

UNIVERSIDADE CATÓLICA DE PELOTAS
CENTRO POLITÉCNICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

**EXTENSÃO VIRTUAL DO MUNDO REAL:
INTEGRAÇÃO SEMÂNTICA
E
INFERÊNCIA**

Dissertação apresentada como
requisito parcial para a obtenção do grau de
Mestre em Ciência da Computação
DM-2010-2-006

LUCIANO EDSON MERTINS

Orientador: Dr. Luiz Antônio Moro Palazzo

Pelotas, fevereiro de 2011.

*Dedico este trabalho ao meu pai (memória póstuma)
que me ensinou o quão importante é a alegria, a humildade,
a tolerância e a dedicação nas ações do dia a dia para a
conclusão do almejado com competência e satisfação.*

AGRADECIMENTOS

*Agradeço a Deus pelo amor e graça concedida;
Aos meus avós pelos exemplos de estrutura familiar;
A meus pais pelo esforço, amor e dedicação a minha criação;
A minha esposa Adriane, pela compreensão, carinho e apoio;
A meus filhos pela ingenuidade e a alegria;
A meu orientador pela paciência e o saber;
E a todos que, direta ou indiretamente, contribuem para o
meu crescimento pessoal e a realização deste trabalho o meu
Muito Obrigado!*

Não me sinto obrigado a acreditar que o mesmo Deus que nos dotou de sentidos, razão e intelecto, pretenda que não os utilizemos.

Galileu Galilei

RESUMO

Este trabalho apresenta um modelo para a extensão do Mundo Real através de sistemas de Mundos Virtuais 3D que garante a integração semântica e a capacidade de inferir novas informações, utilizando-se de uma plataforma comum que agrega a Base de Conhecimento, raciocinadores capacitados a operá-la e interfaces de padrões abertos para conexões com os mais diversos sistemas. Sua motivação origina-se no uso crescente dos sistemas de Mundos Virtuais 3D, que mesclam jogos 3D com a abordagem das redes sociais, e na carência destes sistemas em relação à capacidade de compreender semanticamente o que ocorre com seus avatares e usuários. Esta ausência de capacidade semântica vai na contramão da evolução pretendida para a Internet e considerada benéfica por este autor, o que motivou este trabalho a compreender estas tecnologias e construir um protótipo que agrega ao Mundo Virtual 3D a capacidade de notificar seus usuários com informações que não foram manualmente fornecidas a Base de Conhecimento, mas inferidas da mesma, através de dados de nível mais elementar.

Palavras-chave: Mundos Virtuais, Metaverso, Base de Conhecimento, Ontologia, Inferência.

ABSTRACT

This paper presents a model for the extension of the real world through 3D Virtual Worlds system that ensures semantic integration and the ability to infer new information, using a common platform aggregating the Knowledge Base, reasoners for operate it open standards and interfaces for connections to the most diverse systems. It's motivation stems from the increasing use of 3D Virtual Worlds systems, 3D games that mix with the approach of social networks, and the lack of these systems in relation to the ability to semantically understand what happens to their avatars and users. This lack of semantics goes against the intended for the Internet evolution and considered beneficial by this author. This motivated this work to understand these technologies and build a prototype that adds to the 3D virtual world the ability to notify its users with information that is not manually provided to the Knowledge Base, but inferred from it, using data from the most elementary level.

Keywords: Virtual Worlds, Metaverse, Knowledge Base, Ontology, Reasoning.

LISTA DE ILUSTRAÇÕES

Figura 1: Reunião sobre funcionalidades a desenvolver no <i>Open Wonderland</i> , realizada em 30/06/2010 às 14:00 GMT +3.....	23
Figura 2: Ilha RICESU no <i>Second Life</i>	25
Figura 3: Servidor para Testes na UCPel (http://200.132.45.36:8081).....	26
Figura 4: Camadas da <i>Web Semântica</i>	44
Figura 5: Formato SWRL legível ao Ser Humano.....	45
Figura 6: Exemplo de Regra SWRL.....	46
Figura 7: Sistema de Base de Conhecimento. Fonte: Calvanese e De Giacomo (2003).....	48
Figura 8: Descrições Lógicas.....	52
Figura 9: Teste de Satisfatibilidade.....	52
Figura 10: Teste de Subclassificação.....	53
Figura 11: Teste de Equivalência.....	53
Figura 12: Teste de Disjunção.....	53
Figura 13: Instâncias.....	54
Figura 14: Checagem de Consistência.....	54
Figura 15: Checagem de Instância.....	54
Figura 16: Retorno.....	54

Figura 17: Realização.....	55
Figura 18: Principais componentes do raciocinador Pellet. Fonte Sirin et all, 2007.....	57
Figura 19: Uso da ferramenta de desenvolvimento Netbeans 6.8 no Mundo Virtual 3D.....	66
Figura 20: Termos FOAF agrupados por categorias.....	70
Figura 21: Classes (1º nível) utilizadas pelo Seven (Seven, 2009).....	71
Figura 22: Modelo de integração entre aplicações, ORPlat e Bases de Conhecimento.....	76
Figura 23: Módulos da plataforma ORPlat.....	76
Figura 24: ORPlat implementado em um Servidor JEE. Imagem adaptada do site http://www.e-zest.net/distributed_application.html	78
Figura 25: Base capaz de comunicar-se com o ORPlat.....	80
Figura 26: Ontologia de Ator definindo parentescos.....	81
Figura 27: Propriedade de Dado.....	82
Figura 28: Propriedade de Objeto.....	82
Figura 29: serviços RESTful.....	84
Figura 30: Requisição de Pessoas na Ontologia.....	85
Figura 31: Adicionando uma instância ao conceito Homem.....	85
Figura 32: Associando Pais a um indivíduo.....	86
Figura 33: Pais de um indivíduo.....	86
Figura 34: Arquivo de Propriedades do Módulo <i>Ontology</i>	87
Figura 35: Tela de configuração do <i>Darkstar</i> no <i>Open Wonderland</i>	88
Figura 36: Adicionando a base que utilizará os recursos ORPlat.....	88
Figura 37: Arquitetura de comunicação entre o Módulo instalado no <i>Open Wonderland</i> e o ORPlat.....	89

Figura 38: Instâncias na Ontologia.....	90
Figura 39: Avatares sobre a mesma base.....	90

LISTA DE TABELAS

Tabela 1: Alguns jogos que marcaram evoluções. Traduzido de Messinger et al (2009).....	20
Tabela 2: Comparativo entre <i>Second Life</i> e <i>Open Wonderland</i>	28
Tabela 3: Comparativo Hermit/FaCT++/RacerPro/Pellet/REL. Fonte: Pan, Zhao and Ren, 2009.....	58
Tabela 4: Comparativo Hermit/Pellet/FaCT++. Fonte: Shearer, Motik and Horrocks, 2008..	59
Tabela 5: Raciocinador e SWRL. Fonte: Meech, 2010.....	60
Tabela 6: Cláusulas SWRL.....	83

LISTA DE ABREVIATURAS E SIGLAS

<i>ABox</i>	<i>Assertional Box</i>
<i>AGPL</i>	<i>Affero General Public License</i>
<i>API</i>	<i>Application Programming Interface</i>
<i>DL</i>	<i>Description Logic</i>
<i>GPL</i>	<i>General Public License</i>
<i>LGPL</i>	<i>Lesser General Public License</i>
<i>ORPlat</i>	<i>Ontologia – Raciocinador -Plataforma</i>
<i>OWL</i>	<i>Web Ontology Language</i>
<i>RDF</i>	<i>Resouce Description Framework</i>
<i>REST</i>	<i>Representational state transfer</i>
<i>SQL</i>	<i>Structured Query Language</i>
<i>SWRL</i>	<i>Semantic Web Rule Language</i>
<i>TBox</i>	<i>Terminological Box</i>
<i>XSD</i>	<i>XML Schema Definition</i>
<i>WADL</i>	<i>Web Application Description Language</i>
<i>W3C</i>	<i>World Wide Web Consortium</i>

SUMÁRIO

Introdução.....	14
1.1 Motivação.....	14
1.2 Definição do Problema.....	15
1.3 Objetivos.....	16
1.4 Organização do Trabalho.....	16
2 Mundos Virtuais.....	18
2.1 Origem.....	19
2.1.1 Jogos.....	20
2.1.2 Redes Sociais na Internet.....	21
2.2 Mundos Virtuais 3D.....	22
2.3 Plataformas de Mundos Virtuais 3D.....	24
2.3.1 Second Life.....	24
2.3.2 Open Wonderland.....	26
2.3.3 Comparativo Second Life versus Open Wonderland.....	28
2.4 Considerações.....	29
3 Lógica de Descrição e Programação Lógica.....	30
3.1 Mundo Aberto X Mundo Fechado.....	31
3.2 Lógica de Descrição	32
3.2.1 Conceitos e Construtores básicos da Lógica de Descrição.....	32
3.2.2 Representação de Conhecimento através da Lógica de Descrição.....	35
3.2.3 Regras	36
3.3 Programação Lógica.....	37
3.4 Programação Lógica e Lógica de Descrição.....	38
3.4.1 Modelo de Integração.....	38
3.5 Considerações.....	39
4 OWL e SWRL.....	40
4.1 OWL.....	40
4.1.1 OWL 1.....	41

4.1.2 OWL 2.....	42
4.2 SWRL.....	44
4.2.1 Sintaxe.....	45
4.2.2 Built-Ins.....	46
4.2.3 Restrições.....	46
4.3 Considerações.....	47
5 Raciocinadores adequados a Lógicas de Descrição e Programação Lógica.	48
5.1 Raciocínio sobre Terminologia.....	49
5.1.1 Satisfatibilidade de Conceito.....	50
5.1.2 Subclassificação de Conceito.....	50
5.1.3 Equivalência de Conceito.....	50
5.1.4 Disjunção de Conceito.....	51
5.2 Raciocínio sobre Instância.....	51
5.2.1 Consistência da Instância.....	51
5.2.2 Checagem da Instância	51
5.2.3 Retorno.....	52
5.2.4 Realização.....	52
5.3 Exemplo genérico sobre a utilidade dos raciocinadores.....	52
5.4 Paradigmas de Raciocinadores para Lógica de Descrição.....	55
5.4.1 Algoritmos Estruturados.....	55
5.4.2 Algoritmos Tableau.....	56
5.5 Raciocinadores para Web Semântica.....	56
5.5.1 Pellet.....	56
5.5.2 FaCT++.....	57
5.5.3 HermiT.....	59
5.6 Raciocinadores e SWRL.....	60
5.7 Considerações.....	60
6 Cenários para extensão do Mundo Real.....	62
6.1 Cenário Hospitalar.....	62
6.2 Cenário Educação à Distância.....	63
6.3 Cenário Penitenciário.....	65
6.4 Cenário para desenvolvimento de <i>software</i>	65
6.5 Considerações.....	67
7 Aspectos da Base de Conhecimento para Mundos Virtuais.....	68
7.1 Ontologias	70
7.2 Agrupamento.....	72
7.2.1 Ontologia de Ambiente.....	72
7.2.2 Ontologia de Coisa.....	72
7.2.3 Ontologia de Ator.....	73

7.2.4 Ontologia de Ação.....	73
7.3 Raciocinadores.....	73
7.4 Otimizações na Plataforma para Mundos Virtuais 3D.....	73
7.5 Considerações.....	74
8 Plataforma ORPlat.....	75
8.1 Modelo.....	75
8.2 Arquitetura.....	76
8.2.1 Tecnologias.....	77
8.3 Considerações.....	79
9 Experimento.....	80
9.1 Ontologia.....	81
9.2 Regras.....	83
9.3 Serviços do ORPlat.....	83
9.3.1 Funcionamento dos Webservices RESTful.....	85
9.3.2 Ontologia e Persistência.....	87
9.4 Módulo do Open Wonderland.....	87
9.4.1 Instalação e configuração.....	87
9.4.2 Comportamento.....	89
9.5 Base de Conhecimento.....	90
9.6 Resultados.....	90
9.7 Considerações.....	91
10 Considerações Finais.....	93
10.1 Conclusões.....	93
10.2 Contribuições.....	94
10.3 Trabalhos Futuros.....	95
Anexo A.....	104
Anexo B.....	118
Anexo C.....	121

INTRODUÇÃO

1.1 Motivação

No processo evolutivo das tecnologias computacionais, que a cada ano apresenta novas perspectivas e inovações, uma das áreas que vem recebendo interesse, mas que ainda não conseguiu modificar os mecanismos tradicionais é a área de interface entre o ser humano e o computador. Esta afirmação fica evidente, se avaliarmos que o teclado e o vídeo foram criados praticamente junto com os sistemas computacionais e fizeram parte da primeira geração de computadores liberada para aquisição do grande público. Somente o *mouse* apareceu momentos depois, e mesmo assim já faz parte de várias gerações, sem sofrer grandes modificações. Mas com certeza nos próximos anos viveremos novas formas de interatividade e comunicação entre homens e computadores (Karray et al, 2008).

Além das interfaces propriamente, também uma mudança comportamental sobre como o ser humano realiza encontros e se comunica socialmente está ocorrendo, principalmente com a utilização de Sistemas de Mundos Virtuais 3D (Messinger et al, 2009). Estes sistemas surgiram com a junção dos jogos 3D Online com as tecnologias de Redes Sociais (Messinger, Stroulia and Lyons, 2008), sendo que sua trajetória evolutiva tem sido de concentrar estas e diversas outras tecnologias existentes, permitindo tarefas síncronas e assíncronas de forma remota e interativa. Assim, existem indícios que o surgimento de novas interfaces homem-máquina acrescidos de maiores recursos computacionais farão com que o ser humano utilize o computador de forma diferente da que utiliza nos dias atuais.

Esta nova forma de interação pode ser baseada na utilização dos Mundos Virtuais 3D interconectados, pois atualmente já é possível integrar recursos como voz sobre ip, mensagens instantâneas, visualizações de imagens *online* (*webcam*), compartilhamento de arquivos e aplicativos, tudo através de uma única interface com o Mundo Virtual 3D.

Verifica-se também que diversas propostas nos campos da simulação, do trabalho colaborativo (Nebolsky et al, 2003) e do ensino a distância (Schlemmer et al, 2008) estão considerando o uso intensivo de Mundos Virtuais 3D como as principais ferramentas para a realização das atividades fim. E é neste contexto que percebe-se a utilização dos Mundos Virtuais como extensões do Mundo Real, apresentando de forma digital situações comuns em nosso mundo e novas situações, como por exemplo simulações onde os indivíduos representam personagens de séculos passados ou vivenciam o mundo sobre a perspectiva das formigas.

