante mude de um currículo para outro.

Projeto Navegacional

A camada navegacional representa as telas de comunicação bem como o esquema de navegação entre o FraW e seus aprendizes. Sendo esta uma interface *Web*, o aluno poderá acessar o sistema em qualquer computador que tenha acesso à Internet. Desta forma, o sistema tem a necessidade de disponibilizar uma ferramenta que o estudante possa definir um problema para ser enviado e resolvido pelo sistema. Além disso, há a necessidade de disponibilizar uma instrução guiada pelo sistema tutor inteligente. Com isso, a Figura 5.5 descreve o diagrama Web do sistema FraW.

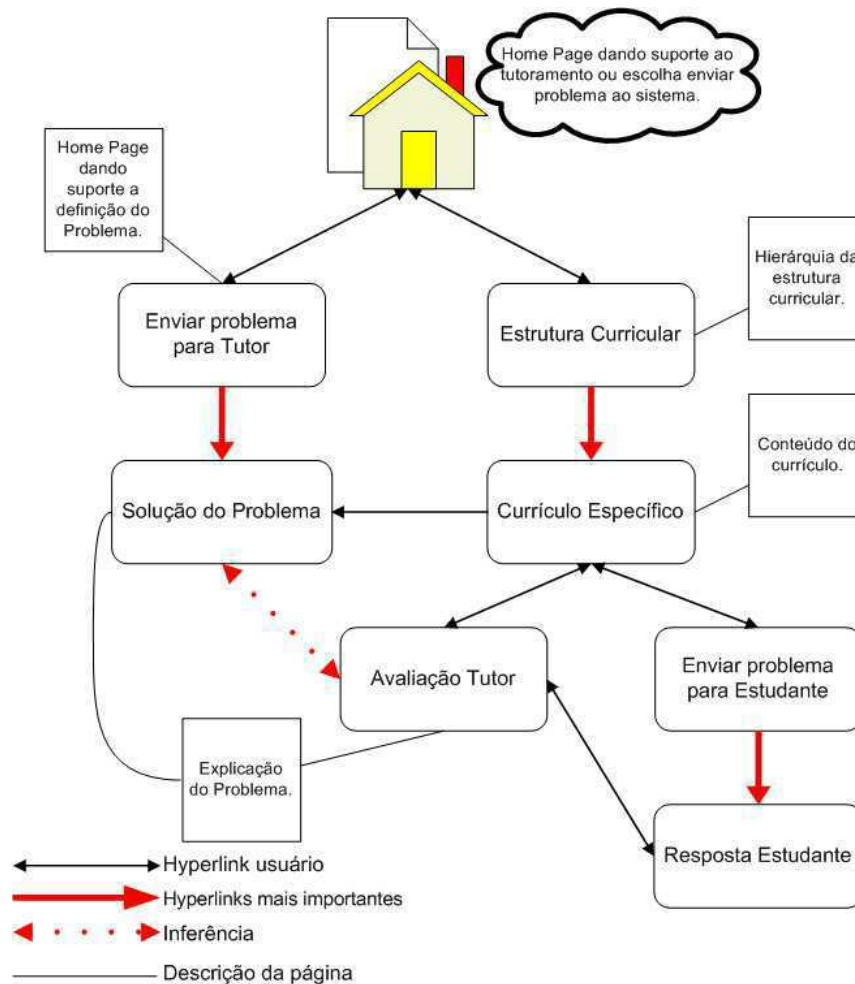


Figura 5.5: Diagrama Web do sistema FraW.

As sete telas presentes na interface são:

- Home Page: página inicial do sistema, onde o estudante se conecta;
- Estrutura Curricular: tela onde é guiado o processo de ensino-aprendizagem. O currículo específico está presente na mesma tela;
- Enviar problema para estudante: esta tela tem um problema para o estudante responder;
- Resposta estudante: para cada problema, o estudante define a solução dada;
- Avaliação do tutor: esta tela mostra a solução do estudante e avaliação do sistema tutor. A avaliação ocorre através da comparação com as soluções presentes no problema;
- Enviar problema ao tutor: nesta tela o estudante descreve a expressão que deverá ser respondida pelo sistema;
- Solução do problema: a solução do problema se dá através da execução do algoritmo de resolução de problemas seguido de sua explanação.

Projeto de Especificação Ontológica

Esta fase representa o projeto de especificação baseado nas ontologias providas pelo ambiente, sendo elas de ensino, adaptação, agentificação e interoperação. Cada uma delas é descrita abaixo.

1. Ontologia do Domínio: a especificação do domínio se dá através da criação de instâncias para as classes de domínio, currículos, unidades pedagógicas e recursos educacionais.

As Figuras 5.6 e 5.7 descrevem instâncias do modelo do domínio. Ressalta-se que foram geradas 53 instâncias dos recursos, sendo um total de 10 conceitos, 5 textos, 14 exercícios, 18 explicações e 6 problemas.

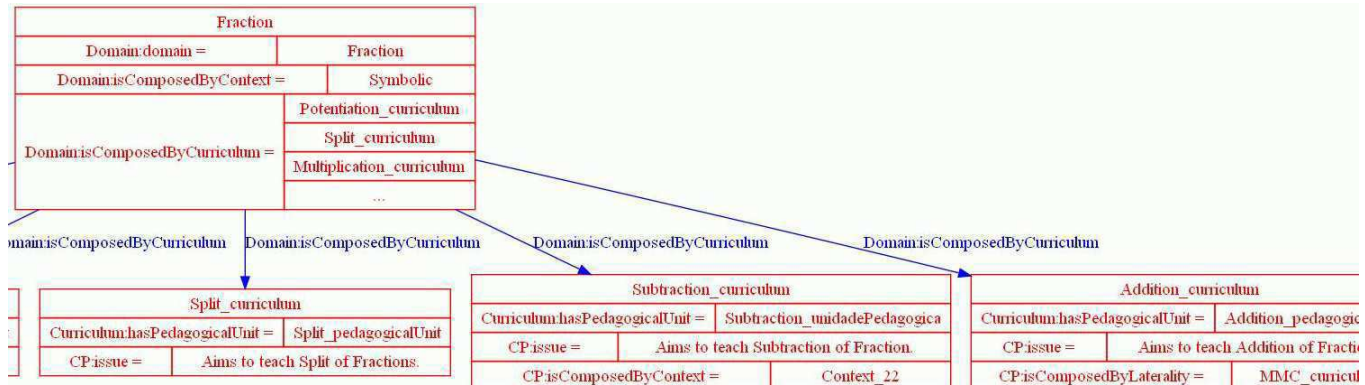


Figura 5.6: Ontologia descrevendo os currículos.

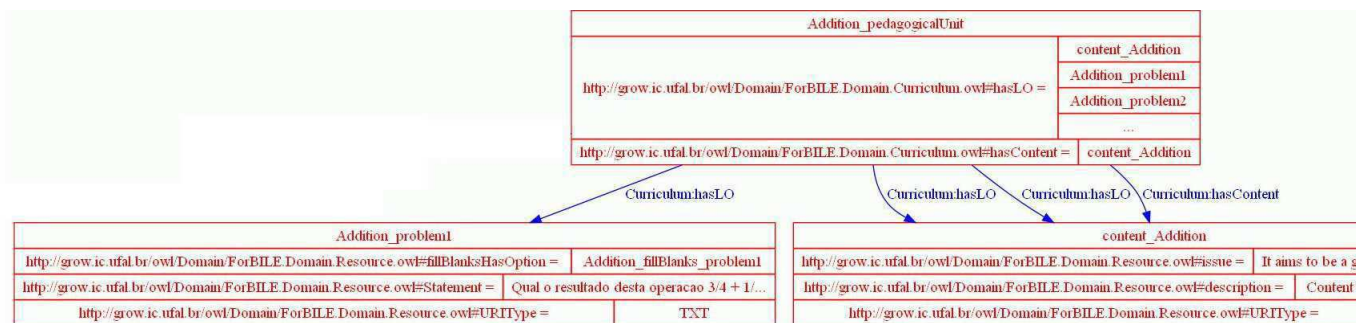


Figura 5.7: Parte da Ontologia descrevendo os recursos presentes em um currículo.

2. Ontologia do Estudante: nesta ontologia são especificadas basicamente as informações sobre o nível de conhecimento esperado que os estudantes devam obter sobre cada currículo presente no modelo do domínio. A Figura 5.8 descreve a interação de um usuário com o sistema.

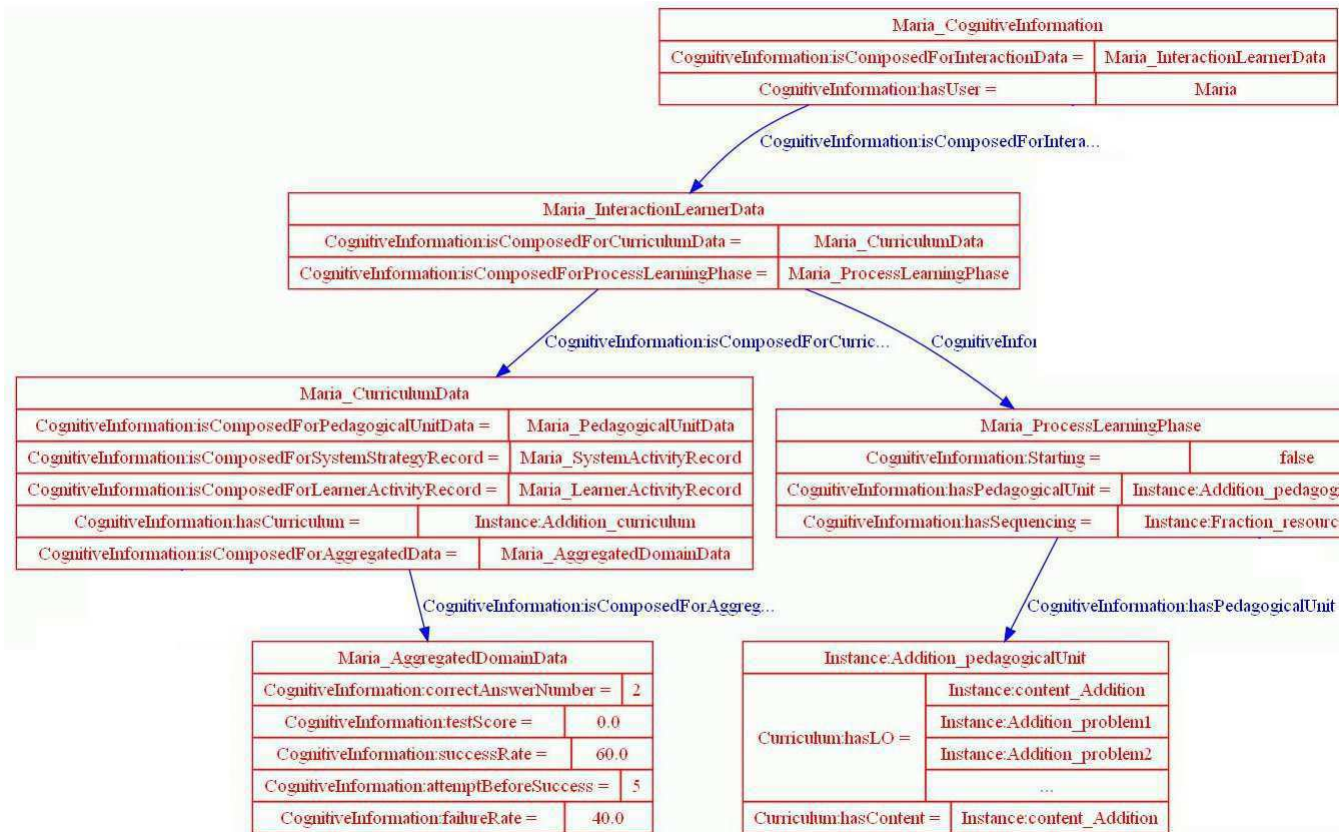


Figura 5.8: Parte da Ontologia descrevendo a interação estudante-sistema-estudante.

3. Ontologia Pedagógica: além da definição da estratégia pedagógica presente no ambiente, é descrito também o plano instrucional do mesmo. As instâncias do plano instrucional são mostradas na Figura 5.9.

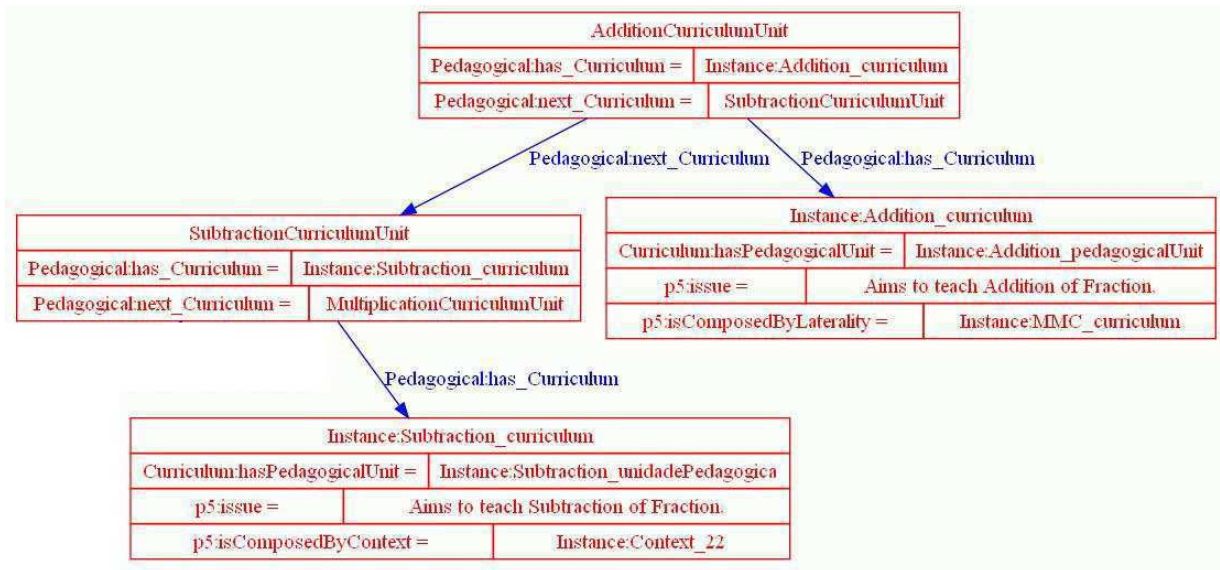


Figura 5.9: Parte da Ontologia com estrutura curricular.

- Regras de mapeamento Domínio-GAIA: a partir do momento que o domínio foi especificado, há a necessidade de executar as regras de mapeamento do domínio para a ontologia de GAIA. Com as regras executadas, os recursos do papel de tutor são definidos, como descritos na Figura 5.10;

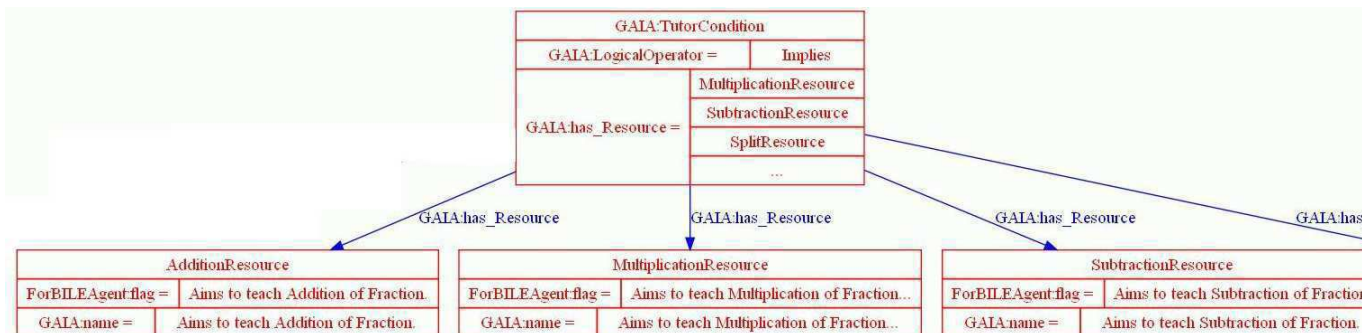


Figura 5.10: Parte da Ontologia GAIA com os recursos gerados a partir das regras de mapeamento.

- Ontologia de GAIA: esta ontologia possuirá a especificação dos agentes presentes no sistema tutor inteligente (vide Figura 5.11). Com isso, além dos agentes tutores que foram definidos de acordo com a ontologia do domínio e a execução das regras de mapeamento, há também os agentes controlador, mediador e de evolução, que são inatos ao MASSAYO. Outros agentes devem também ser especificados.

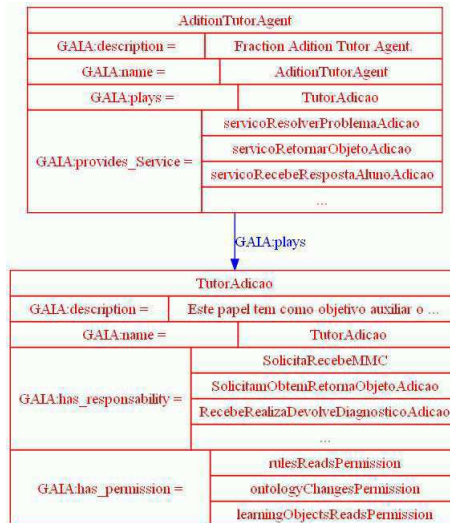


Figura 5.11: Parte da Ontologia de GAIA.

6. Regras de Mapeamento GAIA-JADE/ForBILEAgent: com as instâncias criadas na ontologia de GAIA, pode-se agora mapear as regras de GAIA para um modelo de construção e execução de agentes (ontologia de *ForBILEAgent*). O Código 5.1 descreve uma regra de mapeamento GAIA-JADE/ForBILEAgent;

```

1 GAIA:Resource(?ForBILEAgent:resource) ^
2 GAIA:has_Resource(GAIA:TutorCondition, ?ForBILEAgent:resource) ^
3 abox:hasNumberOfPropertyValue(0, ?ForBILEAgent:resource, ForBILEAgent:flag) ^
4 GAIA:name(?ForBILEAgent:resource, ?ForBILEAgent:resourceName) ^
5 GAIA:name(GAIA:Tutor_Agent, ?ForBILEAgent:tutorName) ^
6 swrlx:createOWLThing(?ForBILEAgent:agent, ?ForBILEAgent:resource) ^
7 swrlx:createOWLThing(?ForBILEAgent:aid, ?ForBILEAgent:resource) ^
8 swrlx:createOWLThing(?ForBILEAgent:services, ?ForBILEAgent:resource)
9 --> ForBILEAgent:TutorAgent(ForBILEAgent:agent) ^
10 JADE:className(?ForBILEAgent:agent, "br.ufal.ic.forbile.agent.ForBILEAgent") ^
11 JADE:AID(?ForBILEAgent:aid) ^
12 JADE:has_AID(?ForBILEAgent:agent, ?ForBILEAgent:aid) ^
13 JADE:is_ContainedContainer(?ForBILEAgent:agent, ForBILEAgent:TutorContainer) ^
14 JADE:Service(?ForBILEAgent:service) ^
15 JADE:serviceName(?ForBILEAgent:service, "Tutoring") ^
16 JADE:serviceType(?ForBILEAgent:service, ?ForBILEAgent:resourceName) ^
17 JADE:isOfferedBy_JADEAgent(?ForBILEAgent:service, ?ForBILEAgent:agent) ^
18 ForBILEAgent:type(?ForBILEAgent:agent, ?ForBILEAgent:tutorName)

```

Código Anotado 5.1: Regra de mapeamento GAIA-JADE/ForBILEAgent no FraW.

7. Ontologia de *ForBILEAgent*: com as regras de mapeamento executadas, faz-se necessário apenas a configuração de algumas propriedades caso a equipe considere

conveniente, como o tempo de percepção do ambiente pelos agentes e a classe de execução de agentes. A Figura 5.12 descreve parte da ontologia de *ForBILEAgent*;

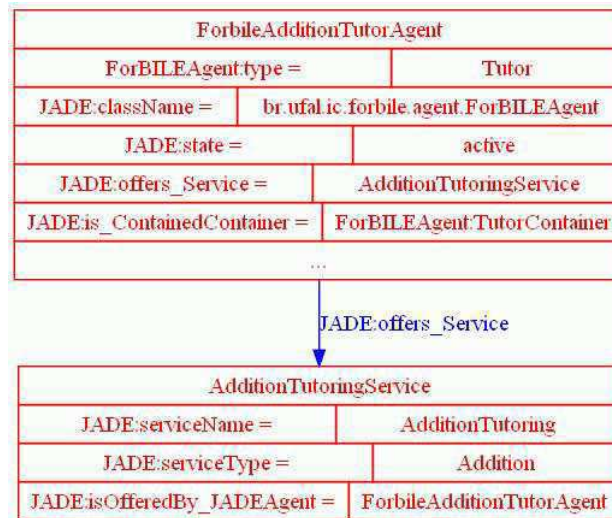


Figura 5.12: Parte da Ontologia ForBILEAgent.

5.1.3 Implementação e testes

Com a modelagem do FraW definida, há a necessidade ainda de implementar os serviços e ferramentas que não foram disponibilizados pelo MASSAYO. Logo, há a necessidade de implementar os serviços de resolução de problemas abertos e explicação, as ferramentas de especificação e explicação de problemas.

Adição de Novos Requisitos

Os novos requisitos que foram adicionados foram de resolução de problemas abertos e recomendação. Para que os novos requisitos sejam adicionados ao ambiente, há a necessidade de fazer os seguintes passos:

1. Implementar o mecanismo de resolução de problemas abertos: o mecanismo de resolução de problema é baseado em regras. Como o ambiente já provê tal mecanismo, há a necessidade apenas de definir as regras a serem utilizadas pelo ambiente e então usá-las. Após a implementação deste mecanismo, o mesmo é encapsulado em um serviço Web semântico;

2. Implementar Recomendação de Estudante e Professor: as duas formas de recomendação são bastante simples, tendo que levar em consideração apenas os dados presentes na ontologia sobre os mesmos. Além disso, não há a necessidade de adicionar nenhuma informação na ontologia, ou seja, basta implementar os dois serviços levando-se em consideração os dados presentes nas ontologias;
3. Transformar em SWS: esta transformação envolve a criação de um serviço Web tradicional e logo após a transformação do mesmo em serviço semântico². Após a transformação, há a necessidade de implantar o serviço para que possa ser descoberto pelo gerenciador de serviço.

É importante destacar que não houve a necessidade de adicionar novos agentes, pois na especificação já foram previstas as implementações dos novos requisitos, isto é, a partir do momento que os requisitos são implementados e transformados para serviços Web semânticos, os serviços são automaticamente descobertos pelos agentes previamente especificados.

Implementar Interfaces

Finalmente, há a necessidade de implementar as interfaces que irão permitir tanto a visualização da resolução de problemas quanto a interface de especificação de um problema. Como o modelo já disponibiliza uma interface padrão, diversos aspectos que deveriam ser considerados foram poupados, como telas de cadastro, *login*, visualização de recursos, entre outros. Desta forma, as interfaces implementadas foram:

- Especificação de problemas: permite que o estudante envie problemas ao tutor. Para que a especificação do problema ficasse mais fácil de ser definida, foi criada uma interface onde o estudante define os tipos de operações presentes em uma determinada expressão;
- Explicação do problema: permite que a solução de um determinado problema seja visualizada pelo estudante. Além disso, a solução do problema é seguida da

²Para transformar um serviço tradicional em serviço semântico, deve-se criar as ontologias baseadas no wsdl e relacionar com a ontologia sobre o domínio tratado no serviço.

explicação detalhada do mesmo. Neste caso, o estudante poderá verificar como ocorre a resolução de um problema e a explicação, como mostram as figuras a seguir.

Figura 5.13: Passos 1 e 2 de resolução de problemas pelo FraW.

Figura 5.13: Passos 1 e 2 de resolução de problemas pelo FraW.

Figura 5.14: Passos 3 e 4 de resolução de problemas pelo FraW.

Figura 5.14: Passos 3 e 4 de resolução de problemas pelo FraW.

© 2009 Massiao - Todos os direitos reservados

ANY Learning Time Where Body

Figura 5.15: Último passo de resolução de problemas pelo FraW.

- Resolução Passo a Passo: possibilita que o estudante faça a resolução do problema passo a passo. À medida que o estudante vai resolvendo o sistema vai verificando como está a resolução de cada passo e informa se está correto ou não. As telas com um exemplo de resolução passo a passo são descritas abaixo.

© 2009 Massiao - Todos os direitos reservados

ANY Learning Time Where Body

Figura 5.16: Passos 1 e 2 de resolução de problemas no FraW pelo estudante.



Figura 5.17: Passos 3 e 4 de resolução de problemas no FraW pelo estudante.



Figura 5.18: Último passo de resolução de problemas no FraW pelo estudante.

As outras interfaces não precisaram ser desenvolvidas, pois já estavam previstas no modelo.

5.2 Estudo de Caso 2 - Ambiente de Educação a Distância em MMC

Esta seção descreve um ambiente de educação a distância adaptativo para o domínio de Mínimo Múltiplo Comum (MMC). O objetivo é fazer com que os estudantes interajam de modo a garantir o aprendizado neste subdomínio da matemática. A idéia básica neste ambiente é que alunos e professores interajam através de ferramenta interativas, tanto síncronas quanto assíncronas, além de permitir a interação entre os próprios estudantes.

Assim, as subseções que seguem descrevem tanto a modelagem do problema quando a modelagem e implementação da solução.

5.2.1 Especificação

O sistema em questão possui os módulos tradicionais de um ambiente de educação a distância, ou seja, ferramentas de interação e módulo de avaliação. A interação no ambiente se inicia após o cadastramento dos estudantes no ambiente e o adcionamento dos mesmos no curso de MMC. Com isso, as características presentes no ambiente são:

- O sistema possui as ferramentas *chat*, fórum, enquete, calendário, *wiki*, biblioteca digital, atividades e ferramenta de resolução de problemas;
- Os recursos educacionais disponibilizados no ambiente são problemas, exemplos, conteúdos e explicação de um problema;
- O processo de ensino ocorre através do desenvolvimento de atividades. Com isso, as atividades podem ser enviadas através das ferramentas de *wiki*, *chat*, fórum, e resolução de problemas;
- Os problemas podem ser de cinco tipos:
 1. Múltipla escolha;
 2. Relacionar Colunas;
 3. Preencher Lacunas;
 4. V ou F;
 5. Questões Abertas.
- Além da possibilidade do sistema enviar problemas para o estudante, o mesmo pode ocorrer por parte dos estudantes. Ou seja, o envio de problemas é bidirecional;
- Os estudantes podem fazer atividades tanto em grupo quanto individualmente;

- A avaliação ocorre tanto por parte do professor, quanto pelo auxílio do sistema. O sistema é responsável por avaliar o estudante nas atividades de resolução de problemas;
- O gerenciamento de atividades é feito através de sistemas tutores inteligentes.

