

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

ELDER RIZZON SANTOS

**Uma Abordagem Baseada em Ontologias
para a Interoperabilidade entre Agentes
Heterogêneos**

Dissertação apresentada como requisito parcial
para a obtenção do grau de Mestre em Ciência
da Computação

Prof^a. Dr^a. Rosa Maria Vicari
Orientadora

Porto Alegre, abril de 2006.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Santos, Elder Rizzon

Uma Abordagem Baseada em Ontologias para Interoperabilidade entre Agentes Heterogêneos / Elder Rizzon Santos – Porto Alegre: Programa de Pós-Graduação em Computação, 2006.

104 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, BR – RS, 2006. Orientador: Rosa Maria Vicari.

1.Ontologias. 2. Sistemas *Multi-agente* 3. Interoperabilidade. I. Vicari, Rosa Maria. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Profa. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Flávio Rech Wagner

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Ao meu pai e minha mãe, por todo o apoio, amor, carinho, dedicação, zelo e tantos outros sentimentos bons que nem consigo expressar em palavras. O exemplo de força, trabalho, vontade e persistência que vocês têm apresentado durante toda a vida foi o que conduziu-me a concluir mais uma etapa de minha vida. Espero, um dia, chegar aonde vocês chegaram!

A minha orientadora, professora Rosa Maria Vicari, por toda a sua atenção, dedicação, paciência, amizade e confiança. Sem sua ajuda esta dissertação não existiria.

A Elisa Boff, agradeço o companheirismo, as trocas de *insights*, a amizade e a paciência para corrigir esta dissertação.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS.....	8
LISTA DE TABELAS	9
RESUMO.....	10
ABSTRACT	11
1 INTRODUÇÃO.....	12
1.1 PortEdu	12
1.2 AMPLIA.....	13
1.3 Agente Social.....	15
1.3.1 Nível Individual.....	15
1.3.2 Nível de grupo	18
1.4 Motivação	19
1.5 Questão de Pesquisa	19
1.6 Objetivos.....	20
1.7 Contribuições	20
2 REFERENCIAL BIBLIOGRAFICO	21
2.1 Linguagens Ontológicas	21
2.1.1 SHOE.....	22
2.1.2 DAML+OIL	23
2.1.3 RDF	24
2.1.4 RDF SCHEMA.....	25
2.1.5 OWL	26
2.2 Web Semântica	30
2.3 Sistemas Multi-Agente	31
2.3.1 Conceituação de agente	31
2.3.2 Conceituação de Sistema Multi-Agente	32
2.3.3 Padrões para Sistemas Multi-Agentes	33
2.3.4 FIPA	34
2.4 Considerações Finais	41
3 UMA ABORDAGEM PARA AUMENTAR A INTEROPERABILIDADE DO AGENTE SOCIAL	42

3.1	Arquitetura Estendida do Agente Social.....	42
3.2	Módulo para Gerenciamento de Ontologias	44
3.2.1	Redes Bayesianas Representadas em OWL	45
3.2.2	Ontologia em OWL para Representar Redes Bayesianas	45
3.2.3	Conversão Automática de uma RB para uma Representação em OWL	49
3.2.4	Processamento de Consultas.....	57
3.3	Módulo de Comunicação Externa.....	58
3.4	Considerações Finais	61
4	CONCLUSÃO E TRABALHOS FUTUROS.....	62
	REFERÊNCIAS	65
	APÊNDICE A ONTOLOGIA DEFINIDA PARA A ESPECIFICAÇÃO DE REDES BAYESIANAS EM OWL.....	72
	ANEXO A ARTIGO SUBMETIDO PARA O CONGRESSO ITS 2006	79
	ANEXO B ARTIGO SUBMETIDO PARA O CONGRESSO ICALT 2006	90
	ANEXO C ARTIGO SUBMETIDO PARA O CONGRESSO ECAI 2006.....	97

LISTA DE ABREVIATURAS E SIGLAS

ACL	Agent Communication Language
AID	Agent Identifier
AMS	Agent Management System
AMPLIA	Ambiente Multiagente Probabilístico Inteligente de Aprendizagem
API	Application Programming Interface
AUML	Agent Unified Modeling Language
CPT	Conditional Probability Table
DAML	DARPA Agent Mark-up Language
DAML-ONT	DAML-Ontology
DAML+OIL	DAML+Ontology Inference Layer
DARPA	Defense Advanced Research Projects Agency
DF	Directory Facilitator
DL	Descriptive Logics
FIPA	Foundation for Intelligent Physical Agents
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
KQML	Knowledge Query Manipulation Language
MASIF	Mobile Agent System Interoperability Facility
MEBN	Multi-Entity Bayesian Networks
MTS	Message Transport Service
OIL	Ontology Inference Layer
OMG	Object Management Group
OWL	<i>Web</i> Ontology Language
OWL-QL	OWL-Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
PR-OWL	Probabilistic-OWL

RB	Rede Bayesiana
RDF	Resource Description Framework
RDFS	RDF Schema
RMI	Remote Method Invocation
SGML	Standard Generalized Markup Language
SHOE	Simple HTML Ontology Extensions
SMA	Sistema Multi-Agente
XML	eXtensible Markup Language
XMLS	XML Schema
W3C	World Wide <i>Web</i> Consortium

LISTA DE FIGURAS

Figura 1.1: PortEdu e sua plataforma de suporte.	13
Figura 1.2: Arquitetura do ambiente AMPLIA.	14
Figura 1.3: Proposta preliminar da rede probabilística de raciocínio do agente social..	17
Figura 1.4: Distribuição inicial de probabilidades da rede probabilística do agente social	18
Figura 2.1: Estrutura das triplas do RDF.	24
Figura 2.2: Arquitetura da extensão OWL e sua relação com a arquitetura do Protégé.	29
Figura 2.3: Visão em camadas da <i>web</i> semântica.	31
Figura 2.4: Esquema com os tópicos contemplados nas especificações da FIPA.....	34
Figura 2.5: Dois mapeamentos da arquitetura abstrata para realizações concretas.....	35
Figura 2.6: Modelo de Referência do Gerenciamento de Agentes FIPA	36
Figura 2.7: Diagrama AUML representando o protocolo de interação FIPA-REQUEST	39
Figura 2.8: Modelo de referência do transporte de mensagens FIPA	40
Figura 3.1: Arquitetura do agente social	43
Figura 3.2: <i>Screenshot</i> do desenvolvimento de ontologias OWL através do Protégé....	46
Figura 3.3: Rede Bayesiana com dois nodos órfãos.....	47
Figura 3.4: Hierarquia de classes da ontologia para representação de redes bayesianas	48
Figura 3.5: Rede bayesiana do agente social para representar o modelo do aluno	56
Figura 3.6: Instâncias geradas automaticamente abertas no Protégé.....	57
Figura 3.7: Protocolo de interação FIPA-Query.....	59

LISTA DE TABELAS

Tabela 2.1: Parâmetros de uma mensagem FIPA-ACL.	38
Tabela 3.1: Tabela de probabilidades condicionais destacando quais são as condições	47
Tabela 3.2: Tabela de probabilidades condicionais com destaque para os indivíduos da classe Conditional Probability.....	48

RESUMO

Esta dissertação apresenta a utilização de ontologias como um primeiro passo em direção à interoperabilidade, no nível semântico, entre agentes de diferentes domínios. O contexto de aplicação é um portal educacional chamado PortEdu. Este portal fornece a infra-estrutura e suporte para ambientes de aprendizagem baseados em agentes.

O foco deste trabalho é em um agente específico, o agente social, no qual são acrescentadas todas as funcionalidades necessárias para que ele possa interagir com agentes fora de seu contexto. O agente social pertence a um ambiente de aprendizagem multi-agente projetado para apoiar o treinamento do raciocínio, diagnóstico e modelagem de domínios de conhecimento incerto e complexo, como a área médica. Esse ambiente de aprendizagem chama-se AMPLIA.

O conhecimento do agente social é modelado através de redes bayesianas, as quais possibilitam ao agente representar seu conhecimento probabilístico e tomar decisões através dele. A representação através de redes bayesianas não foi desenvolvida para ser utilizada no processo de comunicação com outros agentes, o que dificulta a interoperabilidade do agente social com os demais agentes do portal educacional, no qual os ambientes de ensino aprendizagem encontram-se ancorados.

A abordagem proposta para fornecer interoperabilidade amplia a arquitetura do agente social possibilitando-o representar seu conhecimento probabilístico através de OWL (*Web Ontology Language*) e também sua comunicação com agentes externos através de FIPA-ACL (*Foundation for Intelligent Physical Agents – Agent Communication Language*), linguagem para comunicação utilizada no PortEdu. A OWL não foi projetada para representar incerteza, sendo assim, também foi desenvolvida uma estrutura para possibilitar tal representação em OWL.

Palavras-Chave: Ontologias, Interoperabilidade, Sistemas Multi-Agente.

An Ontology-based Approach to Interoperability Among Heterogeneous Agents

ABSTRACT

This dissertation presents the utilization of ontologies as a first step towards interoperability at the semantic level among agents of different domains. Our test bed is an Educational Portal (PortEdu). This portal provides the infra-structure and support for agent-based learning environments.

We focus on a specific agent, the social agent, adding all the necessary functionality for him to interact with agents that aren't fully aware of its context. The social agent belongs to a Multi-agent Learning Environment designed to support training of diagnostic reasoning and modeling of domains with complex and uncertain knowledge, such as the medical area. This learning environment is called AMPLIA.

The knowledge of social agent is implemented with Bayesian networks, which allows the agent to represent its probabilistic knowledge and make its decisions. The representation through Bayesian networks was not designed to be used for communication with other agents, which makes it more difficult for the social agent to interoperate with the agents present in the portal.

The approach proposed to supply interoperability extends the social agent's architecture allowing the representation of his probabilistic knowledge through OWL (*Web Ontology Language*) and also allows this communication with external agents through FIPA-ACL (*Foundation for Intelligent Physical Agents – Agent Communication Language*), which is the communication language adopted in PortEdu. OWL was not designed to represent uncertain knowledge, thus, it was also developed a structure to allow such representation in OWL.

Keywords: Ontologies, Interoperability, Multi-Agent Systems.

1 INTRODUÇÃO

Esta dissertação está diretamente relacionada a questões de interoperabilidade entre agentes heterogêneos. A interoperabilidade entre agentes é abordada no nível semântico, ou seja, na forma como os agentes lidam com o significado de suas interações. Neste contexto, as principais tecnologias envolvidas para a promoção de interoperabilidade são as linguagens para definição de ontologias e os sistemas multi-agente.

Atualmente, o principal desenvolvimento em linguagens ontológicas é motivado pela *web* semântica (LEE, 2001, p.35). A *web* semântica representa o próximo passo na tecnologia da Internet. Atualmente, o conteúdo das páginas *web* é compreensível apenas por seres humanos. O propósito da *web* semântica é agregar significado às páginas de uma maneira que *softwares* possam interpretar seu conteúdo. Para alcançar esse objetivo torna-se fundamental a utilização de ontologias.

No contexto da *web* semântica, as ontologias são utilizadas como um meio para explicar conceitos e os relacionamentos entre eles, permitindo que agentes interpretem seu significado de uma forma flexível e não ambígua (HORROCKS, 2003, p.7). O domínio em que essas tecnologias para interoperabilidade serão utilizadas neste trabalho é um portal educacional, o PortEdu.

1.1 PortEdu

O PortEdu é um portal *web* que fornece acesso a conteúdos e sistemas educacionais. Ele é um sistema multi-agente compatível com os padrões FIPA e sua implementação foi projetada para apoiar ambientes de aprendizagem baseados em agentes. A funcionalidade principal do portal é a busca de informações personalizadas, a qual considera o perfil do usuário, modelo do estudante e ontologias dos ambientes de aprendizagem (NAKAYAMA, 2005, p.49).

No PortEdu, os agentes de um ambiente de aprendizagem podem interagir com outros agentes, tais como o agente perfil de usuário, o agente para aquisição de informações, entre outros. Neste escopo, a utilidade dos agentes é ampliada, uma vez que eles podem obter informações de mais fontes e também fornecer seu conhecimento e serviços para todos os agentes presentes no portal.

Na Figura 1.1 é apresentada uma visão da arquitetura do PortEdu com foco na infra-estrutura de apoio. A plataforma de agentes é uma plataforma FIPA, e todos os serviços obrigatórios são disponibilizados para que as implementações dos agentes não precisem considerar detalhes de comunicação, gestão do ciclo de vida dos agentes e funcionalidades como descoberta de serviços, gerenciamento de agentes, entre outros.

Estes serviços estão representados na Figura 1.1 como Plataforma FIPA, a qual apóia todos os agentes, tanto do PortEdu, quanto dos ambientes de aprendizagem. Cada número indica um agente na plataforma e as setas indicam exemplos de agentes. Os agentes do PortEdu (agente perfil de usuário e agente para aquisição de informações), por exemplo, são representados na figura pelos agentes 1 e 2. Os números 3 e 4 indicam agentes do ambiente AMPLIA e o restante refere-se aos demais agentes dos respectivos sistemas. Todos os agentes do portal podem interagir diretamente entre si.

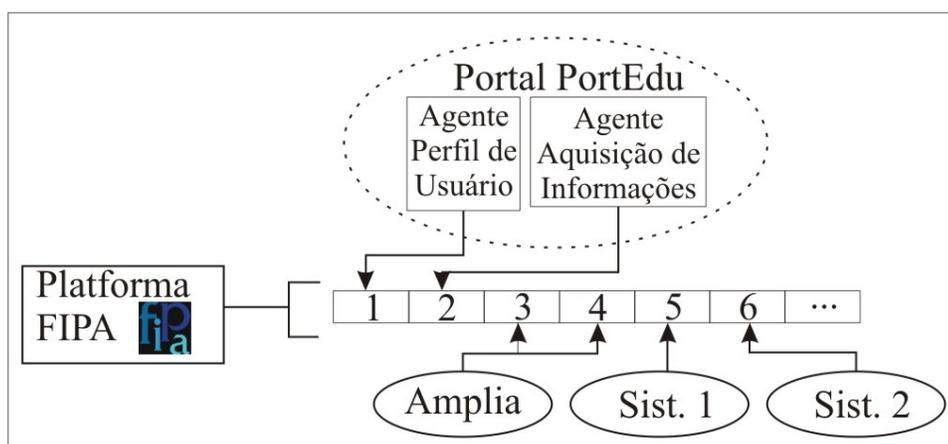


Figura 1.1 PortEdu e sua plataforma de suporte.

Uma das mais úteis funcionalidades dos sistemas multi-agente é a interação entre os agentes, a qual pode auxiliar na resolução de problemas que estão além da capacidade individual de um agente. Tal cooperação é muito difícil de se obter e é um dos objetos de estudos dos pesquisadores de sistemas multi-agente. Com a FIPA, o problema da interação entre agentes heterogêneos é abordado em nível sintático (forma da mensagem), o nível semântico (conteúdo da mensagem) fica a cargo do projetista.

O principal papel das ontologias na *web* semântica é fornecer uma interoperabilidade semântica. Através do uso de uma linguagem ontológica amplamente aceita pretende-se aumentar a interoperabilidade entre os agentes do portal, possibilitando sua interação através de uma linguagem de conteúdo desenvolvida para este fim.

1.2 AMPLIA

O ambiente AMPLIA (Ambiente Multi-agente Probabilístico Inteligente de Aprendizagem) foi projetado para suportar o treinamento do raciocínio diagnóstico e modelagem de domínios de conhecimento incerto e complexo, como a área médica. Este ambiente usa a

abordagem de Redes Bayesianas onde os alunos, que neste contexto são médicos residentes, constroem suas próprias redes para um problema apresentado pelo sistema. A construção destas redes envolve aspectos *qualitativos* (referente a topologia da rede que é uma relação causal entre as variáveis de domínio) e *quantitativos* (referente a distribuição de probabilidades condicionais das variáveis representadas) (VICARI, 2003, p.335).

O processo de negociação dos agentes do ambiente AMPLIA (gerenciado pelo agente inteligente *MediatorAgent*) trata as diferenças de topologia e distribuição das probabilidades entre o modelo construído pelo aprendiz e o construído pelo sistema. A negociação acontece entre os agentes que representam o especialista no domínio do conhecimento (*DomainAgent*) e o agente que representa o aprendiz (*LearnerAgent*), como na Figura 1.2. O agente social, por sua vez, troca mensagens com o agente aprendiz a fim de sugerir colegas com os quais o aluno pode interagir, e com o agente mediador, que conhece o domínio e suporta o processo de negociação.

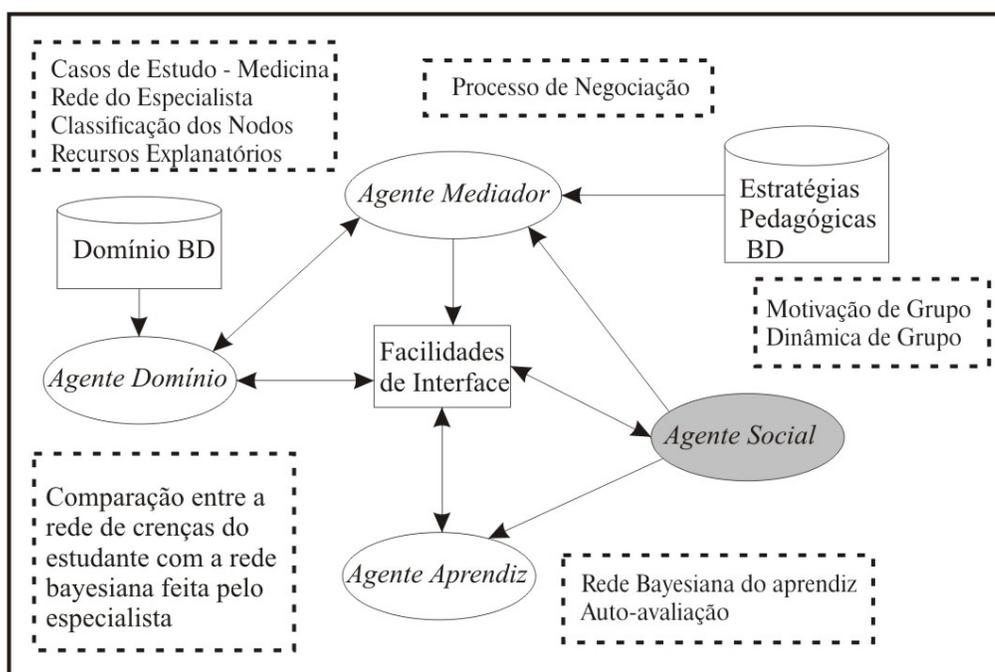


Figura 1.2 Arquitetura do ambiente AMPLIA.

As redes bayesianas têm sido largamente utilizadas para modelar domínios incertos. A incerteza é representada pela probabilidade e a inferência básica é o raciocínio probabilístico, ou seja, o cálculo da probabilidade de uma variável ou mais, face à evidência disponível. Essa evidência é representada por um conjunto de variáveis com valores conhecidos (FLORES, 2005).

A proposta de utilizar as redes bayesianas como uma ferramenta para que os alunos possam representar o seu conhecimento através de modelos probabilísticos deriva da possibilidade de construir e observar concretamente o seu objeto de estudo, formulando e testando as suas hipóteses. Por outro lado, por se constituir um recurso computacional, o processo pode ser facilmente acompanhado por agentes inteligentes, responsáveis pela seleção de

estratégias pedagógicas. Para a construção destes modelos é utilizado o editor Seamed (VICARI, 2003, p.335)).

As informações armazenadas pelo AMPLIA serão complementadas na adaptação do Agente Social ao sistema, visto que atualmente o AMPLIA não armazena dados referentes à natureza social ou afetiva dos alunos. Estas informações também serão acessadas através do ambiente de EAD (PortEdu ou Teleduc) no qual o AMPLIA está vinculado.

A análise do *log* existente do AMPLIA permite inferir algumas características dos alunos, como sua performance em um assunto, sua capacidade de colaboração com os colegas (quantas vezes toma a iniciativa de ajudar, quantas vezes é solicitado, se é solicitado e não atende), o quanto este aluno aceita ou não as sugestões dos colegas e com quais colegas ele costuma interagir mais.

Estas informações serão fundamentais para que o agente forme grupos de usuários no AMPLIA, a partir do modelo do aluno (individual) e do modelo do grupo armazenado.

Como parte do PortEdu, o AMPLIA pode usar as funcionalidades e serviços disponíveis no portal e, mais importante, possibilitar que seus agentes interajam diretamente com agentes de diferentes sistemas de ensino.

1.3 Agente Social

O objetivo principal do agente social é melhorar o aprendizado do estudante através do estímulo à sua interação com seus colegas, monitores e professores. No AMPLIA, cada usuário constrói sua própria rede bayesiana para uma patologia específica. A rede corresponde ao modelo do aluno para uma solução de um problema específico no contexto da saúde.

Durante esta tarefa, o agente social recomenda estudantes para ajudar outros estudantes. O agente cria grupos de trabalho para solucionar problemas cooperativamente. Além disso, o agente social utiliza a estratégia de começar a construção de uma rede para motivar os estudantes a interagir com o ambiente. O raciocínio do agente social é baseado em um nível individual e num nível de grupo. A seguir é apresentada uma breve descrição dos aspectos considerados no nível individual e no nível de grupo. A descrição detalhada bem como a fundamentação teórica dos aspectos apresentados a seguir encontra-se em (BOFF, 2006).

1.3.1 Nível Individual

As características individuais do aluno que serão levadas em consideração para o agrupamento, foram elaboradas com base em estudos da sociologia e não são objeto de estudo deste trabalho (mais informações (BOFF, 2006), (JAQUES, 2004), (MATURANA, 1995)):

Interesse – Dado em função das iniciativas tomadas pelo usuário (qual material ele acessou sem ser recomendado, com quais colegas ele iniciou uma interação).

Performance – Denota a competência do aluno relativo a um assunto específico. O nível de conhecimento do estudante sobre um determinado assunto é obtido pelo número de acertos nos exercícios realizados.

Comprometimento – Alunos que colaboram com seus colegas mais ativamente (é possível realizar uma medição quantitativa das interações) são inseridos em grupos com mais frequência e receberão ajuda de melhor qualidade. Pode-se medir o comprometimento pelo tempo que o aluno ficou no grupo e pela constância e quantidade de interações realizadas com os colegas do grupo.

Familiaridade – As pessoas geralmente procuram interagir com quem têm uma certa familiaridade. Num ambiente virtual, onde as pessoas trabalham de forma distribuída, a aproximação pode ser feita pelo perfil de cada usuário, coletado pelo sistema. Informações que indicam familiaridade podem ser: mesmos interesses, mesmo curso, mesma série e faixa-etária.

Liderança – É possível identificar um líder no ambiente de aprendizagem pela relação entre líderes, seguidores, tarefas e situações. Quando um aluno atua ativamente no ambiente de aprendizagem, estabelece muitas interações e colaborações no editor de redes. Estas informações são analisadas e o agente pode verificar qual estudante mais colaborou na edição da rede e qual aluno teve seu trabalho um menor número de vezes modificado. Este fato pode indicar a presença de um líder, pois são pessoas que conduzem as tarefas e os integrantes do grupo aceitam suas posições. Quanto maior o número de colaborações aceitas, maior o indicador de liderança.

Perfil Social – Traçado durante o processo de comunicação entre os alunos. Este perfil guarda as seguintes informações:

- Iniciativas de comunicação: quantidade de vezes que o aluno teve iniciativa de conversar com outros alunos;
- Respostas a iniciativas de comunicação: numa comunicação inicial, a quantidade de vezes que o aluno respondeu;
- Histórico de interações: alunos com o qual o aluno interage ou já interagiu e quantidades dessas interações;
- Grupo de amigos: alunos com o qual o aluno interage regularmente e a quantidade dessas interações.

Aceitação – O grau de aceitação de um aluno, conferido por outro aluno, é um aspecto levado em consideração para formação de um grupo. Assim, um colega só será indicado novamente para interagir com outro, caso seu grau de aceitação tenha sido alto. Nos casos de baixa aceitação são considerados os assuntos os quais tratavam as interações, uma vez que a baixa aceitação pode ter ocorrido devido ao conhecimento em um assunto específico.

Sociabilidade – Na sociabilidade, a aceitação é utilizada para indicar quão bem um aluno se sai socialmente na visão do aluno que o está avaliando. No entanto, é preciso deixar claro que

o Grau de Aceitação, também pode indicar emoções diversas, tais como amor, inveja, ódio, etc. A média aritmética dos graus de aceitação que os outros alunos têm por um aluno qualquer influi no grau de Sociabilidade.

Humor – O humor é um estado importante no que tange as relações sociais e o aprendizado. Suas causas são tipicamente conceituais ou avaliativas. São exemplos de humor: alegre, triste, irritável e desinteressado. Acredita-se que para o bom funcionamento do grupo os integrantes devem estar com um estado de humor positivo. Neste momento, o que foi definido para o estado de humor são os estágios “mal-humorado ou indisponível” (*Ruim*, na Figura 1.4) e “bem-humorado ou receptivo” (*Bom*, na Figura 1.4). Esses estágios podem ser indicados pelos próprios alunos. Em estudos futuros devem-se analisar formas automáticas de inferência do humor.

A Figura 1.3 ilustra esses aspectos do nível individual dispostos em uma rede bayesiana, mais especificamente, um diagrama de influência para tomada de decisões. O nodo *TraçosDePersonalidade* refere-se aos traços de personalidade que serão considerados neste modelo, aplicado ao contexto de aprendizagem. Os estados afetivos (nodo *EstadoAfetivo*) podem ser considerados como manifestação de emoções em um dado momento.

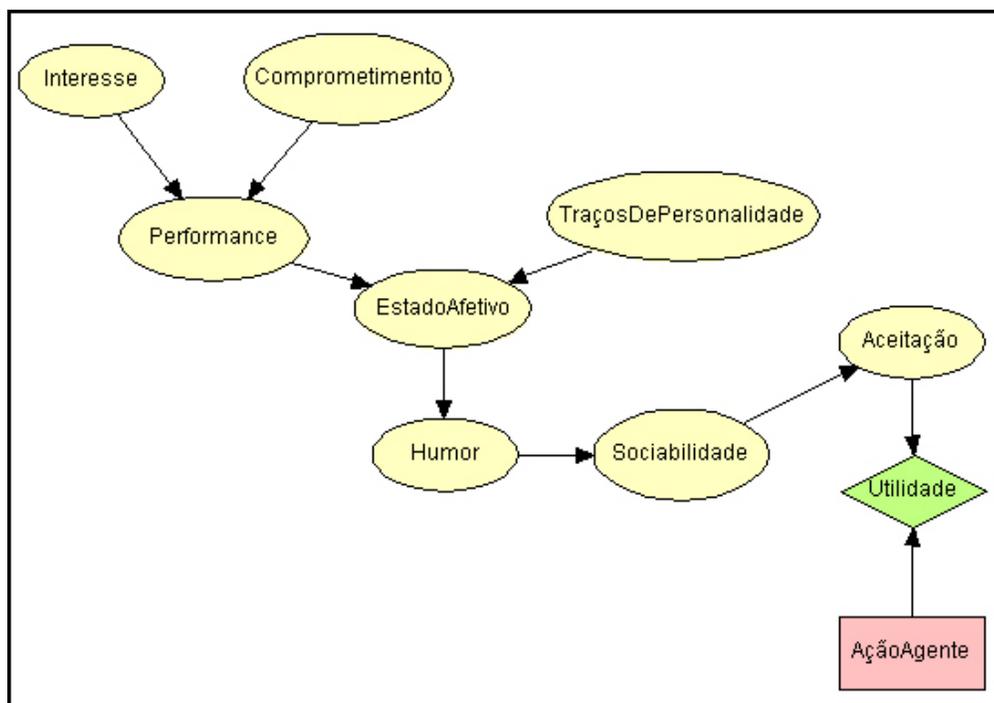


Figura 1.3 Proposta preliminar da rede probabilística de raciocínio do agente social

A distribuição inicial das probabilidades para a rede da Figura 1.3 é apresentada na Figura 1.4.

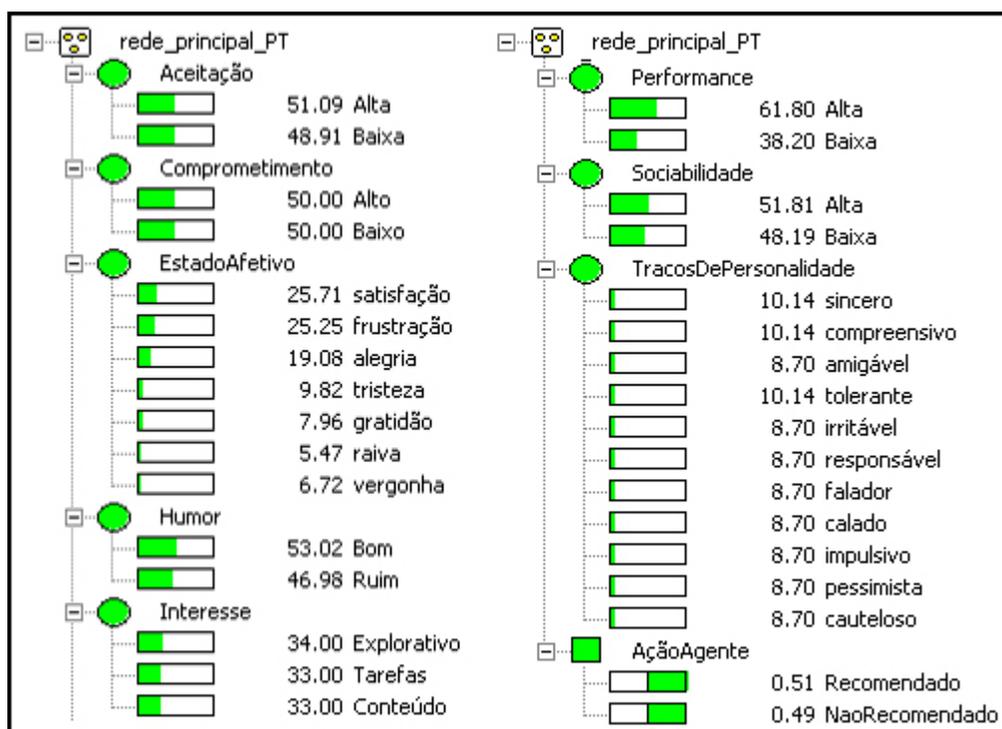


Figura 1.4 Distribuição inicial de probabilidades da rede probabilística do agente social

1.3.2 Nível de grupo

O modelo de grupo armazena os aspectos relacionados ao grupo, fruto da interação social entre os participantes do ambiente de aprendizagem colaborativo. Importantes aspectos para estabelecimento e continuidade de um grupo são a confiança que um indivíduo tem em relação ao trabalho do grupo e a coesão dos membros do grupo. Estes fatores podem indicar o sucesso na realização de tarefas conjuntas.

Confiança – O agente social pode decidir quem juntar em um grupo de forma que traga mais benefício ao grupo. Este tipo de confiança é interessante para ajudar no julgamento do agente (que forma o grupo) a respeito da sua confiança no grupo que outro agente pertence.

A confiança é definida como a crença de um agente nos atributos tais como a confiabilidade, honestidade e competência do agente “confiado” em questão. A reputação de um agente define uma expectativa sobre seu comportamento, que é baseado nas observações do agente ou em informações sobre o comportamento passado do agente em um contexto específico em um dado momento.