1.2 Definição do Problema

É fato comprovado de que a atual *web* precisa agregar semântica aos seus conteúdos (Berners-Lee et al, 2001), de forma que não apenas seres humanos, mas também sistemas computacionais possam compreender e inferir novas informações sobre estes conteúdos. Baseando-se nesta ideia é de supor que tecnologias que agreguem novas formas de interação com o usuário, também necessitem incorporar o aspecto semântico. Entretanto, apesar de alguns esforços isolados em determinadas funcionalidades dos Mundos Virtuais (Otto, 2005) (Gutierrez et al, 2005) (Martínez and Matta, 2006), ainda não se percebe uma efetiva preocupação e utilização integrada destas duas tecnologias.

Portanto, se os Mundos Virtuais pretendem integrar-se ao dia a dia das pessoas, como uma ferramenta capaz de expandir o Mundo Real, então os sistemas de Mundos Virtuais precisarão interligar-se com o Mundo Real e entender o seu significado, seja sobre a perspectiva do ambiente em algumas situações, seja sobre a perspectiva do tempo em outras e ou sobre a perspectiva do indivíduo correlacionado entre o Mundo Real e o Mundo Virtual. Esta interligação entre o Mundo Real e os Mundos Virtuais pode seguir os mecanismos semânticos adequados a *web* e padronizados pelo W3C (W3C, 2009), como a linguagem para representar ontologia OWL (OWL, 2009) e os raciocinadores capazes de operarem sobre ela.

Desta forma, o presente trabalho busca desenvolver um vínculo semântico entre o Mundo Real e os Mundos Virtuais e confirmar a possibilidade dos sistemas computacionais auxiliarem os usuários através da inferência de novas informações sobre a base de conhecimento produzida pela unificação semântica. Sendo esta hipótese viável, então a maioria, senão a totalidade dos algoritmos, métodos e processos utilizados atualmente nas

áreas da *web* semântica, da descoberta de conhecimento e da inteligência artificial poderão ser reutilizados e adaptados para funcionar nos Mundos Virtuais.

1.3 Objetivos

O objetivo geral é apresentar mecanismos semânticos para expandir o Mundo Real através do uso de Mundos Virtuais 3D, de forma que a base de conhecimento produzida possa sofrer inferência e gerar novas informações e conhecimentos, permitindo aos Mundos Virtuais auxiliarem, proporem e exporem novas situações aos seus usuários de forma automática.

Como objetivos específicos destaca-se:

- Definição das tecnologias capazes de garantir a representação de ontologias necessárias aos objetivos do trabalho.
- Definição de mecanismos de inferência que garantam a exploração adequada da base de conhecimento.
- Definição e construção de uma plataforma de ontologias e raciocinadores, capaz de garantir a comunicação semântica entre os Mundos Virtuais e o Mundo Real e causando o menor impacto possível na dinâmica dos Mundos Virtuais, indiferentemente ao sistema utilizado. Esta plataforma deve ser a) modular, para comunicar-se com diversas sistemas externos; b) expansível, permitindo a inclusão de novas ontologias e novas regras e c) evolutiva, suportando o acoplamento de novas tecnologias e abordagens.
- Desenvolvimento de um protótipo para testes, comprovando a viabilidade real de interligar um sistema de Mundos Virtuais com a plataforma de ontologias e raciocinadores produzida.

1.4 Organização do Trabalho

Para realizar os objetivos almejados, o presente trabalho faz uma revisão bibliográfica no capítulo 2 sobre os Mundos Virtuais 3D, resumindo sua história e definindo seu estado atual como sistema utilizável em experimentos reais.

No capítulo 3 a revisão bibliográfica é focada nos conceitos e definições da Lógica de Descrição e da Programação Lógica potencializando as vantagens em combiná-las. O capítulo 4 preocupa-se em estudar e avaliar as linguagens OWL e SWRL.

O capítulo 5 faz o levantamento bibliográfico dos raciocinadores capazes de atender os recursos avaliados nos capítulos anteriores, ponderando sobre os conceitos propostos para os raciocinadores e como a comunidade científica os qualifica. Por fim, apresenta as implementações mais utilizadas atualmente, com suas vantagens e deficiências.

Algumas situações, vislumbradas como possíveis no futuro, são apresentadas no capítulo 6, onde salienta-se que o uso dos Mundos Virtuais 3D tem muito a ganhar, caso estes sistemas sejam incrementados com tecnologias que permitam a percepção semântica do que ocorre em seu interior. Este trabalho entende que esta união irá permitir de fato a extensão do Mundo Real através de Mundos Virtuais 3D.

O capítulo 7 propõe o modelo a ser seguido em relação as ontologias necessárias para o domínio dos Mundos Virtuais e do Mundo Real. Este modelo não pretende especificar detalhadamente as ontologias necessárias, visto a dificuldade de tal feito, mas construir divisões macros que sirvam de marco para a construção incremental das ontologias.

A necessidade de combinar ontologias, raciocinadores e interfaces de comunicação com outros sistemas resultaram na construção de plataforma ORPlat, abordada no capítulo 8. Sua divisão em módulos permite a substituição ou o acréscimo de tecnologias impactando o menos possível nos sistemas que a utilizam, sejam Mundos Virtuais 3D ou outros tipos de sistemas.

O capítulo 9 demonstra os resultados do protótipo que combina o *Open Wonderland* com o ORPlat, especificando uma ontologia de parentescos, instâncias de indivíduos e regras de inferência e demonstrando com sucesso a capacidade de inferência agregada aos Mundos Virtuais 3D.

Por fim, o capítulo 10 apresenta as considerações, contribuições e futuros trabalhos à explorar.

2 MUNDOS VIRTUAIS

Os sistemas de Mundos Virtuais, também conhecidos em algumas esferas por *Metaverso* (Petry, 2009), são sistemas que buscam recriar um mundo sobre determinadas regras e transmitir esta sensação para os usuários que o utilizam, de forma que a percepção do ser humano conceba uma pseudo-realidade, com todas as possibilidades do mundo real acrescidas de situações ainda não imaginadas mas plausíveis no Mundo Virtual.

Seu uso está em franca expansão, motivado pela evoluções tecnológicas em relação às interfaces homem-máquina e pela necessidade de interligar pontos distantes geograficamente, buscando o trabalho colaborativo, treinamento e comércio de forma visual sem a exigência de localização presencial. Tal necessidade é tão evidente que estimasse, segundo Balkin e Noveck (2006a), que no ano de 2006 de 20 a 30 milhões de pessoas acessaram regularmente algum ambiente de Mundo Virtual. Complementa esta informação a estimativa da empresa Gartner Consulting (Gartner, 2007) de que até o final de 2011, 80% dos usuários frequentemente ativos na Internet possuirão uma segunda vida, ou seja um avatar em um sistema computacional.

Com esta interação de milhares de pessoas simultaneamente, já se constata uma nova fronteira para a computação social, vindo a influenciar em larga escala a economia, educação e a sociedade em geral (Messinger, Stroulia and Lyons, 2008). E sobre este aspecto, alguns acertos já efetuados com outras tecnologias estão sendo repetidos, como a utilização em ambientes de aprendizado e interfaces ricas para com o usuário e infelizmente alguns erros, como a falta de semântica do que é realizado, dificultando a interpretação autônoma dos sistemas, apresentando problemas semelhante a atual *web*.

Sobre a perspectiva social pode-se considerar que diversos sistemas computacionais, como comunidades de relacionamento, *blogs*, *fotologs* e comunicadores instantâneos¹ são

1 MSN, Gtalk, Skype

mundos alternativos e paralelos, sendo então tratados como Metaverso (Schlemmer e Bakes, 2008). Porém neste trabalho as tecnologias anteriormente citadas são definidas como sistemas virtuais e o foco fica concentrado na perspectiva tecnológica de Mundos Virtuais 3D, que são sistemas computacionais tridimensionais com interfaces com o usuário que transmitem o sentimento tridimensional de profundidade visual, sonora e comportamental.

Percebe-se que o aspecto visual e sonoro continua evoluindo, mas já faz parte do cotidiano de diversas aplicações e sistemas, porém o aspecto comportamental ainda carece de estudos e tecnologias de apoio, pois é possível fazer com que o usuário perceba o movimento de folhas de uma árvore e deduza que há rajadas de vento, mas a sensação corporal sobre o evento ainda não é de fácil implementação.

2.1 Origem

Os Mundos Virtuais tridimensionais possuem duas linhas de origem tecnológica que em determinado momento se integraram para formar o que hoje considera-se atender a nomenclatura de Mundo Virtual 3D (Messinger et al, 2009). De um lado os ambientes de jogos 3D e do outro lado as redes sociais na Internet.

Os jogos 3D são desenvolvidos por uma indústria forte, competitiva e lucrativa, onde realidade, interação, recursos físicos e inovação são fundamentais para agregar e manter adeptos. Arelado a sua evolução, estão as estruturas e tecnologias de última geração que, muitas vezes, são subsidiadas para entrarem no mercado. Sua lucratividade, via de regra, fica em cima do rendimento retornado pelos jogos, seus adendos e expansões.

As redes sociais na Internet, em contra partida, focam seus esforços em conectar pessoas e fornecer formas de integração e relacionamento entre estas através de recursos assíncronos, como mensagens, vídeos, sons e síncronos como conversas em tempo real, por texto, áudio e ou vídeo. Estes recursos caracterizam-se em tecnologias distintas que se unificam apenas a nível de interface com o usuário, normalmente através de um navegador *web*, apenas para melhor atender as necessidades de usabilidade. No seu núcleo estas tecnologias são bastante segmentadas, dificultando associações e interpretações por parte dos sistemas computacionais. As redes sociais tem no *marketing* e na propaganda sua maior, senão única, fonte de renda, utilizando a interatividade com o usuário e monitorando suas preferências

para oferecer produtos e serviços que vão de encontro com o contexto no qual o usuário se encontra.

2.1.1 Jogos

Das primeiras versões eletrônicas de “consoles” ligados a televisões em que a visão era bidimensional, os jogos eram embutidos no *hardware* e os gráficos eram limitados até os dias de hoje, com consoles como as do Playstation 3² e do Xbox³, em que a visão 3D está sendo expandida para a sensação ocular, os movimentos do corpo do jogador estão sendo interpretados pelos controles e as imagens remetem a cenas quase reais pode-se contabilizar mais de 25 anos e fica evidente a enorme evolução que os jogos sofreram. Um levantamento evolutivo pode ser visto na tabela 1, que foi traduzida de Messinger et al (2009) e onde consegue-se obter maiores informações sobre a história e a evolução do jogos.

Tabela 1: Alguns jogos que marcaram evoluções. Traduzido de Messinger et al (2009).

Jogo	Ano de Lançamento	Plataforma	Característica inovadora
Pong	1972	Arcade	Primeira máquina de jogos eletrônicos a base de moeda a fazer sucesso. Seguida pelos Jogos Tank, Indy 500, Space Invaders e Pac-Man.
Super Mario Brothers	1986	Console Vídeo Game	Jogo lançado nos EUA pela Nintendo Entertainment System.
Populous	1989	PC e Console	Primeiro jogo que dá ao usuário algum controle sobre o mundo do jogo.
Doom	1993	LAN	Jogo em terceira pessoa que permite aos jogadores se conectarem através da rede local e atuarem no jogo de forma colaborativa ou rival.
Quake	1996	Internet	Primeiro jogo que permitiu a conexão dos jogadores através da Internet e trocarem suas criações em um jogo de tiro em primeira pessoa.
Grand Theft Auto	1997	PC e Console	Primeiro jogo de uma série de jogos que permite ao jogador explorá-lo de forma não linear o que implica em roteiros diferentes para a sua conclusão.
The Sims	2000	PC e Internet	Jogo expandido para um ambiente compartilhado, onde grande parte dos Conteúdos são produzidos pelos próprios jogadores.
World of Warcraft	2004	Internet	Um dos mais populares jogos em ambiente compartilhado, com 11 milhões de contas ativas compartilhando um único mundo.

Atualmente técnicas de inteligência artificial estão fazendo com que os jogos simulem inteligência, permitindo ao jogador sensação de experiências, desafios e percepções diferentes de outro jogador em virtude das escolhas e da postura dentro do jogo. Conexões de Internet velozes permitem a interação de dezenas a centenas de jogadores no mesmo evento, compartilhando informações, atuando em grupos com mesmos interesses ou competindo por metas.

² Console da Sony

³ Console da Microsoft

Enfim, todos estes componentes podem ser mesclados aumentando a gama de possibilidades e dos recursos de interação fazendo com que este tipo de entretenimento faça parte do cotidiano das pessoas, indiferente a idade, e conseqüentemente permitindo o crescimento das indústrias desta área.

2.1.2 Redes Sociais na Internet

As redes sociais na Internet, no qual o estudo esta focado, pode ser melhor caracterizado pelos serviços e *software* de relacionamento que estão atualmente disponíveis na Internet do que propriamente com o estudo das Redes Sociais, que por si só é uma área de estudo. Assim, toda a comunicação mediada por computador entre indivíduos se adéqua ao termo Redes Sociais na Internet utilizado neste trabalho.

Cardozo (2008c) centraliza a questão das redes sociais na Internet com a valorização dos elos informais e das relações em si, sobrepujando as estruturas hierárquicas. Neste contexto, as redes sociais na Internet se caracterizam pelos sistemas *web* de fácil utilização, unidos aos valores e interesses que o usuário deseja compartilhar, ou não, com os demais indivíduos ao redor do mundo.

Serviços e *sites* como Facebook⁴, Orkut⁵, Twitter⁶, entre outros, são notoriamente reconhecidos como ferramentas de Redes Sociais, mas outros serviços também se enquadram neste nicho, como *blogs*, mensageiros instantâneos, *photoblogs*, etc. Basicamente, uma pessoa possui um usuário em um *site*, ou aplicação, onde consegue disponibilizar conteúdos diversos, como textos, fotos, vídeos, sons, etc. O acesso a estes conteúdos pode ser aberto e sem restrição ou pode restringir o acesso a grupos de interesse do próprio usuário. Estes lançam convites a outros usuários ainda não relacionados e o convidado é questionado pelo sistema sobre o aceite. Diversos outros recursos e ferramentas podem estar disponíveis, como a classificação dos contatos em categorias, conversas online, envio de mensagens, etc.