Desta forma, com as características fundamentais do sistema descritas, descreve-se na Figura 5.19 o diagrama de casos de uso do ambiente de MMC.

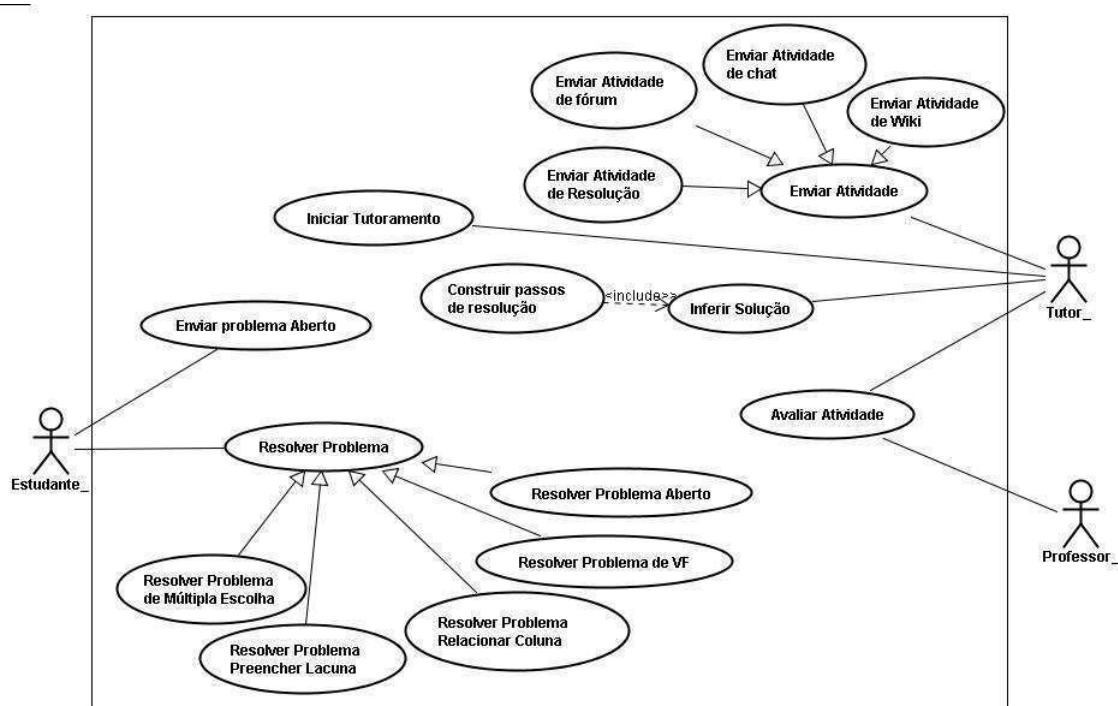


Figura 5.19: Diagrama de Casos de Uso do Ambiente de EaD Adaptativo de MMC.

O diagrama de caso de uso aborda a interação entre três diferentes papéis, sendo eles Estudante, Tutor e Professor. Entretanto, o papel do Tutor é representado pelo próprio sistema educacional a partir da utilização de sistemas tutores inteligentes. Frisa-se ainda que a maior carga de atividades a serem desempenhadas é representada pelo tutor. Desta forma, percebe-se claramente que há automatização de diversas atividades do ambiente. Após o caso de uso, apresenta-se o diagrama de classes conceitual (Vide Figura 5.20) do ambiente de EaD Adaptativo de MMC.

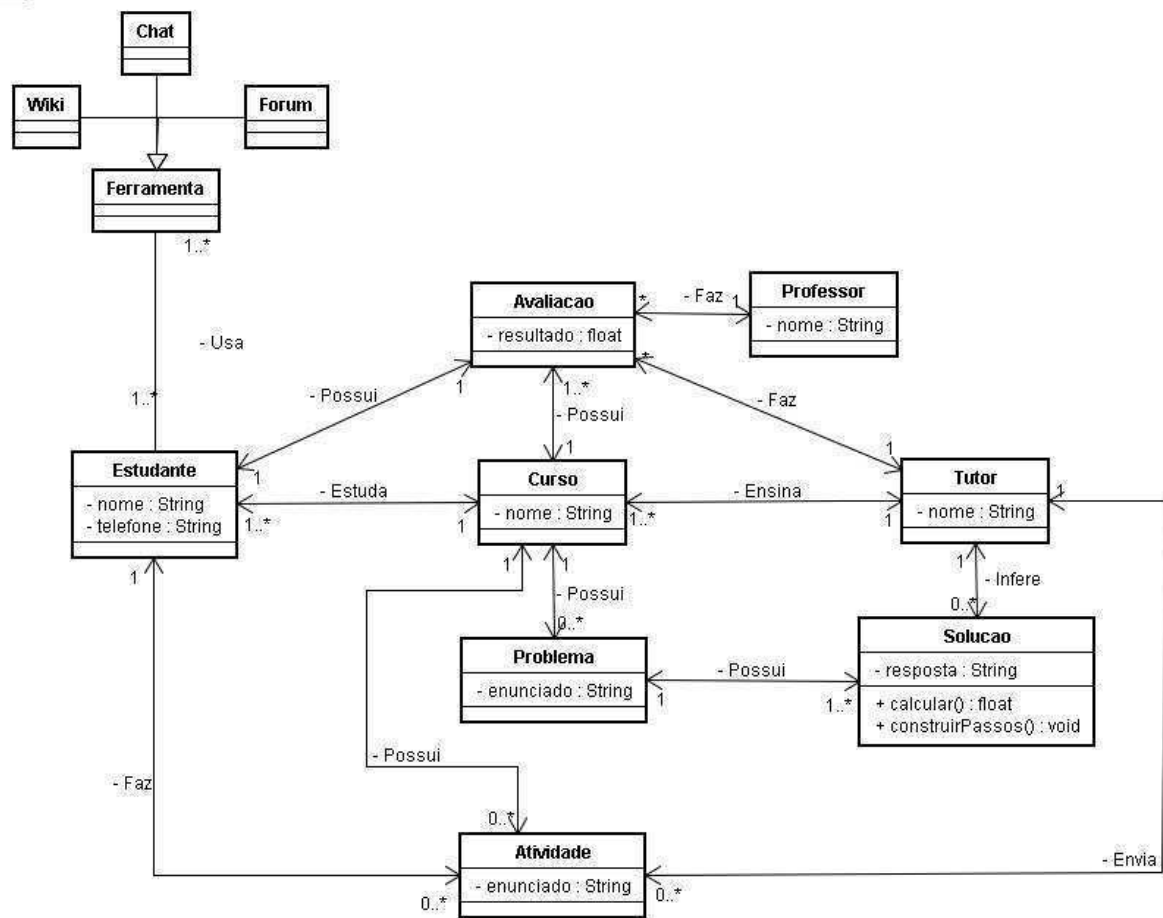


Figura 5.20: Diagrama de Classes Conceitual do Ambiente de EaD Adaptativo de MMC.

Com a especificação definida, identificam-se as entidades presentes no ambiente de acordo com o modelo conceitual descrito na Subseção 4.1. Com isso, as entidades e papéis presentes são:

- Papel do Estudante: interage com outros estudantes, bem como com o professor e o próprio sistema;
- Papel do Professor: define os recursos educacionais presentes no ambiente, bem como a avaliação final do estudante;
- Papel do tutor: auxilia o professor no processo de ensino-aprendizagem;
- Ferramentas de *chat*, fórum, enquete, calendário, *wiki*, biblioteca digital, atividades e ferramenta de resolução de problemas;

- Ferramenta de Visualização de recursos: além das ferramentas descritas anteriormente, há a necessidade de disponibilizar uma ferramenta para descrever os recursos (problemas, exemplos, etc) presentes no ambiente;
- Serviços: os principais serviços presentes no sistema são:
 - Resolução de problemas: estes serviços representam atividades de resolução dos diversos tipos de problemas presentes no sistema;
 - Explicação do problema: a explicação do problema ocorre de acordo com o tipo de problema e com o percentual de acerto do estudante;
 - Avaliação: este serviço calcula o novo nível de conhecimento do estudante de acordo com a resolução de determinado problema³;

Dentre as ferramentas e serviços descritos anteriormente, o modelo disponibiliza as ferramentas de *chat*, fórum, enquete, calendário, *wiki*, biblioteca digital, atividades, visualização de recursos e os serviços de mapeamento de nível. Além disso, o modelo disponibiliza o módulo de resolução de problemas e explicação para todos os tipos de problemas, exceto os problemas abertos.

5.2.2 Modelagem

Esta seção descreve como o ambiente de educação a distância adaptativo pode ser construído baseado no MASSAYO. Da mesma forma que o estudo de caso anterior, descreve-se nas subseções que seguem a modelagem do ambiente.

Projeto Arquitetural

Apresenta-se aqui a arquitetura do sistema baseada no diagrama de classes conceitual e no diagrama de casos de uso apresentados na especificação do ambiente. Desta forma, a Figura 5.21 descreve a arquitetura, apresentando as ferramentas presentes no ambiente de educação distância adaptativo bem como outros módulos fundamentais para o seu funcionamento.

³É importante notar que este serviço é bastante similar ao serviço de mapeamento de nível do estudante, descrito no sistema FraW.

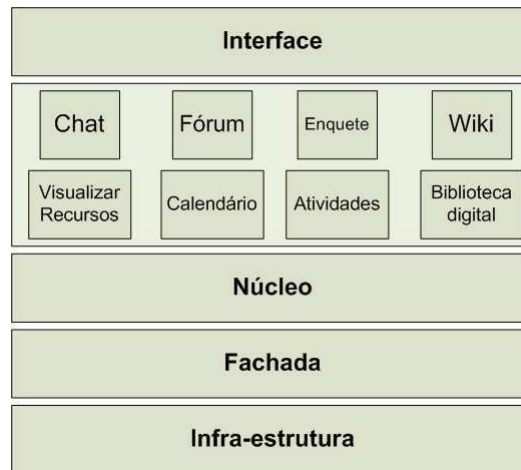


Figura 5.21: Arquitetura descrevendo as ferramentas presentes no Ambiente de EaD Adaptativo de MMC.

Abaixo são descritos cada uma dos módulos presentes na arquitetura do ambiente de MMC.

- Ferramentas: todas as ferramentas, com exceção da ferramenta de visualização de recursos, seguem um modelo tradicional presente em qualquer ambiente de educação a distância;
- Ferramenta de Visualizar Recursos: esta ferramenta permite a visualização de recursos em *flash*, *pdf*, *doc*, *txt* e *html*;
- Núcleo: este módulo possui os requisitos presentes do tutor, como gerenciamento de atividades, resolução de problemas, dentre outros.

Projeto Navegacional

A interface apresenta para os usuários (estudantes e professores) tanto as ferramentas quanto os recursos disponíveis no ambiente educacional. O Diagrama Web é mostrado na Figura 5.22 descrevendo aspectos navegacionais do ambiente educacional.

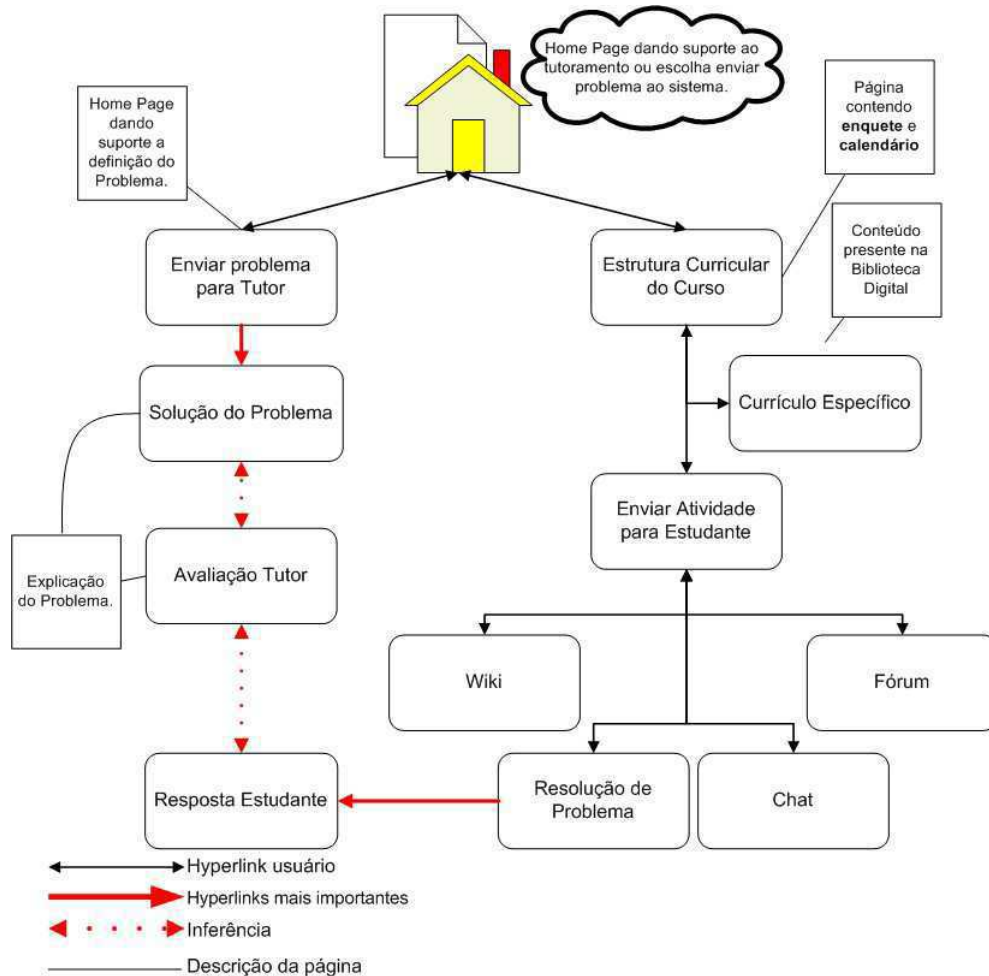


Figura 5.22: Diagrama Web do ambiente de Ead Adaptativo de MMC.

Com isso, as 13 telas presentes no ambiente são descritas a seguir:

- Home Page: página inicial do sistema, onde o estudante se conecta;
- Estrutura Curricular do Curso: é através desta tela que o estudante poderá acessar os recursos educacionais (disponibilizados através da tela de *Currículo Específico*) ou verificar as atividades (disponibilizadas através da tela *Enviar Atividade para Estudante*). Além disso, é através desta tela que o estudante tem acesso ao calendário bem como a enquete;
- Currículo Específico: nesta tela o estudante possui acesso aos diversos recursos educacionais disponibilizados tanto pelo professor quanto pelos sistemas tutores inteligentes. Tais recursos educacionais estão disponibilizados na biblioteca digital presente no ambiente;

- Enviar Atividade para Estudante: aqui o estudante é informado sobre as diversas atividades presentes para serem feitas bem como as atividades que já passaram do prazo de desenvolvimento da mesma. Através desta tela é possível ir diretamente para a tela da atividade específica, como descrita abaixo.
 - Fórum: tela com atividades para serem feitas de forma assíncrona através da utilização de fórum;
 - Wiki: nesta tela, a atividade deve ser desenvolvida em grupo;
 - Chat: esta tela equivale a uma atividade para ser desenvolvida de forma síncrona e através da interação com outros estudantes;
 - Resolução de Problema: esta atividade é feita individualmente, onde o aluno terá que resolver um ou mais problemas.
 - * Resposta Estudante: para cada problema, o estudante define a solução dada;
- Enviar Problema para Tutor: nesta tela o estudante descreve a expressão que deverá ser respondida pelo sistema;
- Solução do Problema: a solução do problema se dá através da execução do algoritmo de resolução de problemas seguido de sua explanação.
- Avaliação do Tutor: esta tela mostra a solução do estudante e avaliação do sistema tutor. A avaliação ocorre através da comparação com as soluções presentes no problema;

Projeto de Especificação Ontológica

Abaixo descreve-se a especificação do ambiente através da instanciação das ontologias.

- Ontologia do Domínio: descrevem-se as características do domínio de MMC, como mostrado na Figura 5.23.

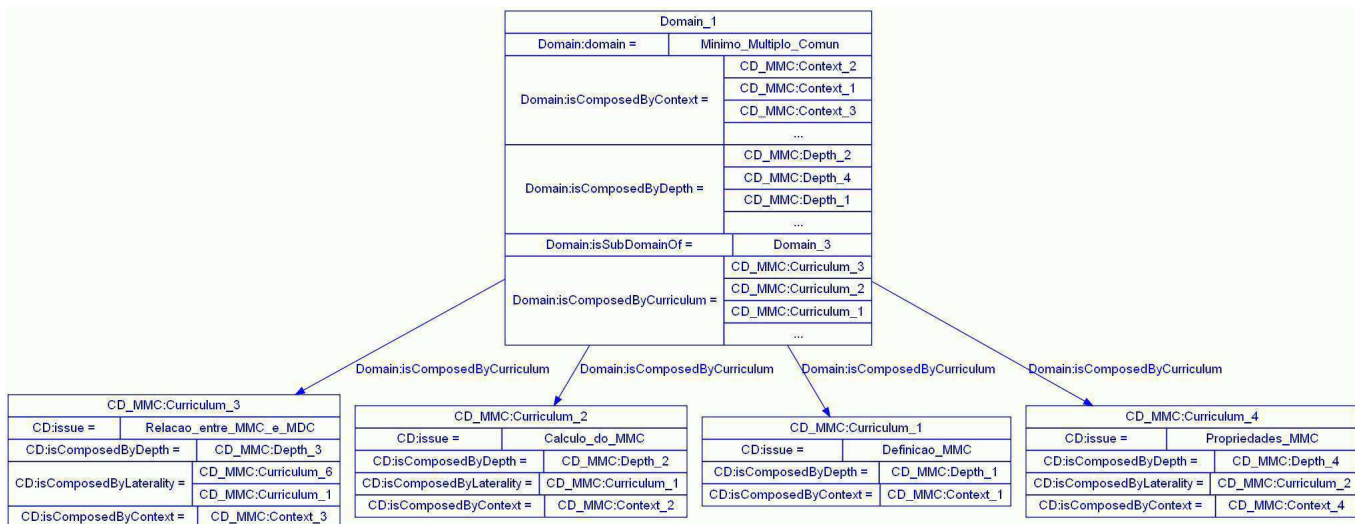


Figura 5.23: Parte da ontologia do domínio com os currículos de MMC.

Além disso, a Figura 5.24 descreve informações referentes a um currículo presente no modelo do domínio de MMC, bem como sua unidade pedagógica e recursos presentes na mesma.

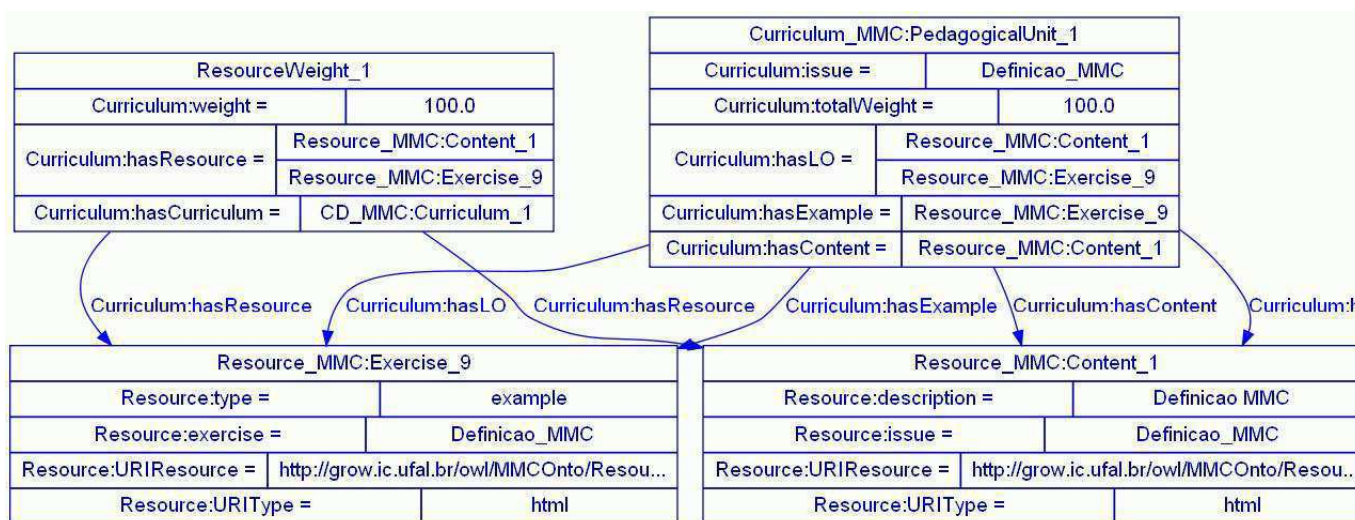


Figura 5.24: Parte da ontologia contendo uma unidade pedagógica com os seus recursos.

- Ontologia do Estudante: informações tanto estáticas quanto sobre a interação do estudante com o sistema. A Figura 5.25 contém informação cognitiva para um determinado usuário do sistema.

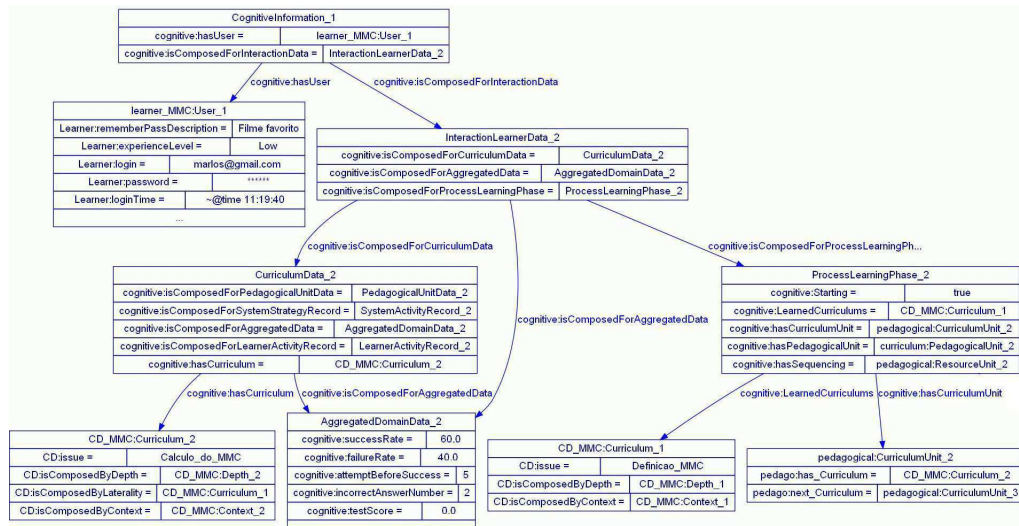


Figura 5.25: Parte da ontologia do estudante com informação cognitiva para um usuário.

- **Ontologia Pedagógica:** descreve o modelo instrucional bem como a abordagem pedagógica do ambiente. As Figuras 5.26 e 5.27 descrevem a estrutura de sequenciamento do tutor.

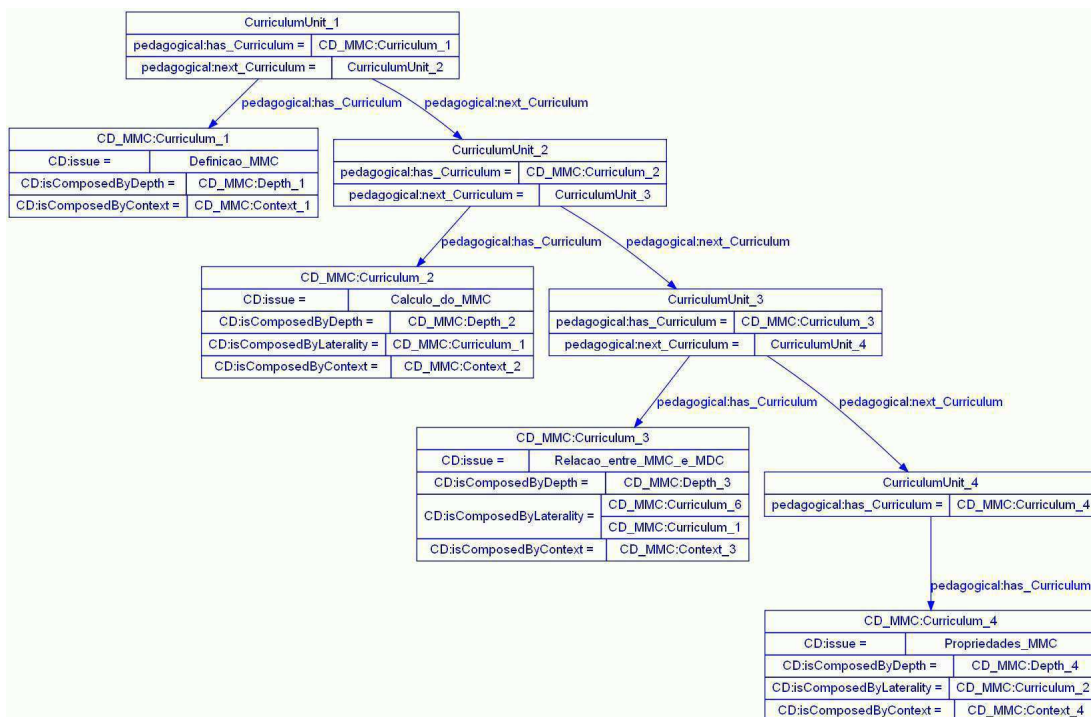


Figura 5.26: Parte da ontologia pedagógica contendo o sequenciamento do sistema (a).