Mecanismos sobre confiança e reputação podem ser utilizados para os agentes distinguirem os bons dos maus (adequados ou inadequados) colaboradores. Pode-se verificar o comprometimento de um aluno pelo tempo que ficou no grupo e pela quantidade de interações que o aluno fez com os colegas do grupo.

Coesão – Grupos coesos apresentam uma grande produtividade. A coesão é a atratividade que o grupo exerce pelos seus membros, que dele desejam continuar a participar, resistindo a idéia de abandoná-lo. Podemos definir como *coesão grupal* a solidariedade e estabelecimento de lealdade em um grupo e medi-la pela quantidade de vezes que as mesmas pessoas escolhem interagir.

A formação de um grupo é uma coalizão para alcançar um objetivo maior, que é a aprendizagem. Esta coalizão será feita baseada no comprometimento (aspecto social), estado afetivo e personalidade (aspectos afetivos). Cada aluno possui uma personalidade que influencia na dinâmica dos grupos. Para levar em consideração os aspectos afetivos, os agentes devem manter uma memória emocional.

1.4 Motivação

A principal motivação para a elaboração deste trabalho é a carência observada nas aplicações de grande porte, projetadas sob abordagem de agentes, de interoperabilidade comunicativa (no nível do conteúdo das mensagens) entre os agentes heterogêneos.

Além disso, pretende-se contribuir para a evolução das pesquisas desenvolvidas pelo grupo orientado pela professora Rosa Vicari, mais especificamente na área de ambientes de aprendizagem. Os resultados deste trabalho podem ser diretamente aplicados no projeto PortEdu e no desenvolvimento do Agente Social, o qual faz parte do projeto AMPLIA e está sendo elaborado pela aluna de doutorado do PPGC – UFRGS, Elisa Boff.

Sob o ponto de vista computacional, a motivação é aplicar as tecnologias convergentes com a interoperabilidade entre agentes heterogêneos para contribuir com o desenvolvimento de sistemas multi-agente abertos e com o intercâmbio efetivo de conhecimento entre agentes.

1.5 Questão de Pesquisa

Para nortear este trabalho, foi levantada a seguinte questão de pesquisa:

“Identificar e implementar uma técnica factível e funcional para interoperabilidade entre agentes heterogêneos”.

Relacionadas a esta questão, foram consideradas as seguintes hipóteses:

Hipótese 1: O uso de ontologias é uma técnica capaz de fornecer meios para alcançar interoperabilidade na comunicação – referente ao conteúdo das interações – entre agentes heterogêneos (neste caso é considerada a heterogeneidade dos domínios de aplicação dos agentes);

Hipótese 2: É possível integrar ontologias em sistemas multi-agente;

Hipótese 3: A ontologia deve modelar o conhecimento dos agentes.

1.6 Objetivos

O objetivo geral deste trabalho é adicionar interoperabilidade ao Agente Social para que ele possa ser inserido no PortEdu e interagir com os demais agentes do portal. A partir deste objetivo geral é possível definir alguns objetivos específicos deste trabalho:

1. Especificar uma estrutura para representar o atual conhecimento do agente social de uma forma acessível e compreensível por agentes externos. Isso é necessário uma vez que a representação utilizada pelo agente social são redes bayesianas, as quais não são adequadas para o intercâmbio de conhecimento.
2. Definir qual a linguagem para definição de ontologias é mais adequada no contexto deste trabalho. Isto significa definir qual a linguagem ontológica é mais adequada para comunicação entre agentes e para especificar recursos na *web*.
3. Definir uma infra-estrutura de apoio para o agente social, possibilitando sua integração ao PortEdu. O PortEdu fornece uma plataforma FIPA para os agentes do portal. Dessa forma, faz-se necessário possibilitar ao agente social comunicar-se através de FIPA-ACL e também fornecer a estrutura necessária para que ele interaja com os serviços básicos da plataforma (registro, serviço de diretório, transporte de mensagens, etc.)
4. Implementar as estruturas para aumento da interoperabilidade, sob a forma de um protótipo, no agente social. Isso significa implementar uma ontologia para representação de redes bayesianas; implementar um mecanismo para representar as redes do agente social em OWL, seguindo a ontologia pré-definida; implementar o suporte à plataforma FIPA (registro e interação com outros agentes).

1.7 Contribuições

Este trabalho visa contribuir com as pesquisas realizadas, na Inteligência Artificial, relativas à interoperabilidade entre agentes heterogêneos, que fazem parte de uma sociedade de Sistemas Multi-Agente. Através deste estudo e desenvolvimento pretende-se apresentar uma maneira para lidar com a interoperabilidade da comunicação entre agentes no nível semântico, ou seja, quanto ao conteúdo das mensagens.

Estudos relativos à interoperabilidade em nível sintático vêm sendo realizados há mais tempo e já existem padrões maduros abordando esse assunto (FIPA e KQML, por exemplo). Já no nível semântico da comunicação, os estudos mais recentes apontam para o uso de ontologias como uma possível solução. Neste contexto está uma contribuição deste trabalho.

Além dessa contribuição, espera-se contribuir, também, com o desenvolvimento da linguagem ontológica OWL. Devido a sua fundamentação lógica formal, ela não oferece suporte a representação de incerteza. A representação desse tipo de conhecimento é fundamental para a *web* semântica uma vez que nem todas as informações disponíveis na Internet são 100% confiáveis, ou seja, não há certeza se o conhecimento é verdadeiro ou não.

O contexto desta dissertação torna necessária a especificação de conhecimento probabilístico, sob a forma de redes bayesianas, em OWL. Esse não é um problema trivial e não pretende-se solucioná-lo com este trabalho, mas assim apresentar uma abordagem utilizada para lidar com essa limitação e algumas reflexões sobre o assunto.

2 REFERENCIAL BIBLIOGRAFICO

Neste Capítulo serão apresentados os principais conceitos e tecnologias utilizadas neste trabalho. Primeiro são estudadas as linguagens ontológicas apresentando um enfoque em sua utilização para interoperabilidade em sistemas abertos. Tal estudo é importante pois fundamenta a escolha da linguagem adotada para representar o conhecimento do agente social de uma maneira mais interoperável.

Em seguida é apresentada uma descrição da *web* semântica, a qual contextualiza o desenvolvimento das principais linguagens ontológicas da atualidade.

Na seção 2.3 são explicados os principais conceitos sobre Sistemas Multi-Agente para contextualizar a utilização de padrões para a interoperabilidade entre agentes. Ainda nesta seção são apresentados os principais padrões para a comunicação entre agentes heterogêneos, com enfoque no padrão FIPA. O enfoque neste padrão deve-se ao fato de ele ser utilizado no PortEdu, portal no qual o agente social está inserido e onde ele interage com os demais agentes através de FIPA-ACL, a linguagem para comunicação entre agentes definida pela FIPA.

No final deste Capítulo são apresentadas as considerações finais sobre os estudos realizados.

2.1 Linguagens Ontológicas

Assim como a maioria dos conceitos muito conhecidos e utilizados por diversas áreas do conhecimento, o conceito de ontologia possui diversas definições. A definição de ontologia adotada neste trabalho é a seguinte: (GRUNINGER, 1993) uma ontologia é uma especificação explícita e formal de uma conceituação compartilhada.

Nesse contexto, uma conceituação refere-se a um modelo abstrato de como as pessoas pensam sobre coisas no mundo, geralmente restrito a um tópico específico. Explícita significa que as categorias de conceitos utilizadas e respectivas suas restrições são definidas explicitamente. Formal refere-se ao fato de que a ontologia possa ser lida e interpretada por uma máquina. Existem vários níveis de formalismos possíveis.

O fato de uma ontologia ser uma teoria formal sobre um domínio implica a necessidade de uma linguagem formal para a sua definição. Estas linguagens utilizadas para definição de

ontologias são chamadas linguagens ontológicas. Neste capítulo serão apresentadas as principais linguagens ontológicas desenvolvidas com o enfoque na utilização em sistemas abertos, como aplicativos *web*, por exemplo.

Optou-se por esse tipo de linguagem devido ao contexto deste trabalho ser um portal *web* para educação. Desta forma, além de aplicar os estudos realizados no agente social, eles podem ser utilizados para agregar significado aos conteúdos existentes no portal. Além disso, o fato de serem desenvolvidas para utilização na *web* já predispõe a necessidade de padronização, o que é um fator importante para a interoperabilidade.

2.1.1 SHOE

Segundo (HORROCKS, 2003), SHOE foi uma das primeiras tentativas para definir uma linguagem ontológica para implantação na *web*. SHOE (*Simple HTML Ontology Extensions*) (HEFLIN, 2000) é uma linguagem para representação de conhecimento baseada em ontologias embutida em páginas *web*.

A SHOE estende o HTML (*HyperText Markup Language*) com um conjunto de marcações (*tags*) que, ao contrário das marcações HTML, fornecem estrutura para aquisição de conhecimento ao invés de apresentação de informações. A SHOE associa significado ao conteúdo das páginas fazendo com que cada página comprometa-se com uma ou mais ontologias. Estas ontologias permitem a descoberta do conhecimento implícito através do uso de taxonomias e regras de inferência.

A interoperabilidade é promovida através do compartilhamento e reuso de ontologias. Para alcançar compatibilidade com os padrões para *web*, a sintaxe da SHOE é definida como uma aplicação de SGML (*Standard Generalized Markup Language*), o predecessor do HTML. Existe uma variação do SHOE que utiliza XML (*Extensible Markup Language*) ao invés do original SGML (HEFLIN, 2000).

Trata-se de uma linguagem baseada em *frames* com uma sintaxe XML que pode ser seguramente incluída em documentos HTML. A linguagem utiliza referências URI (*Uniform Resource Identifier*) para os nomes, o que constitui uma importante inovação que foi utilizada na DAML+OIL (seção 2.1.2) e outras linguagens desenvolvidas posteriormente (HORROCKS, 2003). O fato de ontologias poderem ser interligadas e sujeitas a atualizações com o passar do tempo fez com que fossem incorporadas diretivas para importação de ontologias, informações de compatibilidade entre ontologias e declaração de versão. Esta linha de raciocínio influenciou parte do vocabulário da OWL.

Por tratar-se apenas de um conjunto de marcações adicionais ao HTML, nenhuma semântica é definida na linguagem. Os termos utilizados para definir uma ontologia podem ser definidos pelo usuário, dificultando consultas e o entendimento do conhecimento representado por agentes ou aplicativos externos. Outro aspecto a ser considerado é que a SHOE não possui uma fundamentação lógica formal, desta forma, aspectos simples, porém fundamentais não estão presentes, tais como negação e cardinalidade.

2.1.2 DAML+OIL

DAML+OIL (*Darpa Agent Markup Language + Ontology Interchange Language*) é uma linguagem de marcação semântica para recursos *web*. Ela foi construída com base no RDF (seção 2.1.3) e RDF *Schema* (seção 2.1.4) e estende estas linguagens com primitivas de modelagens mais ricas (CONNOLI, 2001-a). É a precursora da OWL.

DAML foi desenvolvido em um projeto da DARPA¹, chamado DAML-ONT, o qual visava aumentar a acessibilidade e interpretação de dados na *web*. Esta linguagem não é uma recomendação do W3C (*World Wide Web Consortium*), mas contribuiu para o desenvolvimento do DAML+OIL, a qual mais tarde evoluiu para a OWL.

DAML-ONT foi concebido como uma camada adicional acima do RDF, utilizando a mesma sintaxe baseada em XML, porém estendendo as capacidades de modelagem visando incluir descrições de termos mais expressivas e semânticas mais precisas (MCGUINNESS, 2002-b). O desenvolvimento do projeto DAML-ONT aconteceu simultaneamente ao projeto OIL (*Ontology Inference Layer*) (FENSEL, 2000), o qual apresentava objetivos muito similares aos do DAML-ONT. Eventualmente, ambos projetos foram unidos gerando o DAML+OIL.

Basicamente, a DAML+OIL foi desenvolvida para criar ontologias, ou representações formais de um conjunto de dados em particular, de tal forma que a informação possa ser processada e manipulada logicamente. O papel do DAML no DAML+OIL é fornecer um conjunto de primitivas (classes, subclasses, restrições, etc.) para marcar informações e localizá-las dentro da estrutura conceitual do domínio. Já a função do OIL é fornecer os meios para que possam ser realizadas inferências. Sua estrutura é fundamentada em linguagens baseadas em *frames*. A linguagem OIL estende a linguagem baseada em *frames* e permite que *slots* tenham *slots* próprios, possibilitando que eles mesmos sejam objetos. Em outras palavras, uma propriedade de uma classe pode ser uma classe dela mesma (FENSEL, 2000).

Com a união da DAML e da OIL, a linguagem resultante suporta conversão e interpretação utilizando lógica de primeira ordem, a qual é parte importante na interoperabilidade entre ontologias, pois é responsável pela inferência. Na interoperabilidade, este processo é importante pois permite às aplicações chegar a conclusões sobre informações fora de sua ontologia original. Tais conclusões podem ser obtidas utilizando motores de inferência de lógica de primeira ordem para analisar declarações sobre informações fora de seu escopo e agregá-las à sua ontologia original (MCGUINNESS, 2002).

A DAML+OIL foi submetida ao W3C como um ponto de partida para a OWL (*Web Ontology Language*). O grupo de trabalho responsável pela ontologia para a *web* do W3C produziu uma série de requisitos (HEFLIN, 2004) para a sua linguagem ontológica. A DAML+OIL trata quase todos estes requisitos. A descrição da OWL é muito similar a da DAML+OIL. Pode-se dizer que a OWL é a DAML+OIL somada à recomendação da W3C. Entretanto, todos os esforços de desenvolvimento estão focados na OWL e não mais na DAML+OIL.

¹ <http://www.darpa.mil/>

O apoio da W3C a OWL faz com que muitas ferramentas fossem desenvolvidas para trabalhar com essa linguagem. Desta forma, o desenvolvimento é facilitado por implementações prontas de parsers, motores de inferência, APIs, editores, etc. Além disso, o fato de ser uma recomendação W3C dá mais visibilidade à linguagem e padroniza sua utilização em contextos similares, assim como é a utilização do HTML para exibir informações na Internet.

O motivo pelo qual DAML+OIL não é utilizado neste trabalho é pelo fato de que a OWL é uma continuidade dessa linguagem e é uma recomendação da W3C.

2.1.3 RDF

Resource Description Framework (MANOLA, 2004) (arquitetura para descrição de recursos) é uma linguagem para representar informações sobre recursos na *World Wide Web*. Mais especificamente, foi desenvolvida para expressar meta-dados sobre recursos *web*, tais como o título, autor, data de criação de uma página *web*, *copyright* e licenciamento de um documento ou a disponibilidade de algum recurso compartilhado. Generalizando-se o conceito de um recurso *Web*, a aplicabilidade do RDF aumenta bastante, possibilitando seu uso para expressar informações sobre coisas que possam ser identificadas na *WEB*.

Segundo (KLYNE, 2004) o desenvolvimento do RDF foi motivado pelas seguintes utilizações:

- Meta-dados *Web*: fornecer informações sobre recursos e sistemas que os utilizam;
- Aplicações que necessitem modelos de informação abertos;
- Permitir que os dados sejam processados fora do ambiente específico em que foram criados, de uma forma a funcionar na escala da Internet;
- Processamento automatizado de informações da *web* por agentes de software: o foco da *web* está indo em direção a uma rede mundial de processos cooperativos, ao invés de apenas fornecer informações que possam ser lidas por seres humanos.

O projeto do RDF visa representar informações da forma menos restritiva e mais flexível possível. O valor da informação expressa em RDF é maior uma vez que se torna acessível para uma gama maior de aplicações na Internet.

Seguindo os objetivos da *web* semântica, o RDF é dirigido para situações as quais a informação deve ser utilizada por aplicativos de *software* ao invés de ser simplesmente exibida para pessoas. O RDF fornece um *framework* comum para expressar tais informações para que elas sejam compartilhadas entre aplicações heterogêneas sem perda de significado.

A estrutura base de qualquer expressão em RDF é uma coleção de triplas, cada uma composta por um sujeito, um predicado (também chamado de propriedade, o qual expressa a relação) e um objeto. Um conjunto de tais triplas é chamado de grafo RDF, no qual cada tripla é representada por uma ligação nodo-arco-nodo. A Figura 2.1 ilustra a estrutura das triplas do RDF:

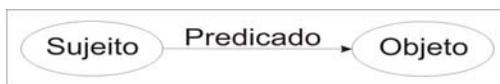


Figura 2.1 Estrutura das triplas do RDF.

O sujeito de uma tripla pode ser um nodo vazio ou então um URI (*Uniform Resource Identifier*). O objeto, além de poder ser um nodo vazio ou um URI também pode ser um literal (simples ou tipado). Os sujeitos e os objetos formam os nodos de um grafo RDF.

Os grafos RDF são armazenados através de uma sintaxe baseada em XML, chamada RDF/XML. Assim como o HTML, o RDF/XML é processável por máquinas e através do uso de URIs é possível conectar pedaços de informação presentes na *web*. Entretanto, ao contrário do hipertexto convencional, as URIs em RDF podem referir-se a qualquer coisa identificável, inclusive algo que não possa ser diretamente obtido na *web*.

Desta forma, além de descrever coisas como páginas *web*, com RDF é possível descrever carros, negócios, pessoas, notícias, etc. Além disso, as próprias propriedades possuem URIs, para identificar precisamente as relações existentes entre os itens conectados.

O RDF pode ser visto como uma aplicação do XML para representar meta-dados. Em outras palavras, o RDF fornece uma forma padrão para representar os meta-dados em XML. Essa linguagem não possui primitivas para declaração de classes, propriedades, relações, etc e nem uma fundamentação formal. Essas primitivas são especificadas no RDF *Schema*.

2.1.4 RDF SCHEMA

Através do RDF é possível expressar sentenças simples sobre recursos, utilizando propriedades nomeadas e valores. Entretanto, também existe a necessidade de definir os vocabulários (termos) a serem utilizados nessas sentenças, especificamente, para indicar a descrição de classes de recursos e suas respectivas propriedades. O RDF em si não fornece meios para definir classes e propriedades relativas a uma aplicação específica. Tais classes e propriedades são descritas como um vocabulário RDF, através de extensões para RDF definidas pelo RDF *Vocabulary Description Language: RDF Schema* (MANOLA, 2004).

O RDF *Schema* (RDFS) fornece as facilidades necessárias para descrever classes e propriedades, resumindo-se, fornece um sistema de tipos para o RDF. Este sistema de tipos é parecido com o das linguagens de programação orientadas a objetos. É possível definir recursos como instâncias de uma ou mais classes, além disso, pode-se organizar as classes em uma estrutura hierárquica de classes e subclasses.

As facilidades do RDF *Schema* são fornecidas sob a forma de um vocabulário RDF, ou seja, um conjunto especializado de recursos RDF pré-definidos com seus respectivos significados. Descrições de vocabulários (*schemas*) escritos com RDF *Schema* constituem grafos RDF válidos. Desta forma, softwares para RDF que não foram escritos para processar o vocabulário adicional do RDF *Schema* ainda podem interpretar um *schema* como grafo válido constituído por vários recursos e propriedades, mas não será capaz de interpretar os significados pré-definidos dos termos do RDF *Schema* (BRICKLEY, 2004).

As primitivas referentes à definição classes em RDFS são `rdfs:Resource`, `rdf:Property` e `rdfs:Class`. Qualquer coisa descrita por expressões RDF é vista como uma instância da classe `rdfs:Resource`. A classe `rdf:Property` é utilizada para caracterizar as instâncias de `rdfs:Resource`, ou seja, cada slot ou relação é uma instância de `rdf:Property`. A primitiva `rdfs:Class` é utilizada para definir conceitos em RDFS, ou seja, cada conceito deve ser uma instância de `rdfs:Class`.

O RDFS pode ser considerada uma linguagem ontológica uma vez que define a semântica para classes e propriedades, limites, restrições de domínio, sub-classes e relações entre sub-propriedades. Entretanto, é uma linguagem muito limitada e é necessário mais poder de expressão para descrever informações em detalhe. Além disso, não oferece nenhum suporte à inferência. O RDFS é utilizado como base para a OIL, DAML+OIL e OWL.

2.1.5 OWL

A *Web Ontology Language* (OWL) (SMITH, 2004) é uma linguagem para definição de ontologias voltadas para a *web*. Esta linguagem faz parte de um esforço da W3C para tornar possível a *web* semântica. Foi projetada para utilização por aplicações que necessitem processar o conteúdo das informações ao invés de apenas apresentar a informação para os usuários. No contexto da arquitetura em camadas da *web* semântica o nível acima do RDF, a OWL, é uma linguagem para ontologias capaz de descrever formalmente o significado dos termos utilizados em documentos *web*.

A OWL utiliza e estende a declaração de fatos do RDF e a estruturação em classes e propriedades do RDF *Schema*. Com OWL é possível declarar classes e organizá-las numa hierarquia de subclasses, como no RDF *Schema*. As classes podem ser definidas como combinações lógicas (intersecções, uniões ou complementos) de outras classes ou como enumerações de objetos específicos, indo além das capacidades do RDF *Schema*.

Assim como no RDFS, é possível também declarar propriedades, estabelecer uma hierarquia de propriedades e subpropriedades e definir domínios para estas propriedades. Os domínios das propriedades em OWL são classes e os limites podem ser tanto classes quanto tipos de dados definidos externamente, como um inteiro ou um literal. As propriedades podem ser definidas como transitivas, simétricas, funcionais ou o inverso de uma outra propriedade (KLYNE, 2004).

Em OWL pode-se especificar a qual classe pertencem determinados objetos (indivíduos) e quais valores de propriedades são de indivíduos específicos. Declarações de equivalência podem ser feitos sobre classes e propriedades, disjunções podem ser declaradas entre classes e igualdade e diferença entre indivíduos.

Apesar destas funcionalidades, segundo (HORROCKS, 2003) a maior extensão com relação ao RDFS é a habilidade de definir restrições na maneira como uma propriedade comporta-se no contexto de uma classe específica. É possível definir propriedades de classes restritas ao contexto de um indivíduo específico, definindo-se sua classe ou tipo de dado desejado. Outras possibilidades são declarar a presença de no mínimo alguns valores específicos; no mínimo um valor de determinada classe; no mínimo ou no máximo um certo número de valores distintos.

Em (HORROCKS, 2003) é apresentada a seguinte comparação prática entre RDFS e OWL:

Através de RDFS é possível:

- Declarar classes como país, pessoa, estudante, canadense, etc.;
- Afirmar que estudante é uma subclasse de pessoa;
- Afirmar que Canadá e Inglaterra são instâncias da classe país;

- Declarar nacionalidade como uma propriedade relacionando as classes pessoas (domínio) e país (escopo);
- Definir que idade é uma propriedade cujo domínio é a classe pessoa e o tipo inteiro é seu escopo;
- Declarar que Peter é uma instância da classe canadense e que sua idade possuiu valor 48.

Com OWL é possível também:

- Definir que país e pessoa são classes disjuntas;
- Definir que Canadá e Inglaterra são indivíduos diferentes;
- Declarar que possuiCidadão é uma propriedade inversa à propriedade Nacionalidade;
- Declarar que a classe SemCidadania é definida pelos membros da classe Pessoa que não possuem valores para a propriedade Nacionalidade;
- Especificar que a classe NacionalidadeMultipla é composta pelos indivíduos da classe Pessoa que possuem no mínimo dois valores para a propriedade Nacionalidade;
- Definir que a classe Canadense é definida pelas instâncias da classe Pessoa que têm Canadá como valor da propriedade Nacionalidade

A utilização da OWL permite uma maior interoperabilidade do que a obtida com o uso de XML, RDF e RDF *Schema*, uma vez que essa linguagem fornece mais vocabulário para a definição de classes e suas propriedades e apresenta uma semântica formal. Existem três sub-linguagens do OWL, cada uma oferece níveis de expressividade diferentes: OWL *Lite* (menos expressiva), OWL DL e OWL *Full* (mais expressiva) (SMITH, 2004).

OWL *Lite* é indicada para quem necessita apenas uma hierarquia de classificação e funcionalidades restritivas simples. Por exemplo, enquanto OWL *Lite* suporta cardinalidade, são permitidas somente as cardinalidades 0 ou 1.

OWL DL fornece expressividade máxima sem perder a completude computacional (todas as deduções são computáveis) e a decidibilidade (todas as computações terminam em um tempo finito) dos sistemas de inferência. Esta sub-linguagem possui todas as diretivas da OWL com restrições como separações de tipo (uma classe não pode ser ao mesmo tempo um indivíduo ou uma propriedade; uma propriedade não pode ser também um indivíduo ou uma classe). O nome OWL DL vêm de sua correspondência com a lógica descritiva².

OWL *Full* fornece expressividade máxima sem nenhuma restrição, ou seja, com a liberdade sintática do RDF, entretanto, sem garantias computacionais. Uma classe descrita em OWL *Full* pode ser tratada simultaneamente como uma coleção de indivíduos e também como um único indivíduo. Com esta sub-linguagem é possível expandir o significado do vocabulário pré-definido do RDF e da OWL.

De acordo com Dean (2004), OWL *Full* pode ser considerada uma extensão do RDF, enquanto OWL *Lite* e DL podem ser vistos como extensões de conjuntos restritos do RDF. Todo

² Lógica Descritiva (*Descriptive Logics*): Campo de pesquisa que estuda um conjunto decidível da lógica de primeira de ordem (BAADER, 2002).

documento OWL, independente de sua sub-linguagem, é um documento RDF e todo RDF é um documento OWL *Full*, entretanto, somente alguns documentos RDF são documentos OWL *Lite* ou DL válidos.

A linguagem OWL visa possibilitar a definição completa de ontologias. Ela fornece construtores ricos para elaborar classes e axiomas para permitir raciocínio nos dados da ontologia. Uma importante funcionalidade da OWL é sua divisão em três níveis de expressividade, possibilitando a definição de ontologias simples ou complexas.

Apesar de uma linguagem nova, a OWL é resultado das pesquisas realizadas com a OIL, DAML-ONT e DAML+OIL. Seu desenvolvimento é mantido pela W3C, a qual especificou a OWL como a linguagem ontológica da *web* semântica, sob a forma de uma recomendação.

Já existem pesquisas para desenvolver uma linguagem de consultas padrão para OWL, a proposta mais atual é a OWL-QL (OWL-*Query Language*) (FIKES, 2005). Por ser uma recomendação da W3C a OWL possui uma visibilidade muito grande na comunidade acadêmica. Isso resulta no desenvolvimento de muitas ferramentas para apoiar a implementação de aplicações utilizando OWL.

Um ponto fraco da OWL é o fato de não conter em sua especificação meios para definição de conceitos probabilísticos, ou seja, não permite a modelagem de incerteza. Isso é um aspecto importante uma vez que o conhecimento do agente social é fundamentado em redes bayesianas. A OWL será utilizada neste trabalho como a linguagem ontológica utilizada pelo agente social visando sua interoperabilidade com os agentes do PortEdu.

A seguir é descrito o editor para ontologias Protégé, o qual foi utilizado para definição da ontologia OWL sobre redes bayesianas. Existem vários editores de OWL disponíveis, entretanto, a maioria deles possibilita o desenvolvimento através de triplas RDF, o que torna a elaboração da ontologia um processo mais demorado e menos intuitivo. Além do Protégé, outro editor OWL de destaque é o SWOOP (KALYANPUR, 2005). Optou-se pelo Protégé devido a sua interface ser mais intuitiva e fornecer a integração direta com ferramentas para inferência, manutenção de indivíduos e visualização da ontologia.

2.1.5.1 Protégé

O Protégé é um editor de ontologias e *framework* para sistemas baseados em conhecimento. Ele é desenvolvido nos moldes de software livre e possui código aberto. É possível modelar ontologias de duas maneiras, com o Protégé-Frames e com o Protégé-OWL. As ontologias criadas nesse editor podem ser exportadas para diversos formatos, tais como RDFS, OWL e XMLS (GENNARI, 2002).

A arquitetura do Protégé é separada em duas partes: modelo e visão. O modelo do Protégé constitui o mecanismo de representação interna para ontologias e bases de conhecimento. Os componentes da visão fornecem uma interface com o usuário para exibir e manipular o modelo de base.

O modelo utilizado no Protégé baseia-se em meta-modelo comparável aos sistemas orientados a objetos e baseados em *frames*. Esse modelo pode representar ontologias compostas

por classes, propriedades (*slots*), características das propriedades (*facets* e restrições) e instâncias. É fornecida uma API Java para consultar e manipular os modelos (GENNARI, 2002).

Através das visualizações da interface com o usuário do Protégé os desenvolvedores de ontologias basicamente criam classes, atribuem propriedades às classes e restringem as *facets* das propriedades a algumas classes. A partir das ontologias resultantes, o editor gera automaticamente interfaces para a criação dos indivíduos (instâncias). Para cada classe o sistema cria uma maneira para editar seus componentes (*widgets*) de acordo com suas propriedades (KNUBLAUCH, 2003).

A extensão para OWL do Protégé pode ser utilizada para editar arquivos OWL e bases de dados OWL. Esse *plugin* oferece uma coleção de interfaces e componentes específicos para OWL. Além disso, fornece acesso a serviços relacionados a OWL, tais como classificação, verificação de consistência e testes de ontologia. Conforme ilustrado na Figura 2.2, o *plugin* OWL estende o modelo do Protégé e sua API com classes para representar a especificação OWL. É fornecido suporte para RDFS, OWL *Lite*, OWL DL (exceto axiomas globais de classes, os quais precisam ser nomeados) e partes importantes da OWL *Full* (KNUBLAUCH, 2005).

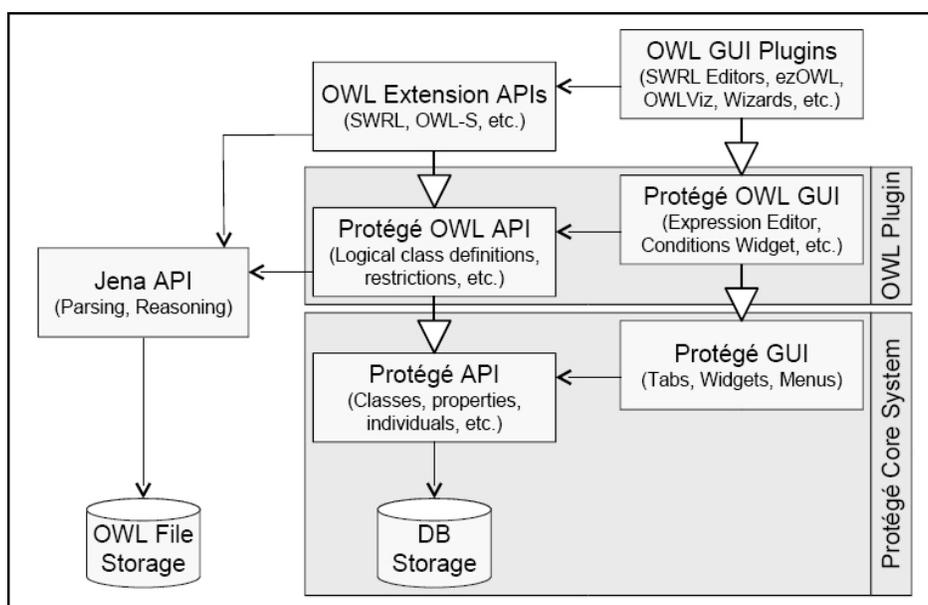


Figura 2.2 Arquitetura da extensão OWL e sua relação com a arquitetura do Protégé (KNUBLAUCH, 2005).