Apesar das Redes Sociais na Internet serem construídas sobre uma tecnologia relativamente simples e barata, se comparada com os investimentos necessários a produção de jogos, elas possuem grande número de adeptos e não por apresentar grandes inovações

4 <http://www.facebook.com/>

5 <http://www.orkut.com.br>

6 <http://www.twitter.com/>

tecnológicas, mas por permitir a integração das pessoas em volta de temas de interesse comuns e de aproximar virtualmente pessoas distantes fisicamente. Assim o ser humano moderno incorpora e substitui comportamentos comuns a anos passados como almoço em família aos domingos por relações remotas pelas redes sociais em virtude da impossibilidade de encontros presenciais.

2.2 Mundos Virtuais 3D

Os sistemas de Mundos Virtuais 3D, como já mencionado, mesclaram as capacidades de interfaces visuais tridimensionais e disponibilizadas nos jogos 3D com a capacidade de construir redes de contato e relacionamento absorvidos das Redes Sociais na Internet. Na prática são sistemas computacionais em um modelo cliente-servidor, de maneira tal, que qualquer indivíduo que deseje utilizá-lo, precisa instalá-lo localmente e possuir uma conexão de boa qualidade com a Internet sempre ativa para utilizar o sistema, que necessariamente precisa se conectar ao servidor. Qualquer atualização em um dos clientes é imediatamente replicado para os outros clientes, sincronizando a visão do ambiente de forma coerente para todos os usuários *online*.

Basicamente a pessoa ou usuário é representado no sistema por um avatar, que pode ou não ser uma imagem baseada na forma humana, mas dono de um perfil e de características que o seu usuário lhe atribui. O termo Avatar em Mundos Virtuais 3D fica melhor esclarecido em (Schlemmer e Bakes, 2008):

“Nos metaversos, MDV3D, os sujeitos são representados por um avatar, termo Hindu para descrever uma manifestação corporal de um ser imortal, ou uma manifestação neste mundo de um ser pertencente a um mundo paralelo, por vezes até do Ser Supremo. Deriva do sânscrito Avatâra, que significa “descida”, normalmente denotando uma (religião) encarnação de Vishnu (tais como Krishna), que muitos hinduístas reverenciam como divindade. Muitos não-hindus, por extensão, usam o termo para denotar as encarnações de divindades em outras religiões. No contexto tecnológico, na área de realidade virtual, o termo refere-se meramente a uma representação gráfica de um sujeito

em um mundo virtual. De acordo com a tecnologia, pode variar desde um sofisticado modelo 3D até uma simples imagem.”

Portanto os avatares para os sistemas computacionais são representantes visuais, das ações empregadas pelo seu usuário, que pode interagir com o ambiente do Mundo Virtual 3D e com outros avatares, conversando através de imagens, mensagens de texto e recursos de voz/vídeo. Neste ponto é importante observar que o ambiente do Mundo Virtual 3D gerencia diversas regras e possibilidades, controlando acesso a recursos e locais, validando usuários, formas de comunicação, acesso a arquivos, vídeos, permissões para leitura, escrita e modificações em objetos de forma geral, com o objetivo de viabilizar o uso do sistema com segurança e controle. Então percebe-se que em relação aos aspectos computacionais, o ambiente em si possui “consciência” de tudo o que ocorre em seu interior e esta capacidade pode fornecer informações úteis e contextualizadas. Como exemplo, na Figura 1 ilustra-se uma reunião em um Mundo Virtual 3D, com a utilização de mensagens de texto e comunicação por áudio, tudo interno e integrado ao sistema. O ambiente pode ser adaptado para registrar quem conversa com quem, por onde os avatares mais circulam, quais os grupos de indivíduos mais frequentes, etc.



Figura 1: Reunião sobre funcionalidades a desenvolver no *Open Wonderland*, realizada em 30/06/2010 às 14:00 GMT +3.

2.3 Plataformas de Mundos Virtuais 3D

O conceito de Mundo Virtual 3D permite a sua implementação em diversas plataformas e tecnologias tornando possível o uso de tecnologias como vídeo conferência⁷, voz sobre ip, compartilhamento de aplicações entre estações de trabalho e uso de áudio e vídeo, ou seja, o Mundo Virtual 3D pode concentrar praticamente toda a gama de conteúdos dispersos pela Internet em uma única aplicação, permitindo o controle centralizado e digitalizado de tudo que acontece no ambiente.

Das diversas plataformas existentes, duas merecem destaque. Uma por ser, além de um produto estável, a mais utilizada no Brasil e a outra por, apesar de estar em desenvolvimento, permitir total modificação no seu código, inclusive participação no seu desenvolvimento.

2.3.1 *Second Life*

O *Second Life* (Second Life, 2009), ou apenas SL, foi idealizado em 1999 por Philip Rosedale e desenvolvido em 2003 pela empresa americana Linden Lab. Sucesso de público mundial, tem recebido atenção de diversas áreas de estudo, como o comportamental, mercadológico, econômico, pedagógico e também o ligado à Computação.

Utilizando o conceito de mundo único segmentado por ilhas, o SL deixa transparente ao usuário a transposição entre servidores de controle destas ilhas e no melhor estilo de uma rede social de Internet, permite localizar amigos *online* em outras ilhas, conversar e se teletransportar até eles.

O SL caracteriza-se também por ser um produto estável, de código fechado e proprietário, que executa sobre diversas plataformas de *hardware* e cujo acesso aos *software* cliente e posteriormente ao sistema é liberado mediante um cadastro no *site*⁸ do SL.

A empresa Linden Lab mantém uma estrutura confiável de servidores, de forma que raramente há indisponibilidade de acesso e seu modelo de negócios baseia-se em liberar acesso ao público em geral e cobrar mensalidades apenas daqueles indivíduos ou organizações que tem interesse em alocar espaço em uma ilha (Santos, 2009). Pode parecer inviável, mas praticamente todas as grandes organizações e entidades do mundo já possuem

7 Uso de câmeras ou webcams para transmissão de imagens reais em tempo real

8 <http://www.secondlife.com>

seus espaço no *Second Life*. Em nota no *blog* a Lindem divulgou números de acessos e de crescimento, como o de 826.000 acessos por usuários diferentes apenas no mês de março de 2010 (Linden, 2010).

Como atrativos deste ambiente verifica-se gráficos e sons de boa qualidade e a utilização de uma moeda única, chamada Lindem. O avatar pode ganhar Lindens, através de atividades exercidas dentro do próprio ambiente ou através da aquisição no próprio *site* do *Second Life*. O usuário também tem acesso a recursos de programação através de uma linguagem de *script* própria⁹ baseada em eventos e estados, mas com uma sintaxe muito similar as linguagens C e Java.

Como ilustração dos gráficos e do ambiente do *Second Life*, a Figura 2 apresenta uma visão panorâmica da ilha da RICESU (RICESU, 2010) presente no mesmo. Salienta-se que a RICESU é uma entidade que agrega instituições de ensino que buscam motivar o uso do *Second Live* como ferramenta importante no desenvolvimento da Educação Digital e que a mesma necessitou alocar espaço nos servidores da Linden (Schlemmer et al, 2010).



Figura 2: Ilha RICESU no *Second Life*

2.3.2 Open Wonderland

O *Open Wonderland* (OpenWonderland, 2010) é um projeto inicialmente denominado Wonderland e sustentado pela Sun Microsystems que incentivou o desenvolvimento de tecnologias necessárias para atender o sistema através do uso da linguagem Java. Desde o início a licença era de *software* livre específica da Sun, mas com a aquisição da Sun pela Oracle, esta última liberou o projeto sobre a licença GPL 2.0 e retirou o incentivo ao desenvolvimento. Porém a abertura do sistema não o enfraqueceu, pois muitas instituições de ensino no mundo e algumas pequenas empresas estão mantendo seu desenvolvimento e sua evolução.

Sua abordagem não é a de um sistema pronto para uso, mas a de uma plataforma robusta e livre para a construção de pequenos Mundos Virtuais independentes, o que permite que cada instituição ou até mesmo indivíduos possuam seu próprio mundo, a custos de sustentabilidade apenas do *hardware* e da conectividade com a Internet. Algo muito similar ao que é feito atualmente com as estruturas de armazenamento de páginas e documentos da *web*. A Figura 3 apresenta a imagem de um cliente em um servidor disponibilizado para testes na Universidade Católica de Pelotas.

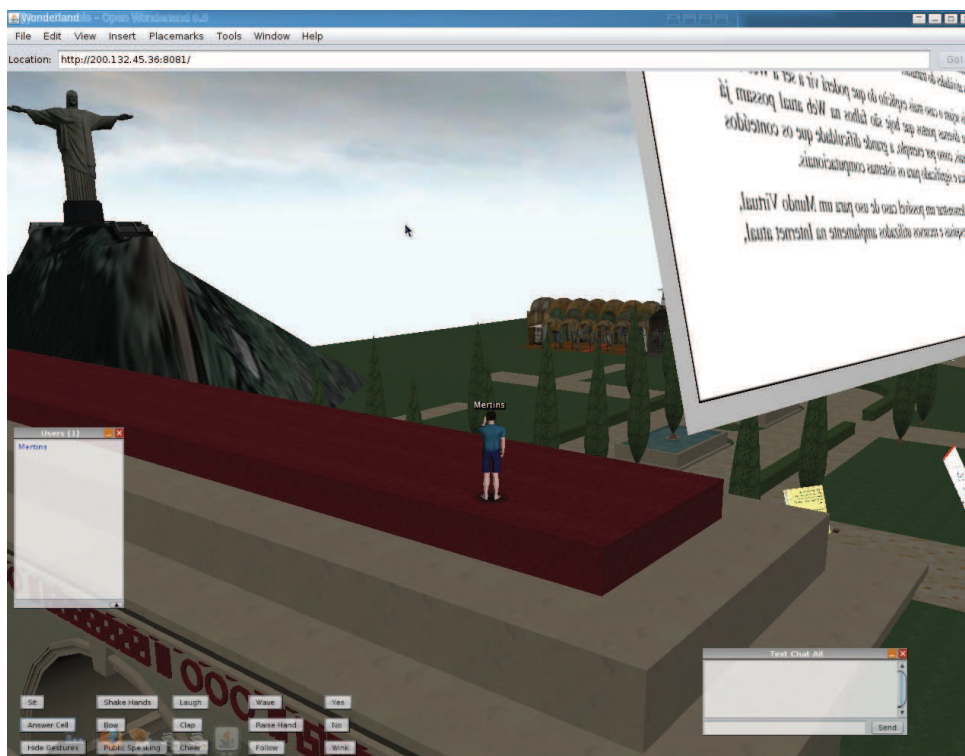


Figura 3: Servidor para Testes na UCPel (<http://200.132.45.36:8081>)

Assim como as páginas *web* que podem possuir conexões para outros servidores (*links*), os mundos do *Open Wonderland* também podem possuir portais para outros servidores, o que viabiliza a navegação entre estruturas distintas.

Dependendo da forma que o servidor de Mundo Virtual está configurado, ele pode permitir acessos anônimos ou apenas de usuários cadastrados e baseado em regras controladas pelo ambiente e configuradas pelo administrador, o avatar poderá ser autorizado a entrar em determinadas regiões, salas ou prédios, a criar ou copiar objetos, usar ou não áudio e vídeo, programar ações através de linguagem de *scripts*, adicionar imagens, textos e a executar aplicações no formato X11¹⁰ armazenadas em um servidor de aplicações interligado ao servidor do *Open Wonderland*. Este inclusive é um recurso interessante, que permite adicionar ao Mundo Virtual 3D aplicações consagradas atualmente como o navegador Firefox¹¹, a ferramenta de desenvolvimento Netbeans¹² e praticamente qualquer aplicação que execute em ambiente XWindow, de maneira que todos os avatares no momento possam interagir entre si e com as respectivas aplicações.

O *Open Wonderland* encontra-se atualmente em sua versão 0.5 *Preview 4* e apesar de ainda não ser considerada pelos desenvolvedores como uma *release*, já é mais estável e interessante que a versão 0.4, cujo desenvolvimento não sofre alterações há mais de dois anos. A estrutura da versão 0.5 é modular, permitindo criações de novos objetos e controles adequados a cada necessidade, sem prejudicar outras instâncias de Mundos Virtuais 3D com esta plataforma.

Também, a partir da versão 0.5, ficou muito fácil importar objetos e estruturas 3D para o sistema, inclusive com o uso de ferramentas amplamente conhecidas e dominadas nesta área, como o *Google Sketchup*¹³ e outras. Também é possível importar objetos 3D já construídos e disponibilizados em diversos *sites* da Internet e onde, em particular, o *3D Warehouse*¹⁴ do próprio *Google Sketchup* auxilia muito.

¹⁰ <http://www.xfree86.org/current/X.7.html>

¹¹ <http://www.mozilla.com/pt-BR/firefox/>

¹² <http://www.netbeans.org/>

¹³ <http://sketchup.google.com/>

¹⁴ <http://sketchup.google.com/3dwarehouse/>

Baseado na experiência prática do autor deste trabalho com este sistema e a troca de informações entre o mesmo e integrantes do desenvolvimento do sistema, alguns pontos são destacados no *Open Wonderland*:

- design focado na colaboração, pois existem diversas formas de compartilhar informação no Mundo Virtual, como aplicações X11, VNC Cliente, Compartilhamento da Tela Cliente, PDF, etc;
- áudio é uma característica integrada ao sistema, apresentando alta definição, qualidade e diversos níveis de integração aos outros objetos do Mundo;
- a Extensibilidade ocorre no servidor e no cliente, de acordo com as necessidades do código a ser produzido, sendo a abordagem de módulos ideal para garantir isolamento do código e reaproveitamento em outros módulos;
- controles de segurança e acesso granulares a nível de objetos e componentes, permitindo interfaces com outros sistemas, como LDAP;
- baseado na adoção de Padrões Abertos, o que facilita a integração com outras ferramentas e sistemas.

2.3.3 Comparativo *Second Life* versus *Open Wonderland*

Um pequeno comparativo (Tabela 2) foi realizado para dar continuidade ao trabalho proposto e validar a possibilidade de aplicação destes dois sistemas.