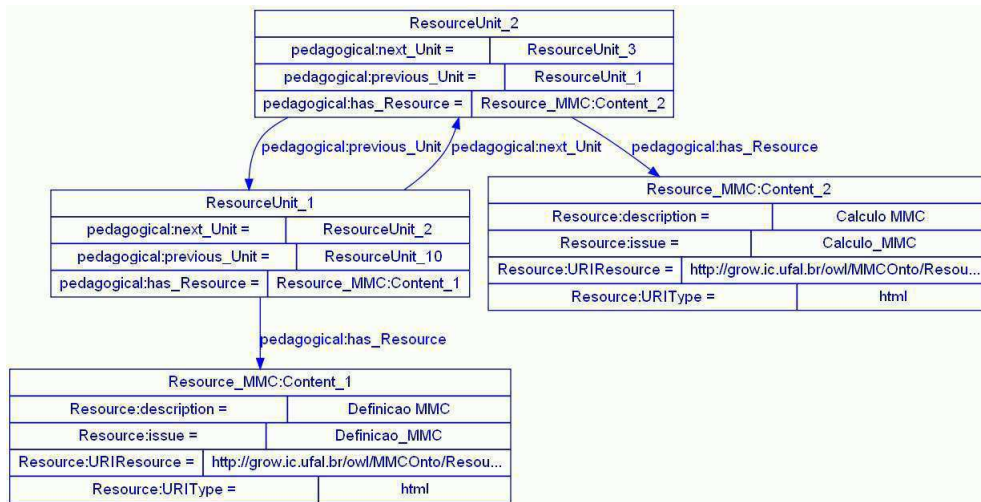


Figura 5.27: Parte da ontologia pedagógica contendo o sequenciamento do sistema (b).

- Regras de Mapeamento Domínio-GAIA: com o domínio definido, há a necessidade de executar as regras de mapeamento de modo a gerar as instâncias da ontologia de GAIA e consequentemente gerar os agentes;
- Ontologia de GAIA: após execução das regras, os agentes são descritos e mapeados na ontologia de GAIA. As Figuras 5.28, 5.29 e 5.30 contêm a modelagem em GAIA, de ambiente, sub-organização, agente e alguns papéis do sistema.

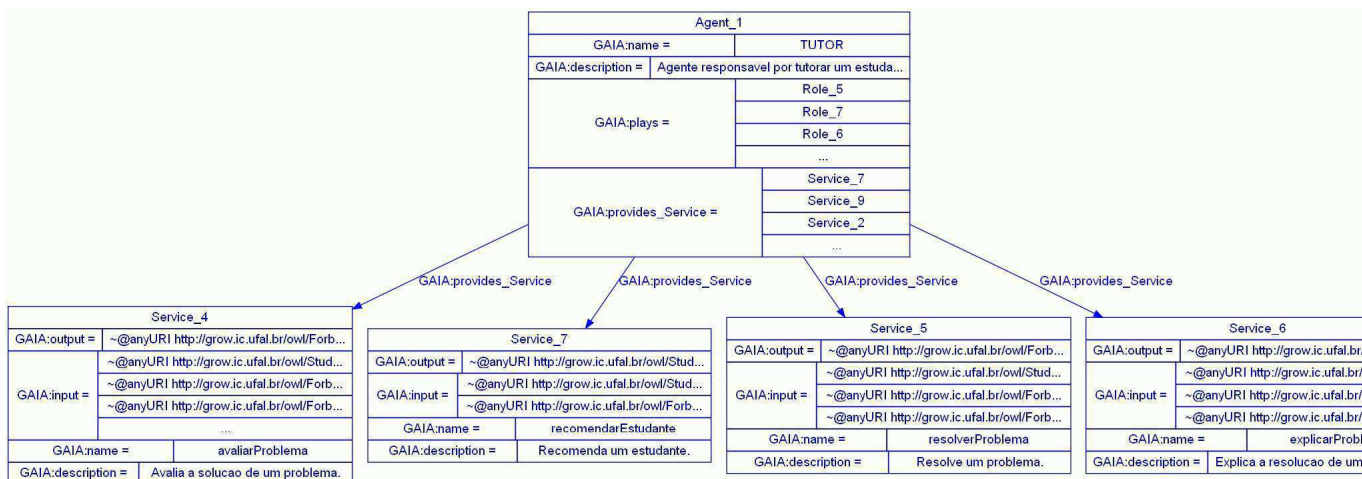


Figura 5.28: Parte da ontologia de GAIA com a especificação de agentes.

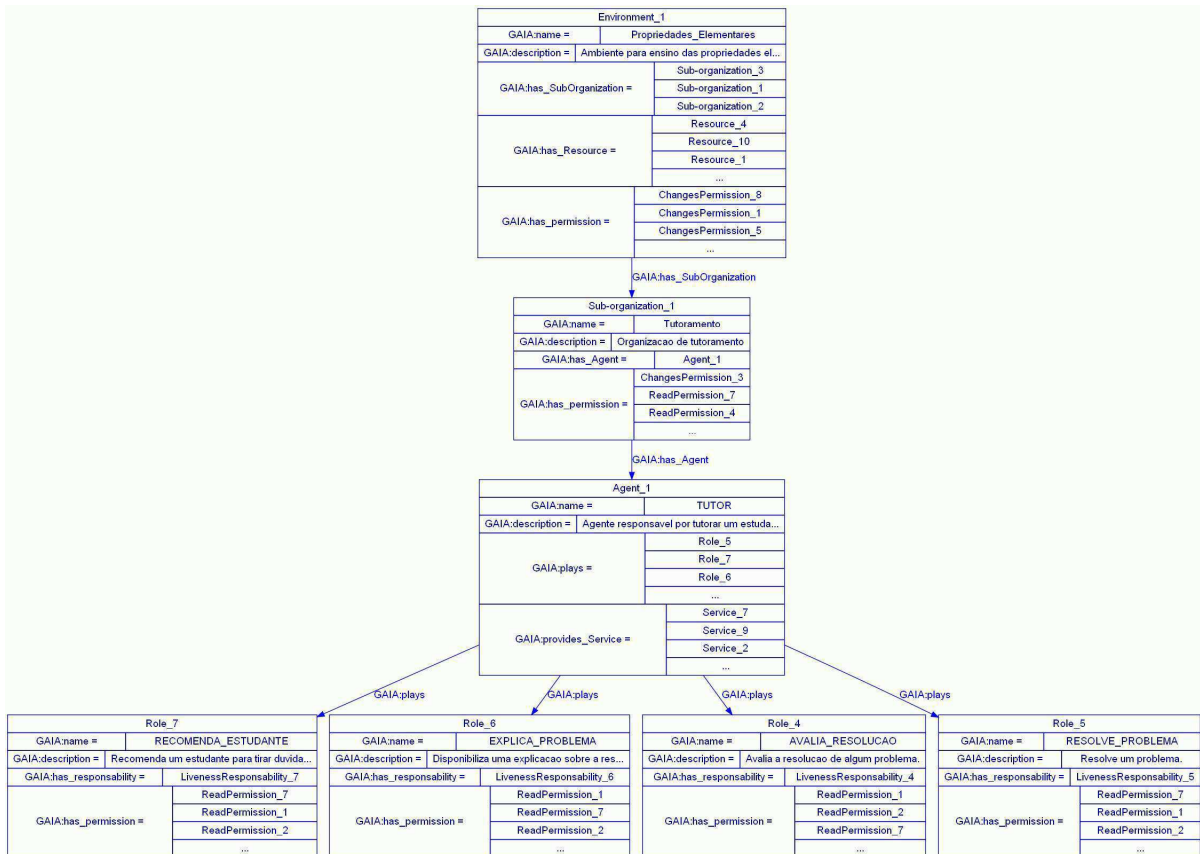


Figura 5.29: Parte da ontologia de GAIA com a especificação do ambiente.

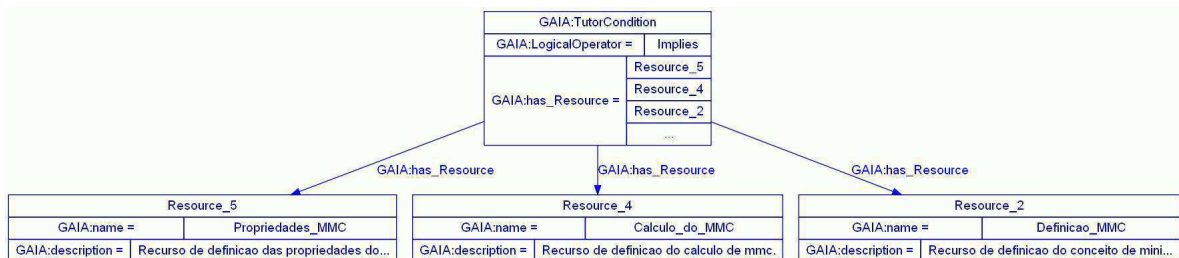


Figura 5.30: Parte da ontologia de GAIA com a especificação do ambiente.

- Regras de Mapeamento GAIA-JADE/ForBILEAgent: após a especificação em GAIA, são executadas as regras de mapeamento da ontologia de GAIA para JADE/ForBILEAgent.
- Ontologia de *ForBILEAgent*: descreve a ontologia do ForBILEAgent com a especificação dos agentes. As Figuras 5.31 e 5.32 descreve parte desta ontologia.

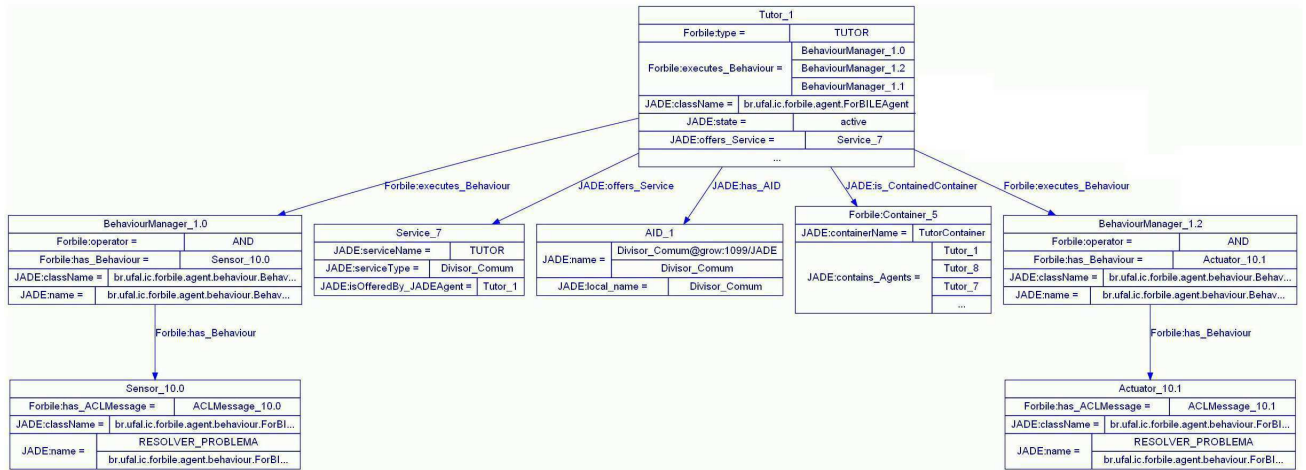


Figura 5.31: Parte da ontologia ForBILEAgent descrevendo um agente tutor.

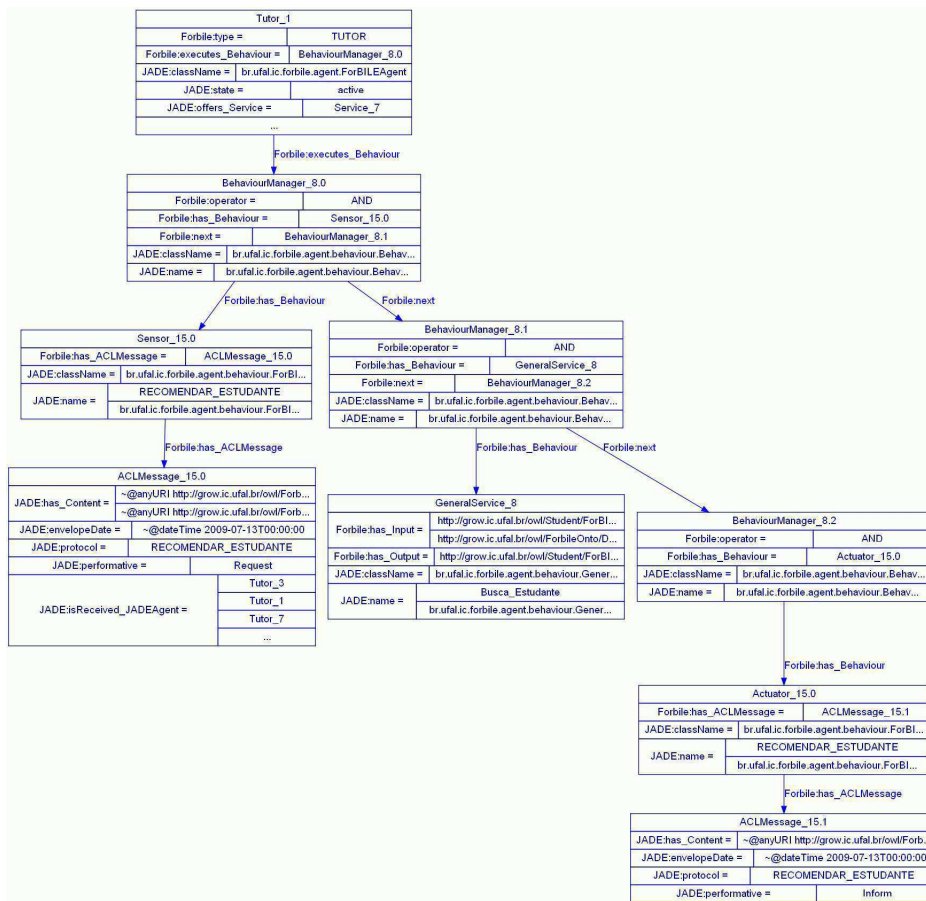


Figura 5.32: Parte da ontologia ForBILEAgent detalhando um comportamento executado por um agente tutor.

5.2.3 Implementação e testes

Esta seção aborda os requisitos não previstos no MASSAYO e descreve como foram implementados.

Adição de Novo Requisito

Para que o ambiente de EaD adaptativo de MMC funcione corretamente, duas funcionalidades devem ser adicionadas. Com isso, os mesmos são descritos como seguem:

- Resolução de Problemas Abertos: para que este novo requisito seja disponibilizado, há a necessidade de:
 1. Alterar ontologia: a alteração na ontologia se deve ao fato de que tal problema não foi previamente especificado. Desta forma, há a necessidade de estender a classe *Resource:Problem* da ontologia *ForBILE.Domain* e adicionar este novo conceito.
 2. Gerar Código: a geração de código é idêntica à forma descrita no estudo de caso anterior.
 3. Implementar do mecanismo de resolução de problema: a partir do momento que as classes de problema e solução foram gerados e disponibilizados, basta apenas descrever como é o processo de resolução de problemas, levando em consideração os diversos tipos de operações que se pode fazer neste tipo de domínio.

5.3 Estudo de Caso 3 - Ambiente Colaborativo em Medicina

Esta seção aborda um sistema colaborativo em Medicina, mas especificamente sobre o estudo do sangue. Este sistema possui as mesmas características de um sistema tutor inteligente tradicional, porém com a adição de um requisito de resolução de problemas de forma colaborativa. Neste estudo de caso, não será mostrada a especificação e modelagem do ambiente educacional, pois segue a mesma abordagem e estrutura que os estudos de caso citados anteriormente.

5.3.1 Implementação e Testes

Desta forma, para que determinado ambiente recomende usuários para interagirem em um processo de resolução de problemas, há a necessidade de agrupá-los primeiro. Os grupos reúnem alunos com mesmas dificuldades e com conhecimentos sobre aquele assunto. A utilização do agrupamento é feita visando maior facilidade e eficiência na recomendação não somente de estudantes, mas também de tutores, conteúdos e outros papéis. Ou seja, o agrupamento permite reunir alunos por interesse, dificuldades de aprendizado, competência em determinado domínio, dentre outros.

Com isso, o ambiente possui o papel do estudante e grupos de estudantes (agrupados pelo nível de conhecimento e currículo) bem como os serviços presentes no processo de tutoramento e de recomendação de estudantes. Entretanto, o processo de recomendação de estudantes envolve vários serviços semânticos e utilização de técnicas de mineração, como mostrado na Figura 5.33

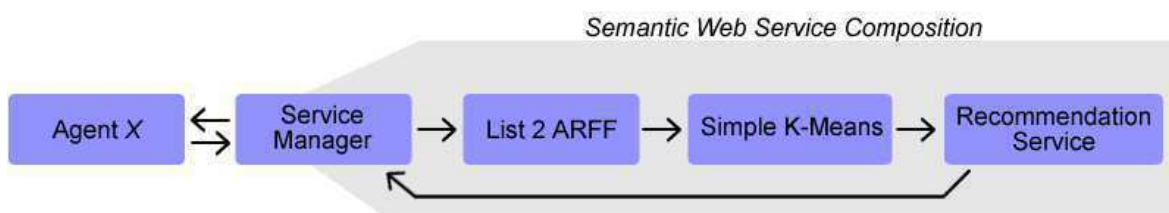


Figura 5.33: Fluxo de execução para recomendação de usuários.

Logo, o ambiente possui um agente de recomendação onde o mesmo inicia o

processo invocando o gerenciador de serviço passando os parâmetros de entrada e saída do serviço desejado. No processo de agrupamento, o parâmetro de entrada será uma lista que contém o histórico de acesso dos usuários; e o parâmetro de saída, grupos de alunos agrupados por interesse e/ou dificuldades. Após o processo de descoberta e composição de serviços realizada pelo gerenciador de serviço, o gerenciador de serviço inicia o processo de invocação dos serviços necessários para realizar o agrupamento de usuários. Este processo é descrito abaixo:

1. Primeiramente, é necessário invocar um serviço de conversão de lista para ARFF⁴. Este serviço é responsável pela transformação da lista contendo os históricos de acessos dos alunos para um ARFF. O ARFF é um tipo compreendido pelos algoritmos disponibilizados pelo WEKA [Witten and Frank, 2005]. Por isso, a necessidade de invocar este serviço;
2. Após a conversão para um ARFF, o gerenciador de serviço invoca o serviço de agrupamento que encapsula o algoritmo *Simple K-Means* (este algoritmo consiste em agrupar os dados fornecidos como entrada gerando um modelo de K agrupamentos). disponibilizado pelo WEKA, gerando os grupos de alunos;
3. Após agrupar os alunos baseado nos seus interesses e/ou dificuldades apresentadas, este serviço usa os grupos gerados para recomendação.

Desta forma, há a necessidade de implementar os serviços descritos na Figura 5.33.

A Figura 5.34 mostra o fluxo de interações entre o estudante e um agente tutor no processo de tutoramento.

⁴Do Inglês: *Attribute-Relation File Format*.

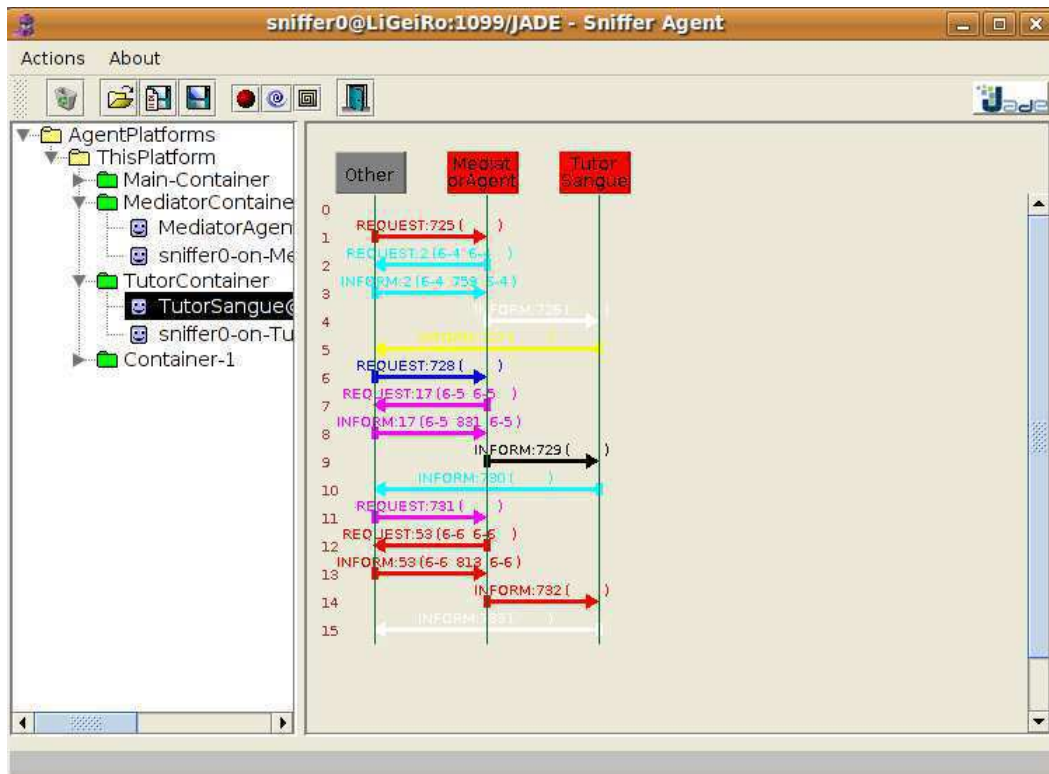


Figura 5.34: Fluxo de Interações entre os agentes no ambiente de Medicina (b).

Algumas telas do ambiente educacional são mostradas como seguem.



Figura 5.35: Tela do ambiente de Medicina com um recurso sendo mostrado.



Figura 5.36: Tela do ambiente de Medicina com descrição segundo a ontologia de páginas Web.

A Figura 5.37 são mostradas as instâncias da ontologia de páginas Web.

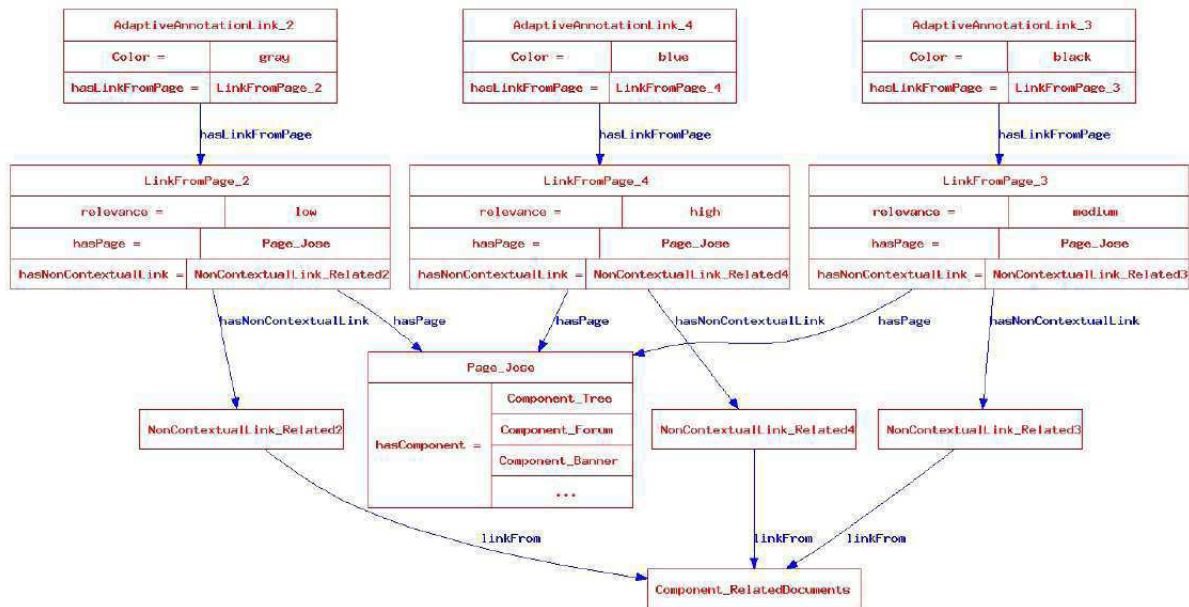


Figura 5.37: Instâncias da Ontologia de páginas Web para o ambiente de Medicina.

5.4 Estudo de Caso 4 - Interoperabilidade

Esta seção descreve como a interoperabilidade entre os dois ambientes do domínio de matemática descritos nos estudos de casos anteriores ocorre. Desta forma, é abordado aqui como este processo de interação ocorre.

É importante ressaltar que o modelo computacional já prevê um mecanismo de interoperabilidade entre ambientes educacionais. Com isso, todo o processo de interação entre os ambientes ocorre de forma automática, pois ambos foram registrados em um servidor bem como descritos semanticamente de modo a permitir a descoberta dos ambientes, isto é, a partir do momento que os ambientes foram distribuídos, as regras de mapeamento entre a ontologia *ForBILEAgent* com a ontologia de aplicações foram executadas. Esta execução fez com que os ambientes fossem descritos e disponibilizados para interoperarem. As Figuras 5.38 e 5.39 descrevem os ambientes do domínio de matemática especificados em suas ontologias de aplicação.

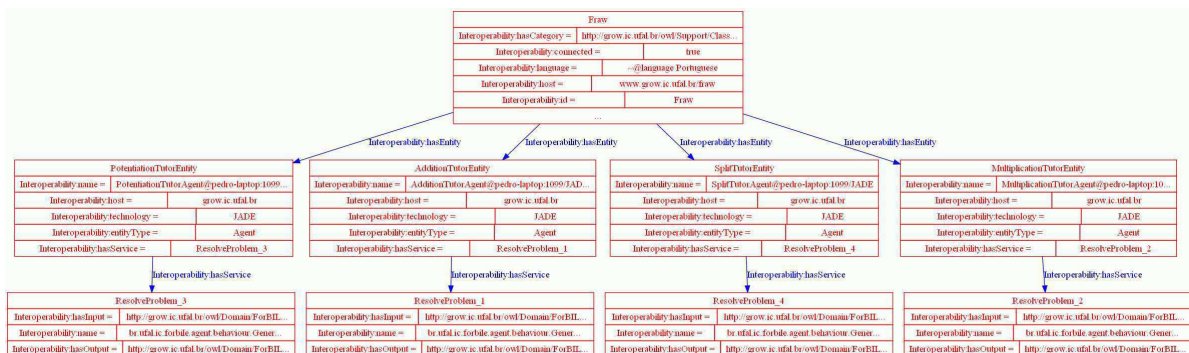


Figura 5.38: Ontologia de aplicação com instâncias do ambiente de FraW.