A utilização do Protégé com a extensão para OWL foi fundamental no desenvolvimento do protótipo, pois possibilitou o desenvolvimento da ontologia para representação de redes bayesianas e também para verificar criação automática dos indivíduos através da API desenvolvida.

2.2 Web Semântica

A maioria das linguagens ontológicas apresentadas na seção 2.1 foram desenvolvidas com motivação na *web* semântica. Para contextualizar o cenário na qual estas linguagens foram desenvolvidas nesta seção são apresentados os principais conceitos sobre a *web* semântica.

A *web* semântica (BERNERS-LEE, 2001) pode ser vista como o próximo passo da tecnologia da Internet. Atualmente, as páginas *web* fornecem informações para seres humanos, através de linguagem natural. Não há uma estrutura adequada para que aplicativos computacionais possam acessar automaticamente tais conteúdos e utilizá-los de acordo com o seu contexto. Com a *web* semântica, a intenção é agregar uma estrutura ao significado das páginas de tal forma que a informação possa ser processada e interpretada automaticamente por *softwares*.

Segundo Berners-Lee (2001) a *web* semântica é uma extensão da *web* atual na qual a informação recebe um significado bem definido, possibilitando que computadores e pessoas trabalhem em cooperação.

A visão inicial da *web* semântica foi exposta por Tim Berners Lee no artigo (BERNERS-LEE, 2001). Os esforços atuais para o desenvolvimento dessa tecnologia são coordenados pelo W3C (*World Wide Web Consortium*) (W3C, 2005). O W3C é um consórcio internacional que trabalha no desenvolvimento de padrões para a *web*, visando garantir sua evolução e utilização.

A iniciativa para o desenvolvimento da *web* semântica pelo W3C (W3C, 2005-a) tem um objetivo tão amplo quanto o da Internet, criar um meio universal para o intercâmbio de informações. Para que a Internet possa utilizar todo seu potencial é necessário que as informações sejam compartilhadas e processadas tanto por aplicativos automatizados quanto por pessoas. Os softwares devem ser capazes de compartilhar e utilizar dados mesmo sendo construídos de forma totalmente diferente.

Para alcançar este ambiente universal para a troca de informações entre pessoas e máquinas a W3C desenvolve especificações abertas das tecnologias necessárias e, através do desenvolvimento com código aberto, identifica os componentes que serão necessários para que a *web* semântica torne-se uma realidade.

As principais tecnologias da *web* semântica são definidas por um conjunto em camadas de especificações. Os componentes atuais são o modelo núcleo do RDF (*Resource Description Framework*), a linguagem RDF Schema e o OWL (*Web Ontology Language*). Acima destes componentes principais está uma linguagem de consulta padrão para RDF, o SPARQL, a qual possibilita a união de coleções descentralizadas de informação em RDF. Todas estas linguagens têm como base URI (*Universal Resource Identifier*), XML (*eXtended Markup Language*) e XML namespaces. A Figura 2.3 exibe essa visão em camadas das tecnologias atuais para o suporte à *web* semântica.

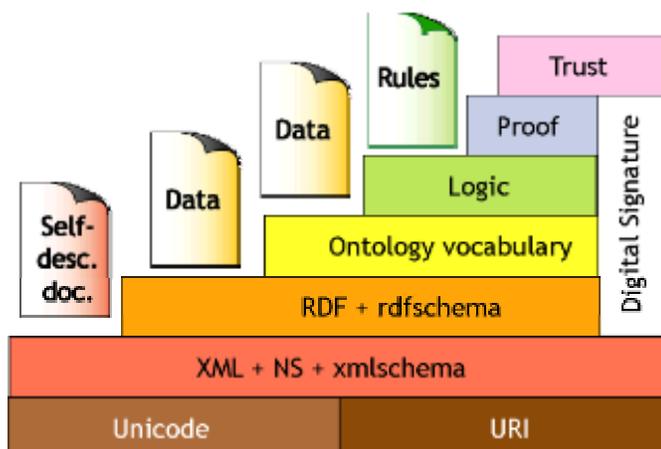


Figura 2.3 Visão em camadas da *web* semântica (BERNERS-LEE, 2001).

2.3 Sistemas Multi-Agente

Nesta seção são apresentados os principais conceitos relativos aos sistemas multi-agente (SMA) visando contextualizar a importância da elaboração e da adoção de padrões para possibilitar interoperabilidade entre agentes heterogêneos. Os principais padrões são apresentados e é dado um enfoque nas especificações da FIPA, visto que elas são utilizadas pelo PortEdu e o agente social está inserido neste portal. Sendo assim, as mensagens enviadas e recebidas pelo agente social no PortEdu seguem as especificações da FIPA, tornando esse padrão um importante objeto de estudo para a realização deste trabalho.

2.3.1 Conceituação de agente

Existem inúmeras definições para o termo agente e, na maioria das vezes, elas não possuem muito em comum. De acordo com Nwana (1996) existem dois motivos principais para a falta de consenso quanto à definição de agente. O primeiro se deve ao fato do termo agente ser utilizado nos mais diferentes domínios e contextos, diferente do termo lógica *fuzzy*³, por exemplo.

O segundo motivo é que, mesmo dentro da ciência da computação, a palavra agente abriga diferentes ramos de pesquisa e desenvolvimento. A grande abrangência do termo agente ocasionou a criação de vários sinônimos, tais como *knowbot*, *softbot*, *userbots*, etc., os quais somente aumentam a confusão. A seguir são apresentadas algumas definições de agente e a definição adotada neste trabalho:

“Quando realmente precisamos, definimos um agente como um componente de *software* e/ou *hardware* capaz de agir precisamente para realizar tarefas em nome de seu usuário. Tendo a

³ A Lógica *fuzzy* baseia-se nos conceitos de conjuntos *fuzzy* e lógica simbólica. Ela pode ser vista com um tipo especial de lógica multi-valorada. Sendo assim, o valor verdade de uma sentença, ao invés de assumir apenas dois valores (0 ou 1), pode assumir qualquer valor no intervalo [0,1] (LEE, 1972).

escolha, preferimos dizer que se trata de um termo “guarda-chuva”, um meta-termo ou uma classe, a qual cobre uma gama de outros tipos mais específicos de agentes, listando-os e definindo o que estes tipos de agentes são” (NWANA, 1996).

“Um agente é um sistema computacional situado em um ambiente onde é capaz de agir autonomamente para atingir os objetivos definidos em seu projeto” (WOOLDRIDGE, 2002).

“Basicamente, um agente é um sistema de *software* situado em um ambiente onde opera continuamente em um ciclo Perceber-Raciocinar-Agir (*Perceive-Reason-Act*)” (LIND, 2001).

“Agente é qualquer coisa que pode ser vista percebendo seu ambiente através de sensores e agindo sobre aquele ambiente através de seus atuadores (*effectors*)” (RUSSEL, 1995).

“A palavra agente refere-se a todo ente que possui a habilidade, capacidade e permissão para atuar em nome de outro” (YEPES, 2003).

Quando o termo agente for mencionado neste trabalho, estará referindo-se a entidades de *software* capazes de agir autonomamente para atingir seus objetivos.

2.3.2 Conceituação de Sistema Multi-Agente

Um SMA faz uso do conceito de “comunidade de agentes”, cujo enfoque baseia-se na existência de uma sociedade, composta por vários agentes que atuam no sistema por meio de cooperação e concorrência, sendo que é devido a esse “comportamento social”, que emerge a inteligência do sistema. Assim, o objetivo maior dos pesquisadores em SMA está na coordenação de tal comportamento social inteligente, uma vez que a coordenação envolve conhecimento, objetivos, habilidade e planejamento sobre os agentes e não necessariamente está voltada para a busca da solução de um problema específico (YEPES, 2003).

Uma possível definição formal para SMA é apresentada em (SYCARA, 1998): “uma rede fracamente acoplada de solucionadores de problemas que trabalham em conjunto para resolver problemas além das capacidades ou conhecimentos de cada um. Os solucionadores de problemas são os agentes, que podem ser autônomos e heterogêneos por natureza”.

A partir da contextualização apresentada, pode-se observar alguns dos principais aspectos presentes nos SMA: comunicação, coordenação, cooperação e negociação. A comunicação, apesar de não ser utilizada de forma explícita em alguns sistemas, é fundamental para promover a interação entre os agentes. A cooperação não significa somente um esforço coordenado para atingir um objetivo, ela implica em um objetivo compartilhado pelos agentes e o desejo de trabalhar em conjunto (WOOLDRIDGE, 1997).

A coordenação é um processo utilizado para garantir que uma comunidade de agentes aja de uma maneira coerente e harmoniosa, evitando interações prejudiciais e explorando interações benéficas (GREEN, 1997), (JENNINGS, 1998). A negociação é um processo realizado por um grupo de agentes visando um acordo aceito por todos (GREEN, 1997).

De acordo com Jennings (1998), a flexibilidade e o alto-nível destes padrões de interações (comunicação, cooperação, coordenação e negociação), são responsáveis pelo poder do paradigma multi-agente, além de diferenciá-lo dos demais paradigmas.

2.3.3 Padrões para Sistemas Multi-Agentes

Além de promover interoperabilidade entre os sistemas multi-agente heterogêneos, um padrão ajuda os desenvolvedores focarem na utilização de uma infra-estrutura, e não no seu desenvolvimento.

Em termos mercadológicos, a principal razão para empresas apoiarem e financiarem padrões *de facto* é estimular o mercado, visando obter consumidores potenciais, aplicativos e produtos a médio e longo prazo. Para que o mercado cresça mais facilmente os padrões devem ser disponibilizados publicamente.

Sendo assim, durante a padronização, as empresas perdem sua vantagem competitiva, a curto prazo, mas ganham-na a médio e longo prazo, devido a existência de produtos que adicionem valor e incrementem o padrão, promovendo um mercado muito mais amplo (POSLAD, 2000). Este é o motivo pelo qual muitas empresas, principalmente de telecomunicações, investem e apóiam padrões para SMA.

O escopo das especificações para SMA geralmente inclui a interpretação e tratamento de mensagens ACL (*Agent Communication Language*), a utilização de agentes facilitadores, o uso (mas não a especificação) de protocolos para o transporte de mensagens, gerenciamento de ciclo de vida dos agentes e uma estrutura para a persistência de mensagens (POSLAD, 2000). Segundo Poslad (2000-a), existem três importantes esforços visando a padronização de sistemas baseados em agentes: a KQML, o MASIF e a FIPA.

A KQML (*Knowledge Query Manipulation Language*) (FININ, 1997) foi uma das primeiras iniciativas para especificar como fornecer o suporte à interação social entre agentes através de um protocolo baseado em atos de fala, atualmente é uma das ACL mais importantes. Entretanto, a KQML não é considerada um padrão *de facto*, uma vez que não existe um consenso da comunidade em uma especificação (ou conjunto de especificações) única. Desta forma, existem diversas variações da KQML. A existência destas variações compromete a interoperabilidade entre SMAs que conversem em dialetos diferentes.

O OMG (*Object Management Group*) MASIF (*Mobile Agent System Interoperability Facility*) (MILOJICIC, 1999) difere-se da KQML e da FIPA por considerar que a característica principal de um agente é a sua mobilidade. Ao contrário do MASIF, tanto a FIPA quanto a KQML enfatizam a agência e a interação social como as características fundamentais dos agentes.

O MASIF não padroniza a comunicação entre agentes em diferentes plataformas. Além disso, a interoperabilidade restringe-se a agentes desenvolvidos em plataformas CORBA, enquanto a FIPA provê a interoperabilidade entre agentes no nível de arquitetura de agentes, podendo ter meios de transporte heterogêneos (POSLAD, 2000-a).

A FIPA (*Foundation for Intelligent Physical Agents*) (FIPA, 2005) é uma organização sem fins lucrativos estabelecida no final de 1996, na cidade de Geneva, Suíça. Seu objetivo é promover o desenvolvimento de especificações de tecnologias genéricas para agentes, procurando maximizar a interoperabilidade dentro e entre aplicações baseadas em agentes.

Devido à importância atual do padrão FIPA, o qual possui mais de 70 participantes, incluindo empresas e universidades, as principais especificações da FIPA serão apresentadas na seção seguinte. Outra razão para o maior enfoque na FIPA se deve à aplicação deste trabalho, que é em um portal educacional cuja plataforma de base é um sistema multi-agente compatível com a FIPA.

2.3.4 FIPA

Segundo Calisti (2003), a idéia fundamental da FIPA é oferecer meios padronizados para a interpretação de comunicações entre agentes de forma a respeitar o significado pretendido da mensagem. Na visão da FIPA, a interpretação correta das comunicações é obtida através da combinação de atos de fala, lógica de predicados e ontologias.

Um aspecto de fundamental importância, relativo às especificações, é que elas não padronizam nem a arquitetura interna do agente nem a maneira como este é implementado. Nelas, são especificadas as interfaces necessárias para promover a interoperabilidade.

O conjunto das principais especificações da FIPA define uma arquitetura abstrata para SMA, um modelo para o gerenciamento de agentes e uma linguagem de comunicação entre agentes. Neste trabalho serão apresentadas apenas as especificações mais relevantes para o desenvolvimento de SMA interoperáveis. A Figura 2.4 mostra um esquema dos tópicos tratados pelas especificações.

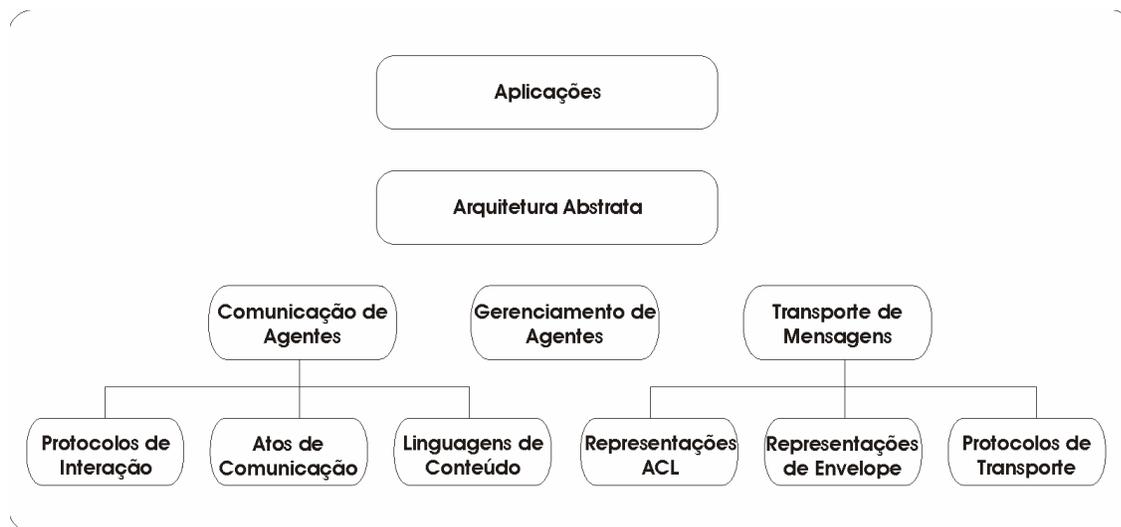


Figura 2.4 Esquema com os tópicos contemplados nas especificações da FIPA

2.3.4.1 Arquitetura Abstrata

O objetivo da especificação de uma arquitetura abstrata é promover a interoperabilidade e a reutilização. Para atingir estes objetivos é necessário identificar os elementos de uma arquitetura que devem ser codificados. Ou seja, se dois ou mais sistemas usam diferentes tecnologias para fornecer a mesma funcionalidade, é necessário identificar as características comuns destas várias abordagens.

As características em comum conduzem à identificação de abstrações de arquitetura: projetos abstratos que podem ser formalmente relacionados com todas as implementações válidas (FIPA, 2005-b).

Através da descrição abstrata de sistemas, é possível explorar as relações entre elementos fundamentais destes sistemas de agentes. A partir das descrições das relações, torna-se mais claro como SMA podem ser criados de forma que sejam interoperáveis. Partindo-se deste conjunto de elementos e relacionamentos arquiteturais, é possível derivar muitas arquiteturas concretas, as quais são capazes de interoperar, pois compartilham o mesmo projeto abstrato.

Apesar de não poder ser diretamente implementada, a arquitetura abstrata da FIPA forma a base para o desenvolvimento de especificações concretas de arquiteturas. Tais especificações definem precisamente como construir um SMA, incluindo os agentes e serviços necessários em termos de artefatos de *software* concretos, tais como: linguagem de programação, APIs (*Application Programming Interface*), protocolos de rede, sistema operacional, etc.

A arquitetura abstrata não impede a adição de outros elementos à arquitetura concreta, ela simplesmente define os requisitos mínimos. A Figura 2.5 ilustra um possível mapeamento da arquitetura abstrata para uma realização concreta.

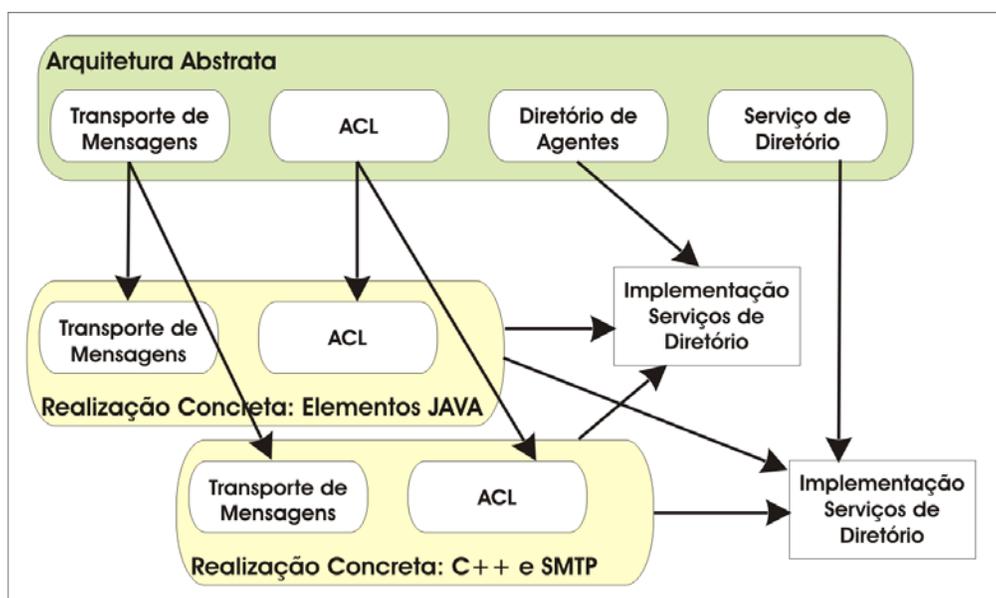


Figura 2.5 Dois mapeamentos da arquitetura abstrata para realizações concretas

Na especificação, todos os elementos da arquitetura são definidos em termos de suas relações com outros elementos, suas ações e sua descrição geral. Alguns elementos mais complexos possuem suas próprias especificações para defini-los, por exemplo, a ACL.

2.3.4.2 Gerenciamento de Agentes

A especificação do gerenciamento de agentes fornece a estrutura na qual os agentes FIPA existem e operam. Ela estabelece o modelo de referência lógico para a criação, registro, localização, comunicação, migração e remoção de agentes. As entidades contidas no modelo de

referência (Figura 2.6), são serviços e não implicam em nenhuma configuração física. Os detalhes de implementação de Plataformas de Agentes (PA) e dos agentes são escolhas de projeto dos desenvolvedores do SMA.

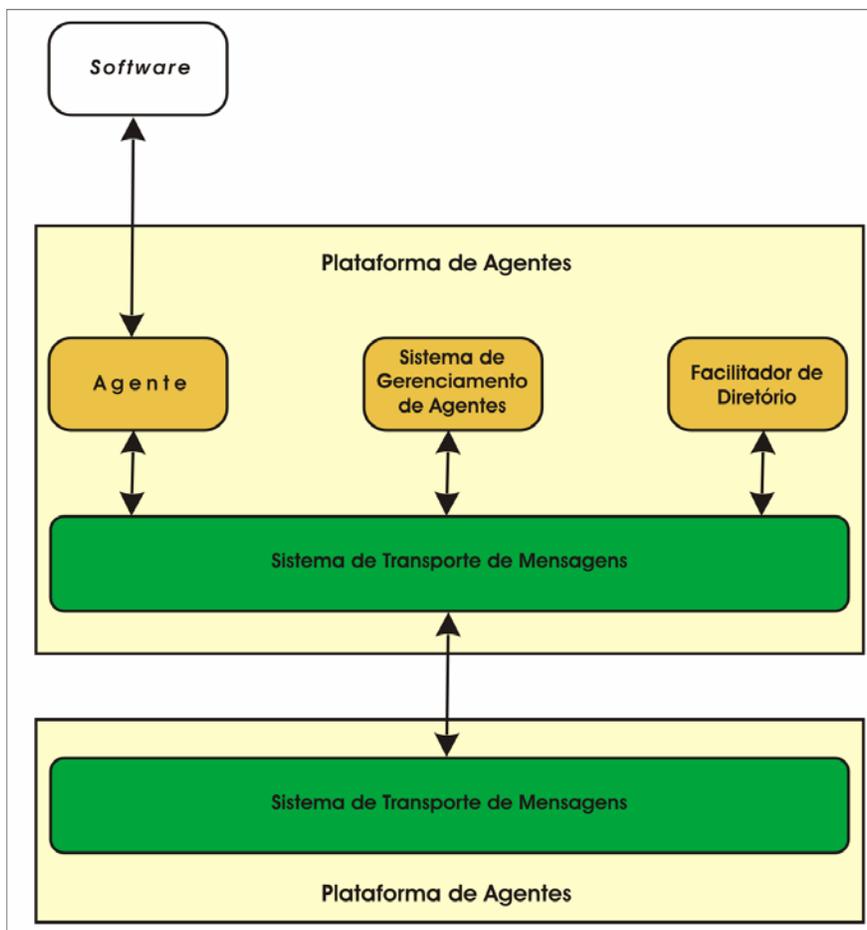


Figura 2.6 Modelo de Referência do Gerenciamento de Agentes FIPA

O modelo de referência consiste dos seguintes componentes lógicos (FIPA, 2005-e):

- **Agente:** processo computacional que implementa a funcionalidade autônoma e comunicativa de uma aplicação. O agente é o ator fundamental numa PA no qual combina uma ou mais capacidades de serviços, conforme publicado na sua descrição de serviço, em um modelo integrado e unificado de execução. Um agente deve ter, no mínimo, um dono e deve fornecer ao menos uma noção de identidade. Esta noção de identidade é o Identificador de Agente (*Agent Identifier* - AID), utilizado para distinguir um agente dos demais.
- **Facilitador de Diretório** (*Directory Facilitator* - DF): fornece um serviço de páginas amarelas (similar ao de uma lista telefônica) aos agentes. Todo agente que desejar publicar

seus serviços para outros agentes deve contatar um DF e solicitar o registro de sua descrição. Da mesma maneira, o agente pode desregistrar-se do DF ou modificar sua descrição. Outra função do DF é executar pesquisas, quando solicitado. O DF não garante a validade das informações registradas na sua base de dados, uma vez que não é imposta nenhuma restrição quanto à informação para o registro.

- **Sistema Gerenciador de Agentes** (*Agent Management System - AMS*): este é um componente obrigatório de uma PA e, em cada uma, existe somente um AMS. Ele é responsável pelo gerenciamento das funcionalidades de uma PA, tais como a criação e remoção de agentes, verificar a migração e emigração de agentes na plataforma, entre outros. O AMS mantém uma lista com todos os agentes residentes na plataforma, incluindo o AID de cada agente. Para estar residente em uma PA, um agente deve estar registrado no AMS. Assim como o DF fornece um serviço de páginas amarelas, o AMS fornece um serviço de páginas brancas.
- **Sistema para Transporte de Mensagens** (*Message Transport System - MTS*): O MTS entrega as mensagens tanto para agentes dentro de uma mesma PA quanto para agentes residentes em outras plataformas. Todos os agentes FIPA possuem, no mínimo, acesso a um MTS.
- **Plataforma de Agentes** (PA): fornece a infra-estrutura física na qual os agentes podem ser implantados. A PA consiste de *hosts*, sistema operacional, aplicativos para suportar agentes, componentes de gerenciamento de agentes da FIPA (DF, AMS e MTS) e agentes. O conceito de uma PA não significa que todos os agentes de uma PA devam estar localizados no mesmo *host*. A FIPA preocupa-se apenas com a maneira que a comunicação é realizada entre agentes de PAs diferentes, dentro da mesma PA, os agentes podem trocar mensagens diretamente de qualquer maneira que possam fazê-lo.
- **Software**: descreve todas as coleções de instruções executáveis acessíveis através do agente, mas que não façam parte do mesmo. Agentes podem acessar aplicativos para, por exemplo, adicionar novos serviços, adquirir novos protocolos de comunicação, utilizar novos protocolos/algoritmos de segurança, acessar ferramentas para suportar migração, etc.

Na especificação do gerenciamento de agentes (FIPA, 2005-e), também é definida toda a ontologia para o gerenciamento de agentes além da linguagem de conteúdo utilizada no gerenciamento, que é a FIPA-SL.

2.3.4.3 Comunicação entre agentes

A FIPA considera que os agentes precisam interagir num nível de discurso semanticamente rico para comunicar declarações significativas a respeito do mundo (O'BRIEN, 1998). A base da comunicação entre agentes FIPA são os atos de fala, também chamados de performativas ou atos de comunicação. A FIPA especifica um conjunto de performativas, tais como *request* (solicitação), *inform* (informar) e *refuse* (recusar) (FIPA, 2005-g), cada um com um significado bem definido e independente do conteúdo geral da mensagem.

A especificação (FIPA, 2005-c) define a mensagem ACL, que fornece os mecanismos para adicionar contexto ao conteúdo da mensagem e ao ato de comunicação. A contextualização da mensagem é feita adicionando informações relativas ao remetente e destinatário, ontologia, protocolo de interação, linguagem, entre outros. A Tabela 2.1 mostra os campos de uma mensagem ACL.

Tabela 2.1: Parâmetros de uma mensagem FIPA-ACL.

Parâmetros	
Performative	Ontology
Sender	Protocol
Receiver	Conversation-id
Reply-to	Reply-with
Content	In-reply-to
Language	Reply-by
Encoding	

Os protocolos de interação (FIPA, 2005-d), (FIPA, 2005-h), (FIPA, 2005-i), (FIPA, 2005-j), entre outros, definem uma seqüência de mensagens as quais representam uma conversação completa ou um diálogo entre agentes. Desta forma, um agente pode antecipar a resposta do outro de acordo com um padrão de conversação pré-definido. A Figura 2.7 mostra o protocolo FIPA-REQUEST elaborado utilizando a AUML (*Agent Unified Modelling Language*) (AUML, 2003).

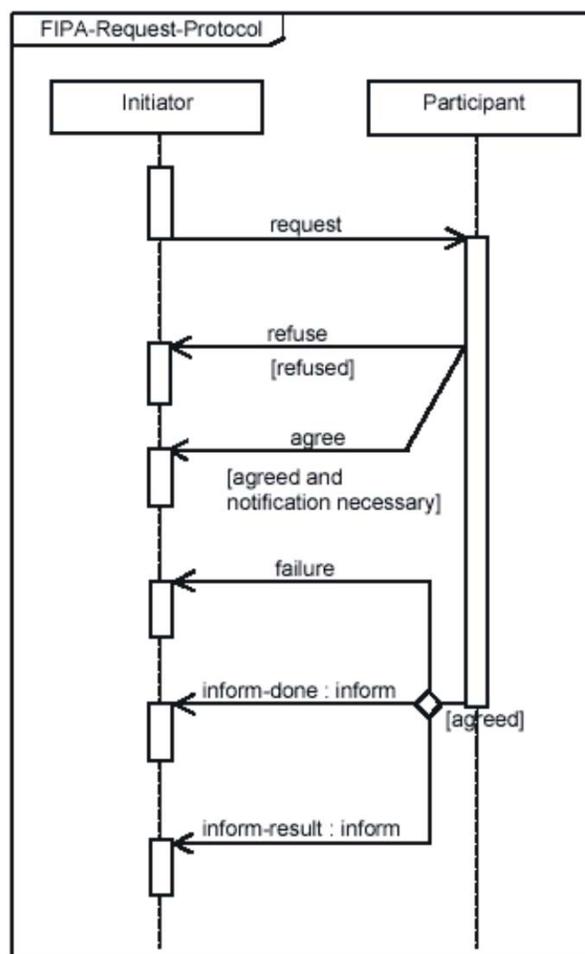


Figura 2.7 Diagrama AUML representando o protocolo de interação FIPA-REQUEST (FIPA, 2005-d)

Vale salientar que a FIPA não limita as interações entre os agentes somente aos seus protocolos pré-definidos. Os desenvolvedores podem adotar seus próprios protocolos ou até mesmo permitir que uma estrutura de conversação surja a partir das interações entre agentes. Entretanto, quando um agente utiliza algum protocolo já definido pela FIPA, este deve ser seguido conforme a especificação. Para expressar o conteúdo da mensagem, a FIPA provê uma linguagem de conteúdo opcional, a FIPA SL (FIPA, 2005-k).

2.3.4.4 Transporte de Mensagens de Agentes

Além das especificações de comunicações em mais alto nível apresentadas na subseção anterior, a FIPA também define o transporte das mensagens entre os agentes (FIPA, 2005-f). Estas especificações tratam da entrega e representação das mensagens entre diferentes protocolos de transporte, incluindo ambientes sem fio (*wireless*). Em (FIPA, 2005-f) é definido o modelo de referência (Figura 2.8) para o transporte de mensagens, o qual é composto por três níveis:

1. O Protocolo para o Transporte de Mensagens (*Message Transport Protocol* - MTP): é utilizado para a transferência física de mensagens entre dois ACCs⁴.

2. O Serviço de Transporte de Mensagens (*Message Transport Service* - MTS): é um serviço fornecido pela PA na qual o agente está inserido. O MTS possibilita o transporte de mensagens FIPA ACL entre agentes na mesma PA e também em plataformas diferentes.

3. A mensagem, em sua forma abstrata, é formada por um envelope, que expressa informações de transporte, e um *payload* (conteúdo), englobando a mensagem ACL da comunicação entre os agentes.