Tabela 2: Comparativo entre *Second Life* e *Open Wonderland*

Características	Open Wonderland	Second Life
Perfil	Plataforma de Desenvolvimento	Sistema Proprietário
Licença de Uso	GPL 2.0	Proprietária
Permite adequações ao código	Total	Restrita a scripts
Qualidade Gráfica/Áudio	Boa, mas inferior	Boa e superior
Construção de prédios e estruturas	Total	Mediante pagamento anual
Infraestrutura	Precisa ser construída e configurada	Previamente existente
Autonomia sobre acessos e compartilhamentos	Total	Restrita a autorização da proprietária
Extensões	Através de Módulos independentes	Não possui
Instalação Cliente	Pela Web	Baixar do site e instalar

As características definidas no comparativo buscam garantir subsídios considerados importantes para que a proposta do trabalho possa ser executada sem transtornos que a

inviabilize. Assim o perfil, a licença de uso, a permissão de adequações no código, a construção de estruturas e a autonomia sobre acessos e compartilhamentos caracterizam o nível de controle e flexibilidade possível. Também a qualidade gráfica e de áudio é muito importante, pois interessa o sistema ao usuário e de nada adianta o sistema ser flexível se sua qualidade visual não for adequada. As características de infraestrutura e instalação no cliente tem o peso de exigência ou não na preparação para utilizar o sistema e o quanto o usuário é impactado. A característica de extensões demonstra o quão modular é a plataforma em relação a propagação e reutilização do que for produzido.

Entretanto, este comparativo não deve desmerecer nenhuma das duas aplicações, mas apenas justificar a adoção do *Open Wonderland* no presente trabalho, pois a liberdade em modificar e adaptar o sistema a necessidade é fundamental para o sucesso do mesmo.

2.4 Considerações

O trabalho ao avaliar duas possibilidades de Mundos Virtuais, não pretende desconsiderar outras plataformas e acredita que tanto o modelo semântico proposto, quanto a plataforma de execução, também proposta, seja genérico o bastante para se adequar a qualquer plataforma de Mundo Virtual 3D que permita acoplamentos de código ao nível de estrutura do sistema. Este também é uma das motivações na adoção de padrões e códigos abertos no desenvolvimento do trabalho, pois favorece a posterior adequação do que foi produzido a qualquer sistema de Mundo Virtual 3D evitando ou minimizando a infração de licenças de uso e patentes.

3 LÓGICA DE DESCRIÇÃO E PROGRAMAÇÃO LÓGICA

A Representação de Conhecimento em sistemas computacionais baseia-se em uma ou mais linguagens, que em conjunto, são capazes de expressar formalmente situações de um domínio e de fornecer mecanismos capazes de inferir sobre o que foi descrito. Desta forma, sistemas que utilizem estas linguagens, são ditos “inteligentes” por sua capacidade em analisar o que foi descrito explicitamente como “conhecimento” e extrair informações implícitas destes (Nardi and Brachman, 2003).

Existem duas abordagens distintas para efetuar a Representação de Conhecimento, uma baseada no formalismo lógico, onde o foco é a resolução do problema de forma inequívoca sobre o domínio e procura através do processo atingir seus objetivos. A outra abordagem baseia-se na representação não-lógica e foca no domínio que pretende abordar e na sua estruturação, utilizando-se de estruturas em rede e de regras para efetuar seus objetivos.

Formalismos baseados em **redes semânticas** (Allen and Frisch, 1982) e **Frames** (Minsky, 1975) são precursores no modelo de representação não-lógico usando interfaces gráficas como forma de expressão. São adequados para especificar e visualizar conceitos mas carecem de recursos para formalizar semanticamente diversos detalhes da lógica através de gráficos, o que dificulta a representação de domínios complexos. Portanto o formalismo lógico apresenta-se como a melhor solução para a representação de base de conhecimento de domínios complexos.

Neste contexto, a Lógica de Descrição é um dos representantes do formalismo lógico capaz de atender a necessidade de definições de conceitos e, por ser um subconjunto da Lógica de Primeira Ordem (Nardi and Brachman, 2003), fornecer adequados recursos computacionais para expressar semanticamente domínios complexos e permitir a obtenção de novas informações.

Porém um aspecto da Lógica de Descrição que falha, quando no processo de inferência sobre as bases de conhecimento, é a dificuldade de representar regras (Sensoy et al, 2010), chegando a inviabilizar seu uso, caso a regra seja complexa e necessite usar a Lógica Não Monotônica (Minh, 2008).

Neste contexto, um outro viés no processo de raciocínio computacional é Programação Lógica que deriva informações na forma de proposições ou predicados lógicos e que esta gradativamente sendo combinada com a bases de conhecimento descritas com a Lógica de Descrição para suprir suas carências (Knorr, Alferes and Hitzler, 2007).

Assim os próximos itens irão transcorrer sobre a Lógica de Descrição, a Programação Lógica e os benefícios e dificuldades em combiná-las.

3.1 Mundo Aberto X Mundo Fechado

Um conceito importante que difere em uso pela **Lógica de Descrição** e pela **Programação Lógica** é o conceito de **Mundo Aberto** e de **Mundo Fechado** (Hebeler et al, 2009).

O conceito de Mundo Aberto implica na impossibilidade de afirmar que uma sentença é falsa em virtude não ser possível comprovar a sua veracidade e por sua vez, o conceito de Mundo Fechado afirma que uma sentença é falsa caso não seja possível comprovar que seja verdadeira.

Por exemplo, tendo a informação de que indivíduos (Pedro, Paulo, Maria, etc) residem em determinados países (Brasil, Argentina, Uruguai, etc), uma consulta sobre bases de conhecimento que opere sobre o conceito de Mundo Fechado buscando confirmar que determinado indivíduo reside no Brasil, retornará a informação verdadeiro ou falso (null). A mesma pergunta realizada em uma base de conhecimento que opere com o conceito de Mundo Aberto só poderá retornar ou sim (verdadeiro) ou desconhecido. Nunca falso, pois o Mundo Aberto considera que a informação é desconhecida do sistema, mas pode ser verdadeira.

3.2 Lógica de Descrição

A Lógica de Descrição é uma extensão às redes semânticas e *frames*, acrescentando recursos viáveis de formalização lógica, sendo capaz de representar um domínio de uma aplicação, definindo conceitos deste domínio e posteriormente utilizando estes conceitos para especificar propriedades de objetos e indivíduos e suas associações (Baader et al, 2003). Sua definição adota o conceito de Mundo Aberto para avaliar a base de conhecimento descrita por ela.

Resumidamente a Lógica de Descrição modela conceitos, papéis e indivíduos, podendo os conceitos ser entendidos como classes ou categorias de algo. Já os indivíduos são a efetiva existência, também conhecida como instância de um determinado conceito. E os papéis são propriedades que os conceitos e os indivíduos possuem. Tanto conceitos como papéis e indivíduos são representados por predicados unários.

De nada vale estas definições se não houver ligações entre elas, e é justamente neste ponto que entram as relações, servindo como elementos conectores entre conceitos, papéis e indivíduos, e sendo representadas por predicados binários.

Além de representar os conceitos, papéis, indivíduos e suas relações, a Lógica de Descrição fornece símbolos para representar também quantificadores e construtores, onde os quantificadores permitem expressar valores delimitadores e os construtores são operadores especiais que permitem a composição de conceitos e novas relações.

Para melhor entender os construtores é preciso observar que a Lógica de Descrição parte da estruturação de axiomas, que são sentenças ou conceitos primitivos, considerados verdadeiros, sem a necessidade de comprová-los para a aceitação de uma teoria. Os axiomas, na figura de conceitos primitivos, servem de base para a utilização de construtores que permitirão a construção de conceitos complexos.

3.2.1 Conceitos e Construtores básicos da Lógica de Descrição

O conceito mais geral para uma linguagem da Lógica de Descrição é o de Verdadeiro ou Cheio/Integral, representado pelo símbolo \top , e o Falso ou Vazio, representado pelo \perp .

Assim a interpretação de um conceito é realizado através da validação de um conjunto de valores que torna este conceito Verdadeiro ou Falso.

Desta forma, supondo que A e B sejam conceitos primitivos, C e D conceitos complexos, Q represente um quantificador e R represente papéis de conceitos, verifica-se os seguintes construtores básicos:

- Negação ($\neg A$). Aplicado a conceitos primitivos representando a inversão de Verdadeiro para Falso e vice-versa;
- Conjunção ($C \sqcap D$). Intersecção entre dois conceitos complexos, extraindo o que há de comum;
- Restrição de Valor ($\forall R.C$). Requer que todo o conjunto de elementos do conceito C obrigatoriamente possuam o papel R. Pode ser associado ao termo “para todos”;
- Quantificação Existencial Limitada ($\exists R.T$). Requer que todos possuam o papel R como verdadeiro;
- Equivalência ($C \equiv D$). C é equivalente a D.

Assim, sendo Pessoa e Masculino conceitos primitivos e existindo o papel temFilho, constrói-se alguns conceitos complexos:

- Homem \equiv Pessoa \sqcap Masculino
- Mulher \equiv \neg Homem
- Pai \equiv Homem \sqcap \exists temFilho.T
- PaiDeMenina \equiv Pai \sqcap \forall temFilho.Mulher

O uso dos construtores definidos acima caracterizam a família de linguagem de Lógica de Descrição básica conhecida como \mathcal{AL} , e representam o conjunto mínimo de construtores para efetiva representação do conhecimento. A partir da \mathcal{AL} , acrescenta-se novos construtores que estendem o poder da linguagem e das deduções que podem ser efetuadas sobre a base representada, sendo os seguintes construtores frequentemente adicionados:

- Negação($\neg C$)[*Complement-C*]. Aplicado a conceitos complexos para representar a inversão de Verdadeiro para Falso e vice-versa;

- Quantificação Existencial ($\exists R.C$) [*Existential Qualification-E*]. Requer que algum elemento do conjunto de elementos do conceito C possuam um papel R. Pode ser associado ao termo “existe algum”;
- Restrição Numérica ($\geq n.R$ e $\leq n.R$) [*Number Restrictions-N*]. Restringe a cardinalidade mínima e máxima dos papéis, onde $\geq n.R$ restringe em pelo menos n relações R e $\leq n.R$ restringe em no máximo n (\leq) relações R;
- Disjunção ($\sqcup D$) [*Union-U*]. União entre dois conceitos.
- Hierarquia de Papéis [*Role Hierarchy-H*]. Permite construir níveis hierárquicos entre os papéis.
- Nominais [*Nominals-O*]. Permite a identificação única de uma instância de um conceito.
- Inversão de Papéis (R^-) [*Inverse Roles-I*]. A inversão e transitividade de um papel permite definir propriedades inversas, como por exemplo, “éParteDe” ser o inverso de “temParte”.
- Papéis reflexivos, irreflexivos e disjuntos [*Reflexivity, irreflexivity and disjointness role-R*]. Trabalha com a reflexividade, irreflexividade e união de papéis.
- Uso de tipos de dados [*DataTypes properties-(D)*]. Valores de dados e valores de tipos.

Para representar a linguagem resultante do acréscimo de um ou mais dos construtores acima, adiciona-se ao \mathcal{AL} a letra inicial dos construtor extra, de acordo com a escrita em inglês, para definir o poder da linguagem. Assim $\mathcal{AL} [U] [E] [N] [C]$ são famílias de linguagens de lógica de descrição.

Segundo Baader, Horrocks e Sattler (2007) o acréscimo de construtores aumentou o poder das linguagens de descrição, mas a mesma abordagem de agregar uma letra a cada nova operação suportada estendeu em demasia a nomenclatura que identifica a linguagem. Assim a combinação \mathcal{ALC} com transitividade nas regras foi abreviada para a letra \mathcal{S} e continua acrescentado as outras letras, de acordo com o suporte a operação. Neste trabalho iremos concentrar a atenção nas linguagens de descrição $\mathcal{SHIF}(D)$, $\mathcal{SHOIN}(D)$ e \mathcal{SHOIQ} .

respectivamente os construtores suportados pela OWL Lite, OWL DL e OWL 2, que serão abordados no capítulo 4.

3.2.2 Representação de Conhecimento através da Lógica de Descrição

Aplicações de representação do conhecimento que utilizam Lógica de Descrição ou suas derivadas são frequentemente divididas em duas estruturas distintas, o do conhecimento intencional e o do conhecimento extensional. O conhecimento intencional trata sobre a generalização do domínio do problema, modelando a base de conhecimento de forma genérica e definindo regras que são válidas em qualquer situação do problema, de forma análoga a definição de entidades em um Sistema Gerenciador de Banco de Dados (SGBD). Em contrapartida, o conhecimento extensional é específico e particular ao problema em questão, focando nos indivíduos envolvidos no domínio, similar às ações de inserir registros nas tabelas do SGBD.

As razões para distinguir estas duas categorias de conhecimento não são significantes quanto a sintaxe da descrição em si, pois neste sentido, tudo se resume a Lógica de Descrição. A distinção se faz necessária, primeiro para facilitar a perspectiva dos modeladores do domínio, que podem segmentar o processo da descrição genérica, do processo de manifestação específica.

Um segundo motivo é que a distinção facilita os algoritmos dos raciocinadores, permitindo que estes processem o conhecimento intencional em separado do conhecimento extensional, visto que alguns problemas a serem inferidos são específicos de uma ou outra categoria.

Estas estruturas em bases de conhecimento desenvolvidas com Lógica de Descrição são transportadas para dois componentes, o TBox e o ABox (Baader et al, 2003).

3.2.2.1 TBox

O TBox (*Terminological Box*) é o responsável pela representação do conhecimento intencional especificando a terminologia do domínio. A definição dos conceitos, tanto primitivos como complexos, acontece neste componente na forma de equivalências lógicas. Porém algumas restrições se fazem necessárias, quando utilizadas na lógica de descrição:

- um conceito pode ter uma única definição;
- um conceito não pode ser definido em termos que se referenciem ou a ele mesmo ou a outros termos que direta ou indiretamente se referenciem a ele.

Este tipo de restrição é necessária para evitar que a base de conhecimento entre em um estado de indecisão e torne-se inválida para uso.

Um exemplo de conceito definido no *TBox*:

- Homem \equiv Pessoa \sqcap Masculino.

3.2.2.2 ABox

O conhecimento extensional está contido no *ABox* (*Assertional Box*) e refere-se a instanciação dos conceitos definidos no *TBox*. Dois tipos de declarações podem ser realizadas:

- de conceitos $C(x)$, onde x é um indivíduo de C ;
- de relações $R(x,y)$, onde x está relacionado com y através de R .

Exemplos de *ABox*:

- Pessoa \sqcap Masculino(João) - declaração de Conceito
- temFilho(João,Pedro) – declaração de Relação.