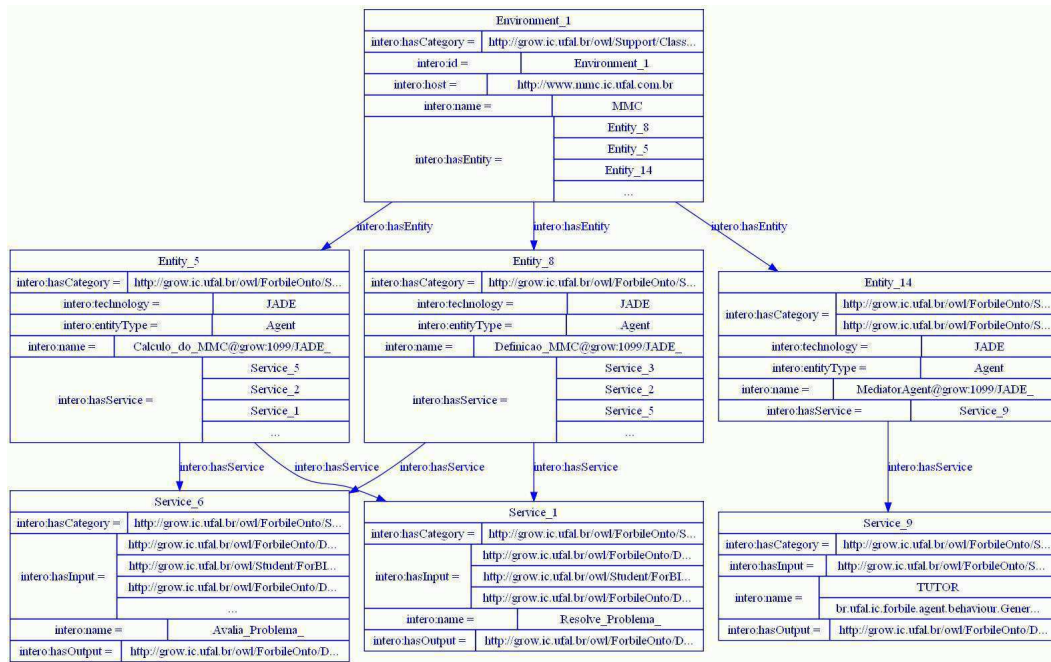


Figura 5.39: Ontologia de aplicação com instâncias do ambiente de MMC.

Com as ontologias definidas e os requisitos implementados no modelo computacional, o processo de interoperabilidade da seguinte forma:

1. Uma situação que necessite de interoperabilidade é identificada e então um determinado agente solicita para o agente mediador ajudar a executar determinada atividade. O agente mediador faz uma invocação de um serviço Web tradicional de *matching* de ambientes educacionais. Este processo é descrito no Diagrama de Sequência que segue.

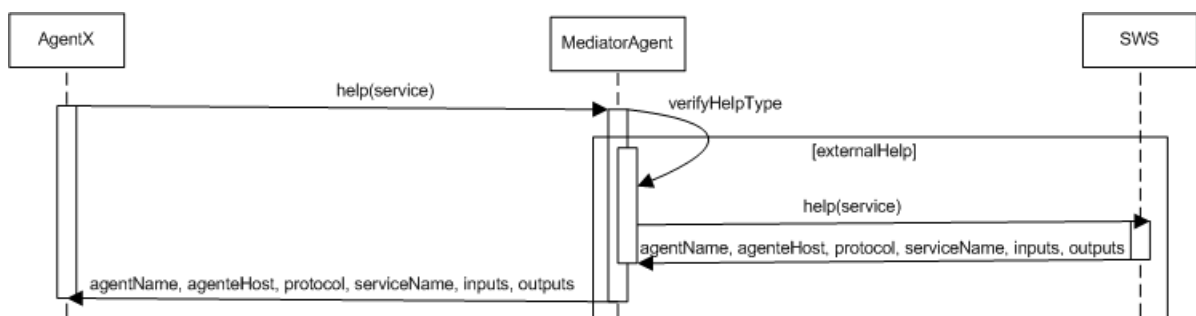


Figura 5.40: Diagrama de Sequência descrevendo solicitação de ajuda externa.

O serviço de *matching* é utilizado para verificar a similaridade entre o serviço solicitado por um determinado agente de um ambiente com os serviços dos

ambientes disponíveis para interoperar. O cálculo é dado pela média ponderada das propriedades disponíveis nas ontologias com o que é passado pelo agente. As propriedades que são levadas em consideração são o nome do serviço, os parâmetros de entrada e saída do serviço, a categoria do serviço, da entidade e do ambiente. Logo após o serviço de matching descobrir um ambiente e consequentemente um agente para interagir, é retornada uma sêxtupla $\langle agentName, agentHost, protocol, serviceName, inputs, outputs \rangle$, onde:

agentName é o nome do agente que irá interagir,

agentHost é o local que o agente está sendo executado,

protocol é o protocolo de comunicação entre agentes (neste caso, *ACL-Message*),

serviceName o nome do serviço do agente que foi descoberto que atende à solicitação feita por determinado agente

inputs são os tipos de parâmetros necessários para que o serviço seja executado e

output é o retorno do serviço que será passado para o agente solicitante.

2. Após o ambiente ter sido descoberto, o agente solicitante irá interagir com o novo agente, respeitando a sêxtupla descrita acima. A Figura 5.41 aborda o Diagrama de Sequência entre os agentes interoperando.

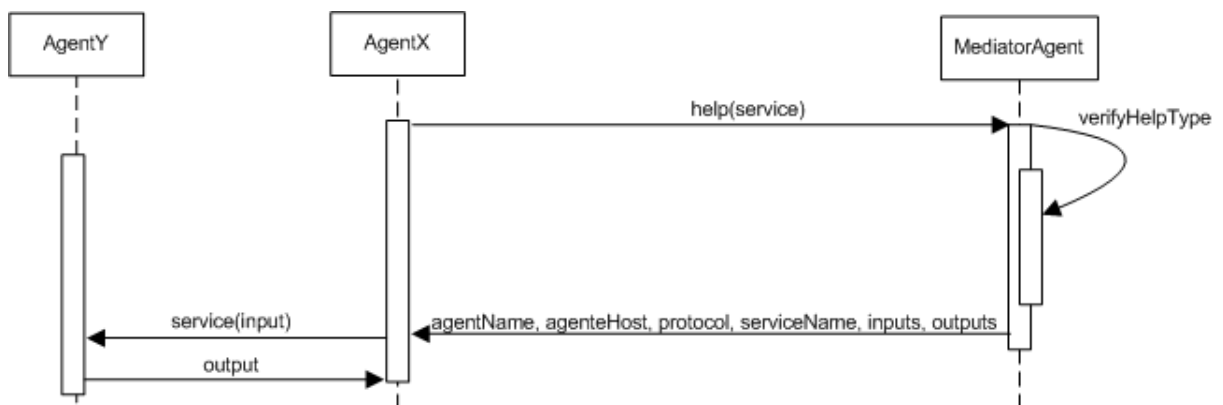


Figura 5.41: Diagrama de Sequência entre dois agentes interoperando.

Com relação aos ambientes no domínio de matemática, foram feitos dois testes onde houve a solicitação de um novo recurso educacional e outro a solicitação de mais

informação sobre determinado tema, ou seja, o ambiente de Fração solicitou informações sobre MMC e como não havia disponível no ambiente, foi feita uma solicitação externa.

A Figura 5.42 descreve parte da interação que necessitou da interoperabilidade.

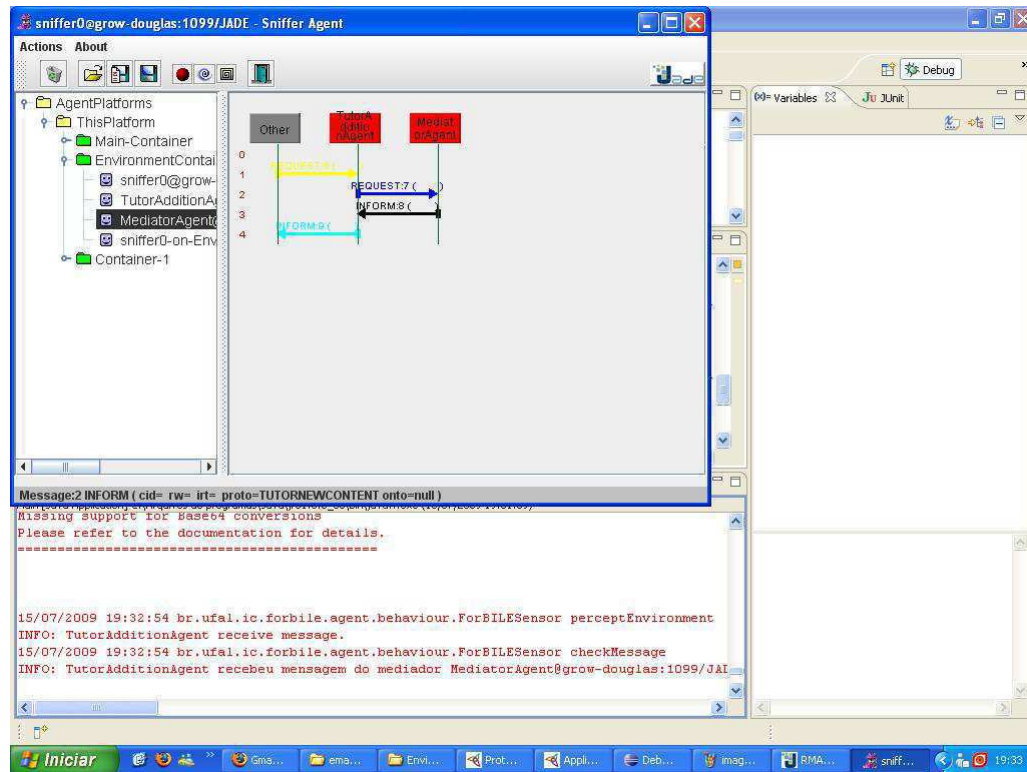


Figura 5.42: Parte da Interação de agentes para descobrir ambiente para interoperar.

5.5 Discussão

Esta seção tem por objetivo fazer uma avaliação dos modelos propostos nesta tese através da análise dos estudos de caso.

5.5.1 MASSAYO-RM

O primeiro passo realizado para a implementação dos três estudos de caso foi a especificação de cada ambiente educacional. A partir do momento que a especificação foi feita, houve a possibilidade de identificar os papéis presentes nos ambientes bem como os recursos computacionais. Isto foi possibilitado através do modelo conceitual, onde os requisitos descritos na especificação são mapeados em serviços Web. Já a

interação entre os papéis humanos com o sistema é possibilitada através da disponibilização de ferramentas interativas.

É importante ressaltar que os estudos de caso contemplaram aplicações educacionais com características diferenciadas, ou seja, um sistema tutor inteligente, um ambiente de educação a distância e um ambiente colaborativo. Independente do tipo de ambiente educacional, os recursos necessários foram identificados no modelo conceitual proposto.

5.5.2 MASSAYO-CM

Com o modelo conceitual definido, juntamente com os papéis e entidades identificados, passa-se a utilizar o modelo computacional para garantir a implementação de cada ambiente. Cada um dos aspectos abaixo foi garantida pelo MASSAYO-CM.

- **Integração de ontologias, agentes e serviços semânticos:** esta integração ocorre de forma transparente para a equipe de desenvolvimento. De fato, em nenhum momento houve a necessidade de implementar agentes de software e em apenas um estudo de caso houve a necessidade de estender a ontologia do domínio para especificar problema. A especificação do sistema através de ontologias permite que os agentes sejam gerados e automaticamente estejam capacitados a descobrir serviços Web semânticos.

Uma ferramenta foi desenvolvida [Oliveira et al., 2009] para facilitar o processo de especificação através de ontologias. Esta ferramenta gera ambientes de autoria dinamicamente. Algumas telas do sistema são mostradas abaixo.

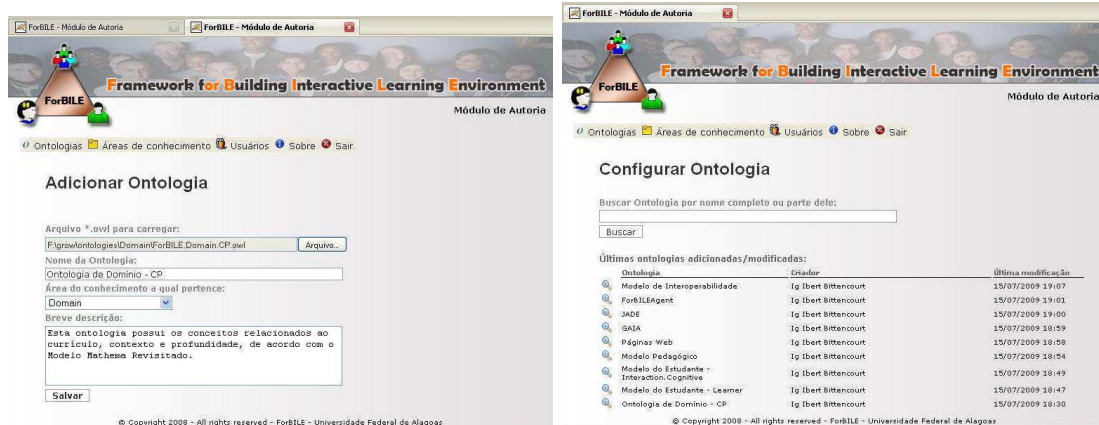


Figura 5.43: Tela para adicionar ontologia (a) e Tela com ontologias cadastradas (b).



Figura 5.44: Tela para especificar passo.

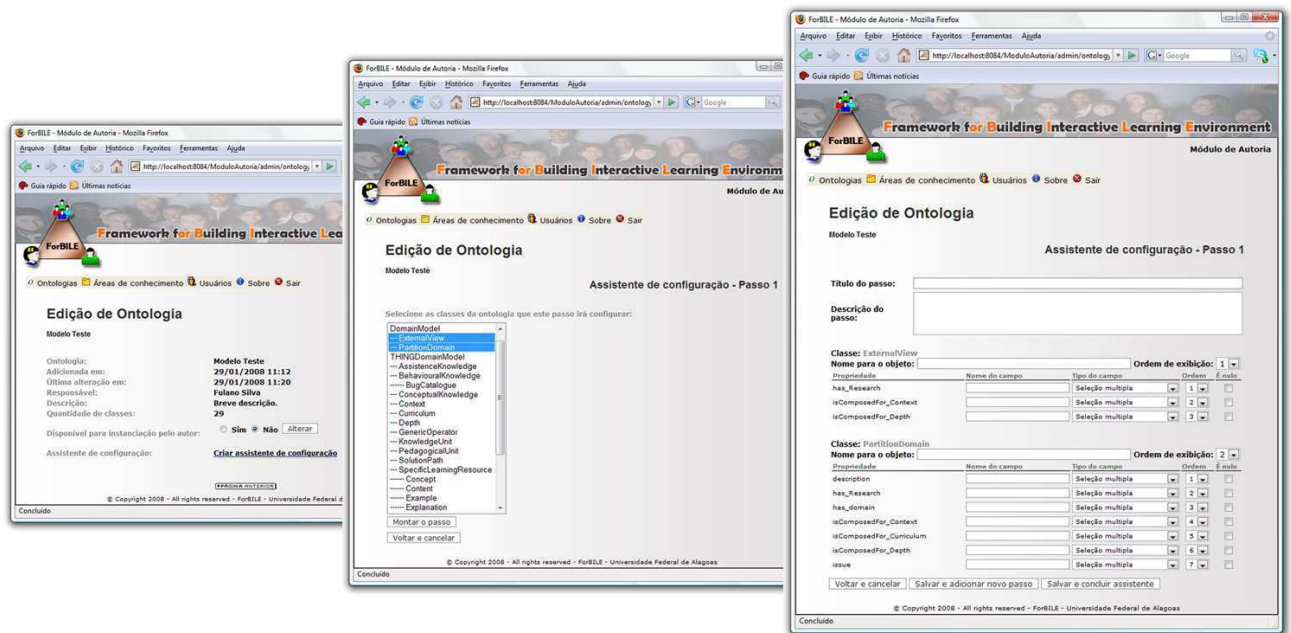


Figura 5.45: Tela com vários passos para construção de assistente.



Figura 5.46: Tela com assistente criado.

- Reúso de Software: o reúso de software foi proporcionado principalmente pela

utilização de serviços Web semânticos e pelos agentes de software, isto é, o ambiente disponibiliza todo o mecanismo de tutoramento (modelo do domínio, estudante e pedagógico), resolução de problemas (para problemas de múltipla escolha, preencher lacuna, relacionar coluna, e V/F), técnicas de inteligência artificial (como raciocínio baseado em regras e técnicas de mineração de dados), dentre outros. Além dos serviços, o ambiente disponibiliza um mecanismo de geração automática de agentes através de ontologias. Assim, toda a implementação de agentes foi abstraída para ontologias e permitindo assim o reúso de software. Nos estudos de caso, poucos foram os requisitos necessários a implementar em cada ambiente. É importante esclarecer que uma análise comparativa sobre o reúso alcançado entre o MASSAYO e os trabalhos relacionados não foi possível fazer, pois apenas dois trabalhos relacionados tinham implementação e mesmo assim não houve acesso ao código fonte dos mesmos.

- Reúso de Conhecimento: o reúso de conhecimento é proporcionado tanto através da utilização de ontologias e na possibilidade destas ontologias serem utilizadas em outros ambientes, ou seja, para que o ambiente de fração use recursos de MMC, não há a necessidade de disponibilizar novos recursos educacionais, mas sim reusar os recursos providos por tal ambiente. Isto pode ser feito de duas formas, reusar diretamente a ontologia e/ou os recursos ou através de interoperabilidade.

5.5.3 MASSAYO-IM

O modelo de interoperabilidade foi avaliado através de um estudo de caso onde foi verificada a comunicação entre os ambientes no domínio de matemática. O ambiente foi capaz de garantir tanto a solicitação de ajuda por um agente do ambiente quanto a descoberta de um novo ambiente para interoperar. O ambiente proporciona seis mecanismos de interoperabilidade de forma automatizada (todos estão divididos nas categorias de recomendação de papéis e na disponibilização de recurso educacional) que são necessárias para ambientes educacionais. Entretanto, para que haja maior efetividade na solução proposta, há a necessidade de implementar mecanismos de mineração de texto e extração de informação, internacionalização e mecanismos automatizados de

alinhamento de ontologias e verificação de satisfatibilidade.

5.5.4 Outras avaliações

Estas avaliações se referem principalmente ao desempenho e usabilidade do ambiente MASSAYO.

Foram gerados gráficos de desempenho referentes à utilização do MASSAYO. O objetivo dos gráficos foi verificar a utilização do processador bem como o tempo para executar cada operação. A semente utilizada para a geração do gráfico abaixo foi com Sistema Operacional Windows XP, Processador Intel Core 2 duo 1.66 GHz, 2GB de Ram e HDD Ata.

O Gráfico 5.47 ilustra tanto o carregamento quanto importação e checagem de ontologias no MASSAYO. No gráfico, a marcação 1 diz respeito ao carregamento do ambiente, importação das ontologias e criação dos agentes (20 primeiros segundos). Já a marcação 2, 3 e 4 equivalem à importação das ontologias e checagem de alguma alteração (10s). Ou seja, o tempo total foi de 50s. É importante ressaltar que este tempo foi utilizado apenas para o carregamento do núcleo da MASSAYO por um usuário, não levando em consideração o carregamento da Interface.

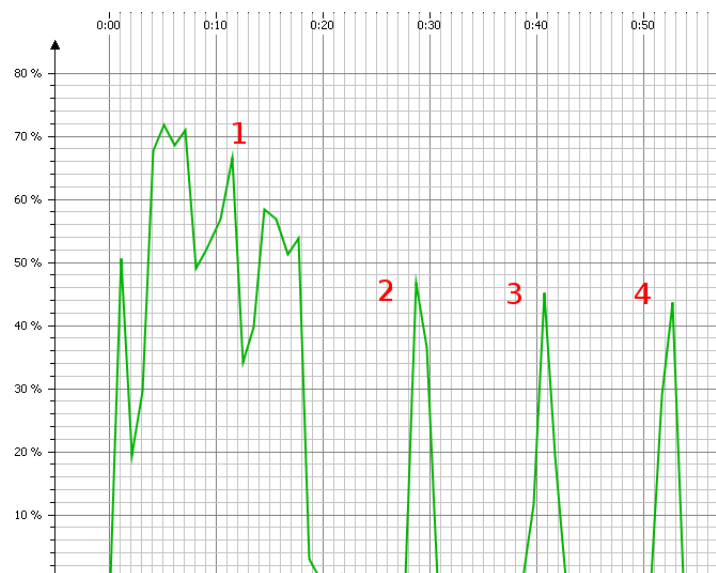


Figura 5.47: Gráfico de desempenho de um agente percebendo o ambiente no MASSAYO.

Esta baixa performance é ocasionada, principalmente, pelos motivos abaixo.

1. agente perceber o ambiente: durante a utilização do sensor de cada agente para perceber o ambiente ocorre um processamento que leva em consideração a utilização de diversos recursos, ou seja, com um ambiente que possui dezenas de agentes com um tempo de sensor baixo vai ocasionar uma baixa qualidade da performance;
2. arquivos: este problema é um dos principais causadores da baixa performance do ambiente, pois o carregamento de ontologias faz com que haja muito consumo da máquina tornando o processo bastante lento. Como o ambiente é todo especificado através do uso de ontologias, a todo momento as ontologias devem ser usadas para executar determinada operação. Além disso, cada serviço semântico faz uso de pelo menos quatro ontologias tornando o processo ainda mais lento.

Como solução para este problema, objetiva-se extinguir a utilização da API do *protégé* e a integração das ontologias com banco de dados. Para isso, foi feita uma análise do Sesame e as mudanças já estão sendo implementadas no MASSAYO. Experimentos para avaliar a performance foram feitos e houve um ganho significativo nas operações (criação, alteração, remoção de instâncias e *query* usando SPARQL) com ontologias, descrevendo o custo computacional utilizando o Sesame e utilizando o *Protégé*, ou seja, como mostrado nos gráficos que o Sesame teve menos uso computacional e menor tempo de execução que o *Protégé*. A semente utilizada foi Sistema Operacional Windows XP, Intel Pentium 4 3.00 GHz, 1 GB de RAM e e HDD Ata.

Além do Sesame, está sendo feita a utilização também do Elmo, que faz geração automática de código, isto é, a solução do Sesame, juntamente com o Elmo, permite que a API do *Protégé* seja utilizada apenas para estender a ontologia.

Com relação à usabilidade, foi feito um estudo sobre diversas tecnologias que estivessem mais em consonância com o ambiente e com os princípios utilizados no trabalhos (neste caso, a Web Semântica Social). Além disso, a solução presente no MASSAYO (com Liferay) faz com que haja uma grande lentidão no carregamento e torna bastante complicada a mudança dinâmica da interface. Com isso, foi decidido fazer

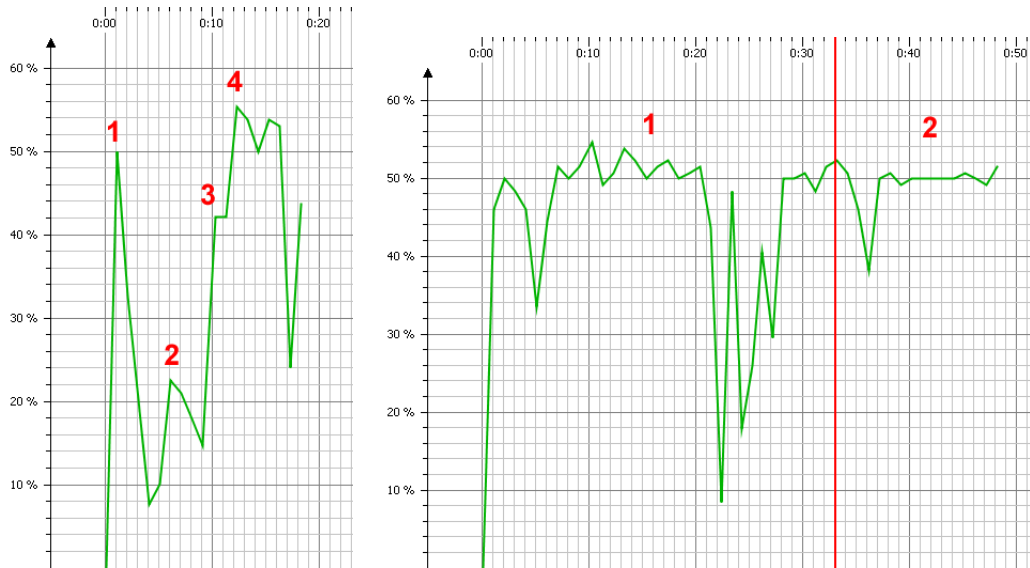


Figura 5.48: Gráfico de desempenho do Sesame (esquerda) e do Protégé (direita).

a mudança para utilizar o GWT (*Google Web Toolkit*), pois o mesmo já dá suporte a várias ferramentas bem como a integração com as tecnologias da Google, como *e-mail*, agenda, *google maps*, entre outras. Apesar de usabilidade ser mais abrangente do que o discutido nesta avaliação, estudos sobre métricas de avaliação estão sendo feitas para garantir um ambiente de fácil utilização.

5.6 Conclusão

Este capítulo fez uma avaliação dos modelos propostos através da descrição de quatro estudos de caso, onde os três primeiros abordaram a construção de ambientes educacionais e o último a possibilidade de interoperar estas aplicações. Além disso, uma discussão foi abordada, onde se levou em consideração aspectos como o reúso de conhecimento e de software, automatização de atividades, dentre outros aspectos, como performance e usabilidade. Além de discussões preliminares referentes a performance e usabilidade, outros aspectos também foram avaliados, porém de forma superficial como a qualidade da composição e a disponibilização de recursos.