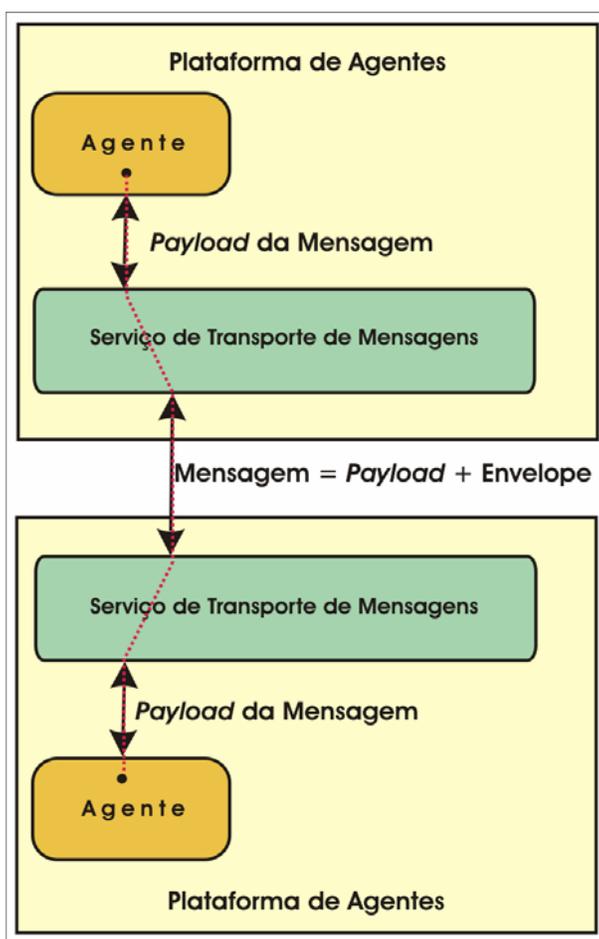


Figura 2.8 Modelo de referência do transporte de mensagens FIPA

O MTS de cada plataforma pode suportar qualquer quantidade de protocolos de transporte e, normalmente, é capaz de efetuar a tradução entre um MTP suportado pela FIPA (utilizado para comunicação entre PA heterogêneas), como o XML sobre HTTP e um MTP usado internamente,

⁴ O ACC (*Agent Communication Channel*) é uma entidade que fornece o serviço de transporte de mensagens (MTS) diretamente aos agentes de uma PA.

tais como objetos JAVA sobre RMI. Conseqüentemente, os componentes do MTS são modulares e extensíveis, para que possam tratar diferentes protocolos de transporte, envelopes de mensagem e diferentes representações ACL.

2.4 Considerações Finais

Neste Capítulo foram apresentadas as principais linguagens ontológicas para utilização em sistemas abertos, uma contextualização destas linguagens na *web* semântica, os principais conceitos sobre SMA e o padrão FIPA. Dentre as linguagens ontológicas estudadas, optou-se pela utilização da OWL como linguagem para elaborar a base de conhecimento do agente social. A OWL é a linguagem recomendada pela W3C para utilização na *web* semântica. Tal recomendação implica uma maior adoção desta linguagem pela comunidade científica, o que resulta no maior desenvolvimento desta linguagem e de ferramentas para sua utilização.

O agente social interage com os agentes presentes no PortEdu, o qual utiliza os padrões da FIPA para comunicação. Desta forma, é necessário fornecer uma infra-estrutura para que o agente possa comunicar-se de acordo com a FIPA-ACL e também as funcionalidades básicas para registrar-se na plataforma, realizar consultas na base de serviços, etc.

3 UMA ABORDAGEM PARA AUMENTAR A INTEROPERABILIDADE DO AGENTE SOCIAL

O propósito deste trabalho é fornecer os meios para que o agente social possa comunicar-se com agentes fora de seu contexto original, o ambiente AMPLIA. Desta forma, a funcionalidade desse agente poderá ser utilizada por outros ambientes de aprendizagem. Ao visualizar o AMPLIA como um ambiente presente em um portal educacional, o PortEdu, onde os agentes de todos os diferentes ambientes podem interagir diretamente entre si, a capacidade do agente social de comunicar informações sobre as interações dos usuários e gestão de grupos torna-se uma funcionalidade útil para toda a sociedade de agentes.

Para alcançar este objetivo é proposta uma abordagem baseada em ontologias. Neste Capítulo será apresentada esta abordagem. Primeiro é apresentada a arquitetura do agente social com a extensão para interoperabilidade. Na seção 3.2 é descrito o módulo para gerenciamento de ontologias. Na seção 3.3 é apresentado o módulo para comunicação externa e na seção 3.4 são apresentadas as considerações finais deste Capítulo.

3.1 Arquitetura Estendida do Agente Social

Na Figura 3.1 é apresentada a arquitetura do agente social, a interação com os agentes do AMPLIA e com outros agentes pertencentes ao PortEdu. Os componentes que tornam possível a comunicação com agentes externos são o Módulo para Gestão de Ontologias e o Módulo de Comunicação Externa. Esta arquitetura modular possibilita a evolução de componentes específicos (suporte a decisões, aquisição de informações, etc.) sem afetar as outras partes da arquitetura.

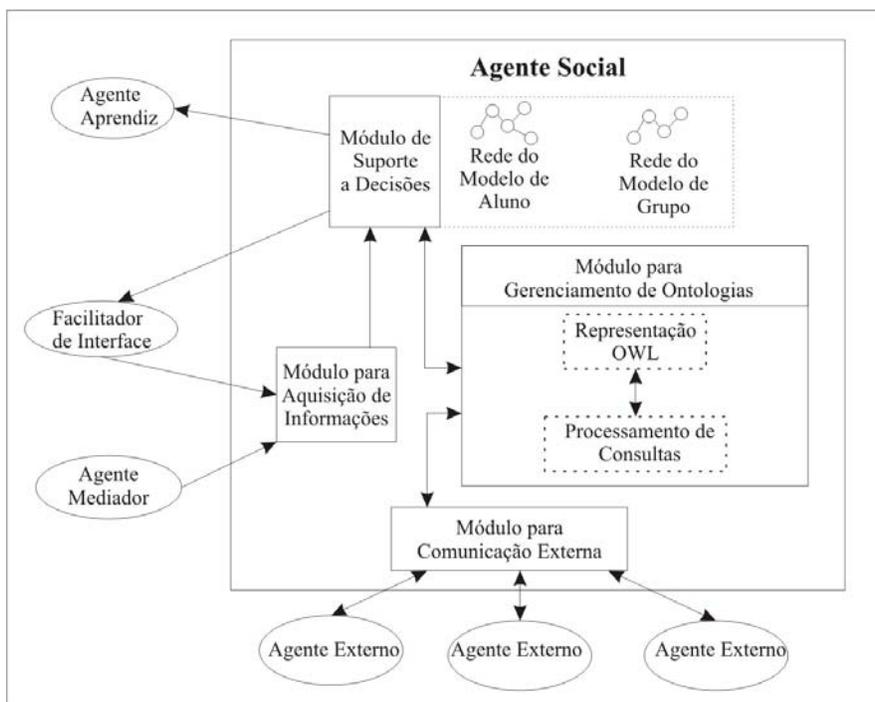


Figura 3.1 Arquitetura do agente social

A funcionalidade principal do agente social está no módulo para suporte de decisões, o qual é utilizado para definir quais alunos farão parte de um grupo, quando esses grupos devem ser criados, etc. Esse módulo representa o conhecimento do agente social sob a forma de redes bayesianas e diagramas de influência.

Através desse módulo, o agente social interage com o agente aprendiz para sugerir qual colega é recomendado para trabalhar em conjunto com o aluno representado pelo agente aprendiz. Essa recomendação também é enviada para o facilitador de interface, o qual exhibe para o usuário a recomendação.

O módulo para aquisição de informações obtém do agente mediador informações sobre o domínio e comparações entre a rede do aluno e a respectiva rede do especialista, como uma medida da evolução do aluno no conteúdo em pauta. Esse módulo também obtém informações relativas às interações do aluno através do facilitador de interface. As informações obtidas por esse módulo são utilizadas para alimentar o módulo para tomada de decisões.

A extensão da arquitetura do agente social é composta pelos módulos para gerenciamento de ontologias e módulo para comunicação externa. O módulo para gerenciamento de ontologias é composto por dois sub-módulos. Um deles é responsável por representar em OWL as redes bayesianas do agente social (módulo de representação OWL). O sub-módulo processamento de consultas é responsável por efetuar as consultas (relativas ao conhecimento do agente social, por exemplo: estudantes de um grupo, grupos disponíveis, etc.) recebidas pelo módulo de comunicação externa na base de conhecimento OWL, gerada pelo sub-módulo representação OWL.

O módulo de comunicação externa constitui a interface do agente social com os agentes externos, os agentes presentes no PortEdu. Esse módulo recebe mensagens dos agentes do portal, analisa-as e passa as solicitações de consulta para o módulo de gerenciamento de ontologias.

Os módulos para gerenciamento de ontologias e para comunicação externa constituem o núcleo da abordagem proposta para ampliar a interoperabilidade do agente social. Por esse motivo, esses módulos serão descritos em detalhe nas seções a seguir.

3.2 Módulo para Gerenciamento de Ontologias

A principal utilidade do módulo para gestão de ontologias é fornecer uma representação sob a forma de ontologia do conhecimento do agente social. Outra funcionalidade desse módulo é o processamento de consultas (*queries*).

A representação ontológica do conhecimento do agente social foi desenvolvida utilizando a linguagem para ontologias OWL. Esta linguagem foi escolhida por ser uma recomendação da W3C, o que confere praticamente o *status* de um padrão a esta linguagem. Sua utilização como linguagem ontológica para a *web* semântica também denota a evolução da linguagem, uma vez que ela foi desenvolvida a partir de pesquisas anteriores em linguagens ontológicas (SHOE, DAML-ONT, OIL, DAML+OIL) voltadas para a *web*, conseqüentemente, com foco em intercâmbio de informações.

A OWL foi desenvolvida em consonância com a visão da *web* semântica proposta por Berners-Lee (2000) e que está desenvolvida principalmente, por um grupo específico da W3C. É uma linguagem para escrever ontologias de forma a possibilitar o intercâmbio do conhecimento descrito entre aplicações (agentes), construídos separadamente, ou seja, sem a noção prévia da existência de informações em outros contextos. Esta funcionalidade da OWL e sua aceitação pela comunidade científica são os fatores principais para sua utilização neste trabalho.

O fato de ser uma recomendação da W3C e ser amplamente aceita pela comunidade da *web* semântica impulsiona o desenvolvimento de ferramentas para dar suporte a essa linguagem. A existência de aplicativos para a edição de arquivos OWL, implementações de APIs, motores de inferência, *parsers*, entre outros, também foi levada em consideração ao optar pela utilização da OWL.

O agente social, descrito na seção 1.3 baseia-se em idéias da psicologia social (para fundamentar os aspectos sociais considerados) e estados afetivos. Internamente, o agente utiliza redes bayesianas (RB) para representar seu conhecimento e tomar suas decisões. Sendo assim, para que seja entendido por agentes externos, é necessária a utilização de uma representação mais adequada à comunicação em ambientes heterogêneos. Neste caso, adotamos a OWL como a linguagem para representar esse conhecimento.

O conhecimento do agente social é um conhecimento probabilístico, representado em redes bayesianas. Em sua concepção, a OWL não suporta representação de incerteza. Desta forma, para representar o conhecimento probabilístico do agente em OWL foi utilizada a abordagem BayesOWL (PAN, 2005) adaptada e estendida para um contexto mais genérico e não somente para mapeamento entre conceitos, objetivo da BayesOWL.

3.2.1 Redes Bayesianas Representadas em OWL

As duas principais abordagens para representação de incerteza em OWL são a BayesOWL (PAN, 2005) e a PR-OWL (COSTA, 2005). A BayesOWL foi desenvolvida para lidar com o problema de mapeamento entre ontologias. A idéia central é acrescentar probabilidades para definir a incerteza entre os conceitos das ontologias sendo mapeadas. Em seguida, as ontologias com as informações adicionais são traduzidas para uma RB, a qual preserva a semântica da ontologia original. A partir da RB, é possível propagar a evidência (o método cria automaticamente as tabelas de probabilidades condicionais) e realizar inferências para realizar o mapeamento considerando a incerteza.

A proposta da PR-OWL é estender a própria linguagem OWL, fornecendo fundamentação para representação de incerteza através de *Multi-Entity Bayesian Networks* (MEBN) (LASKEY, 2005), uma lógica que combina redes bayesianas e lógica de primeira ordem. Inicialmente, foi desenvolvida uma ontologia de alto-nível (*upper level ontology*) para que fosse possível expressar incerteza, sob a forma de MEBN, em uma ontologia escrita em OWL.

Ambas abordagens possuem vantagens e desvantagens. A BayesOWL, por exemplo, foca no mapeamento de ontologias, desta forma, a estrutura é limitada para alcançar este objetivo. Por exemplo, os conceitos expressam somente dois estados, verdadeiro ou falso, uma vez que estão relacionados apenas a questão da compatibilidade entre conceitos. Entretanto, a abordagem proposta utiliza os recursos da OWL atual, sem criar um dialeto específico, desta forma o desenvolvimento torna-se mais rápido.

Já a abordagem da PR-OWL é mais completa, a extensão da linguagem é feita com uma sólida base lógica, possibilitando representação e raciocínio baseado em incerteza diretamente. Apesar de fornecer maior expressividade, existem poucas ferramentas para dar suporte a esta abordagem. Além disso, torna-se necessária a padronização por parte de um órgão como a W3C, já que é proposta uma extensão na linguagem.

Neste trabalho, utilizamos como base inicial para a representação probabilística a abordagem BayesOWL, devido ao fato de manter a OWL acrescentando apenas marcações adicionais, sua conseqüente praticidade e possibilidade de utilização de todas as ferramentas já disponíveis para OWL durante o desenvolvimento.

A PR-OWL modela redes bayesianas de uma forma mais ampla porém com suporte mínimo ao desenvolvimento. Além disso, as redes do agente social encontram-se modeladas através de redes tradicionais, a utilização desta abordagem acarretaria em mudanças em toda a estrutura para tomada de decisão do agente e também levantaria questões quanto à compatibilidade com os demais agentes do AMPLIA.

3.2.2 Ontologia em OWL para Representar Redes Bayesianas

Para representar as redes bayesianas do agente social em OWL é proposta uma ontologia na qual todos os conceitos necessários para representar o conhecimento probabilístico do agente são fornecidos. As classes definidas nesta ontologia permitem a tradução das redes para OWL.

A definição desta ontologia foi feita utilizando o Protégé com um *plugin* para OWL. Desta forma, uma aplicação pode, automaticamente, gerar indivíduos representando suas redes probabilísticas e um especialista pode acrescentar informações à ontologia do seu domínio ou até mesmo atualizar informações sobre as instâncias. A Figura 3.2 ilustra o desenvolvimento de ontologias OWL no Protégé.

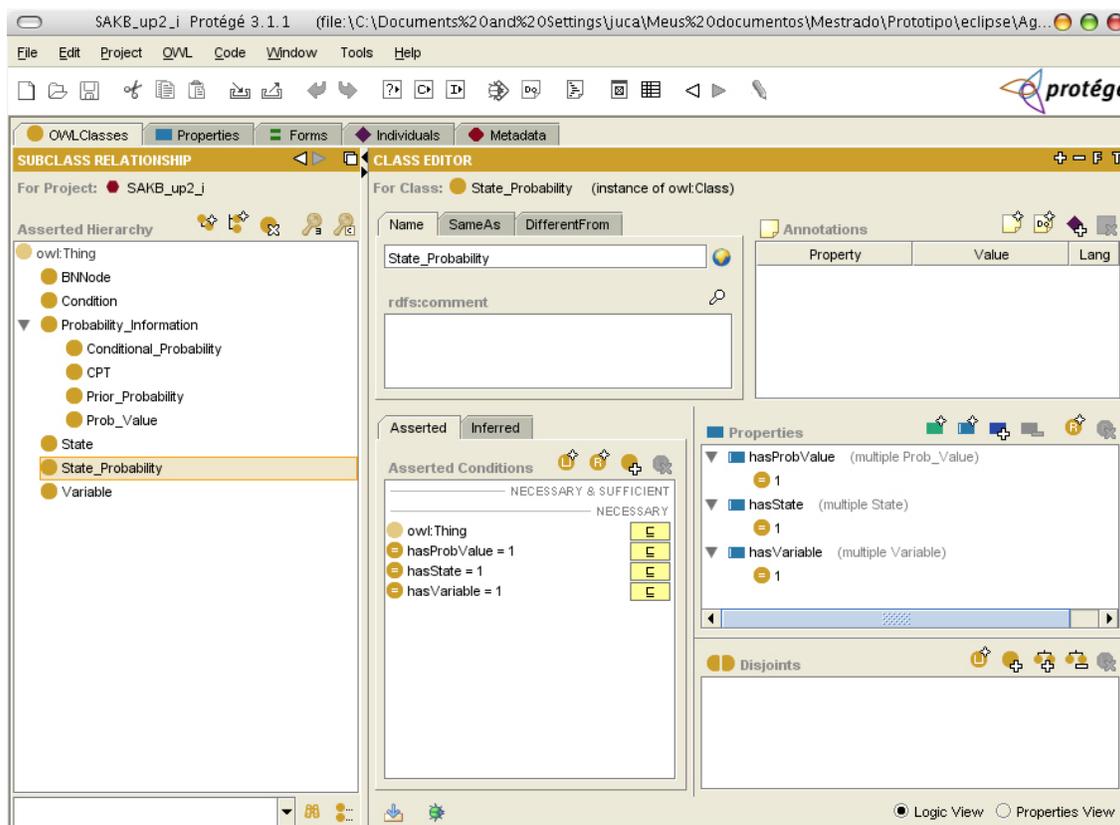


Figura 3.2 Screenshot do desenvolvimento de ontologias OWL através do Protégé.

As classes definidas na ontologia são as seguintes:

- BNNode – representa um nodo de uma rede bayesiana, sua utilização não é obrigatória, se desejar, o usuário pode utilizar diretamente a classe *variable*.
- Variable – representa uma variável probabilística. A implementação atual suporta variáveis probabilísticas discretas. Variáveis de decisão e utilidade não foram implementadas.
- State – utilizada para definir um estado
- State_Probability – especifica o valor probabilístico de um estado de uma variável discreta. Esta classe é utilizada em conjunto com as classes Prior_Probability ou CPT.
- Prior_Probability – expressa as probabilidades de uma variável discreta não condicional, ou seja, que não sofre influência de nenhuma outra variável. São variáveis pertencentes a nodos órfãos. As propriedades são um nome e uma lista de indivíduos da classe

StateProbability. Na rede bayesiana ilustrada na Figura 3.3, os nodos Interesse e Compromisso teriam as informações referentes às probabilidades de seus estados representados por um indivíduo desta classe.

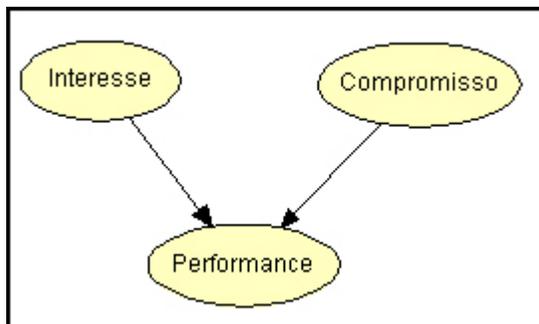


Figura 3.3 Rede Bayesiana com dois nodos órfãos.

- Condition – define uma condição. É utilizada para definir uma condição de uma probabilidade condicional referente a uma variável. Suas propriedades são um nome, uma variável e um estado. Tomando como exemplo a Tabela 3.1, as condições seriam: Estado Afetivo + Satisfação, Estado Afetivo + Frustração, Estado Afetivo + Alegria e assim sucessivamente. A funcionalidade desta classe torna-se expressiva ao ser utilizada nas classes Conditional_Probability e CPT.

Tabela 3.1 Tabela de probabilidades condicionais destacando quais são as condições

Estado Afetivo \ Humor	Satisfação	Frustração	Alegria	Tristeza	Gratidão	Raiva	Vergonha
Bom	0,9	0,1	0,9	0,1	0,7	0,05	0,5
Ruim	0,1	0,9	0,1	0,9	0,3	0,95	0,5

- Probability_Information – uma abstração de alto-nível para as classes Prior Probability, Conditional Probability, CPT e Probability Value.
- Probability_Value – expressa o valor numérico de uma probabilidade.
- Conditional_Probability – utilizada para especificar uma probabilidade condicional. Através dela definem-se as condições (variável e estado) e os respectivos valores probabilísticos dos estados da variável em questão. Indivíduos desta classe geralmente compõem uma tabela condicional de probabilidades, CPT. Na Tabela 3.2 são destacadas as informações representadas nesta classe. Seguindo esta tabela, um indivíduo de Conditional_Probability seria:

Humor: Bom

Condição 1 → Estado Afetivo: Satisfação. Valor = 0,9

Condição 2 → Estado Afetivo: Frustração. Valor = 0,1

Condição 3 → Estado Afetivo: Alegria. Valor = 0,9

Condição 4 → Estado Afetivo: Tristeza. Valor = 0,1

Condição 5 → Estado Afetivo: Gratidão. Valor = 0,7

Condição 6 → Estado Afetivo: Raiva. Valor = 0,05

Condição 7 → Estado Afetivo: Vergonha. Valor = 0,5

Tabela 3.2 Tabela de probabilidades condicionais com destaque para os indivíduos da classe Conditional Probability.

Humor \ Estado Afetivo	Satisfação	Frustração	Alegria	Tristeza	Gratidão	Raiva	Vergonha
Bom	0,9	0,1	0,9	0,1	0,7	0,05	0,5
Ruim	0,1	0,9	0,1	0,9	0,3	0,95	0,5

- CPT – Conditional Probability Table, representa uma tabela condicional de probabilidades. Uma instância desta classe é composta por indivíduos da classe Conditional_Probability. Pode-se considerar que cada indivíduo de Conditional_Probability é uma linha da tabela.

A Figura 3.4 mostra a hierarquia de classes da ontologia proposta. O apêndice A deste trabalho apresenta toda a ontologia em OWL completa.



Figura 3.4 Hierarquia de classes da ontologia para representação de redes bayesianas

Comparando-se esta ontologia com a ontologia proposta pela BayesOWL observam-se poucas semelhanças. Isso se deve ao fato de que a ontologia da BayesOWL foi utilizada como um ponto de partida para elaborar uma ontologia capaz de representar em OWL uma RB.

Tendo o objetivo específico de facilitar o mapeamento de conceitos entre ontologias diferentes, a ontologia da BayesOWL não fornece suporte para definir variáveis com múltiplos estados nem tabelas de probabilidades condicionais, conceitos fundamentais em RB.

A atual ontologia não fornece suporte para variáveis de decisão e funções de utilidade. O primeiro passo deste trabalho é apresentar uma maneira para representar redes bayesianas em

OWL, a partir desta base é possível desenvolver ampliações para suportar diagramas de influência.

O conhecimento sob a forma de diagramas de influência é utilizado pelo agente social para recomendar, por exemplo, quais estudantes devem formar um grupo. Neste caso, a implementação atual possibilita representar toda a rede menos as variáveis de decisão e nodos utilitários. Na seção 3.2.3 será apresentada uma rede utilizada pelo agente social e o respectiva ontologia resultante do processo de tradução.

3.2.3 Conversão Automática de uma RB para uma Representação em OWL

A representação da rede bayesiana do agente social em OWL é realizada automaticamente. Foi desenvolvida uma classe em Java para realizar a tradução. A implementação utiliza a API Jena (CARROL, 2004) para lidar com os aspectos da ontologia (criar instâncias, atualizar valores, armazenar a ontologia, etc.) e a API para Java do Hugin (JENSEN, 2002), a qual é utilizada para obter informações sobre a rede bayesiana.

Jena é um kit de ferramentas para desenvolvimento de aplicações para *web* semântica em Java. São fornecidas APIs para OWL, DAML+OIL e RDFS. Outras funcionalidades são o suporte à inferência para semântica de RDF e OWL, conforme as especificações da W3C e suporte para uma linguagem de consulta, a SPARQL (PRUD'HOMMEAUX, 2006).

As ferramentas do Hugin consistem em um motor de decisões e uma interface gráfica para o usuário. Um conjunto de APIs (*Application Programming Interfaces*) permitem aos desenvolvedores acessar o motor de decisões. APIs são disponibilizadas para C, C++ e Java (JENSEN, 2002).

O módulo de tradução é implementado na classe BN2OWL, a qual fornece métodos para carregar uma rede e representá-la através de objetos, manipular ontologias OWL e construir automaticamente uma ontologia OWL a partir de uma rede bayesiana.

O processo de tradução inicia com a obtenção da rede bayesiana e sua representação em uma estrutura de objetos Java. A implementação atual trabalha com arquivos de redes bayesianas no formato padrão do Hugin, o qual também é utilizado por outros editores. Desta forma, a rede pode ser criada no editor de preferência do usuário, convertida para o formato NET e então ser traduzida para OWL.

O trecho de código a seguir destaca a utilização da API do Hugin para Java, utilizada para carregar a rede e formar sua representação através de objetos Java. O método retorna um objeto do tipo *Domain*, o qual pode representar uma rede bayesiana ou um diagrama de influência. Tendo-se um objeto *Domain* é possível propagar evidência, obter os marginais, calcular as crenças e triangular a rede. Os parâmetros de entrada do método são uma URL, com o endereço da rede e o nome da rede a ser carregada (*className*).

Na primeira linha é definido um *parser* para lidar com erros durante a aquisição da rede. Este *parser* deve ser implementado pelo usuário. Em seguida é criado um objeto *ClassCollection*, o qual é utilizado para manter uma ou mais redes bayesianas, salvá-las em arquivo, criar novas redes (consideradas classes pois esta API já suporta redes orientadas a objetos, para reutilização

de nodos, por exemplo), apagar redes e obter uma rede específica. O método `createDomain()`, utilizado na penúltima linha, transforma a classe carregada em uma rede bayesiana ou diagrama de influência, possibilitando aplicar utilizar as funções específicas destas estruturas.

```
protected Domain abreBN(String url, String className)
{
    ...
    BNParserImpl parseListener = new BNParserImpl();
    ClassCollection cc = new ClassCollection ();
    cc.parseClasses (url, parseListener);
    Domain domain = cc.getClassByName (className).createDomain ();
    return domain;
}
```

A classe `BN2OWL` foi desenvolvida, primeiramente, para ser utilizada pelo agente social, sendo assim, ela disponibiliza um método para carregar a rede do agente e diretamente realizar sua conversão para uma ontologia OWL. Este método será utilizado para explicar o processo de tradução e as principais funcionalidades da API Jena para OWL que foram utilizadas neste trabalho.

A partir de um objeto *Domain*, referente à rede bayesiana, é possível obter uma lista com todos os nodos da rede. Cada objeto *Nodo* armazena todas as informações referentes ao respectivo nodo da rede. Cada nodo é então processado da seguinte maneira:

1. São obtidos o nome e o *label* do nodo (1)⁵ e (2);
2. É obtida a lista de estados do nodo (3);
3. Como tratam-se de nodos probabilísticos discretos, não há uma diferenciação entre nodo e variável, sendo assim, é considerado que o nome do nodo é mesmo que o da variável, sendo assim, com os dados já obtidos, é possível criar uma variável na ontologia OWL (4). Os métodos para manipulação da ontologia serão explicados mais adiante nesta mesma sub-seção.
4. É obtida uma lista dos nodos filhos. São considerados somente os nodos probabilísticos discretos, representados pela classe `LabelledDCNode`. (5)
5. É obtida uma lista dos nodos pais. Esta lista é gerada da mesma forma como a lista dos nodos filhos. (6)
6. Caso trate-se de um nodo órfão, deve-se criar um indivíduo da classe `Prior_Probability` para representar as probabilidades dos estados na ontologia. Conforme apresentado na seção 3.2.2, um indivíduo `Prior_Probability` é formado por objetos `State_Probability`, os quais definem uma probabilidade para um determinado estado. As informações relativas às probabilidades dos estados são obtidas através da tabela de probabilidades do nodo. Para cada estado, obtém-se a respectiva probabilidade e é criado um indivíduo `State_Probability` para representá-lo na ontologia. Conforme estes indivíduos são criados, suas URIs são armazenadas para serem utilizadas (referenciadas) no indivíduo `Prior_Probability`. A partir das probabilidades dos estados, é gerado o indivíduo `Prior_Probability` o qual representa as probabilidades deste nodo. Em seguida, é criada

⁵ A numeração entre parênteses refere-se ao respectivo trecho de código fonte, o qual é apresentado após a descrição do processamento de cada nodo.

na ontologia uma instância de `BNNode` para representar esse nodo utilizando todas as informações obtidas até esse ponto: nome, *label*, estados, variável, pais, filhos e probabilidades dos estados. (7)

7. Tratando-se de um nodo com um ou mais pais, as probabilidades dos seus estados são influenciadas pelas probabilidades dos estados de seus pais, ou seja, trata-se de probabilidades condicionais, as quais formam uma tabela de probabilidades condicionais (CPT). A API do Hugin apresenta as informações sobre as probabilidades e as respectivas condições separadamente. Sendo assim, primeiro são criados as instâncias do conceito `Condição`, as quais representam cada uma das condições relacionadas às probabilidades dos estados deste nodo. Juntamente com as instâncias na ontologia, as URIs são armazenadas, em uma matriz, para serem referenciadas na CPT (8). Nesse ponto, a tabela de probabilidades condicionais possui apenas as condições, agora as probabilidades serão acrescentadas. Este processo é realizado para facilitar o processamento posterior da tabela, uma vez que a representação separada do Hugin não permite o processamento direto dos dados, uma vez que encontram-se separados e sem uma relação explícita entre eles. A relação entre as probabilidades e os respectivos nodos condicionais está na ordem em que os mesmos são apresentados (`NodeList nodosCondicionais = table.getNodes()`). Entretanto, a estrutura de dados que exibe os nodos é uma lista simples, sendo necessário um processamento dessa lista para passá-la para uma tabela, tornando mais intuitiva a geração dos indivíduos que comporão a tabela condicional na ontologia (9). Em seguida as probabilidades são obtidas e estruturadas juntamente com as condições. A partir destas estruturas, para cada condição existente é obtida a respectiva probabilidade de cada estado do nodo e então é criada sua respectiva instância da classe `StateProbability`, indicando para este estado, relativo às respectivas condições, qual a sua probabilidade. Tendo os indivíduos `StateProbability` criados na ontologia, estes são relacionados às instâncias de `ConditionalProbability`, as quais unem as probabilidades dos estados e as condições. O próximo passo é criar a tabela condicional de probabilidades, a qual é composta por todas as instâncias de `ConditionalProbability` criadas anteriormente. Finalmente, todas essas informações são utilizadas para criar uma instância de `BNNode` representando este nodo. Neste caso, ao invés de criá-lo com uma `PriorProbability`, ele é criado com uma CPT (10).