Acima deduzimos que “João”, por ser “Masculino” e “Pessoa”, também é “Homem”, devido a regra *TBox* definida anteriormente. Da mesma forma, “Pedro” é filho de “João”.

3.2.3 Regras

Como já citado, um aspecto da Lógica de Descrição que deixa a desejar no processo de inferência sobre as bases de conhecimento é a dificuldade de representar regras (Sensoy et al, 2010), especificamente quando for interessante o formato “SE x ENTÃO y ”, pois apesar de ser possível em muitos casos, adaptar associações na Lógica de Descrição para atender necessidades específicas de regras, estas associações costumam ser paliativos que reduzem a

capacidade da base de conhecimento em servir como uma estrutura genérica e de amplo aspecto. Acrescenta-se o fato de que mesmo que as adaptações paliativas sejam aceitas, ainda existe a dificuldade em adaptar a regra, que é diretamente proporcional a complexidade da própria regra, ou seja, regras complexas levam a adaptações complexas. Ainda, se a regra necessitar o uso da Lógica Não Monotônica fica inviável a adaptação da regra a Lógica de Descrição (Minh, 2008). Hebler et al (2009) detalham no capítulo 9 as deficiências e carências que a OWL apresenta e como as regras suprem estas carências. A OWL é uma das linguagens baseada na Lógica de Descrição e será abordada no próximo capítulo.

Sobre a Lógica Não Monotônica, verificamos que a Lógica Clássica é dita Monotônica porque, apesar da sua importância para o desenvolvimento da Ciência da Computação, possui duas grandes limitações: 1º) consequências não são revistas como base para novas informações e 2º) Novas informações só podem levar a novas consequências (Huang et al, 2009).

O exemplo que ilustra estas características é a incapacidade da Lógica Clássica de trabalhar com uma definição simples para o ser humano, “de que todas as aves voam”. Em determinado momento futuro percebemos que o “pinguim” é uma ave que não atende a afirmação. Uma exceção simples, mas que na Lógica Monotônica não encontra modo de ser especificada.

A Lógica Não Monotônica, resolve esta questão, admitindo como verdadeiro a inferência realizada na ausência de informação contrária e que poderá ser invalidada, em um futuro, por novas informações. Desta forma, na Lógica Não Monotônica, podemos representar a afirmação de que “toda Ave voa, exceto os Pinguins”. E em um futuro pode-se adicionar novas exceções (avestruz, ema, etc) (Pequeno et al, 2007). Para isto utiliza a abordagem de Mundo Fechado ao invés de Mundo Aberto e possui a propriedade de poder reduzir ou modificar a base de conhecimento, permitindo o raciocínio em um conjunto incompleto de fatos.

3.3 Programação Lógica

A Programação Lógica é um paradigma de programação que faz uso direto da Lógica Matemática, e a principal diferença em relação ao paradigma imperativo é de que, ao invés

de especificar o algoritmo que deve processar, a Programação Lógica especifica fatos e regras, que quando executadas, podem trazer de zero a infinitas respostas válidas.

Usando o conceito de Mundo Fechado e a inferência Não Monotônica, a Programação Lógica (Huang et al, 2009) basicamente constitui seu algoritmo em dois elementos disjuntos: a lógica e o controle. O elemento lógico especifica o que deve ser solucionado e o elemento de controle especifica como a solução pode ser obtida (Palazzo, 1997).

As linguagens que suportam o paradigma de Programação Lógica se valem do uso da cláusula de Horn (WikiHorn, 2010) (Horn, 1951) para descrever e realizar suas regras, o que destaca duas características importantes: 1º) o raciocínio automatizado é eficiente, pois a resolução de duas cláusulas de Horn produz outra cláusula de Horn e 2º) é facilmente reescrita para ser interpretada como uma regra no formato:

- SE (alguma coisa) ENTÃO (outra coisa).

3.4 Programação Lógica e Lógica de Descrição

A combinação entre a Programação Lógica e Lógica de Descrição tem sido considerada como uma proposta interessante para agregar funcionalidades a *Web Semântica* (Huang et al, 2009)(Alsaç and Baral, 2001).

Mas agregar o melhor de ambos, apesar de suas raízes originarem-se da **Lógica de Primeira Ordem** (Dayantis, 1987), apresenta desafios que ora invalidam a Lógica de Descrição, ora invalidam a Programação Lógica (Antonioni and Wagner, 2003). O principal problema é o inconveniente de que a Lógica de Descrição pressupõem o conceito de Mundo Aberto, enquanto que Programação Lógica utiliza o conceito de Mundo Fechado (Bry and Schaffert, 2003). Para amenizar esta incompatibilidade as atuais implementações estão se valendo de subconjuntos e restrições lógicas para viabilizar seu uso.

3.4.1 Modelo de Integração

A integração entre a Programação Lógica e a Lógica de Descrição é classificada em um modelo Híbrido ou Homogêneo (Meditkos and Bassiliades, 2008), onde o Híbrido apresenta uma clara separação entre a **linguagem da ontologia** (definição) e a **linguagem de**

inferência (regras). Por sua vez, o Homogêneo possui a ontologia e as regras embarcadas em uma única linguagem, impedindo a separação entre o que é ontologia e o que são regras.

3.5 Considerações

O presente capítulo tratou de forma genérica duas Lógicas que servirão de base para as linguagens OWL e SWRL descritas no capítulo a seguir e que efetivamente serão combinadas no Plataforma ORPlat responsável por realizar a integração semântica entre os Mundos Virtuais ao Mundo Real.

4 OWL E SWRL

As Lógicas abordadas no capítulo anterior descrevem de maneira genérica o modelo conceitual adotado pelo trabalho para a sua realização. Como é comum, o modelo é representado e implementado com uma ou mais tecnologias que se combinam para providenciar os resultados esperados. Especificamente, este capítulo abordará as tecnologias OWL e SWRL que implementarão na prática os conceitos abordados anteriormente.

4.1 OWL

OWL (OWL, 2009) é uma linguagem especificada através da linguagem XML e proposta pelo consórcio W3C (W3C, 2009) como um padrão na construção de Ontologias. Sua especificação baseia-se nas definições da Lógica de Descrição, comportando conceitos, relacionamentos, quantificadores, propriedades e construtores. Sua construção é realizada sobre a tecnologia RDF¹⁵ e a suas tarefas principais são:

- Formalizar o domínio através do uso de classes e propriedades;
- Definir instâncias e propriedades sobre as classes;
- Inferir e raciocinar sobre o que foi construído.

A OWL, por ser uma Linguagem de Lógica de Descrição funciona com o conceito de Mundo Aberto e é representada através de arquivos no formato XML, apresentando uma aparência que favorece menos o ser humano e mais os sistemas computacionais e que pode ser amenizado com o uso de editores como o Protégé (Protégé, 2010).

A primeira versão recomendada pelo W3C da OWL é de 2004. Seu uso e adoção apontaram diversos benefícios, mas também limitações (Grau et al, 2008) o que provocou a

¹⁵ The Resource Description Framework (RDF)

recomendação em 2009, por parte do W3C da OWL 2 (Motik, Patel-Schneider and Parsia, 2009).

4.1.1 OWL 1

Primeira especificação, foi padronizada com melhorias e revisões sobre a linguagem DAML+OIL e em virtude da complexidade computacional, três sub-linguagens são propostas sendo a distinção entre elas em função do grau de formalismo e expressividade exigido para utilizá-las (Smith et al, 2004). Respectivamente na ordem de menor à maior expressividade, OWL *Lite*, OWL DL e OWL *Full*.

Importante salientar que as três sub-linguagens são compatíveis quanto ao aumento do grau de formalismo, ou seja, uma ontologia válida em OWL *Lite* é válida na OWL DL e uma ontologia válida na OWL DL, também estará válida na OWL *Full*.

4.1.1.1 OWL Full

A OWL *Full* não é propriamente uma sub-linguagem, mas toda a linguagem OWL, o que permite total integração com RDF. Porém esta expressividade máxima sem restrições pode facilmente levar a base de conhecimento a uma situação de indecisão computacional e portanto, inviabilizar seu uso em sistemas computacionais que buscam inferir sobre a mesma.

4.1.1.2 OWL DL

A OWL DL é o meio termo entre expressividade e usabilidade, obtendo a expressividade necessária das lógica de descrições para representar o domínio e mantendo a base de conhecimento sobre regras que garantem a decidibilidade por parte dos raciocinadores que avaliam esta base. Sua principal característica é a distinção entre classes, propriedades e instâncias e o suporte aos construtores da lógica de descrição, o que candidatam esta sub-linguagem como a mais preparada para uso real em sistemas de representação de conhecimento.

As principais regras e restrições são:

- Clara distinção entre classes, tipos de dados, propriedades dos tipos e propriedades dos objetos. Com isto, uma classe não pode ser, ao mesmo tempo, uma instância ou indivíduo.
- O conjunto de propriedades dos objetos são distintos das propriedades dos tipos de dados. Assim propriedades como inverso de, função inversa, simetria, transitividade não podem ser especificada para propriedades de tipos de dados.

Os construtores suportados por esta sub-linguagem em relação a Lógica de Descrição lhe garantem a nomenclatura $\mathcal{SHOIN}(\mathcal{D})$ como visto no item 3.2.1

4.1.1.3 OWL Lite

Por fim, a OWL *Lite* sendo a menos representativa das sub-linguagens, não contempla toda a expressividade da lógica de descrição, atendendo melhor a especificação de *frames*. Sua expressividade basicamente permite hierarquias de classificação e restrições simples, com limitações como a cardinalidade que se resume a 0 ou 1.

Conforme o item 3.2.1, os construtores suportados pela OWL *Lite* definem a nomenclatura $\mathcal{SHIF}(\mathcal{D})$.

4.1.2 OWL 2

A adoção da OWL para a construção de ontologias nas mais diversas áreas da ciência, como Medicina, Biologia, Geografia, Astronomia, e principalmente com uso de aplicações reais e práticas tornou a OWL um padrão de fato como linguagem para desenvolvimento de ontologias e para troca de dados. Alguns exemplos dos tantos existentes vem da NASA¹⁶ e do *National Cancer Institute*¹⁷, onde ambos produzem ontologias de suas áreas de conhecimento.

Obviamente a OWL não representa uma solução ideal e suprema. Seu uso em vários domínios e necessidades revelaram problemas e carências (Grau et al, 2008) que obrigaram o

¹⁶ <http://sweet.jpl.nasa.gov/ontology/>

¹⁷ <http://www.cancer.gov/>

estudo de alternativas e novas funcionalidades que por sua vez resultaram no versionamento da OWL, denominando o desenvolvido até o momento em OWL 1 e definindo a partir de 2009 a OWL 2 (Motik, Patel-Schneider and Parsia, 2009).

Os principais pontos evoluídos com a OWL 2 e detalhados por Grau (Grau et al, 2008).

- Limitações na expressividade em geral– a OWL 1, apesar de expressiva, ainda sofre em alguns pontos a expressar;
- Dificuldades de expressividade nas relações;
- Ausência de expressividade nos tipos de dados;
- Mecanismo de identificação e restrição de chaves.

Assim como a OWL 1, a OWL 2 também é uma linguagem para ontologias com formalismo para auxiliar na construção de uma *Web Semântica*. Para isto, a linguagem permite descrever classes, propriedades, indivíduos e dados. Os construtores da Lógica de Descrição suportados, garante a OWL 2 a nomenclatura $\mathcal{SROIQ}(\mathcal{D})$, conforme descrito no item 3.2.1.

A OWL 2 é segmentada em três *profiles*, mas diferente de OWL 1, em que o foco era permitir maior ou menor expressividade, a segmentação em *profiles* é realizada com o intuito de favorecer a eficiência dos raciocinadores que atuam sobre a ontologia. Os *profiles* são independentes entre si e são denominados OWL 2 EL, OWL 2 QL, OWL 2 RL (OWL2 Profile, 2010)

Os *profiles* na OWL 2 diferenciam-se entre si pelas restrições que cada um possui em relação as estrutura da OWL 2, ou seja, o que irá diferenciar os profiles é a existência, ou não, de determinados construtores e restrições ao seu uso em axiomas e estruturas.

A vantagem em definir *profiles* é a possibilidade de permitir extensões destes, sem exigir a especificação de uma nova versão, o que não fica descartado, mas permite um tempo de vida mais longo e dinâmico para a OWL 2.

Assim o *profile* OWL 2 EL é útil em aplicações com ontologias que contenham um grande número de propriedades e ou classes. Nele o raciocinador deve preocupar-se em como processar o volume de informação baseado no tamanho da ontologia em si.

Por sua vez, o *profile* OWL 2 QL é designado para situações onde o volume de instâncias é alto e o principal objetivo do sistema em relação ao raciocinador é a obtenção de respostas baseadas em perguntas. Os sistemas utilizam a ontologia de uma forma análoga ao uso de sistemas gerenciadores de banco de dados com SQL.

Por fim, o *profile* OWL 2 RL tem como alvo as aplicações que não podem sacrificar a expressividade da linguagem para definir a ontologia e que preferem abdicar da eficiência dos raciocinadores em prol da velocidade em processá-las.

4.2 SWRL

Motik, Horrocks, Rosati e Sattler (2006b) apresentam diversos problemas que são complexos, senão impossíveis, de se modelar utilizando apenas a OWL. Os motivos são oriundos da Lógica de Descrição e foram tratados no capítulo anterior.

Quando o consórcio W3C (W3C, 2009) criou a OWL para preencher uma das lacunas no planejamento e construção de uma *Web Semântica*, o fez em um modelo de camadas, o que permite a construção gradativa da solução. A Figura 4 demonstra o esquema em camadas proposto para a *Web Semântica* e, de acordo com Antoniou et al (2005), a camada *Ontology* (Ontologia) e as inferiores já possuem um grau de maturidade aceitável, em virtude das recomendações publicadas pelo W3C.