Capítulo 6

Conclusões e Trabalhos Futuros

*Não se pode desviar do caminho.
É preciso ir até onde der... e depois continuar.*
Sensei Francisco Araújo, “Chicão”

*A jornada é o que nos traz felicidade...
O resto virá por acréscimo.*
Sócrates, no filme “Poder Além da Vida”

Este trabalho se insere no contexto da pesquisa atual na área de Inteligência Artificial em Educação (IAEd), refletindo-se na busca por sistemas educacionais adaptativos baseados na Web semântica, indo ao encontro de uma nova geração de ambientes virtuais de aprendizagem. Em particular, a presente pesquisa mostra-se relevante e contribui para a construção efetiva desses sistemas.

Neste capítulo são resumidos os principais resultados alcançados neste trabalho, destacando-se as suas principais realizações e contribuições para a área de pesquisa mencionada acima. Além disso, algumas de suas limitações são discutidas, culminando com uma lista de sugestões para estendê-lo em trabalhos futuros.

6.1 Contribuições

Este trabalho de tese abordou problemas significativos relacionados à concepção e desenvolvimento de ambientes educacionais adaptativos e baseados na Web semântica. Assim, enfocaram-se aspectos de sua viabilidade, elaborando-se modelos e ferramentas como parte da resposta às questões levantadas e objetivos estabelecidos no início da pesquisa, tal como descritos no Capítulo 1 deste documento. Neste sentido, investiu-se, por exemplo, em questões sobre reuso de conhecimento e de software, sobre interoperabilidade entre ambientes diferentes. Deste modo, buscou-se suporte técnico nas áreas de Inteligência Artificial, Engenharia de Software e Web Semântica.

Os objetivos descritos no Capítulo 1 para esta tese foram todos alcançados, podendo ser observados no que foi concretizado para o MASSAYO, em termos da elaboração e avaliação dos modelos propostos. Ou seja, foi proposto i) um modelo conceitual (MASSAYO-RM) identificando os recursos, entidades e relações entre eles, bem como gerando categorias para uma melhor classificação tanto dos recursos quanto das entidades. Sendo assim, uma avaliação do modelo foi elaborada, onde cada trabalho relacionado foi descrito segundo o modelo. Além disso, ii) um modelo computacional (MASSAYO-CM) foi proposto e implementado para proporcionar a construção de ambientes educacionais adaptativos e semânticos. Foi feita uma análise do atual estado da prática de ontologias, agentes de software e serviços semânticos, possibilitando descrever como tais abordagens seriam utilizadas. Por fim, iii) foi proposto e implementado um modelo de infra-estrutura para interoperabilidade (MASSAYO-IM), permitindo que aplicações educacionais criadas no MASSAYO-CM pudessem se comunicar de forma automatizada. Foi criado um cenário de uso em matemática para avaliar o modelo. Com isso, esta pesquisa demonstrou a viabilidade em se construir ambientes educacionais adaptativos e semânticos, sendo isto importante para o avanço do estado da arte em IAEd no que diz respeito ao investimento nos rumos para a nova geração de ambientes virtuais de aprendizagem.

No entanto, a presente pesquisa apresentou algumas limitações, tais como o provimento de mecanismos de adaptação automatizada na interface do usuário e a performance do ambiente. Além disso, algumas dificuldades importantes foram

encontradas, como por exemplo, a utilização de serviços Web semânticos e a construção das ontologias e regras SWRL.

Desta forma, espera-se ter contribuído com as seguintes comunidades:

- **Inteligência Artificial e Educação:** pela construção de modelos que permitem a construção de ambientes educacionais adaptativos, bem como na garantia do desenvolvimento e manutenção destes ambientes de forma mais rápida;
- **Educação a Distância:** por disponibilizar modelos que possibilitem a produção em larga escala de tais ambientes e para as diversas áreas do conhecimento. Além disso, tais modelos permitir que instituições de ensino possam cooperar na produção de materiais educacionais e no acesso aos estudantes que necessitem deste tipo de tecnologia, tanto no Brasil quanto no exterior;
- **Engenharia de Software:** na proposição de ontologias que permitiram a especificação de sistemas multi-agentes e na geração dos agentes de software em a necessidade de implementação. Além disso, na integração de ontologias, agentes e serviços semânticos para reduzir o tempo de manutenção das aplicações;
- **Web Semântica:** na implementação de um modelo computacional que integre ontologias, agentes e serviços semânticos de tal forma a facilitar o reuso de conhecimento e automatização de processos nas aplicações. Além disso, pela análise do atual estado da prática de diversas tecnologias provenientes da Web Semântica.

Enfim, foram alcançados alguns outros indicadores como parte do processo de pesquisa associado à realização desta tese. Tais indicadores são: i) Organização dos dois primeiros eventos¹ de Web Semântica e Educação do País; ii) Participação em comitês de avaliação de periódicos²; iii) Consórcio de doutorado³, tendo como mentor o presidente da Sociedade de Inteligência Artificial e Educação, Jim Greer; iv) cooperação

¹Brazilian Workshop on Semantic Web and Education, em conjunto com o SBIE 2007 e 2008.

²*IEEE Multidisciplinary Engineering and Education Magazine* e *Scientia Journal - Special Issue on Semantic Web and Education*.

³*Doctoral Consortium - International Conference on Artificial Intelligence in Education - 2007, Los Angeles, USA*

internacional com pesquisadores renomados (Riichiro Mizoguchi, Julita Vassileva, Jim Greer, Nicola Henze, Darina Dicheva e Ryan Backer) e estudantes de doutorado (Seiji Isotani e Daniel Krause); v) publicação em conferências nacionais [Sibaldo et al., 2008][Isotani et al., 2008] e internacionais [Calado et al., 2009][Bittencourt et al., 2009a][Bittencourt et al., 2008a][Abel et al., 2008] renomadas, além de outras publicações e vi) publicação em periódicos nacionais [Bittencourt et al., 2008d][Isotani et al., 2009] e internacionais [Bittencourt et al., 2009b][Oliveira et al., 2009][Abel et al., 2010][Bittencourt et al., 2008b] renomados.

6.2 Trabalhos Futuros

A pesquisa abordada nesta tese possui vários trabalhos futuros, sendo tais extensões propostas a seguir.

6.2.1 MASSAYO-RM

Como trabalhos futuros do modelo conceitual, sugere-se principalmente:

- Especificação baseada em ontologias: prover ontologias que auxiliem o processo de especificação dos requisitos de tal forma que garanta a identificação automática das entidades e papéis presentes no ambiente a ser construído;
- Protocolos de Interação: descrever modelos de interação que aborde a comunicação entre as entidades presentes no MASSAYO-CM.

6.2.2 MASSAYO-CM

O MASSAYO-CM possui vários trabalhos futuros, nos quais são destacados:

- Serviços Web Semânticos: a disponibilização de mais serviços Web semânticos de forma a permitir maior reusabilidade de código e capacidade de descoberta, composição e invocação de serviços como soluções contingenciais. Além disso, prover mais serviços inteligentes e educacionais ao ambiente, como técnicas de

redes neurais artificiais, mineração de texto, recomendação de pares, adaptação hipermídia, entre outros;

- Performance: melhorar a performance dos agentes de software e serviços Web semânticos;
- Múltiplos Dispositivos: garantir que o MASSAYO possa ser utilizado não somente para aplicações Web, mas também prover facilidades para aplicações móveis e pervasivas e para TV Digital.

6.2.3 MASSAYO-IM

O principal trabalho futuro do modelo de infra-estrutura para interoperabilidade é a implementação de um mecanismo que garanta a interoperabilidade entre ambientes educacionais com diferentes infra-estruturas tecnológicas, como a comunicação de um ambiente com o paradigma de agentes e outro ambiente com a abordagem baseada em componentes.

6.2.4 Outros Trabalhos

Outros trabalhos futuros que são de bastante relevância são:

1. MASSAYO-P: a concepção de um processo de desenvolvimento de aplicações educacionais sob os moldes desta tese. Este processo deve levar em consideração uma abordagem ágil, embutindo no ciclo engenharia ontológica e engenharia de conhecimento, e uma abordagem multi-agente e orientada a serviços semânticos;
2. MASSAYO-E: prover um ambiente de desenvolvimento de aplicações utilizando o MASSAYO. Este ambiente pode ser um *plugin* de uma ferramenta de desenvolvimento como o eclipse.

Bibliografia

- [Des, 2008] (2008). *CharLa'08 - Grand Challenges in Computer Science Research in Latin America Workshop, Buenos Aires, Argentina.*
- [Abazi-Bexheti, 2008] Abazi-Bexheti, L. (2008). Development of a learning content management system. *WSEAS Trans. Info. Sci. and App.*, 5(6):1001–1010.
- [Abel et al., 2008] Abel, F., Bittencourt, I. I., Henze, N., Krause, D., and Vassileva, J. (2008). A rule-based recommender system for online discussion forums. In *AH'08: Adaptive Hypermedia, Hannover, Germany.*
- [Abel et al., 2010] Abel, F., Bittencourt, I. I., Henze, N., Krause, D., Vassileva, J., and Costa, E. (2010). Configurable post recommendations for online discussion forums. *IEEE Transaction on Learning Technologies, Special Issue on Personalization.*
- [Abel et al., 2006] Abel, F., Brunkhorst, I., Henze, N., Krause, D., Mushtaq, K., Nasirifard, P., and Tomaschewski, K. (2006). Personal reader agent: Personalized access to configurable web services. In *LWA*, pages 12–13.
- [Adelstein et al., 2004] Adelstein, F., Gupta, S. K., III, G. R., and Schwiebert, L. (2004). *Fundamentals of Mobile and Pervasive Computing.* McGraw-Hill Professional, 1 edition.
- [AIRES, 1992] AIRES (1992). Laboratory for the advanced researches in intelligent educational systems. Disponível em: http://www.cs.usask.ca/research/research_groups/aries/. Acesso: 23 Ago. 2006.
- [Al-Masri and Mahmoud, 2007] Al-Masri, E. and Mahmoud, Q. H. (2007). Discovering

- the best web service. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1257–1258, New York, NY, USA. ACM.
- [Alhir, 2003] Alhir, S. S. (2003). *Learning UML*. O'Reilly.
- [Anderson and Whitelock, 2004] Anderson, T. and Whitelock, D. (2004). The educational semantic web: visioning and practicing the future of education. *Journal of Interactive Media in Education - JIME*, 7:1–15.
- [Aroyo and Dicheva, 2004] Aroyo, L. and Dicheva, D. (2004). The new challenges for e-learning: The educational semantic web. *Educational Technology and Society*, 7(4):59–69.
- [Aspray et al., 2002] Aspray, W., Goyal, A., Irwin, M. J., Jones, A., Lazowska, E., Patterson, D., Pollack, J., Sproull, B., and Wulf, W. (2002). Grand research challenges in information systems. Technical report, Computing Research Association.
- [Azouaou and Desmoulins, 2006] Azouaou, F. and Desmoulins, C. (2006). Using and modeling context with ontology in e-learning: the case of teacher's personal annotation. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), In Adaptive Hypermedia and Adaptive Web-Based Systems, Dublin, Ireland*.
- [Baader et al., 2003] Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2003). *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- [Baeza-Yates and Ribeiro-Neto, 2005] Baeza-Yates, R. and Ribeiro-Neto, B. (2005). *Modern Information Retrieval*. Addison Wesley, 1 edition.
- [Bayoumi, 2007] Bayoumi, F. (2007). Towards mobile learning. In *IMSA'07: IASTED European Conference on Proceedings of the IASTED European Conference*, pages 202–207, Anaheim, CA, USA. ACTA Press.
- [Benyon and Murray, 1993] Benyon, D. R. and Murray, D. M. (1993). Adaptive systems; from intelligent tutoring to autonomous agents. *Knowledge-Based Systems*, 6(4):197–219.

- [Berners-Lee, 1999] Berners-Lee, T. (1999). *Weaving the Web*. Harper San Francisco.
- [Berners-Lee et al., 2001] Berners-Lee, T., Lassila, O., and Hendler, J. (2001). The semantic web. *Scientific American*, 254(5):34–43.
- [Bittencourt, 2006] Bittencourt, I. I. (2006). Plataforma para construção de ambientes interativos de aprendizagem baseados em agentes. Master's thesis, Universidade Federal de Alagoas.
- [Bittencourt et al., 2009a] Bittencourt, I. I., Bispo, P., Costa, E., Pedro, J., Dermeval, D., Vêras, D., and Pacca, H. (2009a). Modeling jade agents from gaia methodology under the perspective of semantic web. In *ICEIS'09: International Conference on Enterprise Information Systems, Milan, Italy*.
- [Bittencourt et al., 2007a] Bittencourt, I. I., Costa, E., Fonseca, B., Guilherme, J., and Nunes, C. (2007a). Sistema de autoria para construção de ambientes interativos de aprendizagem baseada em agentes. *Revista Brasileira de Informática na Educação*, 15:21–30.
- [Bittencourt et al., 2008a] Bittencourt, I. I., Costa, E., Isotani, S., Mizoguchi, R., and Bittencourt, I. M. (2008a). Towards a reference model for semantic web-based educational systems. In *Second Brazilian Workshop on Semantic Web and Education (WSWEd), Brazilian Symposium on Informatics on Education (SBIE)*.
- [Bittencourt et al., 2009b] Bittencourt, I. I., Costa, E., Silva, M., and Soares, E. (2009b). A computational model for developing semantic web-based educational systems. *Knowledge-Based System*, 22(4):302–315.
- [Bittencourt et al., 2008b] Bittencourt, I. I., Costa, E., Soares, E., and Pedro, A. (2008b). Towards a new generation of web-based educational systems: The convergence between artificial and human agents. *IEEE Multidisciplinary Engineering Education Magazine*, 3(1):17–24.
- [Bittencourt et al., 2007b] Bittencourt, I. I., de Barros Costa, E., de Almeida, H. O., Fonseca, B., Maia, G., Calado, I., and da Silva, A. P. (2007b). Towards an ontology-based framework for building multiagent intelligent tutoring systems. In

- SEAS'07: Third Workshop on Software Engineering for Agent-oriented Systems, João Pessoa, Paraíba, Brazil.*
- [Bittencourt et al., 2008c] Bittencourt, I. I., de Barros Costa, E., Fonseca, B., Maia, G., Melo, J. S. S., Ferneda, E., and Silva, A. P. B. (2008c). *Agent-Based Tutoring Systems by Cognitive and Affective Modeling*, chapter Constructing Intelligent Tutoring Systems based on a Multiagent Architecture, pages 228 – 259. IDEA GROUP INC.
- [Bittencourt et al., 2009c] Bittencourt, I. I., Dicheva, D., Costa, E., Braz, L., Pacheco, H., and Dichev, C. (2009c). Supporting interoperability between web-based educational systems. In *IEEE Frontiers in Education, Texas, USA*.
- [Bittencourt et al., 2008d] Bittencourt, I. I., Isotani, S., Costa, E., and Mizoguchi, R. (2008d). Researchs directions on semantic web and education. *Scientia Journal*, 19(1):59–66.
- [Boley et al., 2001] Boley, H., Tabet, S., and Wagner, G. (2001). Design rationale of ruleml: a markup language for semantic web rules. In E., I., Decker, S., and McGuinness, J. E. D. L., editors, *SWWS'01, The first Semantic Web Working Symposium, Stanford University, California*, pages 381–401.
- [Boticario et al., 2007] Boticario, J. G., Vidal, L. C., Gesa, R. F., Ortega, M., Borrajo, D., and de la Rivaherrera, E. O. (2007). Adaptation based on machine learning, user modelling and planning for complex user-oriented tasks: Adaptaplan (tin2005-08945-c06-00). In *Jornadas de Seguimiento de Proyectos. Programa Nacional de Tecnologías Informáticas*.
- [Brambilla et al., 2007] Brambilla, M., Ceri, S., Facca, F. M., Celino, I., and Cerizza, D. (2007). Model-driven design and development of semantic web services applications. *ACM Transactions on Internet Technology (TOIT)*, 8(1):3.
- [Brooks et al., 2006] Brooks, C. A., Greer, J. E., Melis, E., and Ullrich, C. (2006). Combining its and elearning technologies: Opportunities and challenges. In Ikeda, M., Ashley, K. D., and Chan, T.-W., editors, *Intelligent Tutoring Systems*, volume 4053 of *Lecture Notes in Computer Science*, pages 278–287. Springer.

- [Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. In *Adaptive Hypertext and Hypermedia*, pages 87–129. Kluwer Academic Publishers.
- [Calado et al., 2009] Calado, I., Barros, H., and Bittencourt, I. I. (2009). An approach for semantic web services automatic discovery and composition with similarity metrics. In *SAC*, pages 694–695.
- [Calì et al., 2009] Calì, A., Lukasiewicz, T., Predoiu, L., and Stuckenschmidt, H. (2009). Tightly coupled probabilistic description logic programs for the semantic web. *J. Data Semantics*, 12:95–130.
- [Carmagnola and Cena, 2009] Carmagnola, F. and Cena, F. (2009). User identification for cross-system personalisation. *Information Sciences: an International Journal*, 179(1-2):16–32.
- [Carmagnola et al., 2005] Carmagnola, F., Cena, F., Gena, C., and Torre, I. (2005). A multidimensional semantic framework for adaptive hypermedia systems. In *IJCAI*, pages 1551–1552.
- [Centinia et al., 1995] Centinia, F., Routen, T., Hartmann, A., and Hegartys, C. (1995). Statutor: Too intelligent by half? In Hage, J., Bench-Capon, T., Cohen, M., and van den Herik, H., editors, *Legal knowledge based systems, JURIX'95, Telecommunication and AI & Law, Koninklijke Vermande, Lelystad, Netherlands*, pages 121–132.
- [Chen and Mizoguchi, 2004] Chen, W. and Mizoguchi, R. (2004). Leaner model ontology and leaner model agent. *Cognitive Support for Learning - Imagining the Unknown*, pages 189–200.
- [Chepegin et al., 2004] Chepegin, V. I., Aroyo, L., and Bra, P. D. (2004). Ontology-driven user modeling for modular user adaptive systems. In *LWA*, pages 17–19.
- [Costa et al., 1998] Costa, E., Perkusich, A., and Ferneda, E. (1998). From a

- tridimensional view of domain knowledge to multi-agents tutoring systems. In *Intelligent Tutoring System*, pages 61–72.
- [Costa, 1997] Costa, E. B. (1997). *Um Modelo de Ambiente Interativo de Aprendizagem Baseado numa Arquitetura Multi-Agentes*. Tese de doutorado, Universidade Federal da Paraíba, Campina Grande.
- [Costa et al., 2008] Costa, M. B., Resende, R. F., Nakamura, E. F., and Segatto, M. V. (2008). Software frameworks for information systems integration based on web services. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 777–782, New York, NY, USA. ACM.
- [Cruz et al., 2007] Cruz, A. C., Alejandro Peña, Peredo, R., Sossa, H., and Gutiérrez, A. (2007). Adaptive and intelligent web based education system: Towards an integral architecture and framework. *Expert Systems with Applications*, 33(4):1076–1089.
- [Cruz et al., 2006] Cruz, A. C., Valderrama, R. P., Cano, O. F., and A., H. S. (2006). Architecture for development of wbes based on components and agents. In *15th International Conference on Computing, CIC '06*, pages 223–228.
- [Daconta et al., 2003] Daconta, M. C., Obrst, L. J., and Smith, K. T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Wiley Publishing, Inc., Indianapolis, Indiana.
- [Davis, 2008] Davis, M. (2008). Project 10x's semantic wave 2008 report. Disponível em: <http://www.project10x.com/>. Acesso em: 02 de Mar. 2009.
- [de Barros Costa et al., 2007] de Barros Costa, E., Bittencourt, I. I., Fonseca, B., Calado, I., and Maia, G. (2007). *XIII Brazilian Symposium on Multimedia and the Web*, chapter Desenvolvimento de Sistemas Inteligentes sob a perspectiva da Web Semântica, pages 1–40. SBC.
- [de Leon F. de Carvalho et al., 2006] de Leon F. de Carvalho, A. C. P., Brayner, A., Loureiro, A., Furtado, A. L., von Staa, A., de Lucena, C. J. P., de Souza, C. S., Medeiros, C. M. B., Lucchesi, C. L., e Silva, E. S., Wagner, F. R., Simon, I., Wainer, J., Maldonado, J. C., de Oliveira, J. P. M., Ribeiro, L., Velho, L., Gonçalves, M. A.,

- Baranauskas, M. C. C., Mattoso, M., Ziviani, N., Navaux, P. O. A., da Silva Torres, R., Almeida, V. A. F., Jr., W. M., and Kohayakawa, Y. (2006). Grandes desafios da pesquisa em computação no brasil - 2006 - 2016. In *Seminário sobre os Grandes Desafios da Computação no Brasil*. Sociedade Brasileira de Computação.
- [del Puerto Paule Ruiz et al., 2008] del Puerto Paule Ruiz, M., Díaz, M. J. F., Soler, F. O., and Pérez, J. R. P. (2008). Adaptation in current e-learning systems. *Comput. Stand. Interfaces*, 30(1-2):62–70.
- [Denk et al., 2007] Denk, M., Weber, M., and Belfin, R. (2007). Mobile learning challenges and potentials. *International Journal of Mobile Learning Organization*, 1(2):122–139.
- [Devedzic, 2003] Devedzic, V. (2003). Key issues in next-generation web based education. *IEEE Transaction on Education*, 3(33):339–349.
- [Devedzic, 2004] Devedzic, V. (2004). Education and the semantic web. *International Journal of Artificial Intelligence in Education*, 14:39–65.
- [Devedzic, 2006] Devedzic, V. (2006). *Semantic Web and Education*. Springer, 1 edition.
- [Dicheva and Aroyo, 2004a] Dicheva, D. and Aroyo, L. (2004a). General architecture supporting component-based eis interoperability. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), International Conference on Intelligent Tutoring Systems (ITS), Maceió, Alagoas, Brazil*.
- [Dicheva and Aroyo, 2004b] Dicheva, D. and Aroyo, L. (2004b). A pragmatic approach to support concept-based educational information systems communication. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), International Semantic Web Conference, Hiroshima, Japan*.
- [Dicheva and Dichev, 2007] Dicheva, D. and Dichev, C. (2007). Helping courseware authors to build ontologies: the case of tm4l. In *3th International Conference on Artificial Intelligence in Education, AI-ED, July 9-13, LA, California*.

- [Dicheva and Dimitrova, 2008] Dicheva, D. and Dimitrova, V. (2008). Semantic web and intelligent learning environments. In *The 9th International Conference on Intelligent Tutoring Systems (ITS), Montreal, Canada*.
- [Dietze et al., 2007a] Dietze, S., Gugliotta, A., and Domingue, J. (2007a). Context-adaptive learning designs by using semantic web services. *Journal of Interactive Media in Education (Adaptation and IMS Learning Design)*, 3:1–20.
- [Dietze et al., 2007b] Dietze, S., Gugliotta, A., and Domingue, J. (2007b). A semantic web service oriented framework for adaptive learning environments. In *ESWC '07: Proceedings of the 4th European conference on The Semantic Web*, pages 701–715, Berlin, Heidelberg. Springer-Verlag.
- [Dietze et al., 2007c] Dietze, S., Gugliotta, A., and Domingue, J. (2007c). A semantic web services-based infrastructure for context-adaptive process support. In *ICWS 2007. IEEE International Conference on Web Services, Salt Lake City, UT*, pages 537–543.
- [du Boulay and Luckin, 2001] du Boulay, B. and Luckin, R. (2001). Modelling human teaching tactics and strategies for tutoring systems. *International Journal of Artificial Intelligence in Education*, 12:235–256.
- [Dubost and Herman, 2008] Dubost, K. and Herman, I. (2008). State of the semantic web. In *INTAP Semantic Web Conference, Tokyo, Japan*.
- [FIPA, 2008] FIPA (2008). Ieee foundation for intelligent physical agents. Disponível em: <http://www.fipa.org/>. Acesso: 01 Fev. 2008.
- [Fitzgibbon and Reiter, 2003] Fitzgibbon, A. and Reiter, E. (2003). Memories for life: Managing information over a human lifetime. In *UKCRC - UK Computing Research Committee's. Grand Challenges in Computing Workshop*.
- [Freger, 2001] Freger, H. (2001). Web service conceptual architecture (wsca 1.0). Disponível em: <http://www-3.ibm.com/software/resolutions/webservicespdf/WSCA.pdf>. Acesso em: 02 de Mar. 2009.

- [Garcia-Barrios, 2006] Garcia-Barrios, V. M. (2006). Adaptive e-learning systems: retrospection, opportunities and challenges. In *28th International Conference on Information Technology Interfaces*, pages 53–58.
- [Gasevic et al., 2004] Gasevic, D., Jovanovic, J., and Devedzic, V. (2004). Enhancing learning object content on the semantic web. In *ICALT '04: Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT'04)*, pages 714–716, Washington, DC, USA. IEEE Computer Society.
- [Ghaleb et al., 2006a] Ghaleb, F., Daoud, S., Hasna, A., ALJa'am, J. M., El-Seoud, S. A., and El-Sofany, H. (2006a). E-learning model based on semantic web technology. *International Journal of Computing & Information Sciences*, 4(2):63 – 71.
- [Ghaleb et al., 2006b] Ghaleb, F. F. M., Daoud, S. S., Hasna, A. M., Jaam, J. M., and El-Sofany, H. F. (2006b). A web-based e-learning system using semantic web framework. *Journal of Computer Science*, 2(8):619–626.
- [Giannoukos et al., 2008] Giannoukos, I., Lykourantzou, I., Mpardis, G., Nikolopoulos, V., Loumos, V., and Kayafas, E. (2008). An intelligent mechanism for adaptive peer user modeling in web-based environments. In *SMAP '08: Proceedings of the 2008 Third International Workshop on Semantic Media Adaptation and Personalization*, pages 177–182, Washington, DC, USA. IEEE Computer Society.
- [Gomez-Perez et al., 2004] Gomez-Perez, G.-P., González-Cabero, R., and Lama, M. (2004). Ode sws: A framework for designing and composing semantic web services. *IEEE Intelligent Systems*, 19(4):24–31.
- [Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220.
- [Gómez-Perez and Corcho, 2002] Gómez-Perez, A. and Corcho, O. (2002). Ontology languages for the semantic web. *IEEE Intelligent Systems*, 17(1):54–60.
- [Hatem et al., 2005a] Hatem, M., Neagu, D., and Ramadan, H. (2005a). Context oriented rdf repository for semantic web: Application to sultan qaboos university web services. In *Proc. 6th Informatics Workshop, Bradford University, UK*.