Os trechos de código a seguir foram extraídos do método `processaNodo`, o qual implementa o processamento descrito anteriormente.

```
(1) nomeNodo = node.getLabel();
(2) labelNodo = node.getName();
(3) List<String> estados = new ArrayList<String>();
(3) for (int i = 0, n = node.getNumberOfStates (); i < n; i++)
    {
(3)     estados.add(node.getStateLabel(i));
    }
```

```

(4) geraVariable( getOntModel(), nomeNodo, estados);

(5) List<String> filhos = new ArrayList<String>();
(5) NodeList filhosl = node.getChildren();
(5) for (int i = 0; i < filhosl.size(); i ++)
{
(5)   Object temp = filhosl.get(i);
(5)   if (temp.getClass().equals(LabelledDCNode.class))
{
(5)     LabelledDCNode filho = (LabelledDCNode) filhosl.get(i);
(5)     filhos.add(filho.getLabel());
}
}

(6) List<String> pais = new ArrayList<String>();
(6) NodeList paisl = node.getParents();
(6) for (int i = 0; i < paisl.size(); i ++)
{
(6)   Object temp = paisl.get(i);
(6)   if (temp.getClass().equals(LabelledDCNode.class))
{
(6)     LabelledDCNode pai = (LabelledDCNode) paisl.get(i);
(6)     pais.add(pai.getLabel());
}
}

Table table = node.getTable();
NodeList nodosCondicionais = table.getNodes();
String probabilityInfo;
(7) if (nodosCondicionais.size() == 1)
{
(7)   double [] probs = table.getData();
(7)   List <Double> probsg = new ArrayList<Double>();
(7)   for (int i = 0; i < probs.length; i ++)
{
(7)     probsg.add(new Double(probs[i]));
}
(7)   List<String> stbs = new ArrayList<String>();
(7)   int i = 0;
(7)   for (String estado : estados)
{
(7)     String nome = "STP_"+node.getLabel()+"_" +estado;
(7)     geraStateProbability( getOntModel(), nome, node.getLabel(), estado,
        probsg.get(i));
(7)     stbs.add(nome);
(7)     i++;
}
(7)   probabilityInfo = "PPB_" + node.getLabel();
(7)   geraPriorProbability(getOntModel(), probabilityInfo, stbs);
(7)   geraBNNode(getOntModel(), node.getLabel(), pais, filhos,
        node.getLabel(), "", probabilityInfo, "");
}

else if (nodos.size() > 1)
{
List<LabelledDCNode> nodosq = listaSemProprio(nodos, node);
int colunas = nodosq.size();
int linhas = calculaCondicoes(nodosq);

```

```

String [][] condicoes = new String [linhas] [colunas];
int linha;
(9) for (int col = 0 ; col < colunas; col ++)
{
(9)     linha = 0;
(9)     LabelledDCNode nodoAtual = nodosq.get(col);
(9)     int reps = calculaRepeticoes( obtemNodosRestantes(nodosq, col),
                                nodoAtual);
(9)     while (linha < linhas)
    {
(9)         for ( int st = 0; st < nodoAtual.getNumberOfStates(); st ++)
            {
(9)                 for (int o = 0; o < reps; o ++)
                    {
(9)                        String nome = "COND_" + nodoAtual.getLabel() + "_" +
                                nodoAtual.getStateLabel(st);
(9)                        condicoes[linha][col] = nome;
(9)                        linha++;
(9)                        geraCondition(getOntModel(), nome, nodoAtual.getLabel(),
                                nodoAtual.getStateLabel(st));
                    }
            }
    }
}

int bub = 0;
(10) double [] probso = table.getData();
List<String> nomecps = new ArrayList<String>();
for (int l = 0 ; l < linhas; l ++)
{
    List<String> cps = new ArrayList<String>();
    for (int c = 0 ; c < colunas; c++)
    {
        cps.add(condicoes[l][c]);
    }
(10) List<Double> probs = new ArrayList<Double>();
(10) for (int es = 0 ; es < estados.size(); es++)
    {
(10)         probs.add( new Double(probso[bub]));
(10)         bub++;
    }
(10) int x =0;
(10) List<String> stbs = new ArrayList<String> ();
(10) for (String estado : estados)
    {
(10)         String nomeSTP = "STP_"+node.getLabel()+"_"+estado+"_" + l;
(10)         geraStateProbability(getOntModel(), nomeSTP, node.getLabel(),
                                estado, probs.get(x));
(10)         stbs.add(nomeSTP);
(10)         x++;
    }
(10) String nomeCP = "CP_" + node.getLabel() + "_" + l;
(10) geraConditionalProbability(getOntModel(), nomeCP, cps, stbs);
    nomecps.add(nomeCP);
}
(10) probabilityInfo = "CPT_" + node.getLabel();
(10) geraCPT(getOntModel(), probabilityInfo, nomecps);
(10) geraBNNode(getOntModel(), node.getLabel(), pais, filhos,
                node.getLabel(), probabilityInfo, "", "");
}

```

}

Na classe BN2OWL são disponibilizados métodos utilitários para a criação direta de indivíduos na ontologia. A implementação dos métodos utiliza a API Jena para acessar e modificar a ontologia. Os métodos disponíveis são os seguintes:

- `abreOntologia` – acessa uma ontologia em arquivo ou em através de uma URL e retorna um objeto `OntModel`, da API Jena, o qual representa a ontologia. Através deste objeto é possível realizar todas as operações referentes à manipulação da ontologia (criar instâncias, classes, propriedades, definir hierarquia entre conceitos, realizar consultas, etc.)
- `abreOntologiaPersistente` – tem o mesmo objetivo que o método anterior, porém todas as modificações realizadas na ontologia são refletidas no arquivo original da ontologia.
- `geraVariable` – este método tem como parâmetro o nome da variável e uma lista de estados desta variável. Os estados são criados com o método `geraState` as URIs são então armazenadas nesta instância através da propriedade `hasState`. Após a execução deste método é criado na ontologia um indivíduo da classe `Variable`. O trecho de código OWL a seguir mostra um exemplo de resultado da execução deste método.

```
<Variable rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Interest">
  <hasState rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/ExploratoryInterest"/>
  <hasState rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/ContentInterest"/>
  <hasState rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/TaskInterest"/>
</Variable>
```

- `geraState` – recebe como parâmetro apenas o nome do estado o respectivo estado na ontologia como uma instância da classe `State`. A seguir é apresentado um trecho de código OWL ilustrando o resultado deste método.

```
<State rdf:about="http://gia.inf.ufrgs.br/agenteSocial/compreensivo"/>
```

- `geraStateProbability` – seus parâmetros são o nome da probabilidade, apenas utilizado para ser posteriormente referenciado, o nome do estado e da respectiva variável e a valor, em ponto flutuante, da probabilidade. As instâncias da classe `StateProbability` geradas através deste método não apresentam funcionalidade se observadas sozinhas. Estes indivíduos tornam-se úteis a partir do momento em que são utilizados para definir uma probabilidade condicional ou os valores probabilísticos de um nodo órfão (`PriorProbability`). A seguir é apresentado um exemplo de resultado deste método.

```
<State_Probability
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/STP_PersonalityTraits_falador">
  <hasProbValue>
    <Prob_Value rdf:about="http://gia.inf.ufrgs.br/agenteSocial/P21">
      <hasFloatValue rdf:datatype=http://www.w3.org/2001/XMLSchema#float>
        0.06
      </hasFloatValue>
    </Prob_Value>
  </hasProbValue>
  <hasState>
    <State rdf:about="http://gia.inf.ufrgs.br/agenteSocial/falador"/>
  </hasState>
  <hasVariable rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/PersonalityTraits"/>
</State_Probability>
```

- `geraProbValue` – cria uma instância de `Prob_Value`, o qual representa um valor probabilístico. Recebe como parâmetros um nome e o próprio valor da propriedade. O trecho de código OWL mostra um resultado da execução deste método.

```
<Prob_Value rdf:about="http://gia.inf.ufrgs.br/agenteSocial/P18">
  <hasFloatValue rdf:datatype=http://www.w3.org/2001/XMLSchema#float>
    0.07
  </hasFloatValue>
</Prob_Value>
```

- `geraCondition` – cria um indivíduo da classe `Condition` para representar uma condição. Os parâmetros utilizados são o nome da condição, o nome de uma variável previamente criada e seu respectivo estado. Um resultado deste método é exibido a seguir.

```
<Condition rdf:about="http://gia.inf.ufrgs.br/agenteSocial/COND_Commitment_Low">
  <hasVariable rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Commitment"/>
  <hasState rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Low"/>
</Condition>
```

- `geraPriorProbability` – recebe como parâmetros um nome e uma lista com os nomes de indivíduos da classe `StateProbability`, os quais representam as probabilidades dos estados de uma variável órfã. O indivíduo criado com este método é referenciado na propriedade `hasProbabilityInformation` da classe `BNNNode`.
- `geraConditionalProbability` – gera uma probabilidade condicional, conforme especificado na seção 3.2.2. As entradas deste método são um nome, uma lista de condições (indivíduos da classe `Condition`) e uma lista de probabilidades de estados (`StateProbability`). Cada instância desta classe representa uma linha da tabela de probabilidades condicionais.
- `geraCPT` – a partir de um nome e de uma lista de instâncias da classe `ConditionalProbability` gera um indivíduo da classe `CPT`, o qual representa uma tabela de probabilidades condicionais.
- `geraBNNNode` – este método cria um indivíduo da classe `BNNNode` e seus parâmetros são o nome do nodo, seu label, sua variável, uma lista com os nodos pais, uma lista com os filhos e uma instância da classe `ProbabilityInformation`, podendo ser um indivíduo de `PriorProbability` ou de `CPT`.

A seguir é descrito um dos métodos para a criação de indivíduos em OWL. Os outros métodos seguem a mesma estrutura, sendo modificados os parâmetros e quais conceitos da ontologia são utilizados.

```
void geraStateProbability(OntModel ontModel, String nome, String variavel,
                        String estado, Double prob)
{
  OntClass c = ontModel.getOntClass( NAMESPACE + "State_Probability" );
  Individual inst = ontModel.createIndividual( NAMESPACE + nome, c );
  ObjectProperty p = ontModel.getObjectProperty( NAMESPACE + "hasState" );
  Resource res = ontModel.createResource( NAMESPACE + estado );
  inst.addProperty( p, res );
  p = ontModel.getObjectProperty( NAMESPACE + "hasVariable" );
  res = ontModel.createResource( NAMESPACE + variavel );
  inst.addProperty( p, res );
}
```

```

p = ontModel.getObjectProperty(NAMESPACE + "hasProbValue");
String nomep = "P" + proximaProb();
geraProbValue(ontModel, nomep, prob.floatValue());
res = ontModel.createResource(NAMESPACE + nomep);
inst.addProperty(p, res);
}

```

O primeiro passo para criar um indivíduo de uma classe através da API Jena é obter uma referência a esta classe, a qual é armazenada em um objeto `OntClass`, o qual, neste exemplo, é obtido a partir do modelo de ontologia previamente inicializado, `ontModel`. Executando o método `ontModel.getOntClass(URI da classe)` obtêm-se um objeto representando esta classe. A criação da instância também é realizada a partir do objeto `ontModel`: `inst = ontModel.createIndividual(URI do novo indivíduo, objeto OntClass definindo a classe deste indivíduo)`.

Para atribuir valor a uma propriedade deste novo indivíduo criado, primeiro é obtido um objeto representado esta propriedade, por exemplo:

```
ObjectProperty p = ontModel.getObjectProperty(NAMESPACE + "hasState")
```

Em seguida é criado um recurso, o qual representa o valor a ser atribuído:

```
Resource res = ontModel.createResource(NAMESPACE + estado);
```

Finalmente, o recurso criado é atribuído à respectiva propriedade da instância recém criada:

```
inst.addProperty(p, res);
```

A rede bayesiana utilizada para testar a implementação é apresentada na Figura 3.5.

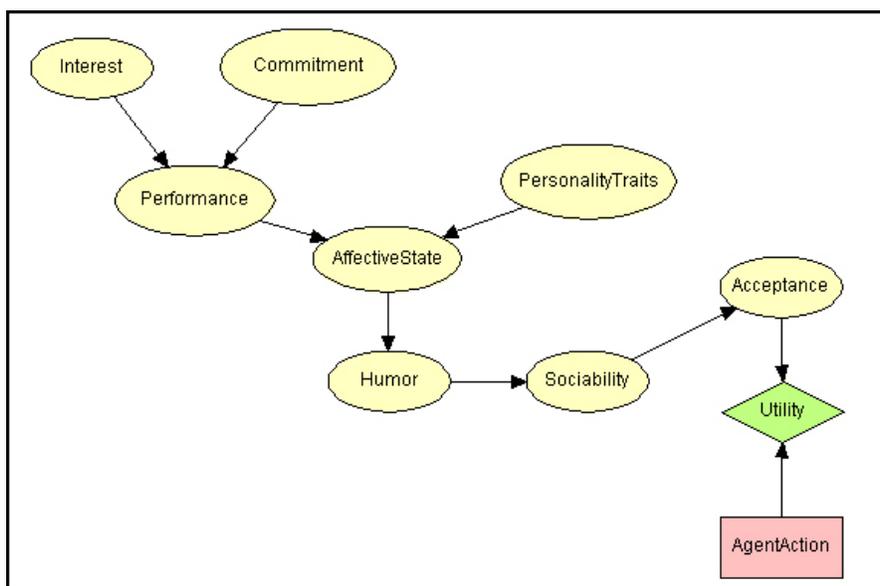


Figura 3.5 Rede bayesiana do agente social para representar o modelo do aluno

Para ilustrar o resultado deste método, a Figura 3.6 é uma *screenshot* do Protégé exibindo os indivíduos da classe BNNode. Também pode-se observar detalhes do indivíduo, como a Performance.

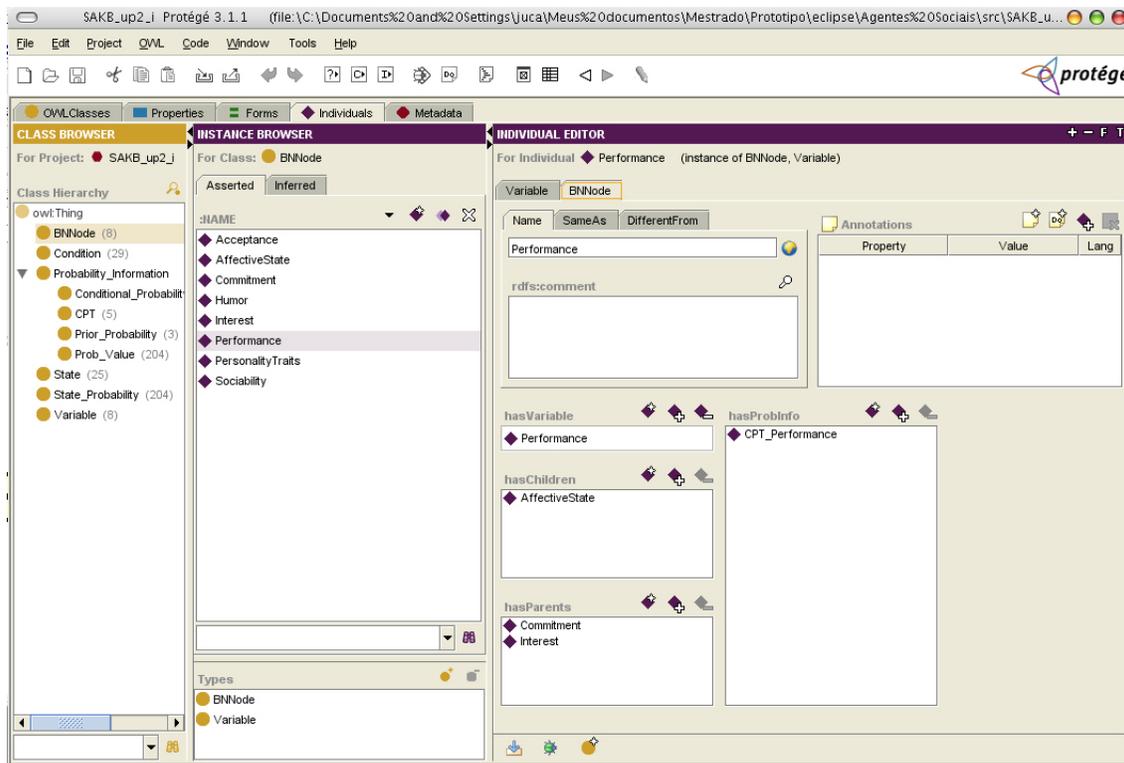


Figura 3.6 Instâncias geradas automaticamente abertas no Protégé.

3.2.4 Processamento de Consultas

Para que a comunicação entre agentes heterogêneos seja efetiva é fundamental que a linguagem de consultada suportada também seja um padrão ou no mínimo uma recomendação, assim como o próprio OWL. Atualmente, não existe nenhuma linguagem de consulta recomendada para ser utilizada com o OWL, linguagem adotada neste trabalho.

A OWL-QL (FIKES, 2005) define uma linguagem formal e estabelece protocolos para interação entre agentes, possibilitando diálogos na forma pergunta e resposta. Esta linguagem foi desenvolvida com o intuito de ser uma candidata para linguagem de consulta padrão para OWL.

Na implementação deste trabalho optou-se por fornecer suporte a linguagem de consulta SPARQL, uma vez que apresenta mais suporte de software do que a OWL-QL e já é um padrão W3C para consultas em RDF. A própria API do Jena fornece uma implementação da SPARQL. Um ponto fraco da utilização de SPARQL é o fato de ser uma linguagem de consulta construída para utilização no nível do RDF, sendo assim, uma consulta SPARQL não considera restrições expressas em OWL, por exemplo.

Na classe BN2OWL foi implementado um método que recebe uma consulta em SPARQL, executá-la em uma ontologia também parametrizada e retorna o resultado como um *string*. Para ilustrar a utilização deste método, a seguir é apresentada uma consulta em SPARQL e seu respectivo resultado. A consulta exemplificada visa obter o valor de uma propriedade (*hasStateProbability*) de um recurso específico (*CP_AffectiveState_18*). Mais especificamente, essa consulta visa obter todas as probabilidades dos estados relacionadas a uma probabilidade condicional da variável *Affective State*. A consulta foi realizada na ontologia gerada a partir da rede bayesiana do agente social.

Consulta:

```
SELECT ?x WHERE
(
  <http://gia.inf.ufrgs.br/agenteSocial/CP_AffectiveState_18>,
  <http://gia.inf.ufrgs.br/agenteSocial/hasStateProbability>, ?x
)
```

Resultado:

```
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_alegria_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_frustração_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_tristeza_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_gratidão_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_vergonha_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_raiva_18)
(x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_satisfação_18)
```

3.3 Módulo de Comunicação Externa

O módulo de comunicação externa fornece as funcionalidades necessárias para que o agente social possa comunicar-se com agentes externos. No contexto deste trabalho, agentes externos são os agentes do PortEdu e dos ambientes de aprendizagem ligados ao portal. O PortEdu tem como base um sistema multi-agente compatível com os padrões da FIPA. A utilização de padrões visa promover interoperabilidade entre os diferentes ambientes de aprendizagem apoiados pelo portal.

Não existe um padrão para a linguagem de conteúdo utilizada nas mensagens dos agentes. O desenvolvedor é livre para adotar a linguagem que melhor suprir suas necessidades. Neste caso, para aumentar a interoperabilidade foi optada pela linguagem OWL.

O módulo de comunicação é capaz de enviar e receber mensagens utilizando FIPA ACL. Ele é responsável por receber consultas dos agentes, interpretá-las e encaminhá-las para o módulo de gestão de ontologias. Ao receber a resposta do módulo de gestão de ontologias, um envelope para a mensagem é montado, de acordo com a especificação da FIPA ACL e a mensagem é respondida ao remetente, de acordo com o protocolo de comunicação utilizado.

A implementação da comunicação externa utilizou o FIPA-OS (FIPA-OS, 2005), um *middleware* desenvolvido para facilitar a implementação de aplicações multi-agente compatíveis com as especificações da FIPA. O FIPA-OS fornece implementações de um serviço de nomes (páginas brancas), serviço de diretório (páginas amarelas), transporte de mensagens, serviços de *parsers* e uma biblioteca de protocolos de interação FIPA.

Todos os componentes obrigatórios de uma plataforma FIPA são disponibilizados pelo FIPA-OS. Para o desenvolvimento de agentes, são fornecidas APIs para acessar os serviços da plataforma, gerenciar o ciclo de vida do agente, acessar serviços de comunicação, etc. Existem outras alternativas além do FIPA-OS, como o JADE (BELLIFEMINE, 2001). Entretanto, a atual implementação do PortEdu utiliza o FIPA-OS. Sendo assim, para facilitar a integração do protótipo no portal foi escolhido o FIPA-OS.

A implementação do protótipo do módulo de comunicação externa foi feita em uma classe Java, a qual utiliza as APIs fornecidas pelo FIPA-OS. Nesta classe, foram implementados métodos para registrar o agente social na plataforma do PortEdu, para realizar consultas no serviço de diretórios do portal e para receber e responder a consultas de outros agentes através do protocolo FIPA-Query. A estrutura que suporta o protocolo para consultas pode ser estendida e dar suporte a quaisquer outros protocolos FIPA, tais como o FIPA-Request e FIPA-ContractNet.

A Figura 3.7, extraída da especificação do protocolo FIPA-Query, mostra o fluxo de mensagens trocadas neste protocolo.

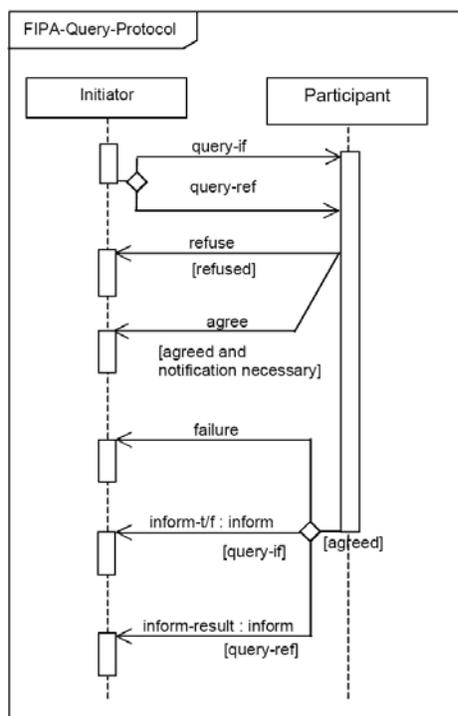


Figura 3.7 Protocolo de interação FIPA-Query (FIPA, 2003).

O iniciante solicita ao participante a realização de uma ação informativa através de dois atos de fala, `query-if` ou `query-ref`. O `query-if` é utilizado quando deseja-se saber se uma proposição é verdadeira ou falsa. Já o `query-ref` é utilizado para consultas mais amplas, cujos resultados podem ser objetos estruturados. Ao receber um `query-if` ou `query-ref`, o participante decide se irá ou não processar a solicitação de consulta.

Caso recuse, ele responde ao iniciante com uma mensagem com o ato de fala `refuse`. Caso aceite, é enviado um `agree`. A mensagem `agree` é opcional, por exemplo, se a ação solicitada for de rápida execução não é necessário enviar o `agree`. Se durante o processamento da consulta houver algum erro e a consulta não seja realizada, é respondida uma mensagem `failure` ao iniciante.

Após uma execução bem sucedida da consulta, o participante pode responder com dois tipos de `inform`: `inform-t/f` em resposta a uma consulta `query-if`, ou `inform-result` em resposta a um `query-ref`.

Na atual implementação do protótipo, ao receber uma consulta, caso o ato de fala seja um `query-if`, a solicitação é recusada. Caso seja um `query-ref` e a linguagem utilizada seja SPARQL a consulta é aceita e então é transmitida para o módulo de gerenciamento de ontologias para que seja processada. Em caso de erro durante a execução, é respondido um `failure` e caso a consulta seja realizada corretamente, é respondido um `inform-result` com o resultado da consulta.

A seguir é apresentado um exemplo de uma consulta recebida pelo agente social e a respectiva mensagem de resposta.

Mensagem solicitando a consulta:

Parâmetros	
Performative: <code>query-ref</code>	Ontology
Sender: <code>tester_agent@PortEdu</code>	Protocol: <code>FIPA-QUERY</code>
Receiver: <code>agente_social@PortEdu</code>	Conversation-id
Reply-to: <code>tester_agent@PortEdu</code>	Reply-with
Content: <code>SELECT ?x WHERE (<http://gia.inf.ufrgs.br/agenteSocial/CP_AffectiveState_18>, <http://gia.inf.ufrgs.br/agenteSocial/hasStateProbability>, ?x)</code>	In-reply-to
Language: <code>SPARQL</code>	Reply-by
Encoding	

Mensagem respondida:

Parâmetros	
Performative: inform_ref	Ontology
Sender: agente_social@PortEdu	Protocol: FIPA- QUERY
Receiver: tester_agent@PortEdu	Conversation-id
Reply-to: agente_social@PortEdu	Reply-with
Content: (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_alegria_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_frustração_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_tristeza_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_gratidão_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_vergonha_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_raiva_18) (x, http://gia.inf.ufrgs.br/agenteSocial/STP_AffectiveState_satisfação_18)	In-reply-to
Language: SPARQL	Reply-by
Encoding	

3.4 Considerações Finais

Neste Capítulo foi apresentada uma abordagem para aumentar a interoperabilidade do agente social. Esta abordagem constitui-se de uma extensão na arquitetura do agente social composta pela adição de dois módulos: o módulo para o gerenciamento de ontologias e o módulo para comunicação externa.

O módulo para gerenciamento de ontologias é responsável por criar uma representação em OWL a partir da rede bayesiana do agente social. Esta representação em OWL constitui uma base de conhecimento, uma vez que é formada por uma ontologia para definição de RB em OWL e as respectivas instâncias criadas a partir da rede.

A criação dessa base de conhecimento é realizada de uma forma automatizada e não é restrita às redes do agente social, qualquer rede RB especificada no formato NET pode ser utilizada. Além disso, o módulo para gerenciamento de ontologias também fornece o suporte para o processamento de consultas feitas na linguagem SPARQL.

O módulo para comunicação externa possibilita ao agente social interagir com os agentes do Portal. Este módulo é capaz de receber e enviar mensagens através de FIPA-ACL, a linguagem para comunicação entre agentes utilizada na PortEdu.

4 CONCLUSÃO E TRABALHOS FUTUROS

A questão de pesquisa utilizada para nortear este trabalho foi a seguinte:

“Identificar e implementar uma técnica factível e funcional para interoperabilidade entre agentes heterogêneos”.

Através das hipóteses pré-estabelecidas e de pesquisas realizadas a partir delas, a questão de pesquisa foi respondida uma vez que foi definida uma abordagem factível e eficiente para a interoperabilidade entre agentes heterogêneos e esta foi implementada como um protótipo no agente social.

Foram apresentadas as principais linguagens ontológicas para especificação de conhecimento na *web*. As linguagens foram analisadas sob o ponto de vista da interoperabilidade e padronização, sendo que a OWL foi adotada para utilização neste trabalho.

Foi proposta uma abordagem para aumentar a interoperabilidade do agente social através da utilização de OWL e FIPA-ACL para a interação com agentes fora de seu domínio. Nesta abordagem foi desenvolvido um mecanismo para representar redes bayesianas através de OWL, um aspecto fundamental já que o conhecimento do agente social é modelado através dessas redes. A abordagem pode ser resumida a uma extensão à arquitetura do agente social constituída por dois módulos: um para a gestão de ontologias e outro para a comunicação externa.

A abordagem proposta foi projetada através de uma extensão ao agente social que fornecesse uma representação possível de ser compartilhada e entendida por agentes alheios ao contexto original do agente social. Além disso, essa extensão trata também os aspectos mais operacionais como a forma utilizada para a troca de mensagens com os agentes externos.

Tais agentes fazem parte de um portal educacional, o PortEdu, cuja arquitetura é fundamentada em uma plataforma para agentes FIPA. Esta plataforma permite que todos os agentes dos sistemas educacionais comuniquem-se através de FIPA ACL. A utilização de uma linguagem comum para a troca de mensagens já fornece um nível significativo de interoperabilidade, mas o fato das linguagens de conteúdo não seguirem nenhum padrão é um ponto fraco.

Foi proposto o uso de OWL como a linguagem de conteúdo utilizada pelo agente social. OWL é a última linguagem ontológica proposta pelo W3C para utilização na *web* semântica. O conhecimento do agente social é expresso através de redes bayesianas, sendo assim foi necessário desenvolver mecanismos para converter a representação bayesiana em uma representação OWL. Essa conversão foi um ponto importante deste trabalho uma vez que a OWL não oferece suporte à representação de incerteza.

A utilização da OWL neste trabalho demonstra uma aplicação de tecnologia da *web* semântica para ampliar a interoperabilidade em um portal educacional multi-agente. Apesar de ser aplicada em um domínio heterogêneo fechado, a abordagem proposta para facilitar a interação entre agentes heterogêneos pode ser aplicada em outros contextos.

A OWL não foi desenvolvida considerando a possibilidade de modelar a incerteza relativa aos conceitos. Tal limitação reduz a aplicabilidade da *web* semântica e é um aspecto importante a ser considerado pelos comitês de padronização da linguagem.

Para lidar com essa limitação da OWL foi desenvolvida uma ontologia capaz de representar redes bayesianas. Além disso, foi desenvolvido um mecanismo para tradução automática de redes bayesianas para uma representação em OWL, utilizando a ontologia previamente definida. Com esse mecanismo, o conhecimento probabilístico do agente social pode ser representado através de uma base de conhecimentos constituída de uma ontologia e os respectivos indivíduos da mesma.

A maneira com que a ontologia foi desenvolvida possibilita sua extensão por parte de especialistas no contexto do agente social. Assim é possível acrescentar informações relativas ao domínio do agente, o AMPLIA, na ontologia.

A abordagem proposta para representar informações probabilísticas em OWL seguiu uma linha mais pragmática. Dessa forma, nota-se a necessidade de mais desenvolvimento para fornecer uma fundamentação formal sólida para o modelo. Uma possibilidade é a utilização de PR-OWL, entretanto a falta de ferramentas para desenvolvimento de redes bayesianas multi-entidade (fundamentação lógica da PR-OWL) constitui um obstáculo para a aplicação dessa tecnologia. Além disso, a PR-OWL propõe modificações na própria linguagem OWL, o que necessita aprovação da W3C para sua utilização em larga escala.

A utilização da OWL pelo agente social constitui um primeiro passo em direção a um portal educacional semântico. A adoção da OWL como um padrão no portal possibilita, por exemplo, o desenvolvimento de assistentes pessoais para os estudantes capazes de, automaticamente, obter informações de fontes heterogêneas (agentes) e fornecer ao estudante recursos educacionais de acordo com suas necessidades.

Como trabalhos futuros podemos apontar os seguintes temas de pesquisa e desenvolvimento:

- O conhecimento do agente social é utilizado para sua tomada de decisões quanto à criação de grupos. Desta forma, além de redes bayesianas tradicionais são utilizados diagramas de influência, os quais não foram contemplados na implementação atual do protótipo.

- Com a estrutura da ontologia atual, uma possibilidade de solução é a criação de uma especialização do conceito BNNode, o qual representa um nodo de uma rede. O mecanismo de tradução automática também necessitaria uma extensão, mas isso já foi previsto na estrutura da implementação, bastando ao desenvolvedor implementar o respectivo método para a geração do código OWL.
- Com o desenvolvimento de ferramentas de software para apoiar a construção de redes MEBN e também para a definição de ontologias com a PR-OWL é muito importante realizar uma implementação utilizando essa linguagem.
- Desta forma, é possível realizar uma comparação significativa entre a PR-OWL e ontologia proposta neste trabalho. Supõe-se que através da PR-OWL seja fornecida diretamente a fundamentação semântica dos conceitos probabilísticos, uma vez a que lógica formal dessa linguagem já incorpora tais conceitos.
- Este trabalho possibilitou vislumbrar o PortEdu com o apoio de tecnologias da *web* semântica. Uma continuação a curto prazo deste trabalho pode ser a utilização de OWL como linguagem de conteúdo entre todos os agentes do Portal. Um agente que beneficiaria-se diretamente desta utilização seria o agente *information retrieval*, o qual poderia acessar e compartilhar suas informações com os demais agentes de uma forma mais adequada à comunicação entre agentes heterogêneos.
- Outra possibilidade é utilizar OWL para especificar o contexto e significado dos conteúdos educacionais armazenados no Portal. Novamente, isso aumentaria a funcionalidade do agente *information retrieval* e possibilitaria a utilização automática do significado desses conteúdos por parte dos agentes presentes no Portal, independente do ambiente de aprendizagem aos quais fazem parte.