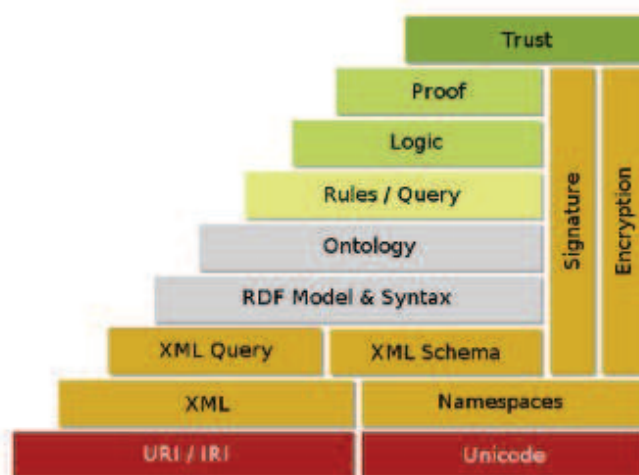


Figura 4: Camadas da *Web Semântica*

Sendo assim, o processo de representar a ontologia e conseqüentemente o conhecimento, parece estar atingindo maturidade e estabilidade suficientes para a propagação de seu uso em sistemas comerciais, como um substituto ou complemento das bases de dados, buscando aumentar o significado e a semântica de suas informações, principalmente quando se tratando de sistemas para Internet, foco principal da linguagem OWL.


Porém, se a descrição do conhecimento parece ter atingido esta maturidade refletida no uso e representação de ontologias escritas com a linguagem OWL, camadas superiores ainda não estão sedimentadas e caracterizadas adequadamente deixando várias questões em aberto, como o que fazer com o significado explícito nos documentos existentes na *Web*? Como raciocinar em cima do significado e obter novos significados? Quais as formas, opções, modelos e ferramentas capazes de efetivar o processamento em cima de uma ontologia?

Estas perguntas precisam ser respondidas pelas camadas de Regras/Questões (*Rules/Query*) e de Lógica (*Logic*) que o consórcio W3C propõem para a *Web Semântica* e neste quesito apenas o processo de *Query* está satisfatório em virtude da padronização da linguagem SPARQL (Sparql, 2010) pela W3C.

A SWRL (*Semantic Web Rule Language*) vem a ser uma proposta feita em 2004 pelo W3C (SWRL, 2004) para compor esta lacuna, na tentativa de agregar a OWL 1 um mecanismo capaz de especificar e interpretar regras. Percebe-se que esta é uma proposta que adota a abordagem homogênea na integração entre a Programação Lógica e a Lógica de Descrição (item 3.4.1).

4.2.1 Sintaxe

A SWRL por fazer parte da ontologia, através de elementos XML específicos torna difícil a interpretação sem algum editor como o Protégé (Protégé, 2010), pois a complexidade do XML tende a crescer com a complexidade da ontologia e das regras. Portanto não pretende-se detalhar o armazenamento XML e sim exemplificar o uso da SWRL de acordo com a representação sugerida pelo W3C (SWRL, 2004) e utilizada no Protégé (Protégé, 2010), que segue sintaxe da Figura 5 baseada na cláusula de Horn (WikiHorn, 2010) (Horn, 1951).



antecedente => resultado

Figura 5: Formato SWRL legível ao Ser Humano

Onde “antecedente” e “resultado” são conjunções de cláusulas.

Variáveis são definidas através da precedência de um ponto de interrogação, como por exemplo: ?x ou ?valor.

A conjunção ou intersecção é explicitada com o símbolo “ \wedge ”. Já a disjunção acontece na construção de outra combinação de cláusulas diferentes que atinjam o mesmo resultado.

Por exemplo a definição de Tio definida na Figura 6 trabalha nas regras da seguinte forma: a) Se X tem um pai Y e o Pai Y tem um Irmão Z então X tem um Tio Z, ou b) se X casado com Y e Z tem tio X então Z tem tio Y. O caso b é representativo de um costume de que casados com tios passam a ser considerados tios também.

$\text{parent}(?x,?y) \wedge \text{brother}(?y,?z) \Rightarrow \text{uncle}(?x,?z)$ $\text{married}(?x,?y) \wedge \text{uncle}(?z,?x) \Rightarrow \text{uncle}(?z,?y)$
--

Figura 6: Exemplo de Regra SWRL

Para os leitores acostumados com a Programação Lógica, o termo “resultado” é melhor reconhecido como a cabeça da cláusula e o “antecedente” equivale ao corpo da mesma.

4.2.2 Built-Ins

A fim de fornecer mecanismos para que métodos definidos pelo usuário possam participar nas cláusulas como conjunções, a SWRL possui um mecanismo denominado *Built-Ins* (O'Connor and Das, 2006).

Algumas definições padronizadas de *Built-Ins* estão disponíveis no *site* do consórcio W3C (SWRL, 2004) e outras podem ser construídas, seguindo regras especificadas na proposta.

4.2.3 Restrições

Para adequar a SWRL à OWL 1, foram necessárias a criação de algumas restrições, que abdicaram de características interessantes da Programação Lógica para evitar que a base de conhecimento entrasse em estado de indecidibilidade (Parsia et al, 2005).

Uma das regras mais restritivas ao poder da Programação Lógica é a proibição que uma variável tem, de aparecer na cabeça da cláusula sem aparecer no corpo. Isto remove um poderoso mecanismo de recursividade da SWRL, mas que se presente na OWL levaria a base de conhecimento a perder a capacidade de decidir (Levy and Rousset, 1998).

4.3 Considerações

A OWL 1, apesar de algumas carências, apresenta-se como um padrão de fato para a representação de bases de conhecimento, permitindo a reutilização e a adequação aos mais diversos modelos. Como evolução natural da OWL1, a OWL 2 acrescenta recursos e poder a linguagem sem no entanto causar perdas, o que favorece sua adoção em novas soluções. A SWRL entretanto, apesar de ser a opção sugerida pelo consórcio W3C, possui alguns pontos ainda em aberto e que merecem estudos, onde mesmo o modelo de integração homogêneo não está consolidado como a melhor abordagem (Drabent, 2010).

Entretanto estas incertezas não prejudicam o presente trabalho, pelo contrário, motivam a clara segmentação do *Framework* para que no futuro seja possível “plugar” e experimentar novas tecnologias e abordagens, apesar de concentrar o atual esforço em integrar a OWL 2 com a SWRL.

5 RACIOCINADORES ADEQUADOS A LÓGICAS DE DESCRIÇÃO E PROGRAMAÇÃO LÓGICA

Sistemas de Base de Conhecimento utilizam linguagens de lógica de descrição não apenas para representar um domínio, o que poderia ser feito com outras tecnologias e linguagens, mas também para permitir o uso de algoritmos de inferência que permitam a extração de conhecimentos implícitos e não percebidos pelo ser humano. Estes novos conhecimentos podem retroalimentar a base de conhecimento, imitando um dos processos humanos para obter conhecimento.

Portanto, sistemas de Base de Conhecimento armazenam informações que descrevem o domínio através do formalismo da Lógica de Descrição e, possuindo volume de informações suficientes, transformam o conhecimento explícito e representado por fatos, em novos conhecimentos existentes no domínio, mas ainda implícitos na base de conhecimento. Os componentes capazes de realizar esta tarefa nos sistemas de Lógica de Descrição adotam o modelo de definição do TBox e do ABox, visto no capítulo anterior, acrescidos de um mecanismo de inferência (Tobies, 2001).

Calvanese e De Giacomo (2003) descrevem um sistema típico de Base de Conhecimento com os elementos TBox, ABox interagindo com o mecanismo de inferência, que por sua vez interagem com a aplicação do usuário.

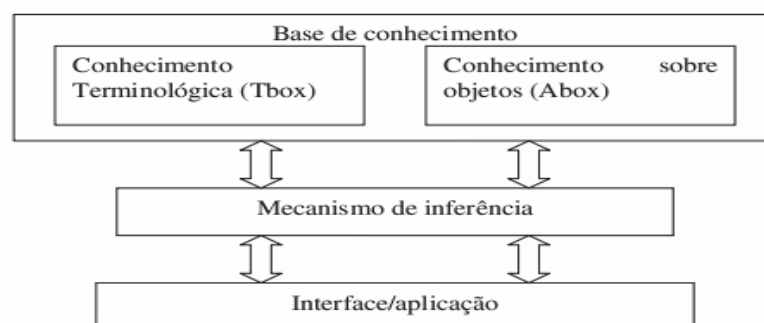


Figura 7: Sistema de Base de Conhecimento. Fonte: Calvanese e De Giacomo (2003)

Os mecanismos de inferência são incorporados aos raciocinadores que podem ser construídos de diversas formas e utilizar diversos algoritmos, desde que respeitem as regras da Lógica de Descrição e sejam eficientes na tarefa de explicitar o conhecimento contido implicitamente na base de conhecimento. Desta forma, a tarefa de explicitar este conhecimento esta interligada a complexidade e a precisão do formalismo utilizado para descrever a base de conhecimento, ou em outras palavras, o grau de complexidade a ser resolvido pelo raciocinador é dependente de qual família de construtores pertence a Lógica de Descrição utilizada para representar o conhecimento. Assim, quanto mais expressiva for a linguagem de descrição, mais complexo é o processo de produzir inferência sobre ela (Fensel et al, 2008).

De forma genérica os raciocinadores segmentam sua atuação em três partes, que podem ou não serem combinadas:

- Efetuar inferência sobre a estrutura da Ontologia;
- Efetuar inferência sobre os relacionamentos entre os conceitos;
- Efetuar inferência sobre as instâncias dos conceitos.

Na prática, percebe-se que os racionadores são construídos gradativamente e vão no decorrer de sua evolução incorporando a capacidade de processar novos construtores e expandindo seu leque de Linguagens de Lógica de Descrição que podem suportar.

Especificamente sobre os serviços que o raciocinador efetua, duas inferências básicas devem ocorrer sobre a terminologia do domínio (TBox), Subclassificação de Conceitos e Satisfatibilidade de Conceito¹⁸ e duas inferências básicas sobre a instância do domínio (ABox), Consistência da Instância e Checagem da Instância¹⁹ (Baader et al, 2003) . Outras formas de inferências são agregadas ao TBox e o ABox como Equivalência, Disjunção, Retorno e Realização.

5.1 Raciocínio sobre Terminologia

Modelar e descrever um domínio para uma base de conhecimento é tarefa dos Engenheiros de Conhecimento que definem novos conceitos, geralmente sobre conceitos já

¹⁸ *Concept Subsumption and Concept Satisfiability*

¹⁹ *Instance Consistency and Instance Checking*

definidos. Sobre esta perspectiva, é importante garantir que o novo conceito faça sentido e não cause contradição nos conceitos já definidos no domínio e esta garantia é fornecida pelos raciocinadores, quando estes atuam sobre o TBox através das verificações de Satisfatibilidade, Subclassificação, Equivalência e Disjunção.

5.1.1 Satisfatibilidade de Conceito

O raciocinador garante a satisfatibilidade de um conceito quando valida as definições deste conceito e garante que elas não afetam outros conceitos de forma negativa, produzindo ambiguidade e ou contradições que invisibilizariam o uso da base de conhecimento.

A satisfatibilidade de conceito é considerada a solução para muitos outros importantes inferências sobre conceitos, pois várias destas inferências podem ser convertidas através de operações lógicas para a inferência da satisfatibilidade.

5.1.2 Subclassificação de Conceito

Os conceitos são organizados através de papéis e de suas relações com outros conceitos, descrevendo muitas vezes situações hierárquicas, similares ao que acontece no mundo real e facilmente exemplificado através dos conceitos de Animal, Mamífero e Gato, onde este último é uma especialização dos primeiros conceitos e garante respostas afirmativas, quando questionamos sobre um Gato ser um Mamífero e ou ser um Animal.

Subclassificação de conceito é justamente a verificação por parte do raciocinador de que determinado conceito é uma subclassificação de outro conceito.

5.1.3 Equivalência de Conceito

Se um ou mais conceitos estão definidos direta ou indiretamente de forma idêntica, o raciocinador define os conceitos como equivalentes. Significa dizer que instâncias de um conceito servirão perfeitamente como instâncias do conceito equivalente.

Como exemplo, se o conceito de Mulher é baseado na conjunção entre os conceitos de Pessoa e Fêmea e dois outros conceitos são formados, o conceito de Homem baseado no conceito Pessoa que não é Mulher e o conceito de Masculino baseado em Pessoa que não é

Fêmea. Existe equivalência de conceito entre Homem e Masculino (Vieira et al, 2005). Assim, qualquer instância do conceito de Homem é equivalente a qualquer instância do conceito de Masculino.

5.1.4 Disjunção de Conceito

Dois conceitos são disjuntos ou desarticulados, se a conjunção (intersecção) entre eles resultar na impossibilidade de instâncias de um conceito pertencerem também a outro conceito.

Como exemplo, se Mulher for um conceito definido como a conjunção entre o conceito Pessoa e Fêmea, e Homem um conceito definido com a conjunção Pessoa e Não Fêmea, conclui-se que o conceito Homem é disjunto do conceito Mulher e que não será possível à uma instância única compartilhar os conceito Homem e Mulher, tendo apenas os conceitos definidos acima.

5.2 Raciocínio sobre Instância

Sendo a terminologia construída pelo Engenheiro do Conhecimento satisfazível e sub-classificável, a próxima etapa são as asserções através das instanciações de conceitos e relações entre estes. Estas são obtidas através das verificações por parte dos raciocinadores apresentados a seguir.

5.2.1 Consistência da Instância

Se uma instância não torna o TBox inconsistente, ou seja, se o TBox é satisfazível e a instância do ABox não o invalida, então a instância é consistente. Esta é uma pergunta que o raciocinador precisa efetuar constantemente, seja para responder ao usuário ou para permitir a especificação de mais uma instância a base de conhecimento.

5.2.2 Checagem da Instância

Dado um indivíduo qualquer, verifica-se se é uma instância de um conceito específico. A checagem serve para confirmar ou não, se uma instância equivale a um conceito específico.

5.2.3 Retorno

Permite encontrar o conceito mais específico em razão de uma instância, servindo para questionar o raciocinador, sobre qual o conceito mais específico que melhor corresponde a determinada instância.

5.2.4 Realização

Questionando o raciocinador sobre determinados conceitos, este localiza na base de conhecimento as instâncias que fazem parte dos referidos conceitos.

5.3 Exemplo genérico sobre a utilidade dos raciocinadores

Para exemplificar a forma como os raciocinadores agregam valor a base de conhecimento, supondo os conceitos da Figura 8, adaptadas do livro *The Description Logic Handbook – Theory, Implementation and Applications* (Baader et al, 2003).