- [Hatem et al., 2005b] Hatem, M., Ramadan, H. A., and Neagu, D. C. (2005b). e-learning based on context oriented semantic web. *Journal of Computer Science*, 4(1):500–504.
- [Hauger and Köck, 2007] Hauger, D. and Köck, M. (2007). State of the art of adaptivity in e-learning platforms. In *15th Workshop on Adaptivity and User Modeling in Interactive Systems*.
- [Hendler, 2001] Hendler, J. (2001). Agents and the semantic web. *IEEE Intelligent Systems*, 16(2):30–37.
- [Herman, 2008] Herman, I. (2008). *2nd European Semantic Technology Conference, Vienna, Austria*, chapter Introduction to the Semantic Web. W3C.
- [Hoare and Milner, 2004] Hoare, T. and Milner, R., editors (2004). *Grand Challenges in Computing Research Workshop*. The British Computing Society.
- [Hoolet, 2004] Hoolet (2004). Hoolet. <http://owl.man.ac.uk/hoolet>.
- [Horrocks et al., 2004] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosz, B., and Dean, M. (2004). Swrl: A semantic web rule language combining owl and ruleml. Disponível em: <http://www.w3.org/Submission/SWRL/>. Acesso em: 13 Abr. 2007.
- [HP, 2006] HP (2006). Jena - a semantic web framework for java. <http://jena.sourceforge.net>.
- [Huang et al., 2006] Huang, W., Webster, D., Wood, D., and Ishaya, T. (2006). An intelligent semantic e-learning framework using context-aware semantic web technologies. *British Journal of Educational Technology*, 37(3):351–373.
- [Huang and Stuckenschmidt, 2005] Huang, Z. and Stuckenschmidt, H. (2005). Reasoning with multi-version ontologies: A temporal logic approach,. In Springer, editor, *4th International Semantic Web Conference (ISWC 2005)*.
- [IIS, 2008] IIS (2008). Intelligent information systems research group. Disponível em: <http://iiscs.wssu.edu/xwiki/bin/view/Main/WebHome>. Acesso em: 23 Dez. 2008.

- [IMS, 2007] IMS (2007). Ims global learning consortium: Specifications. <http://www.imsglobal.org/specifications.html>. 20 Oct, 2007.
- [IS, 2008] IS (2008). Information systems group. Disponível em: <http://www.win.tue.nl/is/doku.php>. Acesso em: 23 Dez. 2008.
- [Ishaya, 2005] Ishaya, T. (2005). A framework for semantic integration of elearning services. In *International Workshop on Applications of Semantic Web Technologies for E-Learning*.
- [Isotani et al., 2008] Isotani, S., Bittencourt, I. I., Costa, E., and Mizoguchi, R. (2008). Os rumos da web semântica e da web 2.0 nos ambientes educacionais. In *Simpósio Brasileiro de Informática na Educação, Fortaleza, Ceará*.
- [Isotani et al., 2009] Isotani, S., Bittencourt, I. I., Costa, E., and Mizoguchi, R. (2009). Estado da arte em web 3.0: Potencia-lidades e tendências da nova geração de ambientes de ensino na web. *Revista Brasileira de Informática na Educação*.
- [Ivanova and Smrikarov, 2004] Ivanova, G. and Smrikarov, A. (2004). Some approaches to implementation of virtual learning environments. In *CompSysTech '04: Proceedings of the 5th international conference on Computer systems and technologies*, pages 1–6, New York, NY, USA. ACM.
- [Jang and Joo-chan, 2004] Jang, M. and Joo-chan (2004). Bossam: an extended rule engine for owl inferencing. In *Proceedings of RuleML, LNCS*, volume 3323.
- [Jennings and Wooldridge, 2000] Jennings, N. R. and Wooldridge, M. (2000). Agent-oriented software engineering. In *ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications, Minneapolis, Minnesota USA*.
- [KAON2,] KAON2. Kaon2. <http://kaon2.semanticweb.org/>.
- [Kazman and Klein, 2009] Kazman, R. and Klein, M. (2009). *Architectural Design Patterns*, volume 1. Addison Wesley.
- [Kieslinger and Simon, 2005] Kieslinger, B. and Simon, B. (2005). Elena project: Creating a smart space for learning. Technical report, Information Society Technology.

- [Koivunen and Muller, 2002] Koivunen, M.-R. and Muller, E. (2002). W3c semantic web activity. In *Semantic Web Kick-Off in Finland - Vision, Technologies, Research, and Applications*.
- [Koper, 2001] Koper, E. J. R. (2001). Modeling units of study from a pedagogical perspective - the pedagogical meta-model behind eml. In *Open University of The Netherlands*.
- [Kosba et al., 2007] Kosba, E. M., Dimitrova, V., and Boyle, R. D. (2007). Adaptive feedback generation to support teachers in web-based distance education. *User Model. User-Adapt. Interact.*, 17(4):379–413.
- [KRRRG, 2008] KRRRG (2008). Knowledge representation and reasoning research group. Disponível em: <http://www.comp.leeds.ac.uk/agc/krgroup.html>. Acesso em: 23 Dez. 2008.
- [Kumar et al., 2004] Kumar, V., Shakya, J., Groeneboer, C., and Chu, S. (2004). Toward an ontology of teaching strategies. In *Proceedings of the ITS'04 Workshop on Modelling Human Teaching Tactics and Strategies, Maceió, 2004*, pages 71–80.
- [Larman, 2004] Larman, C. (2004). *Applying UML and Patterns An Introduction to Object-Oriented Analysis and Design and the Unified Process*. Air Books, 3 edition.
- [Lesser, 1999] Lesser, V. R. (1999). Cooperative multiagent systems: A personal view of the state of the art. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):133–142.
- [lex Sakharov, 2003] lex Sakharov (2003). Horn clause in mathword. Disponível em: <http://mathworld.wolfram.com/HornClause.html>. Acesso em: 10 Mai. 2007.
- [Li et al., 2005] Li, H., Zhang, X., Wu, H., and Qu, Y. (2005). Design and application of rule based access control policies. In *Semantic Web and Policy Workshop. In 4th International Semantic Web Conference*.
- [Lin et al., 2005] Lin, F., Holt, P., Leung, S., Hogeboom, M., and Cao, Y. (2005). A multi-agent and service-oriented architecture for developing integrated and intelligent

- web-based education systems. In *In Proc. International Workshop on Semantic Web Technologies for E-learning, Maceios, Alagoas, Brazil*, pages 11–20.
- [Lin et al., 2006] Lin, F., Holt, P., Leung, S., and Li, Q. (2006). A multiagent and service-oriented architecture for developing adaptive e-learning systems. *Int. J. Cont. Engineering Education and Lifelong Learning*, 16(1/2):77–91.
- [Linwood and Minter, 2004] Linwood, J. and Minter, D. (2004). *Building Portals with the Java Portlet API (Expert’s Voice)*, volume 2. Apress.
- [Loidl, 2006] Loidl, S. (2006). Towards pervasive learning: Welearn.mobile. a cps package viewer for handhelds. *J. Netw. Comput. Appl.*, 29(4):277–293.
- [Lopez-Nores et al., 2006] Lopez-Nores, M., Pazos-Arias, J. J., Garcia-Duque, J., Blanco-Fernandez, Y., and Gil-Solla, A. (2006). A core of standards to support t-learning. In *ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, pages 13–15, Washington, DC, USA. IEEE Computer Society.
- [LORNET, 2008] LORNET (2008). Lornet: Portals and services for knowledge management and learning on the semantic web. Disponível em: <http://www.lornet.org/>. Acesso em 04 de Dez. de 2008.
- [Madmuc, 2008] Madmuc (2008). Multiagent distributed mobile and ubiquitous computing lab. Available at: <http://madmuc.usask.ca/>.
- [Maggio, 2001] Maggio, M. (2001). A importância do tutor no processo de aprendizagem a distância. *Revista Iberoamericana de Educación*, 36(3):1–5.
- [Manuel Rodrigues and Santos, 2005] Manuel Rodrigues, P. N. and Santos, M. F. (2005). Future challenges in intelligent tutoring systems - a framework. In *m-ICTE2005 3rd International Conference on Multimedia and Information and Communication Technologies in Education*. FORMATEX.
- [Markus, 2001] Markus, M. L. (2001). Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. *Journal of Management Information Systems*, 18:57–93.

- [Mcilraith et al., 2001] Mcilraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic web services. *IEEE Intelligent Systems*, 16:46–53.
- [Mcllraith et al., 2001] Mcllraith, S. A., Son, T. C., and Zeng, H. (2001). 2001, semantic web services. *IEEE Intelligent Systems*, 16(2):46–53.
- [Mercer, 2008] Mercer, J. (2008). *Web*. P.D. Publishing.
- [Millard et al., 2006] Millard, D., Doody, K., Davis, H., Howard, Y., Gilbert, L., Tao, F., and Will, G. (2006). (semantic web) services for e-learning. In *2nd International ELeGI Conference on Advanced Technology for Enhanced Learning, Barcelona, Spain*.
- [Mindswap, 2009] Mindswap (2009). Owl-s api. <http://www.mindswap.org/2004/owl-s/api/>. Acesso: 02 Jul. 2009.
- [Mitrovic and Devedzic, 2004] Mitrovic, A. and Devedzic, V. (2004). A model of multitutor ontology-based learning environments. *Int. J. Cont. Engineering Education and Lifelong Learning*, 14(3):229–245.
- [Mizoguchi, 2004] Mizoguchi, R. (2004). Tutorial on ontological engineering - part 2: Ontology development, tools and language. *New Generation Computing*, 22(1):61–96.
- [Mizoguchi and Bourdeau, 2000] Mizoguchi, R. and Bourdeau, J. (2000). Using ontological engineering to overcome common ai-ed problems. *International Journal of Artificial Intelligence in Education*, 11(2):107–121.
- [Mizoguchi et al., 2007] Mizoguchi, R., Hayashi, Y., and Bourdeau, J. (2007). Inside theory-aware and standards-compliant authoring system. In *In Proc. of SWEL: Ontologies and Semantic Web Services for Intelligent Distributed Educational Systems, AIED*, pages 1–18.
- [Murray, 2003] Murray, T. (2003). *Authoring Tools for Advanced Technologies Learning Environments: Toward cost-effective adaptive, interactive and intelligent educational software*, chapter An overview of intelligent tutoring system authoring tools: Updated analysis of the state of the art, pages 493–546. Number 17. Kluwer Academic Publishers, Netherlands.

- [Muñoz and de Oliveira, 2004] Muñoz, L. S. and de Oliveira, J. P. M. (2004). Applying semantic web technologies to achieve personalization and reuse of content in educational adaptive hypermedia systems. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL)*.
- [Neto, 2006a] Neto, W. C. B. (2006a). *Advances in Computer-Supported Learning*, volume 1, chapter Using Semantic Web Technologies Within E-Learning Applications, pages 173–201. Hershey, Pennsylvania, USA: Information Science Publishing, 1 edition.
- [Neto, 2006b] Neto, W. C. B. (2006b). *Web semântica na construção de Sistemas de aprendizagem adaptativos*. PhD thesis, Universidade Federal de Santa Catarina.
- [Neto and Gauthier, 2006] Neto, W. C. B. and Gauthier, F. A. O. (2006). Sharing and reusing information on web-based learning. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL), In Adaptive Hypermedia, Dublin, Ireland*.
- [OKBC, 1995] OKBC (1995). Open knowledge base connectivity. Disponível em: <http://www.ai.sri.com/okbc>. Acesso em: 27 Set. 2006.
- [Oliveira et al., 2009] Oliveira, M., Bittencourt, I. I., Costa, E., and Toledo, T. (2009). An authoring tool for generating wizard-based authoring tools. *IEEE Multidisciplinary Engineering Education Magazine*.
- [Oppermann, 1994] Oppermann, R. (1994). Adaptively supported adaptability. *International Journal on Human-Computer Studies*, 40(3):455–472.
- [OWL, 2006] OWL (2006). Owl web ontology language overview. Disponível em: <http://www.w3.org/TR/owl-features/>. Acesso em: 22 de Setembro de 2006.
- [P. Dillenbourg, 1992] P. Dillenbourg, J. S. (1992). A framework for learner modelling. *Interactive Learning Environments*, 2:111–137.
- [Pagani, 2003] Pagani, M. (2003). *Multimedia and Interactive Digital TV: Managing the Opportunities Created by Digital Convergence*. Irm Press.

- [Payne and Lassila, 2004] Payne, T. and Lassila, O. (2004). Guest editors' introduction: Semantic web services. *IEEE Intelligent Systems*, 19(4):14–15.
- [Plekhanova, 2002] Plekhanova, V. (2002). *Intelligent Agent Software Engineering*. Idea Group Publishing, Hershey, PA, USA.
- [Pressman and Lowe, 2009] Pressman, R. S. and Lowe, D. (2009). *Web Engineering: A Practitioner's Approach*. McGraw-Hill, 1 edition.
- [Protégé, 2007] Protégé (2007). The protégé ontology editor and knowledge acquisition system. <http://protege.stanford.edu>. 10 Oct, 2007.
- [RacerPro, 2000] RacerPro (2000). Racerpro. <http://www.franz.com/products/racer>.
- [Rollett et al., 2007] Rollett, H., Lux, M., Strohmaier, M., Dosinger, G., and Tochtermann, K. (2007). The web 2.0 way of learning with technologies. *Int. J. Learn. Technol.*, 3(1):87–107.
- [Romero et al., 2009] Romero, C., González, P., Ventura, S., del Jesus, M. J., and Herrera, F. (2009). Evolutionary algorithms for subgroup discovery in e-learning: A practical application using moodle data. *Expert Syst. Appl.*, 36(2):1632–1644.
- [Sanchez, 2006] Sanchez, E., editor (2006). *Fuzzy Logic and the Semantic Web*. Elsevier Science Publishers B. V.
- [Santos et al., 2003] Santos, O. C., Gaudioso, E., Barrera, C., and Boticario, J. G. (2003). Alfabet: An adaptive e-learning platform. In *Proceedings of mICTE Multimedia, Information and Communication Technologies, Badajoz, Spain*.
- [Schaffert et al., 2005] Schaffert, S., Bry, F., Besnard, P., Decker, H., Decker, S., Enguix, C. F., and Herzig, A. (2005). Paraconsistent reasoning for the semantic web. In *International Semantic Web Conference (ISWC-URSW)*, pages 104–105.
- [Self, 1995] Self, J. (1995). Computational mathematics: Towards a science of learning systems design. Technical report, Computer Based Learning Unit, University of Leeds.

- [Shadbolt et al., 2006] Shadbolt, N., Lee, T. B., and Hall, W. (2006). The semantic web revisited. *IEEE Intelligent Systems*, 21(3):96–101.
- [Sibaldo et al., 2008] Sibaldo, M. A. A., das Neves, A. F. R., Medeiros, F. M., Bittencourt, I. I., and de Barros Costa, E. (2008). Fraw - ambiente interativo de aprendizagem para o domínio de fração via web. In *Simpósio Brasileiro de Informática na Educação*.
- [Silva, 2003] Silva, M. (2003). *Educação online: teorias, práticas, legislação e formação corporativa*. Edições Loyola, 1 edition.
- [Simic et al., 2004] Simic, G., Gasevic, D., and Devedzic, V. (2004). Semantic web and intelligent learning management systems. In *International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL)*. In *Intelligent Tutoring System, Maceió, Alagoas, Brazil*.
- [Singh and Huhns, 2005] Singh, M. P. and Huhns, M. N. (2005). *Service-Oriented Computing: Semantics, Processes, Agents*. Wiley, 1 edition.
- [Sirin et al., 2007] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical owl-dl reasoner. *Journal of Web Semantics*, 5(2).
- [SMI, 2006] SMI (2006). Stanford medical informatics. Disponível em: <http://www.smi.stanford.edu>. Acesso em: 27 Set. 2006.
- [SPARQL, 2007] SPARQL (2007). Sparql query language for rdf. Technical report, W3C Working Draft.
- [Stollberg and Strang, 2005] Stollberg, M. and Strang, T. (2005). Integrating agents, ontologies, and semantic web services for collaboration on the semantic web. In *First International Symposium on Agents and the Semantic Web*.
- [SWRLTab,] SWRLTab. Swrltab. <http://swik.net/SWRLTab>.
- [Uresti and du Boulay, 2004] Uresti, J. A. R. and du Boulay, B. (2004). Expertise, motivation and teaching in learning companion systems. *International Journal on Artificial Intelligence and Education*, 14(2):193–231.

- [Vasilyeva et al., 2008] Vasilyeva, E., Pechenizkiy, M., and Bra, P. (2008). Adaptation of elaborated feedback in e-learning. In *AH'08: Proceedings of the 5th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 235–244, Berlin, Heidelberg. Springer-Verlag.
- [W3C, 1994] W3C (1994). World wide web consortium. Disponível em: <http://www.w3c.com>. Acesso em: 27 Set. 2006.
- [Wagner et al., 2005] Wagner, G., Giurca, A., and Lukichev, S. (2005). R2ml: A general approach for marking up rules. In Bry, F., Fages, F., Marchiori, M., and Ohlbach, H., editors, *Principles and Practives of Semantic Web Reasoning*.
- [Wang et al., 2004] Wang, E., Kim, S. A., and Kim, Y. S. (2004). A rule editing tool with support for non-programmers in an ontology-based intelligent tutoring system. In *Workshop on Semantic Web for e-Learning (SW-EL), 3rd Int'l. Semantic Web Conference (ISWC), Hiroshima, Japan*.
- [Wang and Chen, 2008] Wang, F.-H. and Chen, D.-Y. (2008). A knowledge integration framework for adaptive learning systems based on semantic web languages. In *ICALT '08: Proceedings of the 2008 Eighth IEEE International Conference on Advanced Learning Technologies*, pages 64–68, Washington, DC, USA. IEEE Computer Society.
- [Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, volume 2. Morgan Kaufmann.
- [Yeo et al., 2006] Yeo, J., Tan, S. C., and Lee, Y.-J. (2006). A learning journey in problem-based learning. In *ICLS '06: Proceedings of the 7th international conference on Learning sciences*, pages 859–865. International Society of the Learning Sciences.
- [Zambonelli et al., 2003] Zambonelli, F., Jennings, N. R., and Wooldridge, M. (2003). Developing multiagent systems: The gaia methodology. *ACM Trans. Softw. Eng. Methodol.*, 12(3):317–370.

Apêndice A

Tecnologias Utilizadas

O único lugar onde sucesso vem antes do trabalho é no dicionário.

Albert Einstein

Este apêndice tem o objetivo de descrever as principais tecnologias reusadas ao longo do desenvolvimento do MASSAYO. Para facilitar o entendimento do papel de cada tecnologia, as mesmas foram separadas por áreas, como Inteligência Artificial e Web Semântica. Além disso, é apresentado o Modelo Mathema revisitado.

A.1 Inteligência Artificial

As tecnologias de inteligência artificial utilizadas nesta tese são descritas nas subseções que seguem.

A.1.1 Jade

O JADE (*Java Agent Development Framework*) é uma plataforma *open source* para desenvolvimento de sistemas multi-agentes distribuídos, possui licença LGPL (*Library General Public License*), é implementado totalmente em Java e segue as especificações propostas pela FIPA[FIPA, 2008](*Foundation for Intelligent Physical Agents*)¹. Por essas

¹Entidade responsável pela padronização em sistemas baseados em agentes.

características, o JADE foi escolhido para o projeto, além de ser uma plataforma de fácil acesso e largamente utilizada no meio acadêmico.

Arquitetura

Um modelo abstrato para uma plataforma de agentes pode ser visto na Figura A.1, onde alguns itens importantes, definidos pela FIPA, estão identificados:

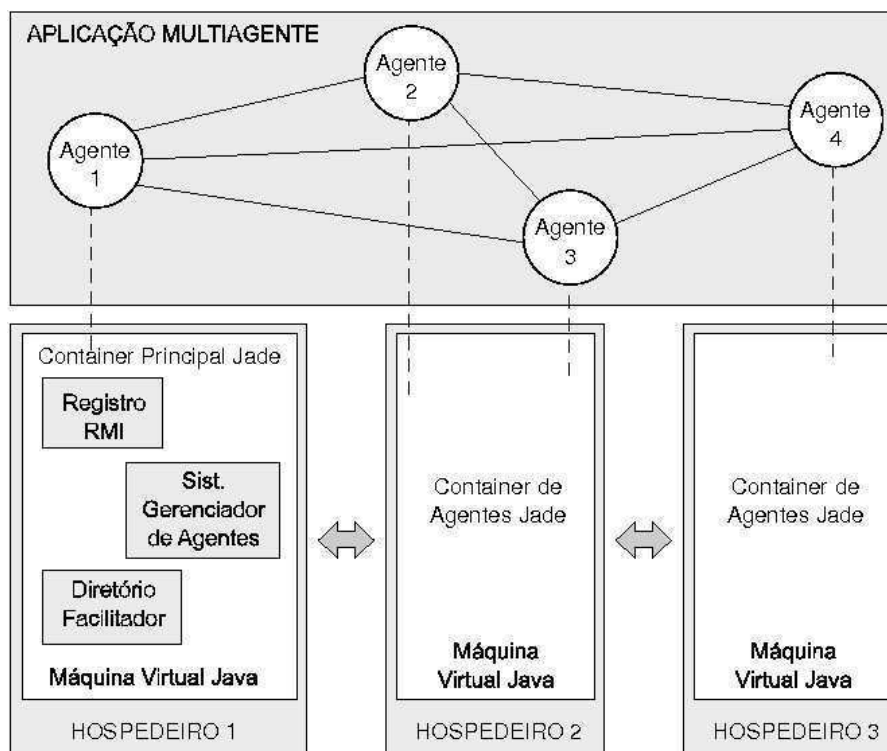


Figura A.1: Modelo de referência da FIPA para plataformas de agentes.

- *Agent Management System (AMS)*: responsável por controlar todo o acesso e uso de uma instância da plataforma. Conhecido como “serviço de páginas brancas”, existe apenas um AMS para cada instância da plataforma. Além disso, é esse agente que gerencia o ciclo de vida de todos os agentes existentes. No Serviço de páginas brancas, que está relacionado ao serviço de endereçamento, cada agente recebe uma identificação (*AID*) única no ambiente.
- *Directory Facilitator (DF)*: esse agente fornece um serviço de “páginas amarelas”, que é a catalogação dos serviços oferecidos pelos agentes na plataforma.

- *Message Transport System (MTS)*: componente para o transporte das mensagens entre os agentes, onde se pode encontrar:
 - *Agent Communication Channel (ACC)*: provê a comunicação entre os agentes, e é responsável pelas as devidas traduções em trocas de mensagens entre agentes distribuídos e localizados, em ambientes heterogêneos.
 - *Message Transport Protocol (MTP)*: utiliza-se do RMI² para troca de mensagens entre as máquinas virtuais Java (com IIOP³ ou HTTP⁴ quando entre plataformas diferentes)

O JADE utiliza-se dessa mesma arquitetura especificada pela FIPA, e um exemplo de um sistema multi-agente em JADE pode ser visto na Figura A.2. Cada *container* é uma máquina virtual Java diferente, e agrupa um conjunto de agentes. As máquinas virtuais podem estar em hospedeiros (*hosts*) diferentes e heterogêneos, graças ao serviço do *ACC*.

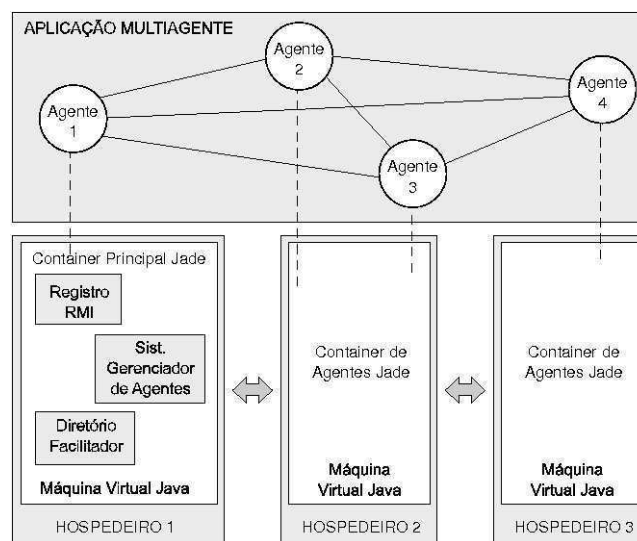


Figura A.2: Arquitetura JADE.

²Remote Method Invocation (Java RMI)

³Internet Inter ORB Protocol

⁴Hypertext Transfer Protocol

Agentes JADE

A Plataforma JADE oferece um arcabouço para construção de agentes, onde a tarefa de trocar mensagens, registro nas páginas amarelas, endereçamento dos agentes estão disponíveis para os usuários do arcabouço. Para que um agente JADE possa ser construído, basta que o usuário do arcabouço implemente a classe abstrata *jade.core.Agent*, que possui um único método abstrato chamado *setup()*, conforme pode mostrar a Figura A.3

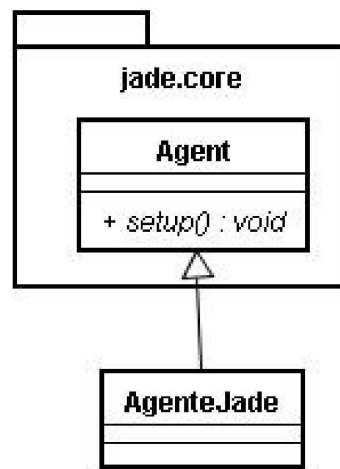


Figura A.3: Estendendo um agente JADE.