REFERÊNCIAS

- AUML. **The AUML Web Site**. Disponível em: <<http://www.auml.org/>> Acesso em: dez. 2003.
- BAADER, F. et al. *The Description Logic Handbook*. Cambridge: Cambridge University Press, 2002.
- BECHHOFER, S. et al. **An Informal Description of Standard OIL and Instante OIL**. 2000. Disponível em: <<http://www.ontoknowledge.org/oil/downl/oil-whitepaper.pdf>> Acesso em: set. 2005.
- BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. Developing multiagent systems with a FIPA-compliant agent framework. **Software - Practice & Experience**, London, p. 103-128, 2001.
- BERNERS LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific American**, [S.l.], v. 284, n.5, p. 35-43, 2001.
- BOFF, E. **Usando agentes sociais para suportar a colaboração em ambientes inteligentes de aprendizagem**. 2006. Proposta de Tese (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- BRICKLEY, D.; GUHA, R. V. **RDF Vocabulary Description Language 1.0**: RDF Schema. 2004. Disponível em: <<http://www.w3.org/TR/rdf-schema/>> Acesso em: dez. 2005.
- BROEKSTRA, J. et al. Enabling knowledge representation on the *web* by extending RDF schema. In: INTERNATIONAL CONFERENCE ON WORLD WIDE WEB, 10., Hong Kong. **Proceedings...** New York: ACM Press, 2001. p. 467-478.
- CALISTI M. **FIPA Standards for promoting interoperability of industrial agent systems**. FIPA Presentation, Agentcities, Information Days, 06 de Fevereiro de 2003, Barcelona, Espanha.
- CARROLL, J. et al. Jena: Implementing the Semantic Web Recommendations. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 13., 2004, New York. **Proceedings...** New York: ACM Press, 2004. p. 74-83.
- CLOUSER, C. **Metadata Overview**. Short paper. 2004. Disponível em: <http://www.pages.drexel.edu/~cbc25/info653/DAML_OIL_review.html> Acesso em: 2005.
- CONNOLY, D. et al. **Annotated DAML+OIL Ontology Markup**. 2001. Disponível em: <<http://www.w3.org/TR/daml+oil-walkthru>> Acesso em: nov. 2005.

CONNOLLY, D. et al. **DAML+OIL Referente Description**. 2001. Disponível em:
<<http://www.w3.org/TR/daml+oil-reference>> Acesso em: nov. 2005.

COSTA, P.; LASKEY, K. B.; LASKEY, K. J. PR-OWL: a Bayesian Ontology Language for the Semantic Web. In: INTERNATIONAL SEMANTIC WEB CONFERENCE, 2005. **Proceedings...** [S.l.:s.n.], 2005.

DEAN, M. et al. **OWL Web Ontology Language Reference** Disponível em:
<<http://www.w3.org/TR/owl-ref/>> Acesso em: nov. 2005.

DING, L. et al. **Using Ontologies in the Semantic Web: A Survey**. [S.l.]: University of Maryland, Baltimore County, 2005. (TR-CS-05-07).

DUŽÍ, M. Semantic Web Ontology and Natural Language from the Logical Point of View. In: INTERNATIONAL SYMPOSIUM IN INFORMATICS FOR GEOINFORMATICS, 2006. **Proceedings...** Ostrava: VŠB-TU Ostrava, 2006.

FARQUHAR, A.; FIKES, R.; RICE J. The Ontolingua Server: A Tool for Collaborative Ontology Construction. **Int. Journal of Human-Computer Studies**, [S.l.], n.46, p.707-727, 1997.

FENSEL, D. et al. OIL in a nutshell. In: EUROPEAN KNOWLEDGE ACQUISITION CONFERENCE, EKAW, 2000. **Knowledge Acquisition, Modeling and Management: proceedings**. Berlin: Springer-Verlag, 2000. p. 1-16. (Lecture Notes on Computer Science, v. 1937).

FENSEL, D. **Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce**, 2nd ed. Berlin: Springer-Verlag, 2003.

FENSEL, D. Relating Ontology Languages and *Web* Standards. **Informatik und Wirtschaftsinformatik**. Modellierung: Foelbach Verlag, 2000.

FIKES, R.; FARQUHAR, A. Large-scale repositories of highly expressive reusable knowledge. **IEEE Intelligent Systems**, Los Alamitos, v. 14, n. 2, 1999.

FIKES, R.; HAYES, P.; HORROCKS, I. OWL-QL - A Language for Deductive Query Answering on the Semantic *Web*. **Journal of Web Semantics**, [S.l.], v. 2, n. 1, 2005.

FININ, T. et al. KQML as an Agent Communication Language. In: INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, CIKM, 3., 1994. **Proceedings...** [S.l.]: ACM Press, 1994.

FININ, T.; LABROU, Y.; MAYFIELD J. KQML as an Agent Communication Language. In: **Software Agents**. Cambridge: AAAI Pres/MIT Press, 1997.

FININ, T.; FRITZSON, R.; MCKAY, D. A Language and Protocol to Support Intelligent Agent Interoperability. In: CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT, 3., 1992. **Proceedings...** Gaithersburg: ACM Press, 1992.

FIPA. **FIPA ABSTRACT ARCHITECTURE SC0001**. Disponível em:
<http://www.fipa.org/specs/fipa00001/>> Acesso em: jul. 2005.

- FIPA. **FIPA ACL Message Structure Specification SC00061**. Disponível em: <http://www.fipa.org/specs/fipa00061/>> Acesso em: jul. 2005.
- FIPA. **FIPA Agent Management Specification SC00023**. Disponível em: <http://www.fipa.org/specs/fipa00023/>> Acesso em: jul. 2005.
- FIPA. **FIPA Agent Message Transport Service Specification SC00067**. Disponível em: <http://www.fipa.org/specs/fipa00067/>> Acesso em: jul. 2005.
- FIPA. **FIPA Communicative Act Library Specification SC00037**. Disponível em: <http://www.fipa.org/specs/fipa00037/>> Acesso em: jul. 2005.
- FIPA. **FIPA Contract Net Interaction Protocol Specification SC00029**. Disponível em: <http://www.fipa.org/specs/fipa00029/>> Acesso em: jul. 2005.
- FIPA. **FIPA Query Interaction Protocol Specification SC00027**. Disponível em: <http://www.fipa.org/specs/fipa00027/>> Acesso em: jul. 2005.
- FIPA. **FIPA Request Interaction Protocol Specification SC00026**. Disponível em: <http://www.fipa.org/specs/fipa00026/>> Acesso em: jul. 2005.
- FIPA. **FIPA Request When Interaction Protocol Specification SC00028**. Disponível em: <http://www.fipa.org/specs/fipa00028/>> Acesso em: jul. 2005.
- FIPA. **FIPA SL Content Language Specification SC00008**. Disponível em: <http://www.fipa.org/specs/fipa00008/>> Acesso em: jul. 2005.
- FIPA. **Foundation for Intelligent Physical Agents**. Disponível em: <<http://www.fipa.org>> Acesso em: jul. 2005.
- FLORES, C. D. **Negociação pedagógica aplicada a um ambiente multiagente de aprendizagem colaborativa**. 2005. 121f. *Tese (Doutorado em Ciência da Computação)* - Instituto de Informática, UFRGS, Porto alegre
- GENNARI, J. et al. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. **International Journal of Human-Computer Studies**, [S.l.], v. 58, n. 1, p. 89-123, 2002.
- GREEN, S. et al. **Software Agents: a Review**. [S.l.]: Trinity College, University of Dublin, 1997 (TCD-CS-1997-06).
- GRUBER, T. R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, London, Academic Press, v. 5, n. 2, p. 199-220, 1993.
- GRUBER, T. R. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. **Int. Journal of Human-Computer Studies**, [S.l.], v. 43, p. 907-928, 1995.
- GRUNINGER, M.; LEE, J. Ontology applications and design: Introduction. **Communications of the ACM**, New York, v. 45, n. 2, p. 39-41, 2002.
- HEFLIN, J. **OWL Web Ontology Language Use Cases and Requirements**. 2004. Disponível em: <<http://www.w3.org/TR/webont-req/>> Acesso em: dez. 2005.

HEFLIN, J.; HENDLER, J. Dynamic Ontologies on the *Web*. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, AAAI, 17.; CONFERENCE ON INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE, IAAI, 12., 2000, Austin. **Proceedings...** Menlo Park: AAAI Press, 2000.

HEFLIN, J.; HENDLER, J.; LUKE, S. **SHOE**: A Knowledge Representation Language for Internet Applications. [S.l.]: Dept. of Computer Science, University of Maryland, 1999. (UMIACS TR-99-71).

HORROCKS, I. et al. **The ontology inference layer oil**. Technical report, Free University of Amsterdam, 2000. Disponível em: <<http://www.ontoknowledge.org/oil/>>. Acesso em: set. 2005.

HORROCKS, I.; PATEL-SCHNEIDER, P. F.; VAN HARMELEN, F. From SHIQ and RDF to OWL: The making of a *web* ontology language. **Journal of Web Semantics**, [S.l.], v. 1, n. 1, p. 7-26, 2003.

JAQUES, P. A.; VICCARI, R. M. A BDI approach to infer student's emotions. In: IBERO-AMERICAN CONFERENCE ON AI, IBERAMIA, 9., 2004, Puebla, México. **Advances in Artificial Intelligence**: proceedings. Berlin: Springer, 2004. p. 901-911. (Lecture Notes in Artificial Intelligence, 3315).

JENNINGS, N.; SYCARA, K; WOOLDRIDGE, M. A Roadmap of Agent Research and Development. **Autonomous Agents and Multi-Agent Systems**, Boston, v.1, n.1, p.7-38, July 1998.

JENSEN, F. et al. HUGIN - The Tool for Bayesian Networks and Influence Diagrams. In: EUROPEAN WORKSHOP ON PROBABILISTIC GRAPHICAL MODELS, 1., 2002. **Proceedings...** [S.l.: s.n.], 2002. p. 212-221.

KALYANPUR, A.; PARSIA, B.; HENDLER, J. A tool for working with *web* ontologies. **International Journal on Semantic Web and Information Systems**, [S.l.], v.1, n.1, Jan./Mar. 2005.

KLYNE, G.; CARROL, J. **Resource Description Framework (RDF)**: Concepts and Abstract Syntax. 2004. Disponível em: <<http://www.w3.org/TR/rdf-concepts/>> Acesso em: nov. 2005.

KNUBLAUCH, H. An AI Tool for the Real World: Knowledge Modeling with Protégé **JavaWorld**, [S.l.], June 20, 2003. Disponível em: < <http://www.javaworld.com/javaworld/jw-06-2003/jw-0620-protege.html> >. Acesso em: nov. 2005.

KNUBLAUCH, H. et al. Editing Description Logic Ontologies with the Protégé OWL Plugin. In: INTERNATIONAL WORKSHOP ON DESCRIPTION LOGICS, DL 2004. **Proceedings...** [S.l.:s.n.], 2004.

KNUBLAUCH, H. et al. The Protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In: INTERNATIONAL SEMANTIC WEB CONFERENCE, ISWC 3., 2004. **Proceedings...** Berlin: Springer-Verlag, 2004. p. 229-243. (Lecture Notes in Computer Science, v. 3298).

- LASKEY, K. B.; COSTA, P. Of Klingons and Starships: Bayesian Logic for the 23rd Century. In: CONFERENCE UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, UAI, 21., 2005. **Proceedings...** [S.l.:s.n.], 2005. p. 346-354.
- LEE R. C. T. Fuzzy Logic and the Resolution Principle. **Journal of the Association for Computing Machinery**, New York, v.19, n.1, p.109-119, Jan. 1972.
- LENAT, D. B.; GUHA, R. V. Building large knowledge-based systems. In: **Representation and inference in the Cyc project**. [S.l.]: Addison-Wesley, 1990.
- LIND J. **Iterative Software Engineering for Multiagent Systems: The MASSIVE Method**. Berlin: Springer-Verlag, 2001.
- MANOLA, F.; MILLER, E. **RDF Primer – W3C Recommendation**. Disponível em: <<http://www.w3.org/TR/rdf-primer/>> Acesso em: nov. 2005.
- MATURANA, H.; VARELA, F. **Árvore do conhecimento**. Campinas: Ed. Psy, 1995.
- MCGUINNESS, D. L. et al. DAML+OIL: An Ontology Language for the Semantic Web. **IEEE Intelligent Systems**, Los Alamitos, p. 72-80, Sept./Oct. 2002.
- MCGUINNESS, D. L.; HARMELEN, F. **Web Ontology Language Overview**. Disponível em: <<http://www.w3.org/TR/owl-features/>> Acesso em: nov. 2005.
- MILOJICIC, D. et al. MASIF: The OMG Mobile Agent System Interoperability Facility. **Personal Technologies**, [S.l.], v.2, n.3, p.117-129, Dec. 1999.
- NAKAYAMA, L.; VICARI, R. M.; COELHO, H. An Information Retrieving Service for Distance Learning. **The IPSI BgD Transactions On Internet Research**, Belgrado, v.1, n.1, p.49-56, Jan. 2005.
- NECHES, R. et al. An Overview of the Darpa Knowledge Sharing Effort. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE REPRESENTATION AND REASONING, KR, 3., 1992. **Proceedings...** San Francisco: Morgan Kaufman, 1992.
- NECHES, R. et al. Enabling technology for knowledge sharing, **AI Magazine**, Menlo Park, v.12, n.3, p.36-56, 1991.
- NOY, N.; MCGUINNESS, D. L. **Ontology Development 101: A Guide to Creating Your First Ontology**. [S.l.]: Knowledge Systems Laboratory, 2001. (Technical Report KSL-01-05).
- NWANA, H. S. Software Agents: an Overview. **Knowledge Engineering Review**, [S.l.], v. 11, n. 3, p. 205-244, Oct./Nov. 1996.
- O'BRIEN, P. D.; NICOL, R. C. FIPA - towards a standard for software agents. **BT Technology Journal**, [S.l.], v. 16, n.3, July 1998.
- ONTOLINGUA. Disponível em: <<http://www.ksl.stanford.edu/software/ontolingua/>> Acesso em: set. 2005.
- PAN, R. et al. A Bayesian Network Approach to Ontology Mapping. In: INTERNATIONAL SEMANTIC WEB CONFERENCE, ISWC, 4., 2005. **Proceedings...** [S.l.]: Springer-Verlag, 2005. p. 563-577. (Lecture Notes in Computer Science v. 3729).

POSLAD, S.; BUCKLE, P.; HADINGHAM, R. The FIPA-OS agent platform: Open Source for Open Standards. In: INTERNATIONAL CONFERENCE AND EXHIBITION ON THE PRACTICAL APPLICATION OF INTELLIGENT AGENTS AND MULTI-AGENTS. Manchester: The Practical Application Company, 2000. p. 355-368.

POSLAD, S.; BUCKLE, P.; HADINGHAM, R. Open Source Standards and Scalable Agencies. In: INTERNATIONAL WORKSHOP ON INFRASTRUCTURE FOR MULTI-AGENT SYSTEMS, 2000. **Agents Workshop on Infrastructure for Multi-Agent Systems**. Berlin: Springer-Verlag, 2000. p. 296-303. (Lecture Notes in Computer Science, v. 1887).

PROTÉGÉ. **The Protégé Ontology Editor and Knowledge Acquisition System**. Disponível em: <<http://protege.stanford.edu/>> Acesso em: nov. 2005.

PRUD'HOMMEAUX, E.; SEABORNE, A. **SPARQL Query Language for RDF**. 2006. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>> Acesso em: fev. 2006.

REYNOLDS, C. W. Flocks, herds, and schools: A distributed behavioral model. **Computer Graphics**, New York, v.21, n.4, p.25-34, July 1987. Trabalho apresentado na Annual conference on Computer Graphics and Interactive Techniques, 14., 1987.

RUSSELL S. J.; NORVIG P. **Artificial intelligence: a modern approach**. New Jersey: Prentice Hall International, 1995.

SMITH, M. K.; WELTY, C.; MCGUINNESS, D. L. **OWL – Web Ontology Language Guide**. Disponível em: <<http://www.w3.org/TR/owl-guide/>> Acesso em: nov. 2005.

SOWA, J. F. Ontology, Metadata and Semiotics. In: GANTER, B.; MINEAU, G.W. (Ed.). **Conceptual Structures: Logical, Linguistic, and Computational Issues**. Berlin: Springer-Verlag, 2000. p.55-81.

STAAB, S. et al. An extensible approach for modeling ontologies in RDF(S). In: **Knowledge media in healthcare: opportunities and challenges**. Hershey: Idea Group Publishing, 2002, p.234-253.

SYCARA K. P. Multiagent Systems. **AI Magazine**, [S.l.], v.19, n.2, 1998.

USCHOLD, M.; GRUNINGER, M. Ontologies: principles, methods, and applications. **Knowledge Engineering Review**, [S.l.], v.11, n.2, p.93-155, 1996.

VICARI, R. M. et al. A Multi-Agent Intelligent Environment for Medical Knowledge. **Journal of Artificial Intelligence in Medicine**, Amsterdam, v.27, p.335-366, 2003.

W3C. **Semantic Web Activity Statement**. Disponível em: <<http://www.w3.org/2001/sw/Activity>> Acesso em: nov. 2005.

W3C. **World Wide Web Consortium**. Disponível em: <<http://www.w3c.org>> Acesso em: nov. 2005.

WOOLDRIDGE, M. Agent-Based Computing. **Interoperable Communication Networks**, [S.l.], v.1, n.1, p.71-97, 1997.

WOOLDRIDGE, M. Intelligent Agents: The Key Concepts. **Multi-Agent-Systems and**

Applications. Berlin: Springer, 2002. p. 3-43. (Lecture Notes in Computer Science, v. 2322).

YEPES, I.; BARONE, D. Inteligência Artificial distribuída: Uma Abordagem ao Comportamento Social Inteligente. In: BARONE, D. et al. **Sociedades Artificiais: A Nova Fronteira da Inteligência nas Máquinas.** Porto Alegre: Bookman, 2003.

ZHIHONG, Z.; MINGTIAN, Z. Web Ontology Language OWL and its description logic foundation. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED COMPUTING, APPLICATIONS AND TECHNOLOGIES, PDCAT, 2003. **Proceedings...** [S.l.]: IEEE Computer Society, 2003. p. 157-160.

APÊNDICE A ONTOLOGIA DEFINIDA PARA A ESPECIFICAÇÃO DE REDES BAYESIANAS EM OWL

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://gia.inf.ufrgs.br/agenteSocial/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xml:base="http://gia.inf.ufrgs.br/agenteSocial/">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Variable">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasState" />
        </owl:onProperty>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >1</owl:minCardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing" />
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
          >1</owl:minCardinality>
        <owl:onProperty>
          <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasClass" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Conditional_Probability">

```

```

<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty>
      <owl:ObjectProperty
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasConditions" />
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Probability_Information" />
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:minCardinality>
    <owl:onProperty>
      <owl:ObjectProperty
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasStateProbability" />
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/State_Probability">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasVariable" />
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasProbValue" />
        </owl:onProperty>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:cardinality>

```

```

    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>
    <owl:onProperty>
      <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasState"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Prob_Value">
  <rdfs:subClassOf
rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Probability_Information"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasProbValue"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Condition">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasVariable"/>
      </owl:onProperty>
    </owl:Restriction>
  </rdfs:subClassOf>
<rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
      >1</owl:cardinality>

```

```

    <owl:onProperty>
      <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasState"/>
    </owl:onProperty>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Prior_Probability">
  <rdfs:subClassOf
rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Probability_Information"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasStateProbability"/>
          </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/BNNode">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:minCardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasProbInfo"/>
          </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
>1</owl:cardinality>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasVariable"/>
          </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl:Class>

```

```

<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/CPT">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasCPs"/>
      </owl:onProperty>
      <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:minCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
  <rdfs:subClassOf
rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Probability_Information"/>
</owl:Class>
<owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/State"/>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasParents">
  <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
  <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasVariable">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Condition"/>
        <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/State_Probability"/>
        <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
  <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Variable"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasCPT">
  <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/CPT"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasProbInfo">
  <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
  <rdfs:range>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Prior_Probability"/>
        <owl:Class
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Conditional_Probability"/>

```

```

        <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/CPT"/>
    </owl:unionOf>
</owl:Class>
</rdfs:range>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasConditions">
    <rdfs:domain
rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Conditional_Probability"/>
    <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Condition"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasState">
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Variable"/>
                <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Condition"/>
                <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/State_Probability"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/State"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasStateProbability">
    <rdfs:domain>
        <owl:Class>
            <owl:unionOf rdf:parseType="Collection">
                <owl:Class
rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Conditional_Probability"/>
                <owl:Class rdf:about="http://gia.inf.ufrgs.br/agenteSocial/Prior_Probability"/>
            </owl:unionOf>
        </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/State_Probability"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasCPs">
    <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Conditional_Probability"/>
    <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/CPT"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasProbValue">
    <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Prob_Value"/>
    <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/State_Probability"/>

```