- | | |
|-----|--|
| (a) | $\text{Homem} \equiv \text{Pessoa} \sqcap \text{Masculino}$ |
| (b) | $\text{Mulher} \equiv \text{Pessoa} \sqcap \neg \text{Homem}$ |
| (c) | $\text{Feminino} \equiv \text{Pessoa} \sqcap \neg \text{Masculino}$ |
| (d) | $\text{Pai} \equiv \text{Homem} \sqcap \exists \text{temFilho.Pessoa}$ |
| (e) | $\text{Mãe} \equiv \text{Mulher} \sqcap \exists \text{temFilho.Pessoa}$ |
| (f) | $\text{Hermafrodita} \equiv \text{Homem} \sqcap \text{Mulher}$ |

Figura 8: Descrições Lógicas

Garante-se que um raciocinador, ao avaliar a Satisfatibilidade no Tbox, rejeitará o item (f), pois este causa uma contradição que impossibilita instanciar o conceito proposto, conforme demonstrado na figura 9.

- | |
|---|
| $\text{Hermafrodita}(\text{José}) \equiv \text{Homem}(\text{José}) \sqcap \text{Mulher}(\text{José})$ |
| $\text{Hermafrodita}(\text{José}) \equiv \text{Homem}(\text{José}) \sqcap \text{Pessoa}(\text{José}) \sqcap \neg \text{Homem}(\text{José})$ |

Figura 9: Teste de Satisfatibilidade

Desconsiderando o item (f) da Figura 8, será possível avaliar a subclassificação no TBox, onde o raciocinador percebe que o conceito Pai(d) e Mãe(e) são tipos especializados de Homem(a) e Mulher(b), respectivamente.

Assim, conforme a Figura 10, nesta base de conhecimento, mesmo não estando explicitamente definido, todo Pai é um Homem e toda a Mãe é um Mulher.



Figura 10: Teste de Subclassificação

A equivalência no TBox é percebida pelo raciocinador, quando ele, por exemplo, aplica operações de simplificação nos conceitos e obtém definições iguais para conceitos distintos, como demonstrado na Figura 11. Desta forma o raciocinador conclui que instâncias de Mulher, são também instâncias de Feminino, e vice-versa.

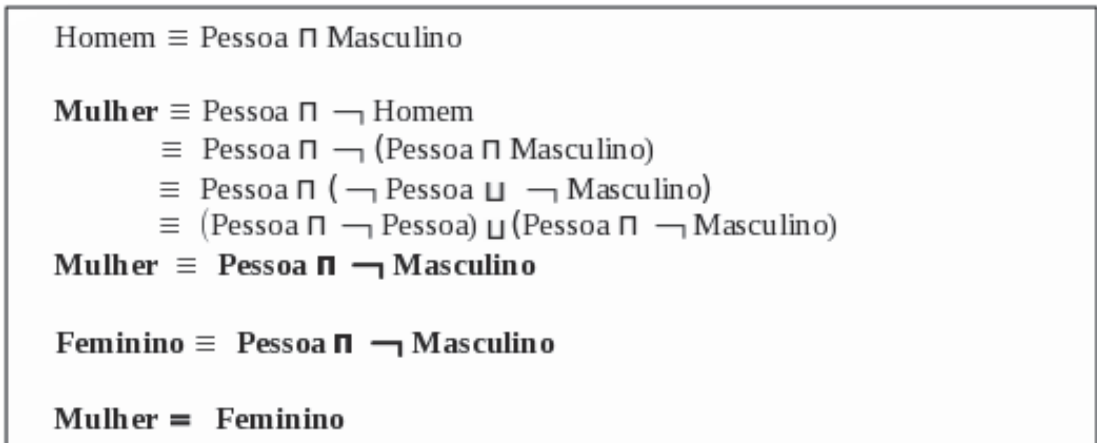


Figura 11: Teste de Equivalência

Para perceber a disjunção no TBox, o raciocinador efetua operações que irão encontrar conceitos contraditórios, como demonstrado na figura 12, onde percebe-se que uma instância de Homem não pode ser uma instância de Mulher e vice-versa.

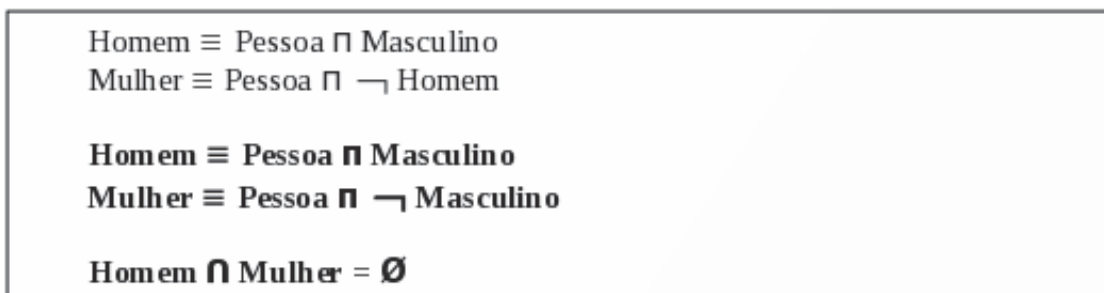


Figura 12: Teste de Disjunção

Para exemplificar as tarefas que os raciocinadores efetuam no ABox será utilizada as instanciações da figura 13.

- | |
|-------------------------------|
| (a) Homem(João) |
| (b) Homem(Francisco) |
| (c) Mulher(Maria) |
| (d) Mulher(Helena) |
| (e) temFilho(João,Francisco) |
| (f) temFilho(Maria,Francisco) |
| (g) Pai(Francisco) |

Figura 13: Instâncias

Os raciocinadores realizam a checagem de consistência no ABox avaliando as instâncias e verificando sua satisfatibilidade, conforme exemplificado na Figura 14, onde João é uma instância de Homem que exige, com sucesso, que João também seja uma instância de Pessoa e Masculino.

Homem(João)
Homem(João) \equiv Pessoa(João) \wedge Masculino(João)

Figura 14: Checagem de Consistência

No exemplo da Figura 15, procura-se descobrir se a instância de João faz parte também do conceito Pai. Como o relacionamento temFilho (letra e, Figura 13) liga a instância João à instância Francisco e o conceito Pai aceita a instância João (letra a, Figura 13), fica caracterizado como verdadeiro a checagem de instância no ABox.

Homem(João)
Pai(João) \equiv Homem(João) \wedge \exists temFilho.Pessoa

Figura 15: Checagem de Instância

O serviço de retorno no ABox, busca encontrar o conceito mais específico de determinada instância, conforme representado na Figura 16, onde o conceito Pai (letra b, Figura 13) é determinado como o conceito mais específico de Francisco, Homem (letra a, Figura 13) é definido como o conceito mais específico de João e Mulher (letra d, figura 13) como o conceito mais específico de Helena.

Francisco ==> Pai
João ==> Homem
Helena ==> Mulher

Figura 16: Retorno

Os raciocinadores fornecem o serviço de realização, devolvendo as instâncias adequadas ao questionamento de determinado conceito, como exemplificado na Figura 17, onde as instâncias João e Francisco representam o conceito Homem e as instâncias Maria e Helena representam o conceito Mulher.

Homem	==>	João, Francisco
Mulher	==>	Maria, Helena

Figura 17: Realização

5.4 Paradigmas de Raciocinadores para Lógica de Descrição

Os raciocinadores para Lógica de Descrição atualmente implementados com código disponível para uso e capacitados a operacionalizar ontologias em bases de conhecimento utilizam diversas abordagens para efetuarem suas tarefas (Tobies, 2001). Apesar destas abordagens não serem modernas, ainda não existe um modelo ideal para qualquer situação, principalmente para inferências em bases de conhecimento volumosas na quantidade de instâncias, com grande quantidade de conceitos, inúmeros relacionamentos entre estes e com alta complexidade de situações a resolver.

Assim o desenvolvimento operacional dos raciocinadores esta basicamente restrito em variações e otimizações de dois modelos de algoritmos, que apesar de imperfeitos por não atenderem toda e qualquer situação de base de conhecimento, podem ser implementados e utilizados em sistemas reais.

5.4.1 Algoritmos Estruturados

O processo clássico de programação, utilizado nos primeiros racionadores, onde comparações sintáticas são utilizadas para definir as respostas para os serviços, como a subclassificação, apresenta dificuldades em atender linguagens de descrição com maior expressividade. Esta dificuldade ocorre em virtude da complexidade exigida dos algoritmos e, conseqüentemente, do tempo necessário que o raciocinador requer para avaliar a ontologia e atender a solicitação.

5.4.2 Algoritmos Tableau

Os algoritmos Tableau e suas variações, funcionam através da refutação da prova de um teorema, provando que a sua negação é falsa, ao invés de provar o teorema diretamente. Foi proposto para raciocinadores para atuar na Lógica de Descrição \mathcal{ALC} e desde então tem recebido evoluções e adaptações para suportar novos construtores (Tobies, 2001).

Baader, Horrocks e Sattler (2007) afirmam que esta é a abordagem algorítmica mais utilizada para implementar raciocinadores para Lógicas de Descrição e possui boa performance em situações reais, servindo como base para mecanismos otimizados, como FaCT++ (Tsarkov and Horrocks, 2006), Racer (Haarslev and Möller, 2001) e Pellet (Sirin et al, 2007).

5.5 Raciocinadores para *Web Semântica*

Diversas implementações e variações de algoritmos tem sido propostos e implementados para a realização de inferência sobre linguagens de Lógica de Descrição, porém o presente trabalho levantou apenas implementações focadas em atender em parte ou na totalidade a OWL 2 (OWL 2, 2010), desconsiderando outras implementações que não processem tal versão da OWL e ou cujos códigos não sejam disponibilizados sobre alguma licença para testes com implementações de código.

5.5.1 Pellet

Pellet (Pellet, 2010) foi originalmente desenvolvido para servir como ferramenta de prova da viabilidade de implementação de raciocinadores para os conceitos sugeridos pelo W3C para a OWL 1. Obtendo a aprovação como ferramenta para inferir sobre a OWL 1, tornou-se popular em vários casos de estudo, vindo a ser o primeiro a suportar toda a expressividade da OWL DL (Sirin et al, 2007).

Seu código é implementado com a linguagem Java e sua licença de uso é dupla, onde uma é de software aberto²⁰, sem custo financeiro mas exigindo que seu uso também se realize em códigos de software aberto e uma de software proprietário, que exige remuneração aos

²⁰ Open Source - <http://www.opensource.org/>

proprietários para aplicações de códigos fechados. Maiores informações podem ser obtidas no próprio *site* do Pellet (Pellet, 2010).

Atualmente o Pellet suporta parte da expressividade da OWL 2 ($SR_{OIQ}(\mathcal{D})$), com os construtores necessários e todos os serviços de inferência básicos são providenciados pelo algoritmo, como Consistência da Instância, Satisfatibilidade de Conceito, Subclassificação de Conceito e Realização.

Existem diversas interfaces para uso do Pellet, que variam de um *site* para *web* ao uso por API de programação. A API é muito útil na integração com aplicações desenvolvidas para trabalhar sobre bases de conhecimentos específicas. Outra interface muito interessante e bastante utilizada é o conector de integração²¹ com o editor de ontologias Protégé (Protégé, 2010). A arquitetura do Pellet pode ser visualizada na Figura 18, onde percebe-se o uso do algoritmo baseado em Tableau para realizar parte das inferências.

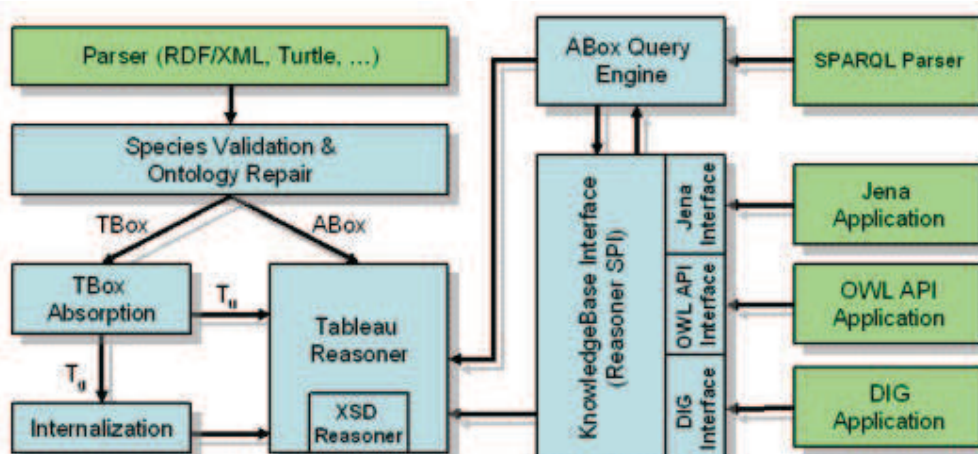


Figura 18: Principais componentes do raciocinador Pellet. Fonte: Sirin et al, 2007.

As demais estruturas demonstradas na Figura 18 não são relevantes para o escopo deste trabalho, exceto para demonstrar a complexidade inerente de qualquer raciocinador, mas podem ser melhor esclarecidas em Sirin et al (2007).

5.5.2 FaCT++

Fast Classification of Terminologies é o significado da palavra FaCT e do raciocinador em questão, que originalmente foi desenvolvido na linguagem C e que no seu lançamento inclui-a de fato dois raciocinadores, um para lógica SHF , e outro para a lógica $SHIQ$. Sua

evolução para o FaCT++(FaCT++, 2010) implicou na mudança para a linguagem de programação C++ e a capacidade de manipular linguagens de descrição mais elaboradas como *SHOIQ*, dando suporte a OWL DL. Atualmente o suporte a OWL 2 está em desenvolvimento atendendo parcialmente as exigências da linguagem *SHOIQ(D)*(Tsarkov and Horrocks, 2006).

Disponibilizado com código aberto, sobre as licenças GNU GPL/LGPL utiliza o compilador C++ GNU versão 3.3. ou superior (Manchester, 2010) e encontra-se na versão 1.4.0. lançada em 29 de abril de 2010.

A base do processamento é o algoritmo de tableaux com otimizações e as interfaces possíveis são a própria API, o DIG²² (DIG 2, 2010), o Protégé (Protégé, 2010), o Jena (Jena, 2010) e OWL API versão 2 (OWL API).

As principais pendências para suportar a OWL 2 e ainda não desenvolvidas no FaCT++ versão 1.4.0 são (FaCT++ 1.4.1, 2010):

- Abordagem *Top/Bottom* na propriedade do Objetos e dos Dados;
- Suporte a chaves;
- Suporte a totalidade dos tipos de dados.

No comparativo de Pan, Zhao and Ren (2009), vide Tabela 3, verifica-se que o FaCT++ tem bom resultado comparativo em testes com ontologias já definidas e descritas no referido artigo. Este mesmo artigo apresenta o raciocinador REL, que obteve os melhores resultados, mas que em momento algum é disponibilizado para avaliação por terceiros.