É importante frisar que é um agente JADE deve possuir conhecimento sobre três aspectos:

1. Comportamentos: cada agente pode executar diversos tipos de atividades diferentes através dos comportamentos;
2. Serviços: cada agente deve ter a capacidade de i) cadastrar os seus serviços e/ou ii) buscar serviços de outros agentes. O objetivo deste é devido a necessidade de comunicação entre os agentes;
3. Mensagens: para que um agente possa se comunicar com outro, ele deve ter a capacidade de receber e/ou enviar mensagens para outros agentes.

Nas subseções que seguem, comportamentos, serviços e mensagens são abordados.

Comportamentos

Cada agente JADE pode executar mais de uma tarefa, as quais podem ser mapeadas em comportamentos. O arcabouço faz todo o controle da execução dos comportamentos, que podem ser executados concorrentemente. JADE possui 9 comportamentos padrões que estendem da classe `jade.core.behaviours.Behaviour`, dentre os quais, destacam-se `TickerBehaviour` (sendo executado periodicamente, através de um tempo pré-estabelecido) e `OneShotBehaviour` (sendo executado apenas no momento que é chamado), como mostrado na Figura A.4. Quando um comportamento é atribuído a um agente, o mesmo solicita ao comportamento a execução do método `action()`, que se repete enquanto o método `done()` estiver retornando `false`.

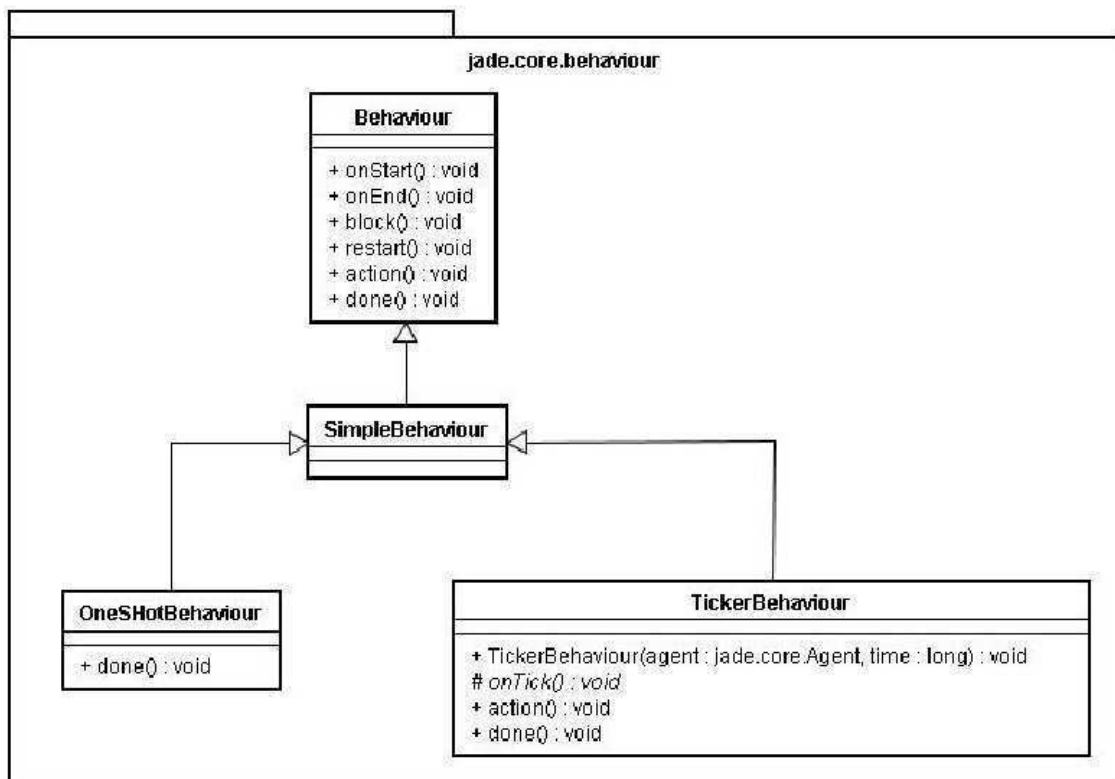


Figura A.4: Estendendo um comportamento Jade.

No Código Anotado A.1 um comportamento é adicionado ao agente, chamado `BehaviourAgent`, através do método `addBehaviour(Behaviour behaviour)` da classe `jade.core.Agent`. A implementação desse comportamento segue em seguida.

```
1 public class AgenteJade extends Agent {
2     public void setup() {
3         this.addBehaviour(new BehaviourAgent(this));
4     }
5 }
```

Código Anotado A.1: Associando um comportamento a um agente.

Serviços

Para que dois agentes possam trocar mensagens, é necessário que cada um deles conheça o endereço JADE um do outro, que está abstraído na classe *AID*. Para que isso aconteça de forma dinâmica, pelo menos um dos agentes deverá ser cadastrado nas páginas amarelas (fornecido pelo agente *Directory Facilitator*), descrevendo quais seus serviços. No Código Anotado A.2, é criado um método que tem por objetivo cadastrar serviços nas páginas amarelas.

```
1 public boolean insertIntoYellowPages
2     (String serviceType, String serviceName) {
3     DFAgentDescription dfAgentDescription =
4         new DFAgentDescription();
5     ServiceDescription serviceDescription =
6         new ServiceDescription();
7     serviceDescription.setName(serviceName);
8     serviceDescription.setType(serviceType);
9     dfAgentDescription.addServices(serviceDescription);
10    try {
11        DFService.register(this, dfAgentDescription);
12        return true;
13    } catch (FIPAException e) {
14        return false;
15    }
16 }
```

Código Anotado A.2: Cadastro nas Páginas Amarelas.

Esse cadastro é executado pelo método *insertIntoYellowPages(String serviceType, String serviceName)*, que executa os seguintes passos:

1. Na linha 3, é criada uma instância de *DFAgentDescription*. Essa classe representa uma descrição de um agente para as páginas amarelas;
2. Na linha 5, foi criada uma instância de *ServiceDescription*. Essa classe representa uma abstração de um serviço nas páginas amarelas. Nas linhas 7 e 8, foram

- atribuídas a essa instância, o nome e o tipo de serviço desejado;
3. Na linha 9, a descrição do serviço (*serviceDescription*) foi atribuída à instância da classe *DFAgentDescription*, chamada *dfAgentDescription*;
 4. Na linha 11, o serviço descrito em *dfAgentDescription* foi registrado em *DFService*. A classe *DFService* é uma abstração do agente páginas amarelas (*Directory Facilitator*).

Além de cadastrar um serviço nas páginas amarelas, o agente pode fazer consultas para saber quais agentes são responsáveis por determinados serviços. Tal implementação é mostrada no Código A.3

```

1   public AID[] getAIDsFromYellowPages(String serviceType,
2                                     String serviceName) {
3       DfAgentDescription dfAgentDescription = new DfAgentDescription();
4       ServiceDescription serviceDescription = new ServiceDescription();
5
6       serviceDescription.setName(serviceName);
7       serviceDescription.setType(serviceType);
8       dfAgentDescription.addServices(serviceDescription);
9
10      try{
11          DfAgentDescription results[] =
12              DFService.search(this,dfAgentDescription);
13          AID aid[] = new AID[results.length];
14          for (int i = 0; i < results.length; i++) {
15              aid[i] = results[i].getName();
16          }
17          return aid;
18      }catch(FIPAException e) {
19          return null;
20      }
21
22  }
```

Código Anotado A.3: Cadastro nas Páginas Amarelas.

Esse busca é executada pelo método *getAIDsFromYellowPages(String serviceType, String serviceName)*, que executa os seguintes passos:

1. Da linha 3 a 8, ocorre da mesma forma que o cadastro de um serviço nas páginas amarelas;

2. Nas linhas 11 e 12, o método *DFService.search* faz uma busca pelos agentes que fornecem um determinado serviço;
3. Das linhas 13 a 20, é feito o tratamento do método *DFService.search*, que retorna um *Array* de *AID*'s, onde cada *AID* representa um endereço JADE;

Mensagens

A interação entre os agentes ocorre através da troca de mensagens. Para que um agente possa interagir, o mesmo deve criar um comportamento que tem a capacidade de receber e/ou enviar mensagens para outros agentes, através de uma linguagem de comunicação entre agentes⁵, como mostrado no Código A.4.

```
1 public class MessageReceiver extends TickerBehaviour{
2     public MessageReceiver (Agent agent) {
3         super (agent, 1000);
4     }
5
6     public void onTick() {
7         ACLMessage receivedMessage = agent.receive();
8         if (receivedMessage != null) {
9             //Treatment of the message
10        }
11    }
12 }
```

Código Anotado A.4: Comportamento (*Behaviour*) *MessageReceiver*.

O comportamento *MessageReceiver* foi estendido da classe *TickerBehaviour*, cuja qual possui um único método abstrato chamado *onTick()*, que deverá ser implementado. Esse método é executado a cada período de tempo definido em seu construtor. Neste caso específico (no código anterior), a cada um segundo (1000 milissegundos) o comportamento *MessageReceiver* executa as seguintes tarefas:

1. Na linha 7, o comportamento solicita ao seu agente que receba a última mensagem, através do método *agent.receive()*.
2. Na linha 8, o comportamento verifica se a mensagem recebida não é nula. Não sendo, ele tratará a mensagem a sua maneira.

⁵Esta linguagem é chamada Agent Communication Language (ACL)

Para que o tratamento seja feito, o agente precisa recuperar o conteúdo. O conteúdo de uma mensagem pode ser adquirido pelo agente de duas formas:

1. O agente recebe o conteúdo em uma *String*, através do método *getContent()*. Para enviar o conteúdo de uma *String*, deve-se usar o método *setContent()*;
2. O agente pode receber o conteúdo em um Objeto, através do método *getContentObject()*. Tal objeto só irá funcionar se o mesmo implementar a Interface *Serializable*. Para enviar o conteúdo de uma *String*, deve-se usar o método *setContentObject()*;

Por último, segue o código do agente *AgentMessageSender*, exibido no Código Anotado A.5, que está enviando uma mensagem para o agente *AgentMessageReceiver*, que executa essa tarefa segundo os seguintes passos:

1. Na linha 5, o agente *AgentMessageSender* procura nas páginas amarelas quem está recebendo mensagens com o nome *AgentMessageReceiver* e do tipo *ReceiveMessage*, através do método *getAIDsFromYellowPages*, mostrado no Código A.3.
2. Na linha 7, uma mensagem no formato JADE é representada por uma instância da classe *ACLMessage*, para ser posteriormente enviada ao agente *AgentMessageReceiver*.
3. Na linha 10, o agente *AgentMessageSender* atribui o endereço do remetente à mensagem, nesse caso, o endereço do agente *AgentMessageReceiver*.
4. Na linha 11, a mensagem recebe como conteúdo a *String* contendo "*Message Content*".
5. Na linha 12, o agente *AgentMessageSender* envia a mensagem, através do método *send* disponibilizado pela classe *Agent*.

```
1 public class AgentMessageSender extends Agent {
2
3     public void setup() {
4
5         AID[] aid = this.getAIDsFromYellowPages
6             ("ReceiveMessage", "AgentMessageReceiver");
7         ACLMessage message = new ACLMessage();
8
9         if (aid[0] != null) {
10            message.addReceiver(aid[0]);
11            message.setContent("Message Content");
12            this.send(message);
13        }
14
15    }
16
17 }
```

Código Anotado A.5: Agente que envia mensagens.

A.2 Web Semântica

As tecnologias da web semântica utilizadas nesta tese são descritas nas subseções que seguem.

A.2.1 OWL

OWL⁶ [OWL, 2006] é a linguagem de ontologias utilizada para Web Semântica, a qual foi padronizada pela W3C [W3C, 1994]. OWL possui um alto poder de expressividade e capacidade de inferência, sendo garantido por três dialetos diferentes [Baader et al., 2003]:

- *OWL Lite*: equivale a uma sub-linguagem bastante simples, podendo ser usada quando somente uma simples hierarquia ou restrições são necessárias;
- *OWL-DL*⁷: possui maior expressividade que *OWL-Lite*, sendo baseado na Lógica de Descrição⁸. Com *OWL-DL* é possível verificar inconsistências na ontologia e inferir a classificação da hierarquia;

⁶Do inglês: Ontology Web Language

⁷DL significa *Description Language*, ou seja, lógica de descrição.

⁸Lógica de Descrição representa uma parte da lógica de primeira ordem.

- *OWL FULL*: representa a mais expressiva das sub-linguagens OWL, devendo ser utilizada em situações onde a necessidade de expressividade é maior que a garantia de computabilidade.

Os componentes presentes em OWL são três, sendo eles: *i) Classes*: equivalendo a uma representação concreta de termos. Além disso, são representadas como um conjunto de contém indivíduos; *ii) Propriedades*: equivalem a relações binárias de indivíduos; e *iii) Indivíduos*: representam objetos em um determinado domínio que foi formalizado, sendo referenciados como instâncias de classes.

Propriedades

Um destaque maior será dado às propriedades, pois possuem características fundamentais para a construção de qualquer ontologia em OWL. Uma propriedade pode ter diversas características, dentre as quais são citadas:

Funcional: uma propriedade é funcional quando o seu indivíduo pode possuir no máximo um valor;

Funcional inversa: significa que a propriedade inversa é funcional, ou seja, significa que esta propriedade pode ter no máximo um indivíduo através da propriedade inversa;

Transitiva: se uma propriedade é transitiva, isto quer dizer que em situações onde um indivíduo **a** tem relação com um indivíduo **b**, e **b** tem relação com **c**, então, pode-se inferir que o indivíduo **a** tem relação com o indivíduo **c**;

Simétrica: propriedades simétricas significam que, quando um indivíduo **b** está relacionado com **a**, quer dizer que o indivíduo **a** também está relacionado com o indivíduo **b**.

Além disso, as propriedades possuem restrições, especificadas em lógica de descrição. Os tipos de restrições são:

Quantificadores: as restrições deste tipo podem ser duas: *i)* quantificador existencial (\exists), tendo que possuir "pelo menos um" indivíduo e *ii)* quantificador universal (\forall), sendo lido como "somente" estes tipos de indivíduos;

Cardinalidade: as restrições deste tipo são de três tipos: i) Cardinalidade mínima, sendo dito qual é a mínima quantidade de indivíduos da propriedade; ii) Cardinalidade Máxima, sendo dito qual é a máxima quantidade de indivíduos da propriedade e iii) Cardinalidade Exata, informando quantos indivíduos exatamente a propriedade deve ter.

hasValue: esta restrição, denotada pelo símbolo (\exists), descrevendo que um indivíduo, através de uma propriedade, tem pelo menos um relacionamento com outro indivíduo.

A.2.2 Implementações de SWRL

SWRL é implementada por uma série de frameworks, porém ela é raramente implementada completamente porque ela não é decidível. Algumas implementações famosas encontram-se descritas na Tabela A.1.

Tabela A.1: Construtores de OWL

Framework	Descrição
SWRLTab [SWRLTab,]	Extensão da ferramenta Protégé que suporta edição e execução de regras
R2ML [Wagner et al., 2005]	Ferramenta que suporta
Bossam [Jang and Joo-chan, 2004]	Uma engine que suporta SWRL
Hoolet [Hoolet, 2004],	Uma implementação de um raciocinador de uma OWL-DL que usa um aprovador de lógica de prim
Pellet [Sirin et al., 2007]	Uma raciocinador open-source OWL DL em Java que tem suporte a
KAON2 [KAON2,]	É uma infra-estrutura para mapear OWL DL, SWRL e ontologias em
RacerPro [RacerPro, 2000]	Suporta processamento de regras em uma sintaxe baseada em SWRL por trad

A.2.3 Protégé

Protégé [Protégé, 2007] é um ambiente integrado para a modelagem e a aquisição de conhecimento voltado para o desenvolvimento de ontologias (Vide Figura A.5), desenvolvido pelos pesquisadores do *Stanford Medical Informatics* [SMI, 2006]. O desenvolvimento de ontologias no protégé pode ser feito de duas maneiras, utilizando o i) *Protégé-Frames*: o usuário pode construir e povoar ontologias que são baseadas em *frames*, em conformidade com *Open Knowledge Base Connectivity* [OKBC, 1995] criar

frames ou ii) *Protégé-OWL*: o usuário pode criar ontologias para a Web Semântica, em particular em W3C's *Ontology Web Language*.

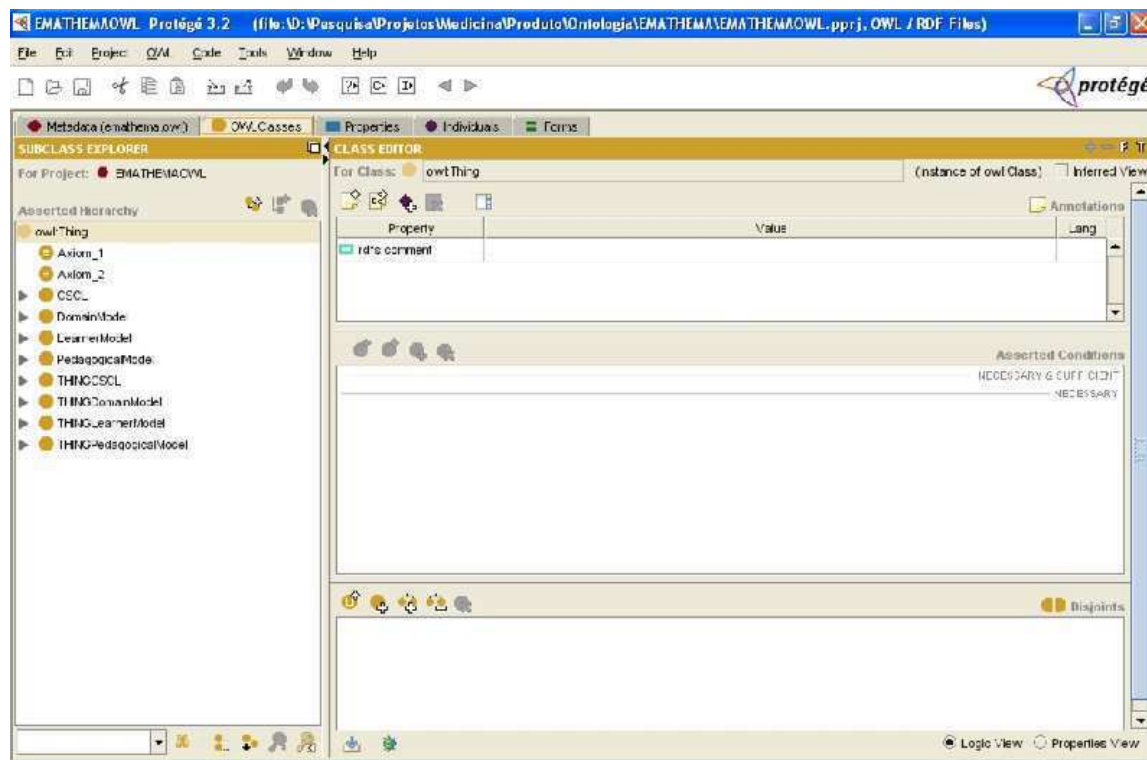


Figura A.5: Tela Inicial do Protégé.

Esta ferramenta é usada por desenvolvedores de sistemas e especialistas do domínio para construir sistemas baseados em conhecimento, pois oferece vantagens como:

- Livre;
- Código aberto;
- Pode ser exportado em uma variedade de formatos, como RDF(S), OWL, XML *Schema*, outros;
- Baseado em Java;
- É extensível e disponibiliza um ambiente baseado em plug-ins, tornando-o flexível para o desenvolvimento de aplicação e rápida prototipação;
- Disponibiliza diversos plug-ins para uso;

- Possui documentação abordando o uso da ferramenta;
- Possui listas de discussões e uma equipe de desenvolvimento dando continuidade na geração de novas versões;
- É suportado por uma comunidade de mais de 62.000 (sessenta e dois mil) usuários.

Atualmente o protégé está na versão 3.2.1., onde o principal foco desta versão é o desenvolvimento de ontologias em *OWL*. Além de criar ontologias em *OWL*, o desenvolvedor poderá fazer o gerenciamento de mudanças, depurar ontologias em *OWL-DL*, dentre outras características. O grupo de desenvolvedores responsável pelo protégé já está desenvolvendo a versão 4.x.

Protégé-OWL

O Protégé-OWL é uma extensão do Protégé para dar suporte a *Web Ontology Language* (*OWL*). Dentre diversas funcionalidades, o editor para o Protégé-OWL permite ao desenvolvedor:

- Carregar e salvar ontologias nos formatos *OWL* e *RDF*, como mostrado na Figura A.6;

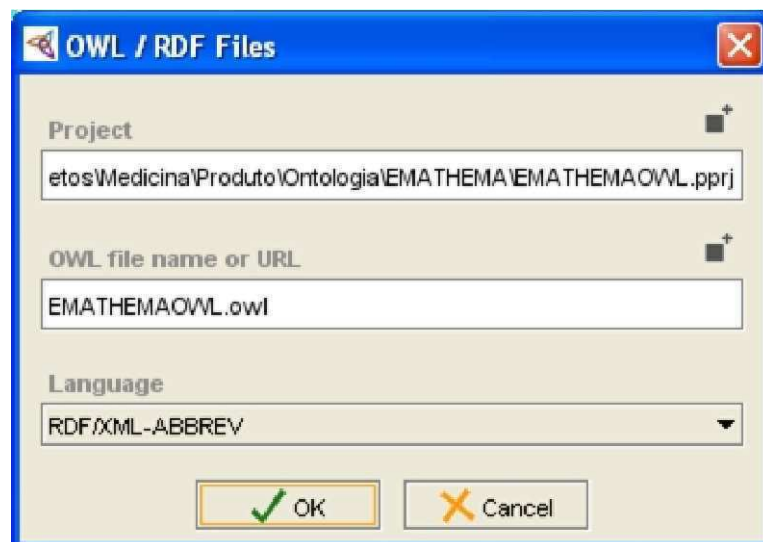


Figura A.6: Tela de Salvar do Protégé.

- Editar e visualizar classes, propriedades (Vide Figura A.7) e regras (*SWRL*⁹);

⁹Do Inglês: Semantic Web Rule Language.

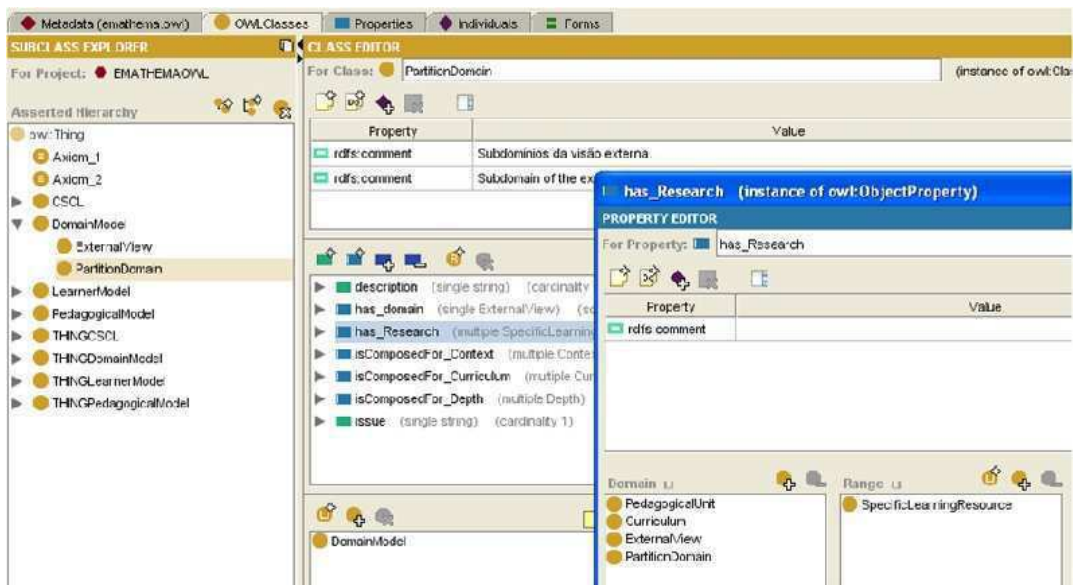


Figura A.7: Editor de Classes e Propriedades do Protégé.

- Definir características lógicas das classes através de expressões OWL-DL, como mostrado na Figura A.8;

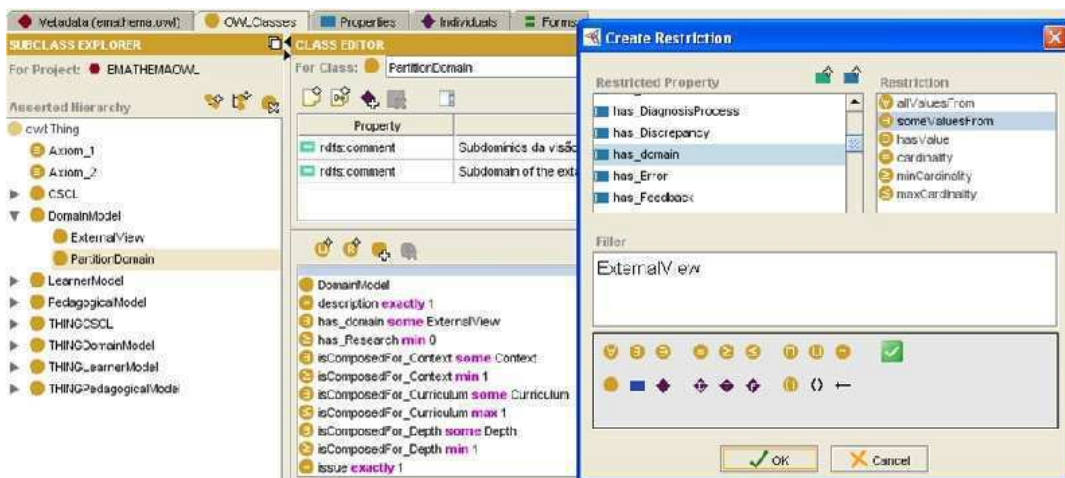


Figura A.8: Tela do Protégé com especificação em OWL-DL.

- Editar indivíduos OWL (Vide Figura A.9).