```
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasChildren">
  <rdfs:range rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
  <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/BNNode"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasClass">
  <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Variable"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:about="http://gia.inf.ufrgs.br/agenteSocial/hasFloatValue">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
  <rdfs:domain rdf:resource="http://gia.inf.ufrgs.br/agenteSocial/Prob_Value"/>
</owl:DatatypeProperty>
<owl:AllDifferent/>
<owl:AllDifferent/>
<rdf:List/>
<owl:DataRange/>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 2.1, Build 284) http://protege.stanford.edu →
```

ANEXO A ARTIGO SUBMETIDO PARA O CONGRESSO ITS 2006

Semantic Web Technologies Applied to Interoperability on an Educational Portal

Elder Rizzon Santos¹, Elisa Boff^{1,2} and Rosa Maria Vicari¹

¹ Computer Science Institute – Federal University of Rio Grande do Sul (UFRGS)
POBox 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
{ersantos, rosa}@inf.ufrgs.br

² Computer Science Department, Caxias do Sul University (UCS), CEP 95.001-970 – Caxias do Sul – RS – Brazil
eboff@ucs.br

Abstract. This paper describes an approach to promote interoperability among heterogeneous agents that are part of an Educational Portal (PortEdu). We focus on a specific agent, the social agent, adding all the necessary functionality for him to interact with agents that aren't fully aware of its context. The social agent belongs to a Multi-agent Learning Environment designed to support training of diagnostic reasoning and modeling of domains with complex and uncertain knowledge, AMPLIA. The knowledge of social agent is implemented with Bayesian networks, which allows the agent to represent its probabilistic knowledge and make its decisions. However, to communicate with agents outside AMPLIA, it is necessary to express such probabilistic knowledge in a way that all agents may process. Such requirement is addressed using OWL, an ontology language developed by W3C to be used on the Semantic *Web*.

1 Introduction

The semantic *web* [7] represents the next step on Internet technology. Nowadays, the content of *web* pages is understandable by humans only. The purpose of the semantic *web* is to aggregate meaning to the pages, in a way that computer software may interpret its content. There are several technologies involved in this process, but the main one is the utilization of ontologies.

Ontologies are expected to be used to provide means to explicate concepts and the relationships among them, allowing agents to interpret their meaning flexibly and unambiguously [6]. W3C (World Wide *Web* Consortium) is developing a stack of recommendations related to semantic *web* in the effort of making the semantic *web* a reality.

Currently, the base of the stack is XML, which provides syntax for the *web* documents; following is XML Schema, a language for restricting the structure of a XML document; next is RDF, which offers a simple graph model consisting of nodes (resources) and relations (statements) between them, provides a small amount of built-in semantics for the data model; RDF Schema augments RDF providing the means to describe properties and classes of RDF resources; last is OWL (*Web* Ontology Language), a semantic markup language which makes possible the description of classes, properties and their instances, besides that, it allows the definition of relation among classes (joint, disjoint), cardinality (none, exactly one), equality, characteristics of properties, among others. [8] [9]

In this work, we apply OWL as the main mechanism to provide semantic interoperability between agents on different contexts. The domain in which these agents exists is an educational portal, Portedu [10], a *web* portal that provides access to educational contents and systems. It provides a platform for agent based educational systems and several services, among them an information retrieval agent, that search for information on local resources and the *web* considering the context of the search. Also present as a core service of Portedu is the user profile agent [10], responsible for maintaining user-related information. One of the educational systems present in portedu is AMPLIA.

AMPLIA is an Intelligent Multi-agent Learning Environment designed to support training of diagnostic reasoning and modeling of domains with complex and uncertain knowledge [15]. As part of Portedu, AMPLIA may use the functionalities and services available in the portal and, most important, allow its agents to interact directly

with agents from different intelligent educational systems. Besides, the agents may use the portal's service discovery, agent management, communication facilities, etc.

AMPLIA focuses on the medical area. It is a system that deals with uncertainty under the Bayesian network approach, where learner-modelling tasks will consist of creating a Bayesian network for a problem the system will present. The construction of a network involves qualitative and quantitative aspects. The qualitative part concerns the network topology, that is, causal relations among the domain variables. After it is ready, the quantitative part is specified. It is composed of the distribution of conditional probability of the variables represented.

A negotiation process (managed by an intelligent *Mediator Agent*) will treat the differences of topology and probability distribution between model built by the *Learner Agent* and the one built-in in the system. That negotiation process occurs between the agents that represent the expert knowledge domain (*Domain Agent*) and the agent that represents the learner knowledge (*Learner Agent*). The AMPLIA architecture is shown in Figure 1.

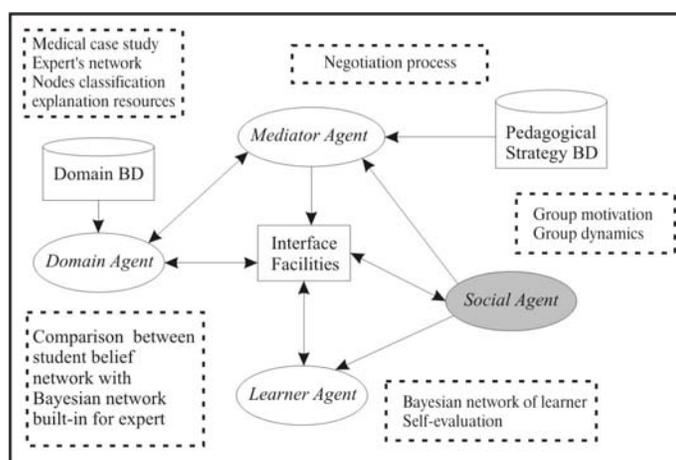


Fig. 1. The AMPLIA Architecture

The *Social Agent* interacts with *Learner Agent* to suggest which classmate is recommended to work together with him. It also interacts with *Mediator Agent* that knows the domain and supports the negotiation process. Such communication is done through FIPA ACL [12] and a FIPA-OS [13] platform, which implements the FIPA standards, supports the whole system.

In this paper we describe how we applied OWL as the ontology language used by the social agent in order to communicate with agents outside its domain. The rest of this paper is organized as follows: section 2 contextualizes where this research is applied, the social agent; section 3 presents the related research with this work; in section 4 it is presented our approach to enable a richer communication between social agent and other agents in the portal; section 5 presents our conclusions and future work.

2. Social Agent

The main goal of the social agent is improve student's learning stimulating his interaction with other students, tutors, professors, etc. At AMPLIA, Each user builds his own Bayesian network for a specific pathology. During this task, the social agent will recommend students to help other students. The social agent creates workgroups to solve tasks cooperatively. Besides, the agent uses the strategy of initiate a network construction to motivate students to interact with the environment.

The social agent reasoning is based on *individual level* and *group level*.

Individual Level

The individual level has the student features. The information collected that is important to define the right student to recommend are: Social Profile; Acceptance Degree; Sociability Degree; Mood State; Interest; Commitment Degree; Leadership and Performance.

The *Social Profile* (SP) is built during the students' interaction through a synchronous mechanism (e.g. chat tool). The following information is collected during the students' interaction: number of times that a student had the initiative to talk with another; number of times that a student answered a communication request; individuals with whom the student interacts or has interacted, and number of interactions and individuals with whom the student interacts regularly, and number of interactions.

Based on Maturana [16] we defined the *Acceptance Degree* (AD), which measures the acceptance between students. Such data is collected through a graphical interface that enables each student to indicate his/her acceptance degree for other students. This measurement may also be considered from a point of view of Social Networks. As the AD is indicated by the students themselves based on their affective structures, the measurement can indicate different emotions [22], such as love, envy, hatred, etc. The average of all ADs received by a student influences his/her *Sociability Degree* (SD).

The *Mood State* (MS) represents our belief in the capability of a student to play the role of a tutor if he/she is not in a positive mood state (although the student may have all the technical and social requirements to be a tutor). We consider three values for the MS: "bad mood", "regular mood" and "good mood". These states are indicated by the students in a graphical interface through corresponding clip-arts.

After a helping session, a small questionnaire is submitted to the student who got assistance, with the purpose of collecting information about the performance of the tutor. The questions made are based on concepts from Social Networks and Sociometry, and may be answered by four qualitative values: "excellent", "good", "regular", and "bad". The questions are presented below:

How do you classify the sociability of your class fellow?

How do you classify the help given by your class fellow?

The answer to the first question together with the average of the ADs of a student, form his/her *Sociability Degree* (SD). This grade indicates how other individuals see the social capability of this student.

Based on [21], the agent's preferences over world states S are expressed by a utility function $U(S)$, which assigns a single number to express the desirability of a state. Furthermore, for each action a available to the agent, and for each possible outcome state S' of that action, $P(S'|E, a)$ represents the agent's belief that action a will result in state S' , when the action is performed in a state identified by evidence E . The expected utility (1) of an action a is computed as follows

$$U(A) = \sum_{S'} P(S'|E, a)U(S') \quad (1)$$

The socio-affective agent selects the action that maximizes this value when deciding how to act. The influence between nodes is shown in Figure 2.

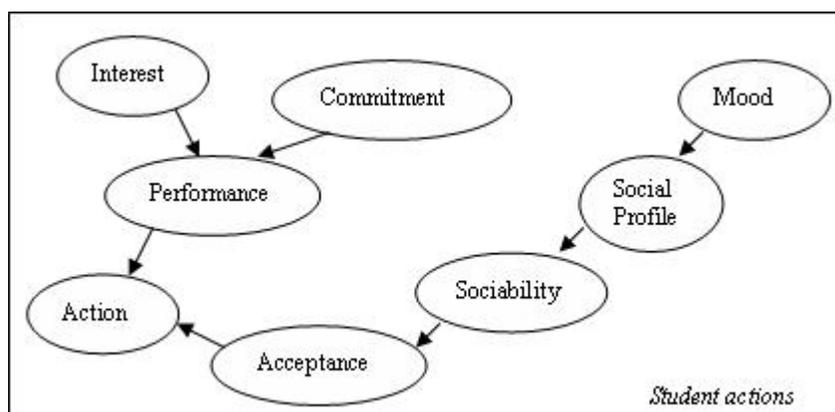


Fig. 2. Decision network of student model

The *Interest* feature is given by the initiatives taken for the user (which material it had access without being recommended, with which student he/she initiated an interaction).

The *Commitment Degree* is based on the consistency theory (theory of persuasion of Social Psychology) places that after the people to assume a public commitment, they probably will act in more consistent way with its commitment. The idea is to use this theory as strategy to motivate the students to collaborate with others. Thus, students who collaborate with other more actively (a quantitative measure of interactions can be made here, is not the ideal, but most viable at the moment) they will be inserted in groups with more frequency and will receive aid from better quality.

Leadership occurs when a person if becomes capable to modify the beliefs, attitudes and behaviors of other individuals, organizing them and guiding its action for objectives that start to desire to reach [17]. Leadership also is related to the personality of the person, the psychological attributes of the followers and to the situations where if it uncurls the leadership process. We can identify a leader in the environment of learning for the relation between leaders, followers, tasks and situations. When a student acts actively in the learning environment, establishes many interactions and contributions in the publisher of networks the agent records these information and verifies which student collaborated more actively in the network construction and which student had its work a lesser number of times modified. This fact can indicate the presence of a leader, therefore they are people who lead the tasks and the integrant ones of the group accept its position. How much bigger the number of accepted contributions, greater the leader pointer.

Group Level

The group level takes into account the *cohesion* and the *confidence* (or trust) in a workgroup as presented in Figure 3. The figure shows the agent knowledge, that is supported by ontologies. The *confidence* node influences the *cohesion* node, like group cohesion is based on individual confidence in a group.

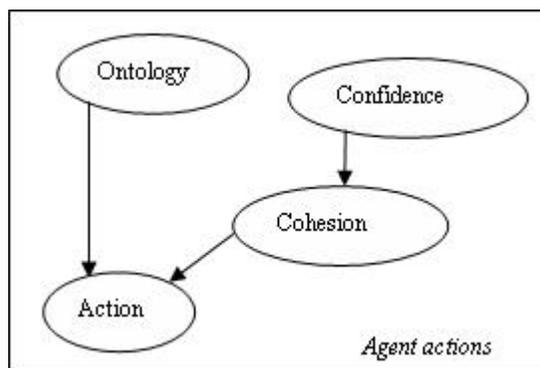


Fig. 3. Decision network of socio-affective agent

The group formation agent can decide who is joining in a group that brings more benefit to the group. This reliable type is interesting to help in the judgment of the agent regarding its confidence in the group that another agent belongs. The confidence is defined as the belief of an agent in the attributes such as the trustworthiness, honesty and ability of the agent "trusted" question. The reputation of an agent defines an expectation on its behavior that is based on the comments of the agent or information on the last behavior of the agent in a specific context in data moment. He assumes that he has two agents, the agent and the agent B. When the agent A does not have no direct interaction with agent B or it is not certain on the trustworthiness of B, the agent A can take decisions based on the reputation of the agent B (gotten with other agents, or either, asking to other agents on the reputation). Once agent A interacted with agent B, it can verify or establish its confidence in the agent B in accordance with its degree of satisfaction in the interaction and use this confidence to take decisions for future interactions [18] [19]. Mechanisms on confidence and reputation can be used by agents to differentiate the good ones of bad (adequate or inadequate) the collaborating ones.

Joined groups have a great productivity. The cohesion is the attractiveness that the group exerts for its members, that of it they desire to continue to participate, resisting the idea to abandon it [17]. The cohesion is analyzed using the sociometric test of Jacob Moreno [20]. We can define as group cohesion the solidarity and establishment of loyalty in a group and measure it for the amount of times that the same people choose to interact.

Portedu

In PortEdu, the social agent may interact with other agents, such as the User Profile Agent, the Information Retrieval Agent [10], among others. In this scope, the utility of the agents is enhanced, since they can gather information from more sources and also provide its information and services to all the agents in the portal.

The education portal is a multiagent system that complies with the FIPA standards and is designed to support agent based learning environments. The main feature of the portal is its personalized information retrieval, which considers student's profiles, student's models and the learning environments' ontologies. [10]

In figure 4 it is presented an architectural view of Portedu focusing on the provided infra-structure. The agent platform is a FIPA compliant platform, and all mandatory services are available, so that the agent implementations doesn't need to consider communication details, agent life cycle, core functionalities such as service discovery, agent management and so on.

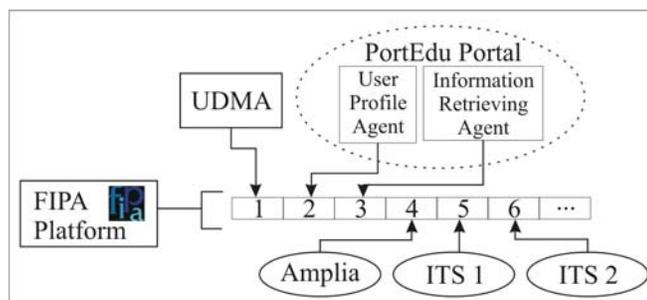


Fig. 4. Portedu and its base platform

One of the most useful feature of multi-agent systems is the interaction among agents, which may help agents solve problems that are beyond its original abilities. Such cooperation is very difficult to achieve and is one of the goals of the multi-agent systems researchers. With FIPA, the problem of interaction between heterogeneous agents is solved, but not on the semantic level, the content of the messages. The main role of ontologies at the semantic *web* is to provide semantic interoperability. Applying OWL as the content language used by social agent we aim to increase the interoperability, allowing the agent to interact with a content language that was designed to provide semantic interoperability.

3. Related Research

Since internally the social agent uses a bayesian network to represent its knowledge and make its decisions, it is necessary to provide another representation of his knowledge that is understandable by all others agents. To obtain such interoperability, we use ontology as a bridge between the social agent internal representation and a representation that other agents may interpret.

Before using an ontology as a bridge for the interpretation gap, it is necessary to represent social agent's decision networks (Bayesian networks) in OWL. A probabilistic extension to OWL is proposed in [1]. To support representation of uncertainty and the ability to reason on scenarios where only partial information about a concept is available it is proposed to incorporate Bayesian networks on OWL ontologies.

The approach presented in [1] first augments OWL so that it may allow additional markups that can add probabilities to concepts, individuals, properties and its relationships. The second step is to define a set of translation rules to convert the probabilistic annotated ontology into a Bayesian network. For more details please refer to [1], here we will only present the specific part that fits in our problem. The translation rules are not applicable to our scenario, since we already have a Bayesian network.

In our case, the most useful part is the extension to OWL, since it allows us to represent social agent's probabilistic knowledge. The encoding of probabilistic information into an existing ontology consists of defining three OWL classes, since a probability is treated as a kind of resource. The classes are:

“PriorProbObj” – used to represent a probability on the form $P(A)$. This class has two mandatory properties: “hasVariable”, used to define which class, property or individual has uncertain information; and “hasProbValue”, a datatype property that defines the probability value.

“CondProbObjT” – used to represent a probability on the form $P(A|B)$.

“CondProbObjF” – used to represent a probability on the form $P(A|\bar{B})$.

Both “CondProbObjT” and “CondProbObjF” have three mandatory properties: “hasVariable”, “hasProbValue” and “hasCondition”. The use of “hasVariable” and “hasProbValue” is the same as it is in the property “PriorProbObj”. To explain the use of the “hasCondition” property we use the following example: “If male is a

class, then $P(\text{male} \mid \text{animal})=0.5$, the conditional probability that an individual object belongs to class male given it is an animal is 0.5. In this case, “hasVariable” has the value male and “hasCondition” has the value animal.

Other approach to represent probabilistic knowledge through OWL is presented in [11], where it is defined PR-OWL, an extension of OWL to express probabilistic knowledge. This approach differs from the first one in the fact that it is based on Multi-Entity Bayesian Networks (MEBN). MEBN [11] combines Bayesian probabilities with First Order Logic. Accordingly to the authors, the use of MEBN logic to base PR-OWL allows the expression of a probability distribution over models of any finitely axiomatizable first-order theory. As a consequence, there are no guarantees that exact reasoning with a PR-OWL ontology will be efficient or even decidable.

In our context, we apply the solution proposed in [1], using only the OWL classes in order to translate from BN to OWL. In addition to representing the agent’s knowledge in a way that other agents may understand, the OWL ontology is also used to define concepts related to the social agent domain. Such definitions help contextualizing the information and services that the agent handle and supply to the agent platform.

The use of semantic *web* technology on interoperability is a topic that is receiving a lot of attention from the SW community. TAGA [23] uses OWL as a content language for the FIPA ACL messages exchanged by the agents, as a language to implement agent’s knowledge bases and to describe and reason about services. The architecture implemented on TAGA is similar to ours, since both are concerned with the communication among heterogeneous agents, providing the means for an open multi-agent system.

The way we use OWL is a little different, since we add the support to represent probabilistic knowledge, which we use to provide another representation of social agent’s knowledge. A future application of such functionality is using a Bayesian approach on ontology merging and integration [1] [11], that might be a feasible solution when conflict on concepts definition arises.

4. Applying Ontologies to Promote Interoperability

The goal of this work was to provide means to the social agent to communicate with agents outside its original scope, the AMPLIA environment. To achieve this goal, we propose the use of semantic *web* technology. In this section we will present how OWL was applied in order to move towards our goal.

In figure 5 it is shown the social agent’s architecture and its interaction with AMPLIA’s agents and other agents belonging to the PortEdu. The components that make possible the communication with external agents are the Ontology Management Module and the External Communication Module.

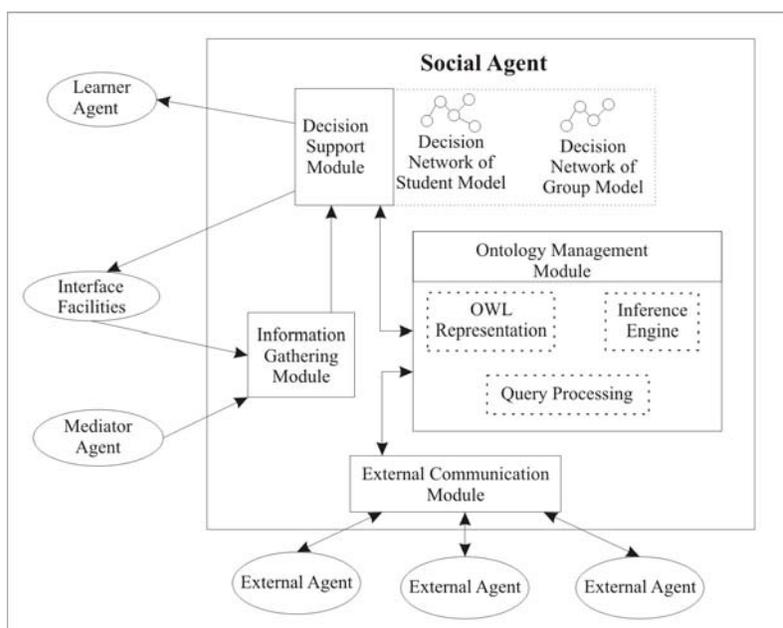


Fig. 5. The social agent architecture

Ontology Management Module

The OWL representation of the agent's knowledge is obtained using a probabilistic extension to OWL, as presented in [1]. This extension aims to solve some limitations of OWL, increasing the language's expressive power, support probabilistic queries and facilitate semantic integration. As seen on section 3, this approach augments OWL language and defines a set of translation rules to convert the ontology into a Bayesian network.

In our case, we don't need the translation between OWL ontology and Bayesian network. It is only necessary the means to represent the probabilistic knowledge in OWL. Thus, we only use the extension to OWL, which allows us to represent the decision networks that social agent uses to make his decisions. In the extension, a probability is treated as a resource, and three OWL classes are defined to express such resource. Such approach doesn't modify the language itself, making it fully understandable by other agents with OWL parsers. Another use of the owl representation is to provide information about the root domain of the social agent, AMPLIA.

The conversion of the Bayesian networks to the ontology is done using Jena [2], which is a semantic *web* toolkit for Java. Jena provides APIs for OWL, DAML+OIL and RDFS. Other features of Jena are the inference support for both RDF semantics and OWL semantics, as defined by W3C, and support for a query language RDQL [14], that can be used on the results of RDFS or OWL reasoning. Using the OWL API we are able not only to easily update the agent's ontology representation but also to provide direct support for other agents to make queries using RDQL.

Although OWL is built on top of RDF, a query language on the RDF level, RDQL, isn't capable of considering the restrictions expressed in OWL, for example. There is no standard query language for the semantic *web* yet. OWL-QL [3] is a formal language and protocol for agents, whose knowledge is expressed in OWL, to use on query-answer dialogs. This query language is intended to be a candidate standard query language for OWL and is designed to be suitable for a broad range of query-answering services and applications.

Since the goal of this work is to provide an interoperable way for the social agent to communicate, it is also fundamental that the supported query language to be a standard, as OWL. As soon as a W3C standard or

recommendation on query language is released we plan on updating our architecture. Since the agent implementation is modular, it won't be a problem adding support for another query language.

External Communication Module

The external communication module provides the means to communicate with external agents. In our case, external agents are agents from Portedu and from others learning environments that are attached to the portal. Portedu is developed as multi-agent system that complies with the FIPA standards, in order to promote interoperability dealing with the heterogeneous nature of the different learning environments that it supports. There is no standard on the content language used on the message. The developer is free to use the language that best suit his needs. In our case, to increase the interoperability, we use OWL.

Thus, the communication module is capable to send and receive messages using FIPA ACL. It is responsible for receiving the queries from agents, parsing them and forwarding the content to the ontology management module. Upon receiving a response from the ontology module, already in OWL, a message envelope is assembled, accordingly to the FIPA ACL specification and the message is returned to its sender, following the message protocol.

The implementation of the communication was made easy by using JADE [5], a middleware developed to facilitate the implementation of multi-agent applications that complies with the FIPA specifications. The architecture provides a names service, directory facilitator, message transport, parser services and a library of FIPA interaction protocols. All the mandatory components of a FIPA platform are available. For agent development, JADE supplies the necessary Java APIs.

4. Conclusions

In this paper we aimed to increase the interoperability of the social agent making possible his communication with agents outside his domain. Such agents are part of an educational portal, Portedu, whose architecture is grounded on a FIPA agent platform. Such platform allows all agents from the educational systems to communicate with each other through FIPA ACL. The use of common language to exchange the messages already provides a significant level of interoperability, but the fact that the content languages doesn't follows any standards can be a drawback.

Our work proposed the use of OWL as the content language used by the social agent. OWL is the latest ontology language proposed by W3C towards the semantic *web*. The several tools available to work with OWL made it easy to provide several ontology related services, such as query processing and inference engine. Since the social agent's knowledge is expressed using Bayesian networks, it was necessary to provide an alternate OWL representation, which enables our agent to communicate with a standard content language.

This demonstrates one application of the semantic *web* technology as to improve the interoperability on a multi-agent educational portal. Although being applied to a close heterogeneous domain, our approach to facilitate the interaction among heterogeneous agents may be applied on other contexts.

The utilization of OWL by the social agent is a first step towards a semantic educational portal, since it is feasible to consider adopting this ontology language as a standard on the portal, making possible, for example the development of students' personal assistants that automatically gather information from heterogeneous sources (agents) and provide the student with educational resources accordingly to its current needs. Once all the agents present in the portal are able to communicate with one another, apply techniques to merge and exchange concepts [1] [23] [24] from its ontologies, the functionality of one specific educational system will increase due to the additional knowledge available to its agents.

5. References

1. Ding, Zhongli; Peng, Yun.. A Probabilistic Extension to Ontology Language OWL. In: Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37), January 2004.
2. Carroll, Jeremy J. *et al.* Jena: Implementing the Semantic *Web* Recommendations. In: Proceedings of the 13th international World Wide *Web* Conference, p. 74-83. ACM Press, 2004.
3. Fikes, Richard; Hayes, Patrick; Horrocks, Ian. OWL-QL - A Language for Deductive Query Answering on the Semantic *Web*. In: Journal of *Web Semantics*, vol. 2, issue 1. Science Direct, 2005.
4. Grau, Bernardo Cuenca; Parsia, Bijan; Sirin, Evren. Working with multiple ontologies on the semantic *web*. In: Proceedings of the Third International Semantic *Web* Conference (ISWC2004). Volume 3298 Lecture Notes in Computer Science, 2004.
5. Bellifemine, Fabio; Poggi, Agostino; Rimassa, Giovanni. Developing multiagent systems with a FIPA-compliant agent framework. In: Software - Practice & Experience, John Wiley & Sons, Ltd. vol no. 31, 2001, pg. 103-128.
6. Horrocks, Ian; Patel-Schneider, Peter F.; Harmelen, Peter F. From SHIQ and RDF to OWL: The making of a *web* ontology language. Journal of *Web Semantics*, vol. 1, pg7--26, 2003.
7. Berners Lee, Tim; Hendler, James; Lassila, Ora. The Semantic *Web*. Scientific American 2001; 284(5), pg. 35-43.
8. Ding, Li; Kolari, P.; Ding, Zhongli; Avancha, Sasikanth. Using Ontologies in the Semantic *Web*: A Survey. In: Ontologies in the Context of Information Systems, Springer-Verlag. October, 2005.
9. Dean, Mike; Schreiber, Guus, eds. OWL *Web* Ontology Language Reference: <http://www.w3.org/TR/owl-ref/>
10. Nakayama, L.; Viccari, R. M.; Coelho, H. An Information Retrieving Service for Distance Learning. Transactions On Internet Research, v. 1, p. 49-56, 2005.
11. Costa, Paulo; Laskey, Kathryn B.; Laskey, Kenneth J. PR-OWL: A Bayesian Ontology Language for the Semantic *Web* In: Proceedings of the Workshop on Uncertainty Reasoning for the Semantic *Web*, International Semantic *Web* Conference, 2005. to appear.
12. FIPA. FIPA ACL Message Structure Specification SC00061. <http://www.fipa.org/specs/fipa00061/>
13. FIPA-OS Agent Toolkit. <http://fipa-os.sourceforge.net/index.htm>
14. Miller, L.; Seaborne, A.; Reggiori, A. Three Implementations of SquishQL, a Simple RDF Query Language. In: Proceedings of the First International Semantic *Web* Conference. Springer-Verlag, 2002, pg 243.
15. Vicari, R. M., Flores, C.D., Seixas, L., Silvestre, A., Ladeira, M., Coelho, H.: A Multi-Agent Intelligent Environment for Medical Knowledge. In: Journal of Artificial Intelligence in Medicine, Vol.27. Elsevier Science, Amsterdam, (2003) 335-366.
16. Maturana, H. and Varela, F. Tree of Knowledge: The Biological Roots of Human Understanding. Shambhala Publications, Boston, MA, 1998.
17. Aronson, E., Wilson, T. D., & Akert, R. M. (2005). Social Psychology (5th ed.). Englewood Cliffs, NJ: Prentice Hall.
18. Vassileva, J. Multi-agent architectures for distributed learning environments. In: AIED Proceedings, vol. 12, pg.1060-1069, 2001.
19. Castelfranchi, C. e Falcone, R. Principles of Trust for MAS: Cognitive Anatomy, Social Importance and Qualification. ICMAS'98.
20. Moreno, Jacob L. (1953). Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Psychodrama. Beacon, NY: Beacon House.
21. Conati, C. Probabilistic assessment of user's emotions in educational games. Journal of Applied Artificial Intelligence. 16(7-8):555-575. 2002.4
22. Jaques, Patricia A.; Viccari, R. M. A BDI Approach to Infer Student's Emotions in an Intelligent Learning Environment. In: IBERAMIA 2004, pgs 901-911.
23. Zou, Youyong; Finin, Tim; Ding Li; Chen, Harry; Pan Rong. Using Semantic *Web* Technology in Multi-Agent Systems: A Case Study in the TAGA Trading Agent Environment. In: Proceeding of the 5th International Conference on Electronic Commerce ICEC 2003, ACM Press, Oct 2003, pgs 95-101.
24. Wiesman, Floris; Roos, Nico. Domain Independent Learning of Ontology Mappings. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2, IEEE, 2004, pgs. 346-853.

ANEXO B ARTIGO SUBMETIDO PARA O CONGRESSO ICALT 2006

Social Agents to improve collaboration on an Educational Portal

Elisa Boff

*Computer Science Department, Caxias do Sul University (UCS), CEP 95.001-970 – Caxias do Sul – RS – Brazil. Phone/Fax: +55 54 2182159
eboff@ucs.br*

Elder Rizzon Santos, Rosa Maria Vicari

*Computer Science Institute – Federal University of Rio Grande do Sul (UFRGS)
POBox 15.064 – 91.501-970 – Porto Alegre – RS – Brazil. Phone/Fax: +55 51 33166161
{ersantos,rosa}@inf.ufrgs.br*

Abstract

Cooperative learning is changing the learning environments design. Intelligent Tutoring Systems, Multi-agent Systems, Affective Computing and Virtual Characters, are techniques and resources to improve the individual and personalized learning. In this research direction, this paper presents a Socio-Affective Agent whose reasoning is based on individual aspects of the student, as affective states, personality and acceptance, and group aspects, as cohesion. This socio-affective agent is being integrated in the AMPLIA environment, an Intelligent Probabilistic Multi-agent Environment to support the diagnostic reasoning development and the diagnostic hypotheses modeling of domains with complex and uncertain knowledge, like medical area. In order to provide semantic interoperability between the environment's agents we are using an OWL (Web Ontology Language).

1. Introduction

The use of Intelligent Tutoring Systems (ITS) and Intelligent Learning Environments (ILE) constitutes a very attractive alternative to distance learning programs based on the Internet. These ITS and ILE try to adapt to each student's features through the construction and the analysis of models that reflect both behavioral and cognitive aspects of them. These

systems represent more advanced pedagogical tools and provide more individualized learning experiences.

However, despite all the sophistication of these systems, situations may occur where the course material given by them is not sufficient to supply pedagogical student necessities that comes to acquire during learning process. In these cases, it is important to have tools that allow human-human interactions, where students can communicate themselves with tutors, or other students, and can jointly supply necessities and learning.

The importance of social interactions in the learning process is already known by the educational theoreticians. Some studies in this field are: the Socio-Cultural approach of Vygostky [1], some works of Piaget [2], theories of Collaborative Learning [3], Maturana [13] and others.

Recent advances in ITS and ILE fields have proposed the use of agent's society-based architectures [4]. The principles of multi-agent systems have showed a very adequate potential in the development of teaching systems, due to the fact that the nature of teaching-learning problems are more easily solved in a cooperative way.

The development of cooperative learning systems takes into account social factors, like presented in Vassileva's and Cao's work [8]. They conclude that is very important consider sociological aspects of cooperation, to discover and describe existing relationships among people, existing organizational structures and incentives for cooperative action. Hence,

the learning environment can detect and solve some conflicts, help to perform tasks and motivate learning and cooperation.

Based on presented ideas, our research group has been developing some intelligent learning environments to promote cooperative learning. The AMPLIA environment [6] is a multi-agent system to medical area that provides a collaborative Bayesian network editor to allow students build their own networks and compare them with the expert network. This construction is made in a cooperative way between medicine students to infer cardiologic pathologies. AMPLIA is an education system that is running at *web* portal that provides access to educational contents. It provides a platform for agent based educational systems and several services. AMPLIA may use the functionalities and services available in the portal and, most important, allow its agents to interact directly with different intelligent educational systems. Besides, the agents may use the portal's service discovery, agent management, communication facilities, the OWL (*Web Ontology Language*) is used as the main mechanism to provide semantic interoperability between these agents.

This paper presents some related work in section 2, the social agent modeling in section 3, the use of ontologies in section 4, and conclusions in section 5.

2. Related Work

The group dynamic has been addressed by many researchers and in different areas. The multi-agent approach is adjusted to the problem of group formation and coordination.

The Vassileva's research is about strategies and techniques of groups. In [7] they propose a motivation strategy for user participation based on persuasion theories of social psychology. In [8] the goal is finding out how people develop attitudes of liking or disliking other people when interacting in a CSCW environment in a collaborative-competitive situation, how they change their attitudes towards others when they realize their attitudes towards themselves and how the design of the environment influences the emergent social fabric of the group.

Prada [9] has developed a model that supports the dynamics of a group of synthetic agents, inspired by theories of group dynamics developed in human social psychological sciences, driven by a characterization of

the different types of interactions that may occur in the group.

A Bayesian network based appraisal model was used in the Conati work to deduce a student's emotional state based on their actions [10].

The social agent, described in section 3, is based on social psychology ideas (to support social aspects) and affective states.

Since internally the social agent uses a Bayesian network to represent its knowledge and make its decisions, it is necessary to provide another representation of his knowledge that is understandable by all other agents. To obtain such interoperability, we use ontology as a bridge between the social agent internal representation and a representation that other agents of the environment may interpret.

Before using an ontology as a bridge for the interpretation gap, it is necessary to represent social agent's decision networks (Bayesian networks) in OWL. A probabilistic extension to OWL is proposed in [12]. To support representation of uncertainty and the ability to reason on scenarios where only partial information about a concept is available it is proposed to incorporate Bayesian networks on OWL ontologies.

The approach presented in [12] first augments OWL so that it may allow additional markups that can add probabilities to concepts, individuals, properties and its relationships. The second step is to define a set of translation rules to convert the probabilistic annotated ontology into a Bayesian network.

Other approach to represent probabilistic knowledge through OWL is presented in [12], where it is defined PR-OWL, an extension of OWL to express probabilistic knowledge. This approach differs from the first one in the fact that it is based on Multi-Entity Bayesian Networks (MEBN). MEBN [12] combines Bayesian probabilities with First Order Logic. Accordingly to the authors, the use of MEBN logic to base PR-OWL allows the expression of a probability distribution over models of any finitely axiomatizable first-order theory. As a consequence, there are no guarantees that exact reasoning with a PR-OWL ontology will be efficient or even decidable.

In our context, we apply the solution proposed in [12], using only the OWL classes in order to translate from BN to OWL. In addition to representing the agent's knowledge in a way that other agents may understand, the OWL ontology is also used to define concepts related to the social agent domain. Such

definitions help contextualizing the information and services that the agent handle and supply to the agent platform.

3. Social intelligent agent

The main goal of the social agent is improve student's learning stimulating his interaction with other students, tutors and professors. At AMPLIA, each user builds his own Bayesian network for a specific pathology. The Bayesian network corresponds to the student model for a particular problem solution in the health context. During this task, the social agent will recommend students to help other students. The social agent creates workgroups to solve tasks cooperatively. Besides, the agent uses the strategy of initiate a network construction to motivate students to interact with the environment.

The social agent reasoning is based on individual level and group level.

3.1 Individual Level

The individual level represents the student features. The information collected that is important to define the right student to recommend are: Social Profile; Acceptance Degree; Sociability Degree; Mood State; Interest; Commitment Degree; Leadership and Performance.

The Social Profile (SP) is built during the students' interaction through a synchronous mechanism (e.g. chat tool and collaborative editor). The following information is collected during the students' interaction: number of times that a student had the initiative to talk with another; number of times that a student answered a communication request; individuals with whom the student interacts or has interacted, and number of interactions and individuals with whom the student interacts regularly, and number of interactions.

Based on Maturana's ideas [13] we defined the Acceptance Degree (AD), which measures the acceptance between students. Such data is collected through a graphical interface that enables each student to indicate his/her acceptance degree for other students. This measurement may also be considered from a point of view of Social Networks. As the AD is indicated by the students themselves based on their affective

structures, the measurement can indicate different emotions [14], such as love, envy, hatred, etc.

The Mood State (MS) represents our belief in the capability of a student to play the role of a tutor if he/she is not in a positive mood state (although the student may have all the technical and social requirements to be a tutor).

After a helping session, a small questionnaire is submitted to the student who got assistance, with the purpose of collecting information about the performance of the tutor. The questions made are based on concepts from Social Networks and Sociometry, and may be answered by four qualitative values: "excellent", "good", "regular", and "bad". The questions are presented below:

- How do you classify the sociability of your class fellow?
- How do you classify the help given by your class fellow?

The answer to the first question together with the average of the ADs of a student, form his/her Sociability Degree (SD). This grade indicates how other individuals see the social capability of this student.

Based on [14], the agent's preferences over world states S are expressed by a utility function $U(S)$, which assigns a single number to express the desirability of a state. Furthermore, for each action a available to the agent, and for each possible outcome state S' of that action, $P(S'|E, a)$ represents the agent's belief that action a will result in state S' , when the action is performed in a state identified by evidence E . The expected utility (1) of an action a is computed as follows

$$U(A) = \sum_{S'} P(S'|E, a)U(S') \quad (2)$$

The socio-affective agent selects the action that maximizes this value when deciding how to act. The influence between nodes is shown in Figure 1.

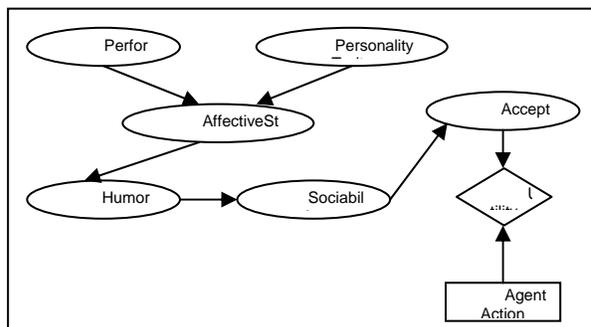


Figure 6. Agent reasoning probabilistic network.

In order to infer the personality traits we are using the model proposed in [9], based on OCC Model. The affective states can be considered as emotion manifestation in a specific time. Conati [10] modeled a Bayesian network to infer emotions. This research use both models to give us states values to *PersonalityTraits* and *AffectiveState* nodes (Figure 1).

The Interest feature is given by the initiatives taken for the user which material it had access without being recommended.

The Commitment Degree is based on the consistency theory (theory of persuasion of Social Psychology) places that after the people to assume a public commitment, they probably will act in more consistent way with its commitment. The idea is to use this theory as strategy to motivate the students to collaborate with others. Thus, students who collaborate with other more actively (a quantitative measure of interactions can be made here) they will be inserted in groups with more frequency and will receive aid from better quality.

Leadership occurs when a person if becomes capable to modify yours beliefs, attitudes and behaviors of other individuals, organizing them and guiding its action for objectives that start to desire to reach. Leadership also is related to the personality of the person, the psychological attributes of the followers and to the situations where if it uncurls the leadership process. When a student acts in the learning environment, establishes many interactions and contributions in the publisher of networks the agent records these information and verifies which student collaborated more actively in the network construction and which student had its work a lesser number of times modified. This fact can indicate the presence of a leader, therefore they are people who lead the tasks and the integrant ones of the group accept its position. How much bigger the number of accepted contributions, greater the leader pointer.

3.2 Group Level

The group level takes into account the cohesion and the confidence (or trust) in a workgroup. The agent group reasoning is a probabilistic network, where confidence node influences the cohesion node, like group cohesion is based on individual confidence in a group.

The group formation agent can decide who brings more benefit when joining. This reliable type is interesting to help in the judgment of the agent regarding its confidence in the group that another agent belongs. The confidence is defined as the belief of an agent in the attributes such as the trustworthiness, honesty and ability of the agent "trusted" question. The reputation of an agent defines an expectation on its behavior that is based on the comments of the agent or information on the last behavior of the agent in a specific context in data moment. He assumes that he has two agents, the agent and the agent B. When the agent A does not have no direct interaction with agent B or it is not certain on the trustworthiness of B, the agent A can take decisions based on the reputation of the agent B (gotten with other agents, or either, asking to other agents on the reputation). Once agent A interacted with agent B, it can verify or establish its confidence in the agent B in accordance with its degree of satisfaction in the interaction and use this confidence to take decisions for future interactions [8] [14]. Mechanisms on confidence and reputation can be used by agents to differentiate the good ones of bad (adequate or inadequate) the collaborating ones.

Joined groups have a great productivity. The cohesion is the attractiveness that the group exerts for its members, that of it they desire to continue to participate, resisting the idea to abandon it. The cohesion is analyzed using the sociometric test of Jacob Moreno [5]. We can define as group cohesion the solidarity and establishment of loyalty in a group and measure it for the amount of times that the same people choose to interact.

4. Applying Ontologies to Promote Interoperability

The use of semantic *web* technology on interoperability is a topic that is receiving a lot of attention from the Semantic *Web* community. TAGA [23] uses OWL as a content language for the FIPA ACL messages exchanged by the agents, as a language to implement agent's knowledge bases and to describe

and reason about services. The architecture implemented on TAGA is similar to ours, since both are concerned with the communication among heterogeneous agents, providing the means for an open multi-agent system.

The way we use OWL is a little different, since we add the support to represent probabilistic knowledge, which we use to provide another representation of social agent's knowledge. A future application of such functionality is using a Bayesian approach on ontology merging and integration [11] [12], that might be a feasible solution when conflict on concepts definition arises.

In order to the social agent to communicate with agents outside its original scope, the AMPLIA environment, we propose the use of semantic *web* technology. In this section we will present how OWL was applied in order to move towards our goal.

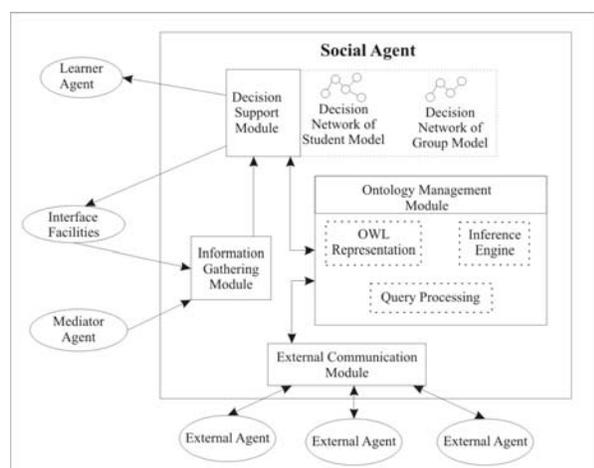


Figure 2. The social agent architecture

In Figure 2 it is shown the social agent's architecture and its interaction with AMPLIA's agents and other agents belonging to the educational portal. The components that make possible the communication with external agents are the Ontology Management Module and the External Communication Module.

The ontology management module is responsible for representing the agent's knowledge in OWL, providing all the conversion tools necessary. Along with the OWL representation, this module supplies query processing through RDQL and a

pluggable inference engine interface through the JENA API.

The External Communication Module provides the means to communicate with external agents. In our case, external agents are agents from educational portal. Thus, the communication module is capable to send and receive messages using FIPA ACL. It is responsible for receiving the queries from agents, parsing them and forwarding the content to the ontology management module. Upon receiving a response from the ontology module, already in OWL, a message envelope is assembled, accordingly to the FIPA ACL specification and the message is returned to its sender, following the message protocol.

5. Conclusions

In this paper we aimed to improve collaboration in AMPLIA through the Social Agent. In order to integrate this agent with AMPLIA system, we proposed the use of OWL as the content language used by the social agent. OWL is the latest ontology language proposed by W3C towards the semantic *web*. The several tools available to work with OWL made it easy to provide several ontology related services, such as query processing and inference engine. Since the social agent's knowledge is expressed using Bayesian networks, it was necessary to provide an alternate OWL representation, which enables our agent to communicate with a standard content language.

This demonstrates one application of the semantic *web* technology as to improve the interoperability on a multi-agent educational portal. Although being applied to a close heterogeneous domain, our approach to facilitate the interaction among heterogeneous agents may be applied on other contexts.

The design of social and affective agents is been highlighted as an effective way to motivate and to improve group formation among students and also to promote collaborative learning. We are using this agent to identify suitable students that can play the role of a tutor, and to recommend them to other students needing assistance. In this way, we aim to contribute to the design of learning environments centered in students' features and collaborative learning.

6. References

- [1] L.S. Vygotsky, *The collected works of L. S. Vygotsky*. New York: Plenum Press, c1987-c1999 v. 1-6.
- [2] J. Piaget, Explanation in sociology. In: J. Piaget (1995), *Sociological studies*, New York: Routledge.
- [3] P. Dillenbourg, M. Baker, A. Blaye, C. O'Malley, The evolution of research on collaborative learning. In: P. Reimann & H. Spada (Eds). *Learning in humans and machines. Towards an interdisciplinary learning science*, 189-211. London: Pergamon, 1995.
- [4] L.M. Giraffa, R.M. Viccari, J. Self, Multi-Agent based pedagogical games. *Proceedings of ITS*, 4, 1998.
- [5] J.L. Moreno, *Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Psychodrama*. Beacon, NY: Beacon House. 1953.
- [6] R.M. Vicari, C.D. Flores, L. Seixas, A. Silvestre, M. Ladeira, H. Coelho, A Multi-Agent Intelligent Environment for Medical Knowledge. In: *Journal of Artificial Intelligence in Medicine*, Vol.27. Elsevier Science, Amsterdam, (2003) 335-366.
- [7] R. Cheng, J. Vassileva, User Motivation and Persuasion Strategy for Peer-to-peer Communities. *Proceedings HICSS'2005* (Minitrack on Online Communities in the Digital Economy/Emerging Technologies), Hawaii, Jan 3-6, 2005.
- [8] Y. Cao, G. Sharifi, Y. Upadrashta, J. Vassileva, Interpersonal Relationships in Group Interaction in CSCW Environments, *Proceedings of the User Modelling UM03 Workshop on Assessing and Adapting to User Attitudes and Affect*, Johnstown, June 22, 2003.
- [9] R. Prada, A. Paiva, Believable Groups of Synthetic Characters. In: *AAMAS'05*, July 2529, 2005.
- [10] C. Conati, Probabilistic assessment of user's emotions in educational games. *Journal of Applied Artificial Intelligence*. 16(7-8):555-575. 2002.
- [11] Z. Ding, Y. Peng, A Probabilistic Extension to Ontology Language OWL. In: *Proceedings of the 37th Hawaii International Conference On System Sciences (HICSS-37)*, January 2004.
- [12] P. Costa, K.B. Laskey, K.J. Laskey, PR-OWL: A Bayesian Ontology Language for the Semantic Web In: *Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web*, International Semantic Web Conference, 2005. to appear.
- [13] H. Maturana, F. Varela, *Tree of Knowledge: The Biological Roots of Human Understanding*. Shambhala Publications, Boston, MA, 1998.
- [14] C. Castelfranchi, Affective Appraisal versus Cognitive Evaluation in Social Emotions and Interactions. Paiva, A. (Ed.): *Affective Interactions, Towards a New Generation of Computer Interfaces*. Lecture Notes in Computer Science, 1814, Springer, 2000. pp. 76-106.

**ANEXO C ARTIGO SUBMETIDO PARA O CONGRESSO
ECAI 2006**

An Ontology-based Approach to Increase Interoperability among Heterogeneous Agents

Elder Rizzon Santos¹, Elisa Boff^{1,2} and Rosa Vicari¹

Abstract. This paper describes the utilization of ontologies as a first step towards interoperability at the semantic level among agents of different domains. Our test bed is an Educational Portal (PortEdu) which provides the infrastructure and support for agent-based learning environments. We focus on a specific agent, the social agent, adding all the necessary functionality for him to interact with agents that aren't fully aware of its context. The social agent belongs to a Multi-agent Learning Environment designed to support training of diagnostic reasoning and modeling of domains with complex and uncertain knowledge, named AMPLIA. The knowledge of social agent is implemented with Bayesian networks, which allows the agent to represent its probabilistic knowledge and make its decisions. However, to communicate with agents outside AMPLIA, it is necessary to express such probabilistic knowledge in a way that all agents may process. Such requirement is addressed using OWL (*Web Ontology Language*), an ontology language developed by W3C (*World Wide Web Consortium*) to be used on the *Semantic Web*.

1 INTRODUCTION

The utilization of ontologies on Computing Science is receiving a lot of attention nowadays due to the *semantic web*. The *semantic web* [1] represents the next step on Internet technology. Nowadays, the content of *web* pages is understandable by humans only. The purpose of the *semantic web* is to aggregate meaning to the pages, in a way that computer software may interpret its content. There are several technologies involved in this process, but the main one is the utilization of ontologies.

Ontologies are expected to be used to provide means to explicate concepts and the relationships among them, allowing agents to interpret their meaning flexibly and unambiguously [2]. W3C (*World Wide Web Consortium*) is developing a stack of recommendations related to *semantic web* in the effort of making the *semantic web* a reality.

One of this recommendations is OWL (*Web Ontology Language*), a semantic markup language which makes possible the description of classes, properties and their instances, besides that, it allows the definition of relation among classes (joint, disjoint), cardinality (none, exactly one), equality, characteristics of properties, among others [3] [4].

In this work, we apply OWL as the main mechanism to provide semantic interoperability between agents on different contexts. The domain in which these agents exist is an educational portal, Portedu [5], a *web* portal that provides access to educational contents and systems. It provides a platform for agent based educational systems and several services, among them an information retrieval agent, that search for information on local resources and the *web* considering the context of the search. Also present as a core service of Portedu is the user profile agent, responsible for maintaining user-related information. One of the educational systems present in portedu is AMPLIA.

AMPLIA environment [6] is a multi-agent system to medical area that provides a collaborative Bayesian network editor to allow students build their own networks and compare them with the expert network. This construction is made in a cooperative way between medicine students to infer cardiologic pathologies. As part of Portedu, AMPLIA may use the functionalities and services available in the portal and, most important, allow its agents to interact directly with agents from different intelligent educational systems. Besides, the agents may use the portal's service discovery, agent management, communication facilities, etc.

In this paper we describe how we applied OWL as the ontology language used by an AMPLIA's agent, the social agent, in order to communicate with agents outside its domain. The rest of this paper is organized as follows: section 2 presents the related research with this work; section 3 contextualizes where this research is applied, the social agent; in section 4 it is presented our approach to enable a richer communication between social agent and other agents in the portal; section 5 presents our conclusions and future work.

2 RELATED RESEARCH

The social agent, described in section 3, is based on social psychology ideas (to support social aspects) and affective states. Internally the social agent uses a bayesian network to represent its knowledge and make its decisions thus, it is necessary to provide another representation of his knowledge that is understandable by all others agents. To obtain such interoperability, we use ontology as a bridge between the social agent internal representation and a representation that other agents of the environment may interpret.

Before using an ontology as a bridge for the interpretation gap, it is necessary to represent social agent's decision networks (Bayesian networks) in OWL. A probabilistic extension to OWL, called BayesOWL, is proposed in [7]. To support representation of uncertainty and the ability to reason on scenarios where only partial information about a concept is available it is proposed to incorporate Bayesian networks on OWL ontologies.

The approach presented in [7] first augments OWL so that it may allow additional markups that can add probabilities to concepts, individuals, properties and its relationships. The second step is to define a set of translation rules to convert the probabilistic annotated ontology into a Bayesian network.

BayesOWL was developed to be a methodology for automatic ontology mapping. In this context, the ontologies are translated into BNs and the concept mapping between the two ontologies are treated as evidential reasoning between the two translated BNs. Probability tables are automatically created during the translation in order to measure the similarity between concepts.

The focus on ontology mapping limits the BayesOWL markups since it was not necessary to represent variables with states different than true or false. The reason for this is that the probabilistic knowledge associated with each ontology concept was used only for telling if two concepts from different ontologies were the same.

Other approach to represent probabilistic knowledge through OWL is presented in [8], where it is defined PR-OWL, an extension of OWL to express probabilistic knowledge. This approach differs from the first one in the fact that it is based on Multi-Entity Bayesian Networks (MEBN). MEBN [9] combines Bayesian probabilities with First Order Logic. Accordingly to the authors, the use of MEBN logic to base PR-OWL allows the expression of a probability distribution over models of any finitely axiomatizable first-order theory.

The use of MEBN logics as the fundamental semantics for OWL provides both expressiveness and flexibility in order to represent probabilistic knowledge. It allows the representation of the structural (graph) information of the model. As OWL-Full, the actual version of PR-OWL focuses on achieving the most expressiveness, as a consequence, in some cases there are no guarantees that a query will be traceable or decidable.

Current implementation of PR-OWL provides an upper-class ontology in order to allow the development of probabilistic ontologies using RDF syntax, which is compatible with OWL. Although a Protegé plugin is under development, the lack of tools for MEBN logic and the need for standardization represents a drawback for short-term solutions but also points to a very interesting medium to long term solution, as it fits well (providing the formal foundation of a first-order logic) the W3C model of standards.

3 SOCIAL INTELLIGENT AGENT

The main goal of the social agent is improve student's learning stimulating his interaction with other students, tutors and professors. At AMPLIA, each user builds his own Bayesian network for a specific pathology. The Bayesian network corresponds to the student model for a particular problem solution in the health context. During this task, the social agent will recommend students to help other students. The social agent creates workgroups to solve tasks cooperatively. Besides, the agent uses the strategy of initiate a network construction to motivate students to interact with the environment.

The social agent reasoning is based on individual level and group level.

3.1 Individual Level

The individual level represents the student features. The information collected that is important to define the right student to recommend are: Social Profile; Acceptance Degree; Sociability Degree; Mood State; Interest; Commitment Degree; Leadership and Performance.

The Social Profile (SP) is built during the students' interaction through a synchronous mechanism (e.g. chat tool and collaborative editor). The following information is collected during the students' interaction: number of times that a student had the initiative to talk with another; number of times that a student answered a communication request; individuals with whom the student interacts or has interacted, and number of interactions and individuals with whom the student interacts regularly, and number of interactions.

Based on Maturana's ideas [10] we defined the Acceptance Degree (AD), which measures the acceptance between students. Such data is collected through a graphical interface that enables each student to indicate his/her acceptance degree for other students. This measurement may also be considered from a point of view of Social Networks. As the AD is indicated by the students themselves based on their affective structures, the measurement can indicate different emotions [11], such as love, envy, hatred, etc.

The Mood State (MS) represents our belief in the capability of a student to play the role of a tutor if he/she is not in a positive mood state (although the student may have all the technical and social requirements to be a tutor).

After a helping session, a small questionnaire is submitted to the student who got assistance, with the purpose of collecting information about the performance of the tutor. The questions made are based on concepts from Social

Networks and Sociometry, and may be answered by four qualitative values: "excellent", "good", "regular", and "bad". The questions are presented below:

- How do you classify the sociability of your class fellow?
- How do you classify the help given by your class fellow?

The answer to the first question together with the average of the ADs of a student, form his/her Sociability Degree (SD). This grade indicates how other individuals see the social capability of this student.

Based on [11], the agent's preferences over world states S are expressed by a utility function $U(S)$, which assigns a single number to express the desirability of a state. Furthermore, for each action a available to the agent, and for each possible outcome state S' of that action, $P(S'|E, a)$ represents the agent's belief that action a will result in state S' , when the action is performed in a state identified by evidence E . The expected utility (1) of an action a is computed as follows

$$U(A) = \sum_{S'} P(S'|E, a)U(S') \quad (3)$$

The socio-affective agent selects the action that maximizes this value when deciding how to act. The influence between nodes is shown in Figure 1.

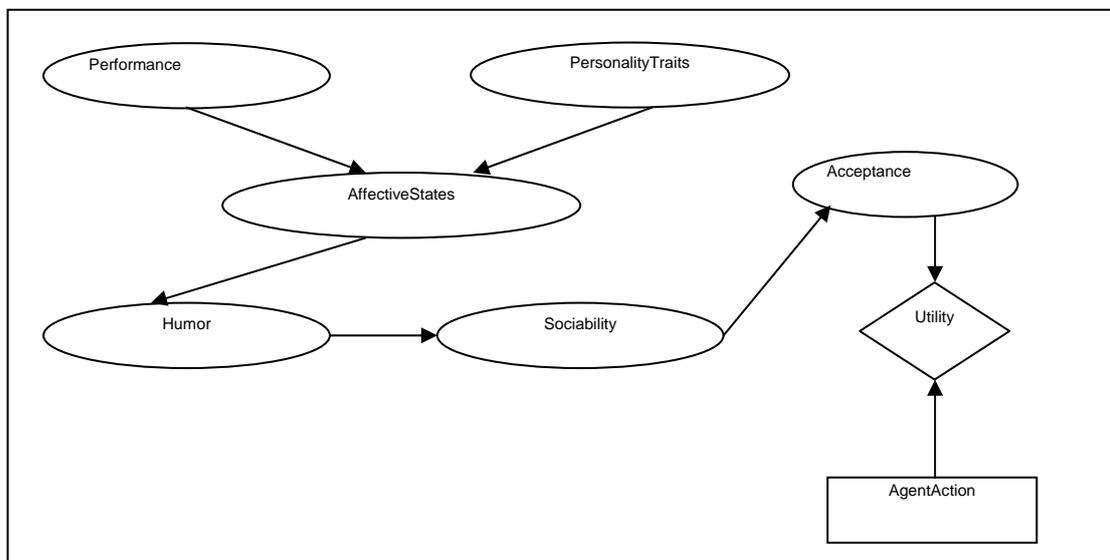


Figure 7. Agent reasoning probabilistic network.

In order to infer the personality traits we are using the model proposed in [12], based on OCC Model. The affective states can be considered as emotion manifestation in a specific time. Conati [13] modeled a Bayesian network to infer emotions. This research uses both models to give us states values to *PersonalityTraits* and *AffectiveState* nodes (Figure 1).

The Interest feature is given by the initiatives taken for the user which material it had access without being recommended.

The Commitment Degree is based on the consistency theory (theory of persuasion of Social Psychology) places that after the people to assume a public commitment, they probably will act in more consistent way with its commitment. The idea is to use this theory as strategy to motivate the students to collaborate with others. Thus, students who collaborate with other more actively (a quantitative measure of interactions can be made here) they will be inserted in groups with more frequency and will receive aid from better quality.

Leadership occurs when a person if becomes capable to modify yours beliefs, attitudes and behaviors of other individuals, organizing them and guiding its action for objectives that start to desire to reach. Leadership also is related to the personality of the person, the psychological attributes of the followers and to the situations where if it uncurls the leadership process. When a student acts in the learning environment, establishes many interactions and contributions in the publisher of networks the agent records these information and verifies which student collaborated more actively in the network construction and which student had its work a lesser number of times modified. This fact can indicate the presence of a leader, therefore they are people who lead the tasks and the integrant ones of the group accept its position. How much bigger the number of accepted contributions, greater the leader pointer.

3.2 Group Level

The group level takes into account the cohesion and the confidence (or trust) in a workgroup. The agent group reasoning is a probabilistic network, where confidence node influences the cohesion node, like group cohesion is based on individual confidence in a group.

The group formation agent can decide who brings more benefit when joining. This reliable type is interesting to help in the judgment of the agent regarding its confidence in the group that another agent belongs. The confidence is defined as the belief of an agent in the attributes such as the trustworthiness, honesty and ability of the agent "trusted" question. The reputation of an agent defines an expectation on its behavior that is based on the comments of the agent or information on the last behavior of the agent in a specific context in data moment. He assumes that he has two agents, the agent and the agent B. When the agent A does not have no direct interaction with agent B or it is not certain on the trustworthiness of B, the agent A can take decisions based on the reputation of the agent B (gotten with other agents, or either, asking to other agents on the reputation). Once agent A interacted with agent B, it can verify or establish its confidence in the agent B in accordance with its degree of satisfaction in the interaction and use this confidence to take decisions for future interactions [14] [11]. Mechanisms on confidence and reputation can be used by agents to differentiate the good ones of bad (adequate or inadequate) the collaborating ones.

Joined groups have a great productivity. The cohesion is the attractiveness that the group exerts for its members, that of it they desire to continue to participate, resisting the idea to abandon it. The cohesion is analyzed using the sociometric test of Jacob Moreno [15]. We can define as group cohesion the solidarity and establishment of loyalty in a group and measure it for the amount of times that the same people choose to interact.

4 APPLYING ONTOLOGIES TO PROMOTE INTEROPERABILITY

The goal of this work was to provide means to the social agent to communicate with agents outside its original scope, the AMPLIA environment. To achieve this goal, we propose the use of semantic *web* technology. In this section we will present how OWL was applied in order to move towards our goal.

In figure 5 it is shown the social agent's architecture and its interaction with AMPLIA's agents and other agents belonging to the PortEdu. The components that make possible the communication with external agents are the Ontology Management Module and the External Communication Module.

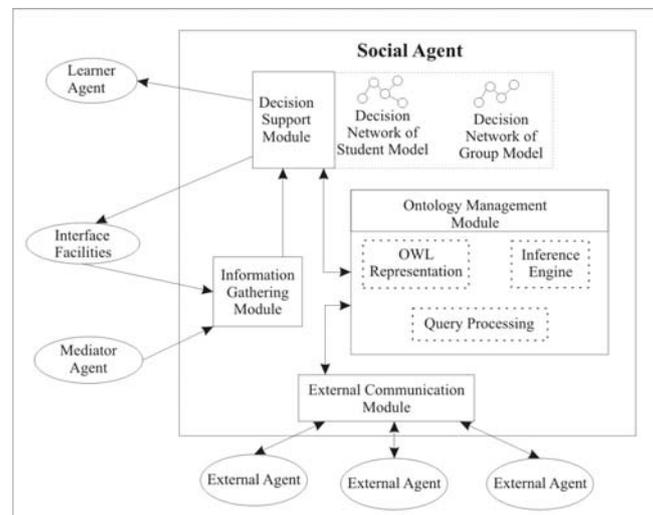


Figure 2. The social agent architecture

4.1 Ontology Management Module

The OWL representation of the agent's knowledge is obtained extending and adapting the structure presented in [7]. As seen on section 2, BayesOWL was developed to solve the ontology mapping problem. Such focus led their approach to a practical but very specific solution. For our case, we used the core concepts of BayesOWL's OWL extension, and added the support to represent multiple state variables and conditional probabilities tables.

We used Protegé [16] to develop the core OWL ontology where all the concepts necessary for representing the social agent's probabilistic knowledge are supplied. The classes defined in this ontology allow the translation from the agent's bayesian networks to OWL. The main classes are:

- Probability Information – a high level abstraction which is the superclass of Prior Probability, Conditional Probability and CPT (Conditional Probability Table)
- Prior Probability – represents the probabilities of states whose nodes doesn't have any parents.
- Conditional Probability – represents a bayesian conditional probability. This class has two properties (a) and (b): State Probability (a), which defines the probability for every state of the node's variable taking into account its (b) conditions; a Condition (b) is a concept that specifies a variable and a state of this variable in order to be used in conjunction with State Probability to represent a Conditional Probability.

In the ontology other concepts such as bayesian network node, conditional probability table, variable, state, among others, are defined, but the core functionality lies on the concept that defines the probability information. As stated earlier, our solution extends and adapts BayesOWL to our context, trying to provide a more generic approach for representing probabilistic knowledge in OWL. Besides the probabilistic information of the agent, this ontology may be used by experts to provide more information about AMPLIA's domain to the other systems in PortEdu.

With the concepts defined in our ontology we can now provide a representation of the agent's networks in OWL. Every node of the network is processed and all information about it (parents, children, CPT, etc.) is converted to OWL creating instances of the pre-defined concepts of our ontology.

The conversion of the Bayesian networks to the ontology is done using Jena [17] and the Hugin Java API [18]. The Hugin API is used to gather information about the bayesian network and Jena is used to create the instances based on the data collected. Jena is a semantic *web* toolkit for Java, it provides APIs for OWL, DAML+OIL and RDFS. Other features of Jena are the inference support for both RDF semantics and OWL semantics, as defined by W3C, and support for a query language RDQL [19], that can be used on the results of RDFS or OWL reasoning. Using the OWL API we are able not only to easily update the agent's ontology representation but also to provide direct support for other agents to make queries using RDQL.

Although OWL is built on top of RDF, a query language on the RDF level, RDQL, isn't capable of considering the restrictions expressed in OWL, for example. There is no standard query language for the semantic *web* yet. OWL-QL [20] is a formal language and protocol for agents, whose knowledge is expressed in OWL, to use on query-answer dialogs. This query language is intended to be a candidate standard query language for OWL and is designed to be suitable for a broad range of query-answering services and applications.

Since the goal of this work is to provide an interoperable way for the social agent to communicate, it is also fundamental that the supported query language to be a standard, as OWL. As soon as a W3C standard or recommendation on query language is released we plan on updating our architecture. Since the agent implementation is modular, it won't be a problem adding support for another query language.

4.2 External Communication Module

The external communication module provides the means to communicate with external agents. In our case, external agents are agents from Portedu and from others learning environments that are attached to the portal. Portedu is developed as multi-agent system that complies with the FIPA [21] standards, in order to promote interoperability dealing with the heterogeneous nature of the different learning environments that it supports. There is no standard on the content language used on the message. The developer is free to use the language that best suit his needs. In our case, to increase the interoperability, we use OWL.

Thus, the communication module is capable to send and receive messages using FIPA ACL. It is responsible for receiving the queries from agents, parsing them and forwarding the content to the ontology management module. Upon receiving a response from the ontology module, already in OWL, a message envelope is assembled, accordingly to the FIPA ACL specification and the message is returned to its sender, following the message protocol.

The implementation of the communication was made easy by using FIPA-OS [22], a middleware developed to facilitate the implementation of multi-agent applications that complies with the FIPA specifications. The architecture provides a names service, directory facilitator, message transport, parser services and a library of FIPA interaction protocols. All the mandatory components of a FIPA platform are available. For agent development, FIPA-OS supplies the necessary Java APIs. There are other alternatives to FIPA-OS, which is no longer under development, but the current PortEdu implementation is supported by FIPA-OS. Thus, to facilitate integration we chose to implement our prototype with FIPA-OS.

5. CONCLUSIONS

In this paper we aimed to increase the interoperability of the social agent making possible his communication with agents outside his domain. Such agents are part of an educational portal, Portedu, whose architecture is grounded on a FIPA agent platform. Such platform allows all agents from the educational systems to communicate with each other through FIPA ACL. The use of common language to exchange the messages already provides a significant level of interoperability, but the fact that the content languages doesn't follows any standards can be a drawback.

Our work proposed the use of OWL as the content language used by the social agent. OWL is the latest ontology language proposed by W3C towards the semantic *web*. The several tools available to work with OWL made it easy to

provide several ontology related services, such as query processing and inference engine. Since the social agent's knowledge is expressed using Bayesian networks, it was necessary to provide an alternate OWL representation, which enables our agent to communicate with a standard content language.

This demonstrates one application of the semantic *web* technology as to improve the interoperability on a multi-agent educational portal. Although being applied to a close heterogeneous domain, our approach to facilitate the interaction among heterogeneous agents may be applied on other contexts.

OWL was not developed considering the possibility of modeling uncertain knowledge. Such limitation reduces the applicability scope of semantic *web* and is a very important issue to be addressed by standardization committees, such as W3C.

Our approach for representing probabilistic information with OWL is very practical, further work needs to be done to provide solid semantic background to the model. A possible way is using PR-OWL, but more tools are necessary to develop both the multi-entity Bayesian networks and for the utilization of the extended OWL language.

The utilization of OWL by the social agent is a first step towards a semantic educational portal, since it is feasible to consider adopting this ontology language as a standard on the portal, making possible, for example the development of students' personal assistants that automatically gather information from heterogeneous sources (agents) and provide the student with educational resources accordingly to its current needs. Once all the agents present in the portal are able to communicate with one another, apply techniques to merge and exchange concepts [7] [8] [22] from its ontologies, the functionality of one specific educational system will increase due to the additional knowledge available to its agents.

6 REFERENCES

- [1] Berners Lee, Tim; Hendler, James; Lassila, Ora. The Semantic *Web*. Scientific American 2001; 284(5), pg. 35-43.
- [2] Horrocks, Ian; Patel-Schneider, Peter F.; Harmelen, Peter F. From SHIQ and RDF to OWL: The making of a *web* ontology language. Journal of *Web Semantics*, vol. 1, pg7--26, 2003.
- [3] Ding, Li; Kolari, P.; Ding, Zhongli; Avancha, Sasikanth. Using Ontologies in the Semantic *Web*: A Survey. In: Ontologies in the Context of Information Systems, Springer-Verlag. October, 2005.
- [4] Dean, Mike; Schreiber, Guus, eds. OWL *Web* Ontology Language Reference: <http://www.w3.org/TR/owl-ref/>
- [5] Nakayama, L.; Vicari, R. M.; Coelho, H. An Information Retrieving Service for Distance Learning. Transactions On Internet Research, v. 1, p. 49-56, 2005.
- [6] Vicari, R. M., Flores, C.D., Seixas, L., Silvestre, A., Ladeira, M., Coelho, H.: A Multi-Agent Intelligent Environment for Medical Knowledge. In: Journal of Artificial Intelligence in Medicine, Vol.27. Elsevier Science, Amsterdam, (2003) 335-366.
- [7] Rong Pan *et al.*, A Bayesian Network Approach to Ontology Mapping. In: Proceedings of the Fourth International Semantic *Web* Conference, November 2005.
- [8] Costa, Paulo; Laskey, Kathryn B.; Laskey, Kenneth J. PR-OWL: A Bayesian Ontology Language for the Semantic *Web* In: Proceedings of the Workshop on Uncertainty Reasoning for the Semantic *Web*, International Semantic *Web* Conference, 2005.
- [9] Laskey, Kathryn B.; Costa, Paulo. Of Klingons and Starships: Bayesian Logic for the 23rd Century. In: Uncertainty in Artificial Intelligence: Proceedings of the Twenty-first Conference. 2005, AUA Press: Edinburgh, Scotland.
- [10] H. Maturana, F. Varela. Tree of Knowledge: The Biological Roots of Human Understanding. Shambhala Publications, Boston, MA, 1998.
- [11] C. Castelfranchi. Affective Appraisal versus Cognitive Evaluation in Social Emotions and Interactions. Paiva, A. (Ed.): Affective Interactions, Towards a New Generation of Computer Interfaces. Lecture Notes in Computer Science, 1814, Springer, 2000. pp. 76-106.
- [12] R. Prada, A. Paiva, Believable Groups of Synthetic Characters. In: AAMAS'05, July 2529, 2005.
- [13] C. Conati, Probabilistic assessment of user's emotions in educational games. *Journal of Applied Artificial Intelligence*. 16(7-8):555-575. 2002.
- [14] Y. Cao, G. Sharifi, Y. Upadrashta, J. Vassileva, Interpersonal Relationships in Group Interaction in CSCW Environments, *Proceedings of the User Modelling UM03 Workshop on Assessing and Adapting to User Attitudes and Affect*, Johnstown, June 22, 2003.
- [15] J.L. Moreno, *Who Shall Survive? Foundations of Sociometry, Group Psychotherapy and Psychodrama*. Beacon, NY: Beacon House. 1953.

- [16] The Protegé Ontology Editor and Knowledge Acquisition System. Available at: <http://protege.stanford.edu>
- [17] Carroll, Jeremy J. *et al.* Jena: Implementing the Semantic *Web* Recommendations. In: Proceedings of the 13th international World Wide *Web* Conference, p. 74-83. ACM Press, 2004.
- [18] Madsen, A. L., Jensen, F., Kjærulff, U. B., Lang, M. (2005). HUGIN - The Tool for Bayesian Networks and Influence Diagrams, *International Journal of Artificial Intelligence Tools* 14 (3), pages 507-543.
- [19] Miller, L.; Seaborne, A.; Reggiori, A. Three Implementations of SquishQL, a Simple RDF Query Language. In: Proceedings of the First International Semantic *Web* Conference. Springer-Verlag, 2002, pg 243.
- [20] Fikes, Richard; Hayes, Patrick; Horrocks, Ian. OWL-QL - A Language for Deductive Query Answering on the Semantic *Web*. In: *Journal of Web Semantics*, vol. 2, issue 1. Science Direct, 2005.
- [21] Foundation for Intelligent Physical Agents. Available at: <http://www.fipa.org>
- [22] Wiesman, Floris; Roos, Nico. Domain Independent Learning of Ontology Mappings. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2, IEEE, 2004, pgs. 346-853.