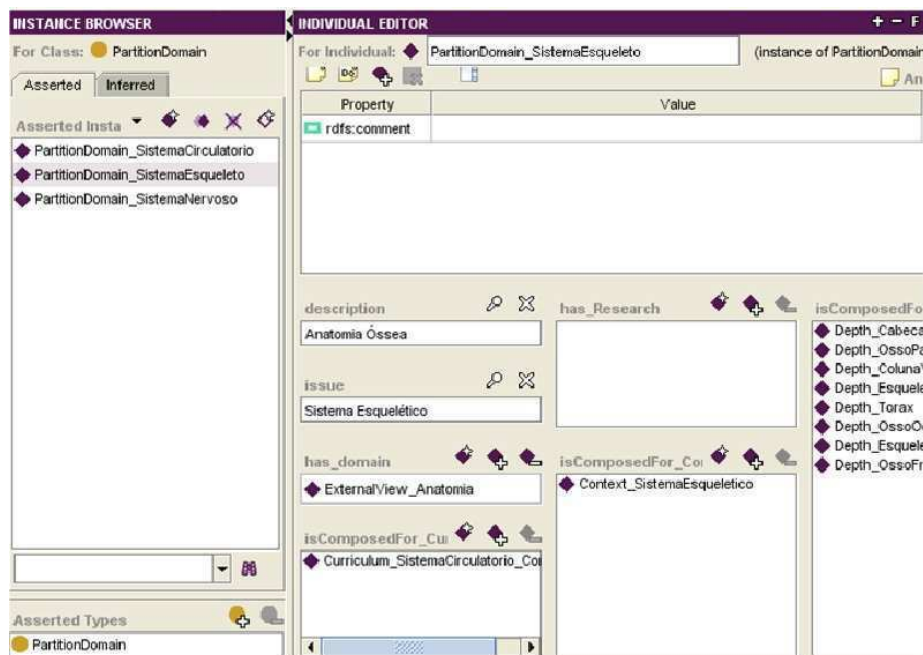


Figura A.9: Tela de Indivíduos do Protégé.

ProtégéOWL-API

O Protégé disponibiliza uma API para o desenvolvedor poder trabalhar diretamente da linguagem Java. Esta API provê classes e métodos para carregar e salvar arquivos OWL, consultar e manipular modelos de dados OWL e executar inferências. Uma das grandes vantagens em se utilizar esta API é devido a abstração de Jena¹⁰, ou seja, o desenvolvedor não precisará implementar diretamente utilizando Jena, pois a API gera uma camada que facilita a manipulação de arquivos OWL. O Código A.6 aborda o carregamento e salvamento de um arquivo OWL utilizando a API do Protégé.

O método *load(final String owlFile)* é responsável pelo carregamento de um arquivo OWL. Os passos para o carregamento são:

1. Na linha 3, é feito o carregamento do arquivo através do método *loadProjectFromFile(owlFile, getErrors())* da classe Project. Esta classe permite

¹⁰Jena equivale a um *framework* para a Web Semântica [HP, 2006].

```
1 public void load(final String owlFile) {
2     this.errors = new ArrayList();
3     this.project = Project.loadProjectFromFile(owlFile, getErrors());
4     this.owlModel = (OWLModel) getProject().getKnowledgeBase();
5 }
6
7 public void save() {
8     this.project.save(getErrors());
9 }
```

Código Anotado A.6: Carregamento e Salvamento de um arquivo OWL.

manipular projetos criados em *Protégé*¹¹. Já o método possui dois atributos, sendo o primeiro o caminho do arquivo do projeto OWL e o segundo uma *Collection* que será preenchida caso algum erro ocorra no carregamento do arquivo;

2. Na linha 4, a base de conhecimento é passada para uma propriedade do tipo *OWLModel*. *OWLModel* é uma interface equivalendo a uma base de conhecimento com métodos para manipular o arquivo OWL.

Além disso, o desenvolvedor pode fazer todo o mapeamento da ontologia em código Java. Para isso, o desenvolvedor deve seguir os seguintes passos:

1. Gerar a classe com o "esquema" de toda a ontologia OWL, ou seja, toda a estrutura do OWL será mapeada nesta classe. A Figura A.10 aborda onde ocorre o processo. Para o mapeamento, o desenvolvedor deve informar o local e o pacote que a classe será salva;

¹¹Observe que a classe *Project* faz com que haja necessidade do arquivo de projeto do Protégé (*.pprj) para o carregamento do arquivo OWL.

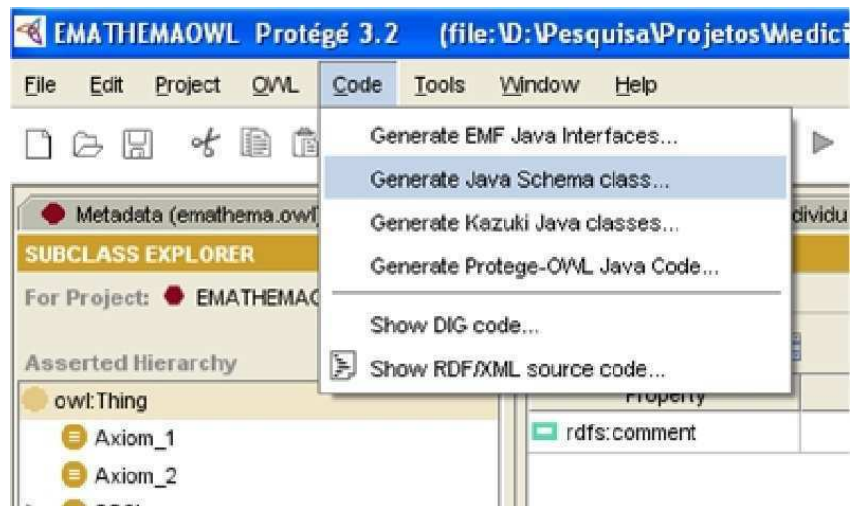


Figura A.10: Gerar a classe com o "esquema" de toda a ontologia OWL.

2. Gerar as classes e interfaces. O Protégé possui um *plugin* (Vide Figura A.11) que faz o mapeamento da ontologia em classes Java.

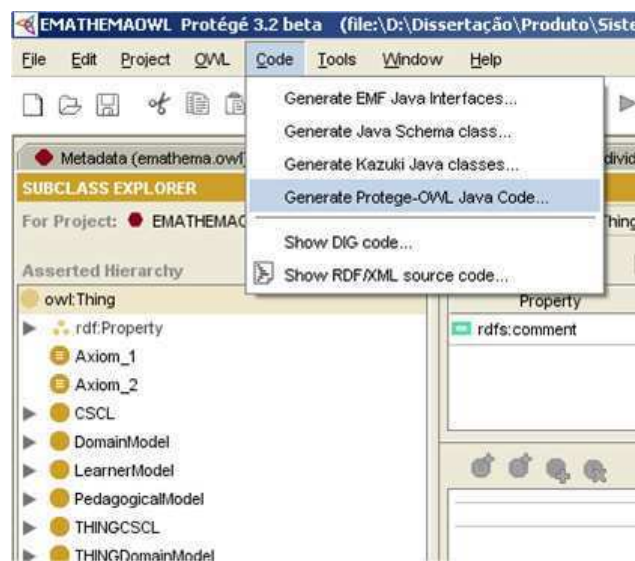


Figura A.11: Mapeamento Ontologia-Objeto através do Protégé.

Para a geração do código (Vide Figura A.12), é necessário configurar *i*) a pasta de geração do código, *ii*) o pacote e *iii*) o nome da classe de fábrica.



Figura A.12: Tela do Protégé com o Mapeamento Ontologia-Objeto.

É gerada, para cada classe da ontologia, uma interface¹² e uma classe concreta. A interface possui a assinatura dos métodos da classe, onde estes métodos equivalem aos *getters* e *setters*¹³ dos *slots*. A classe possui as regras da classe definidas em lógica de descrição e a implementação dos métodos da interface, é mostrada na figura A.13.

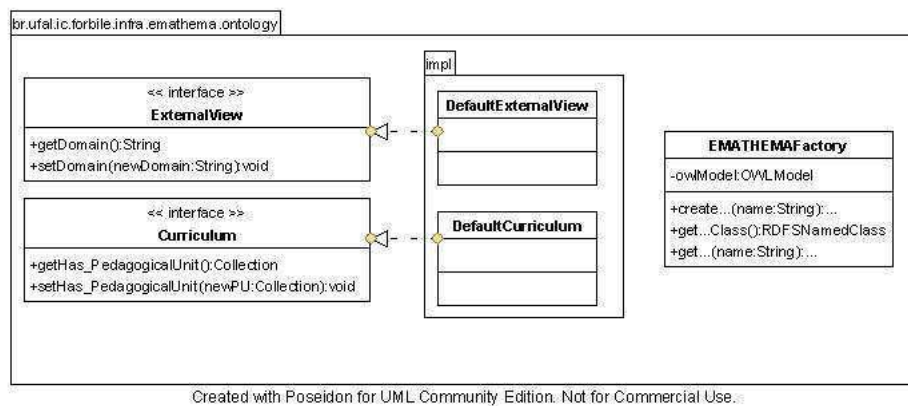


Figura A.13: Pacote com as classes do Mapeamento Ontologia-Objeto.

A classe *EMATHEMAFactory* é responsável pela construção dos objetos da ontologia. Além disso, esta classe possui o modelo OWL da ontologia (*OWLModel*). A propriedade

¹²É importante frisar que toda interface gerada pela API herda da Interface *OWLIndividual*.

¹³São métodos padrões utilizados no desenvolvimento de software orientados a objetos, onde objetiva-se atribuir e recuperar valores de propriedades.

OWLModel equivale a instância da ontologia. O Código A.7 mostra um trecho da Classe *EMATHEMAFactory*.

```
1 public class EMATHEMAFactory {
2
3     public EMATHEMAFactory(OWLModel owlModel) {
4         this.owlModel = owlModel;
5     }
6
7     public ExternalView createExternalView(String name) {
8         final RDFSNamedClass cls = getExternalViewClass();
9         return (ExternalView) cls.createInstance(name).as(ExternalView.class);
10    }
11
12    public ExternalView getExternalView(String name) {
13        return (ExternalView) owlModel.getRDFResource(name)
14            .as(ExternalView.class);
15    }
16 }
```

Código Anotado A.7: Classe *EMATHEMAFactory* gerado através da API do Protégé-OWL.

Além disso, pode-se fazer consultas através da implementação de forma bastante simples¹⁴, sendo estas de duas formas:

1. Recuperando indivíduos de alguma classe da ontologia, como mostra o Código A.8;

```
1 public retrievePartitionDomain(String owlFile, String individual) {
2     this.errors = new ArrayList();
3
4     this.project = Project.loadProjectFromFile(owlFile, this.getErrors());
5     this.owlModel = (OWLModel) this.getProject().getKnowledgeBase();
6
7     this.emathema = new EMATHEMAFactory(getOwlModel());
8
9     PartitionDomain c = this.emathema.getCurriculum(individual);
10 }
11
```

Código Anotado A.8: Recuperação de indivíduo da classe *PartiionDomain*.

2. Através de SPARQL¹⁵, como mostrado no Código A.9.

¹⁴Tal simplicidade é garantida devido a camada de abstração de Jena, que não foi diretamente utilizada.

¹⁵SPARQL equivale a uma linguagem de consulta para Web Semântica [SPARQL, 2007]

```
1 public void retrieve(String querySPARQL) {
2
3     QueryResults qr = null;
4
5     qr = this.owlModel.executeSPARQLQuery(querySPARQL);
6
7 }
```

Código Anotado A.9: Recuperação de indivíduos através de SPARQL.

A.2.4 Modelo Mathema Revisitado

Esta subseção aborda características do modelo Mathema para AIA baseado em Agentes, proposto por Evandro Costa em sua tese de Doutorado [Costa, 1997] bem como as adaptações necessárias do modelo. A seguir, são abordados os aspectos conceituais do modelo Mathema, seguidos das adaptações e integrações.

Modelo Mathema

O Modelo Mathema foi utilizado como uma das bases conceituais deste trabalho, principalmente por apresentar um modelo de ambiente interativo de aprendizagem baseado numa arquitetura multi-agentes, sendo um representante adequado e em conformidade com as concepções de tais ambientes de suporte ao aprendizado mediado por computador.

A especificação do modelo mínimo de aprendizagem do Mathema equivale à idéia de um jogo. O jogo ocorre quando os dois jogadores (Estudante e Sistema Tutor) iniciam uma interação através da troca de mensagens, como mostrado na Figura A.14, visando alcançar os critérios de satisfação de ambas as partes.

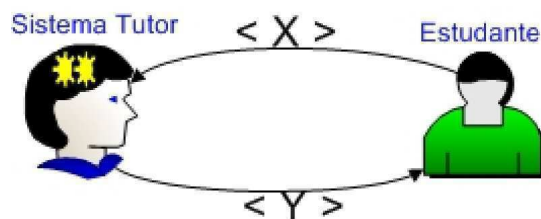


Figura A.14: Modelo Mínimo do MATHEMA

Fonte: Extraída e adaptada de [Costa, 1997].

De uma maneira mais abstrata, o processo de interação é visto como ocorrendo

dentro de um esquema de troca de mensagens (X, Y) entre as duas entidades envolvidas, obedecendo a certo protocolo [Costa, 1997]. O grau de aprendizagem do estudante se dará através de ações sobre o conteúdo das mensagens. Dentre as exigências necessárias para garantir a efetividade no processo de interação, torna-se necessário, do lado do sistema tutor, suporte aos seguintes aspectos:

- conhecimento sobre o domínio;
- conhecimento pedagógico;
- conhecimento sobre o estudante;
- capacidade de interação.

As necessidades supracitadas remetem primeiramente à demanda por uma modelagem do conhecimento através de uma visão dimensional (visão externa) sobre este domínio, a qual ajudará o seu posterior particionamento (conduzindo a uma visão interna). A visão externa equivale a uma interpretação de um dado domínio de conhecimento, enquanto a visão interna equivale a um particionamento do domínio. Tal investimento de modelagem conduziu a uma visão tridimensional associada ao domínio, como mostrado na Figura A.15, onde as dimensões são [Costa et al., 1998]:

1. Contexto: que equivale a um ponto de vista sobre um domínio de conhecimento, ou seja, cada contexto diz respeito a um domínio;
2. Profundidade: é definida relativamente a um dado contexto, sendo concernente a alguma forma de refinamento na linguagem de percepção, ou seja, a estratificação dos vários níveis epistemológicos de percepção do objeto de conhecimento em apreciação;
3. Lateralidade: diz respeito aos conhecimentos afins de suporte a um dado objeto de conhecimento do domínio alvo, proveniente duma visão particular de contexto e profundidade, constituindo-se, no escopo do presente trabalho, em pré-requisitos da referida unidade.

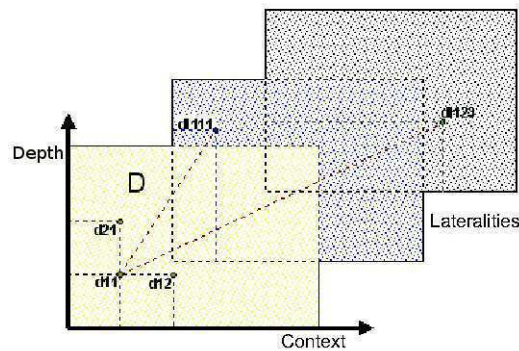


Figura A.15: Visão tridimensional do modelo do domínio no Mathema.

Fonte: Extraída de [Costa, 1997].

Cada partição de D origina subdomínios, os quais passam a ser mapeados em estruturas curriculares. Uma visão interna do currículo é composta por unidades pedagógicas (up), da forma

$$Curric = \{up_1, up_2, \dots, up_n\}, \quad (A.1)$$

onde $Curric$ denota um currículo, considerando que cada up_i denota uma unidade pedagógica do $Curric$. Estas unidades estão relacionadas segundo uma ordem definida com base em critérios pedagógicos. A cada up_i corresponde um conjunto de problemas. Cada problema está associado a um conhecimento de suporte à sua resolução, incluindo basicamente conceitos e resultados, como mostrados na Figura A.16.

Cada subdomínio será representado por dois tipos de agente tutor (AT), um equivalendo ao currículo do subdomínio e outro pela lateralidade/pré-requisito do currículo (ATL). Com isso, o conjunto de todos os agentes do domínio forma uma sociedade de agentes tutores autônomos, sendo:

$$SATA = AT \cup ATL. \quad (A.2)$$

A inserção dos agentes faz com que a interação presente no jogo (Figura A.14) evolua como mostrada na Figura A.17

Além do estudante e do sistema tutor multi-agente, dois aspectos são fundamentalmente importantes para o processo de aprendizagem, sendo eles: *i*) um mediador extra-ambiente para interação com estudantes, e *ii*) a manutenção do

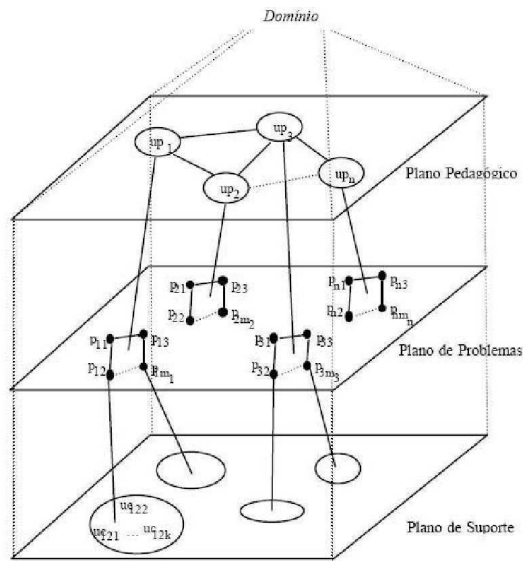


Figura A.16: Estrutura pedagógica do conhecimento do domínio no Mathema.

Fonte: Extraída de [Costa, 1997].

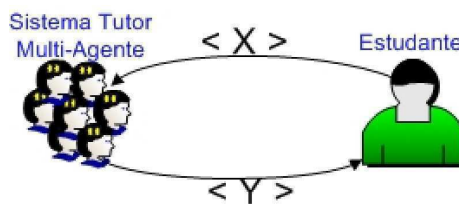


Figura A.17: Modelo de Ambiente de Aprendizagem baseado em agentes no Mathema.

Fonte: Extraída e adaptada de [Costa, 1997].

conhecimento curricular presente nos agentes tutores por especialistas humanos. Tais necessidades evoluem para a arquitetura do Mathema (Figura A.18).

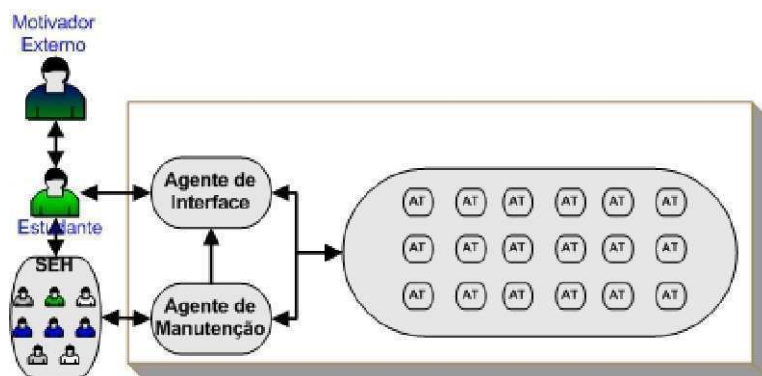


Figura A.18: Arquitetura do Mathema.

Fonte: Extraída e adaptada de [Costa, 1997].

Ressalta-se ainda que a especificação interna do agente tutor é um aspecto elementar para possibilitar a interação estudante-tutor. Internamente, este agente é formado por três camadas (vide Figura A.19), sendo elas:

1. Sistema Tutor: responsável pelas atividades de tutoria no processo de interação estudante-tutor. O resolvidor de tarefas representa as unidades pedagógicas no processo de resolução de problemas;
2. Sistema Social: possibilita ao Sistema Tutor interagir com outros agentes no processo de resolução de problemas. Além disto, é nesta camada que a manutenção dos agentes ocorre;
3. Sistemas de Distribuição: responsável pela manipulação das mensagens recebidas e enviadas.

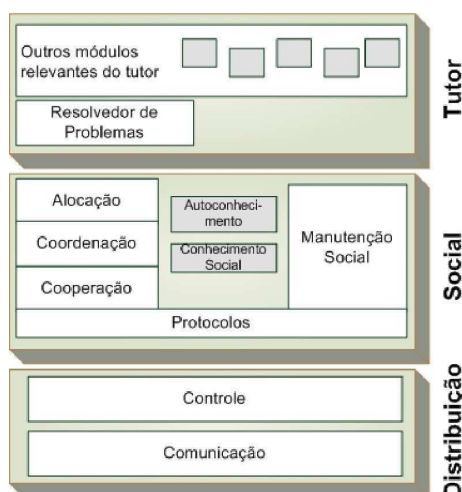


Figura A.19: Arquitetura de um Agente Tutor no Mathema: visão interna

Fonte: Extraída e adaptada de [Costa, 1997].

Com a especificação do modelo Mathema definida, serão abordados a seguir as adaptações e integrações necessárias para possibilitar a construção do arcabouço nos moldes definidos anteriormente.

A.2.5 Adaptações e Integrações ao Modelo MATHEMA

A primeira adaptação que ocorreu no modelo Mathema foi a definição das informações curriculares. Antes, um currículo era associado apenas a um subdomínio relacionado a

uma dada partição, porém, duas necessidades foram identificadas: *i*) uma partição do domínio pode ter subpartições do domínio; *ii*) uma lateralidade passa a ser um currículo.

A mudança na lateralidade foi motivada pelo seguinte aspecto: antes, a lateralidade vista como um currículo não era uma partição do domínio. Porém, tal lateralidade, em outro domínio, equivalerá a uma partição do domínio. Conseqüentemente, esta lateralidade deverá ser mapeada em um currículo, no outro domínio.

A segunda mudança ocorreu na estrutura pedagógica do conhecimento do domínio (Vide Figura A.16). Dois aspectos foram adicionados à estrutura: *i*) uma camada abaixo de unidade do conhecimento, chamada de Plano Comportamental, e *ii*) uma camada presente em todos os níveis da hierarquia, chamada de Recurso de Aprendizagem.

O Plano Comportamental foi definido através da integração com [P. Dillenbourg, 1992], onde é feita uma separação entre o comportamento conceitual e conhecimento conceitual. O Comportamento conceitual possui uma estrutura de inferência para o processo de resolução de problemas. O Conhecimento conceitual contém descrições conceituais dos problemas, tanto em largura quanto em profundidade.

A camada de Recurso de Aprendizagem foi adicionada para dar suporte, tanto as diversas estratégias pedagógicas quanto ao caráter evolutivo dos recursos de aprendizagem. Com isso, tal mudança faz com que a estrutura pedagógica fique como mostrada na Figura A.20.

As duas mudanças anteriormente citadas fazem com que haja um efeito cascata, tanto na sociedade de agentes quanto na arquitetura do Mathema.

A SATA equivalia à união dos agentes tutores com os agentes tutores de lateralidade (Vide Figura A.2), porém, a primeira mudança faz com que o agente tutor de lateralidade seja um agente tutor. Além disso, a segunda mudança motiva a utilização de agentes inteligentes para a resolução de problemas. Com isso, a sociedade de agentes é definida como sendo:

$$SA = ATA \cup ADA, \quad (\text{A.3})$$

onde *ATA* equivale aos agentes tutores autônomos e *AS* aos agentes de suporte. Além disso, um agente mediador foi adicionado para dar subsídio à sociedade de agentes, para a coordenação do processo de resolução de problemas e manutenção do conhecimento

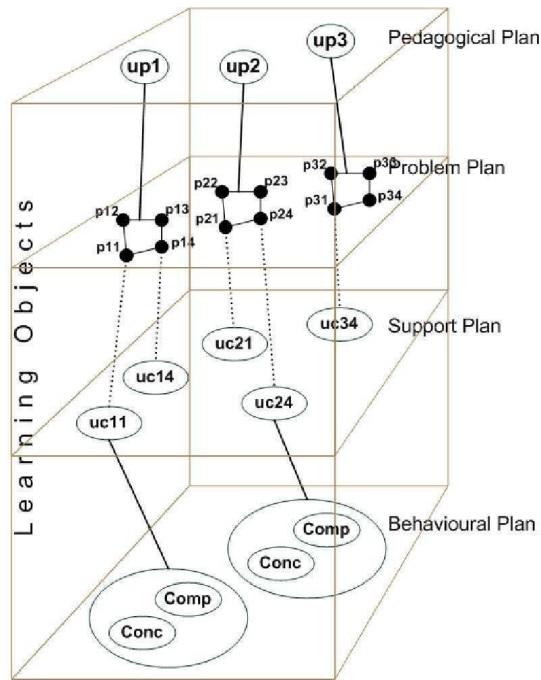


Figura A.20: Estrutura pedagógica do conhecimento do domínio adaptada

social.

Outra mudança ocorreu na arquitetura interna do agente tutor. Anteriormente, um agente tutor possuía três camadas, sendo elas: distribuição, social e tutor (Vide Figura A.19). A disponibilização de diversas infra-estruturas de agentes atualmente, faz com que a camada de distribuição não seja interna ao agente tutor. Além disso, a camada social sofreu algumas mudanças com a adição do agente mediador. Com isso, a arquitetura do agente tutor é definida na Figura A.21.

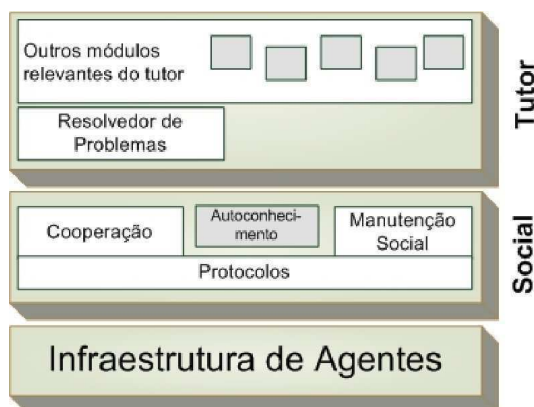


Figura A.21: Arquitetura de um Agente Tutor adaptada

Para finalizar, a arquitetura do agente mediador é definida na Figura A.22. O agente mediador, além de possuir funcionalidades para dar subsídio à sociedade de agentes, passou a possuir funcionalidades presentes no agente de interface no modelo clássico do Mathema.



Figura A.22: Arquitetura do Agente Mediador