Tabela 3: Comparativo Hermit/FaCT++/RacerPro/Pellet/REL. Fonte: Pan, Zhao and Ren, 2009.

Ontology	Hermit	FaCT++	RacerPro	Pellet	REL
Wine	1,297	0,829	19,469	1,719	0,579
Cyc	3,609	1,329	56,047	99,672	5,016
Tambis	1,265	erro	2,126	erro	0,625
FMA	erro	erro	+5 min	erro	21,718
DLP	151,517	0,922	3,922	4,735	0,594

5.5.3 HerMiT

O HerMiT (HerMiT, 2010) é um raciocinador relativamente novo que encontra-se na versão 1.2.3, sendo construído com a linguagem Java versão 5 ou superior e respeitando a licença de *software* livre GNU LGPL, o que permite seu uso tanto em *software* livres como em códigos proprietários e fechados. As interfaces atualmente disponíveis são a própria API, o Protégé (Protégé, 2010) e a OWL API versão 3 (OWL API).

A base é um algoritmo denominado *hypertableau* (Motik, Shearer and Horrocks, 2010) que é uma variação do algoritmo *tableau*. Este algoritmo tem apresentado resultados muito superiores a outros algoritmos e abordagens, reduzindo em segundos a classificação de ontologias que em outros raciocinadores levava minutos e até horas. Em Shearer, Motik and Horrocks (2008a), vide Tabela 4, um comparativo entre várias ontologias aplicadas a raciocinadores conceituados apresenta grande vantagem ao HerMiT.

Tabela 4: Comparativo HerMiT/Pellet/FaCT++. Fonte: Shearer, Motik and Horrocks, 2008.

Ontologia	Tempo de classificação em segundos			
	HerMiT	HerMiT-Anc	Pellet	FaCT++
Fly Taxonomy	1,1	1,2	1,2	5,3
COTermDB	1,6	1,8	364	19,2
Biological Process	24	1,6	107	79,2
NCI	28	3,7	17	30,2
MGED	57	11,2	0,8	0,29
BXPPOBOL	87	85	505,1	1742,3
OWL Guide Food	19,3	29,6	142	1388,1
FMA Lite	43,8	erro	erro	erro
DLPExtDis	95,8	erro	7,1	0,1
FMA-constitutional part	erro	erro	erro	erro
GALEN-horrocks	1,5	1,5	135	156,9
Nt-GALEN	1,6	1,8	54,1	20,4
GALEN-doctored	3,9	4,9	erro	2836,1
GALEN-original	11,9	erro	erro	erro
GALEN-module1	erro	erro	erro	erro
GALEN-full	erro	erro	erro	erro

Em relação a Tabela 4 é importante verificar que outros autores e artigos possuem valores conflitantes em relação ao apresentado nesta ilustração e que os detalhes dos testes e ontologias encontra-se no artigo referenciado. Porém, além da performance, que pode ser

questionada, o Hermit apresenta a seu favor a conformidade com toda a expressividade da OWL 2.

5.6 Raciocinadores e SWRL

Dos raciocinadores avaliados, o trabalho descartou o FaCT++ em virtude da linguagem de implementação C++ ser diferente da linguagem Java utilizada na plataforma desenvolvida. Tal diferença pode ser transposta em futuros trabalhos.

Entre o Hermit e o Pellet, encontrou-se um empasse, pois se o Hermit contempla toda a OWL 2, este ainda não suporta a SWRL por completo. Especificamente para o teste desenvolvido, o Hermit não permite o uso de *Built-Ins* na SWRL. Em contrapartida, o Pellet atende praticamente toda a expressividade e especificação proposta para a SWRL.

Meech (2010a) produziu um comparativo direto entre o Pellet e o Hermit, verificando a situação dos Raciocinadores em relação a SWRL e esclarece os inúmeros pontos a serem ainda estudados para adequar as atuais implementações de raciocinadores a todo o potencial proposto para a SWRL. O resumo encontra-se na Tabela 5.

Tabela 5: Raciocinador e SWRL. Fonte: Meech, 2010.

Raciocinador	Pellet 2.0.0-RC7	Hermit 1.0
OWL		
Nº de Testes executados	347	347
Nº de Testes aprovados	346	347
SWRL		
Nº de Testes executados	38	38
Nº de Testes aprovados	36	11

5.7 Considerações

Os Raciocinadores para ontologias que atualmente são implementados para o desenvolvimento de aplicações reais, ainda não apresentam total capacidade de suportar ontologias construídas com toda a expressividade da OWL 2 unidas com a capacidade de executar regras no modelo homogêneo sugerido pelo consórcio W3C para a SWRL.

Em relação ao modelo, homogêneo ou híbrido, ainda é necessário explorar e verificar qual abordagem de integração entre a OWL e as regras é mais interessante e eficaz.

Provavelmente a própria SWRL necessita sofrer atualização para se ajustar a OWL 2, pois sua especificação é anterior a especificação da mesma.

Porém para o presente trabalho, os raciocinadores Pellet e HermiT estão adequados para serem incorporados a plataforma desenvolvida e atenderem a aplicação que valida a real capacidade de estender com semântica o Mundo Real através dos Mundos Virtuais permitindo o raciocínio sobre as ontologias.

6 CENÁRIOS PARA EXTENSÃO DO MUNDO REAL

A fim de construir e visualizar situações possíveis e úteis para estender o Mundo Real através do uso de Mundos Virtuais 3D foi realizado um exercício mental, desconsiderando condições atuais de *hardware* e de estruturas e prevendo que estas serão resolvidas no futuro.

Os cenários baseiam-se na existência de um Mundo Virtual 3D capaz de compreender semanticamente o que se passa no seu interior e através de inferências construir novos conhecimentos, propor alternativas aos avatares, sugerir caminhos diversos e demonstrar ocorrências e fatos não percebidos pelos avatares e seus usuários.

6.1 Cenário Hospitalar

Uma aplicação imaginada seria baseada na extensão das atividades e tarefas realizadas em um Hospital, produzindo um Mundo Virtual 3D que estenderia uma ala, setor ou mesmo todo o Hospital. Seria possível adaptar e criar serviços, como uma sala para saúde preventiva, sala de esclarecimentos sobre epidemias e central de atendimento, onde o público não necessitaria realizar o deslocamento físico ao Hospital, mas manteria o contato pessoal e profissional. Em alguns casos, recursos audiovisuais poderão ser suficientes para a resolução das questões.

Funcionários do Hospital, utilizarão o Mundo Virtual 3D para realizar suas tarefas, incluindo atendentes, enfermeiros, médicos e pessoal administrativo, usando um único ponto de registro de informação. Laudos, baixas, altas, tratamento, alterações nos quadros clínicos e troca de informações serão registradas no sistema do Mundo Virtual 3D.

O paciente possuirá um avatar, conectado automaticamente a seu histórico de saúde permitindo inclusive o uso de algum mecanismo padronizado, como o *Google Health* (GHealth, 2009) e ou *Microsoft HealthValue* (MHealth, 2009), por exemplo. Visitantes

poderão entrar em contato com os pacientes através de avatares devidamente autorizados pelo sistema que também controlará o acesso a alas e setores virtuais. Recursos audiovisuais estariam liberados sobre rígido controle, porém, situações que hoje são de risco, como visitas a pacientes com doenças contagiosas seriam possíveis e salutares através do Mundo Virtual 3D.

Equipamentos para monitoramento de pacientes alimentariam diretamente o Mundo Virtual e o médico ao entrar em contato com o paciente, mesmo que de outra localização geográfica, terá acesso instantâneo a dados do paciente, conversar por teleconferência com ele, verificar sua imagem física, consultar enfermeiros, observar variações históricas nos monitores e no tratamento. E naturalmente poderá corrigir distorções no tratamento e solicitar novos exames.

Situações inversas poderão ocorrer, como no caso onde pacientes em situações graves de saúde mas que recebem autorização para tratamento e internamento em sua própria residência, manteriam-se virtualmente sobre controle do Hospital e seus médicos.

Enfim, o Mundo Virtual permitirá que remotamente o médico e o paciente estejam conectados e sejam capazes de trocar informações úteis para a melhoria da saúde do paciente.

Mesmo pesquisas e estudos sobre comportamento de pacientes e reações a medicações estarão sendo registradas e poderão frequentemente ser reavaliadas, agregando as percepções do paciente, seu humor, seu estado emocional e físico. Com estas informações, médicos e enfermeiros poderão potencializar o tratamento.

Ainda sobre o ponto de vista administrativo, o Mundo Virtual 3D do Hospital pode favorecer a customização de processos e procedimentos, pelo fato de registrar atividades dos avatares, retrabalho e duplicidade de ações tudo de forma integrada.

6.2 Cenário Educação à Distância

A utilização de Mundos Virtuais 3D como ferramenta de educação à distância tem recebido diversas pesquisas e abordagens, desde adaptações de modelos de aprendizado já existentes até a proposta de novos modelos de aprendizados melhor adequados a esta ferramenta. Mas indiferente a proposta metodológica, o que este trabalho propõe é a captação

de ações e interlocuções que o aprendiz realiza, seja com outros aprendizes ou seja com os professores que coordenam o aprendizado.

Supondo que o aprendizado de determinado conteúdo esteja subdividido em módulos consecutivos e que cada módulo possui uma série de atividades capazes de capacitar o aluno a avançar ao módulo seguinte, toda a informação que o aluno traz consigo é percebida pelo sistema que pode sugerir abordagens diferentes sobre o mesmo aprendizado. Em uma aula de língua estrangeira, as palavras podem estar disponíveis com sua sonoridade. O aluno pode conversar com um colega e ao não entender determinados sons, solicitar a escrita e novamente escutar a pronúncia.

Em uma aula sobre anatomia humana, alunos poderiam caminhar internamente ao corpo humano, escutar sons internos, como o do pulmão, verificar a simulação dos movimentos do coração e ou do aparelho motor. A simulação seria possível em muitos aspectos, desde a proposta pelo ambiente de pacientes com alguns problemas em seus órgãos internos onde os alunos poderiam verificar os sons característicos e os movimentos falhos, buscando diagnosticar corretamente o problema.

Estudos de Física e Matemática poderiam valer-se destes mesmos recursos de simulação, verificação e prova para estimular o aluno ao aprendizado, com o ambiente controlando e observando onde encontram-se os maiores problemas e a forma com que melhor o aprendiz responde aos desafios.

Ou seja, além de fornecer ao aluno e ao professor uma ferramenta que permita viver o aprendizado, o sistema também estará captando semanticamente o que ocorre, permitindo que o mesmo sugira abordagens diferentes para o mesmo aprendizado, baseado nas dúvidas e questões apresentadas pelos alunos ao professor, ou realizada apenas entre alunos. As dúvidas podem inclusive terem sido feitas e respondidas por outras turmas e outros professores, que deixaram suas informações registradas no ambiente. Enfim, uma espécie de tutor autômato e personalizado pode se apresentar ao aluno, oferecendo abordagens agradáveis ao aluno e que busquem suprir carências educacionais percebidas pelo sistema.

6.3 Cenário Penitenciário

Imaginemos a situação de que um determinado presídio está mapeado em um Mundo Virtual 3D e que cada indivíduo que entra no local real é reconhecido por sensores que o identificam de forma única. O Mundo Virtual recria os avatares validando as informações dos sensores e visualmente podemos perceber os agrupamentos, seja de detentos, seja de carcereiros, visitantes, advogados, etc. O sistema monitora as ações e pode prever porcentualmente a chance de uma rebelião, fuga ou agressão aos detentos.

Visitas a detentos podem acontecer também virtualmente, passando por autorização prévia do controle administrativo e todos os recursos utilizados podem ser monitorados e controlados. Da mesma forma, o judiciário pode efetivar audiências e verificações com os detentos sem a necessidade de locomoção real, mas garantindo todas as funcionalidades existentes no Mundo Real, como visão frente a frente, reações humanas, etc.

Sobre esta perspectiva, ao mesmo tempo que as instituições carcerárias e judiciárias ganham transparência por permitir uma visão de seu ambiente, também aumentam a segurança e o controle do que acontece nele.

6.4 Cenário para desenvolvimento de *software*

A área de Engenharia de Software estuda e proporciona as mais diversas abordagens no desenvolvimento de *software*, permitindo a aplicação de metodologias tradicionais, onde os artefatos são obrigatórias e o ciclo é bastante planejado até metodologias mais ágeis e flexíveis. A complexidade das metodologias requerer apoio de ferramentas de *software* para auxílio no desenvolvimento dos artefatos e recursos, e neste aspecto, as metodologias mais tradicionais costumam prover seus desenvolvedores adequadamente. Porém as metodologias ágeis mudaram o foco do desenvolvimento, dos tradicionais artefatos e cronogramas rígidos, para as pessoas envolvidas, a flexibilização e a tolerância a falha. Estas metodologias ao invés de formalizar o processo buscam organizar a informalidade, fazendo com que as pessoas envolvidas absorvam os conhecimentos e sirvam sempre de base para as decisões e mudanças.

No contexto das metodologias ágeis, praticamente todas exigem reuniões diárias entre os desenvolvedores do projeto, que caracterizam-se por serem rápidas (15 a 30 minutos) e

informais. O comum são anotações em quadros brancos e folhas de papel. O código é observado e definições são tomadas sem a preocupação de registrá-las, sempre acreditando que as pessoas da reunião vão levar adiante este compromisso.

Diversos casos de desenvolvimento de *software* estão obtendo êxito com estas abordagens ágeis de desenvolvimento, mas alguns pontos tem ficado em aberto e o Mundo Virtual 3D poderá agregar registro informal ao processo. Por exemplo, o desenvolvimento com equipes remotas fica prejudicado, o *product owner* (principal interessado no sistema) precisa frequentemente participar de reuniões e acompanhar o andamento do projeto, deslocando-se do seu ambiente de trabalho para o ambiente de desenvolvimento. Com o uso dos Mundos Virtuais 3D, todos os recursos de áudio e vídeo estão disponíveis trazendo a sensação de presença real. Dependendo da plataforma, as ferramentas de desenvolvimento de *software* podem ser incorporadas ao ambiente e utilizadas pela equipe para melhorar o entendimento e sanar dúvidas, como demonstrado na Figura 19. E o processo apesar de continuar informal, como a metodologia exige, passa a ser digitalizado automaticamente, garantindo um registro importante da atividade.

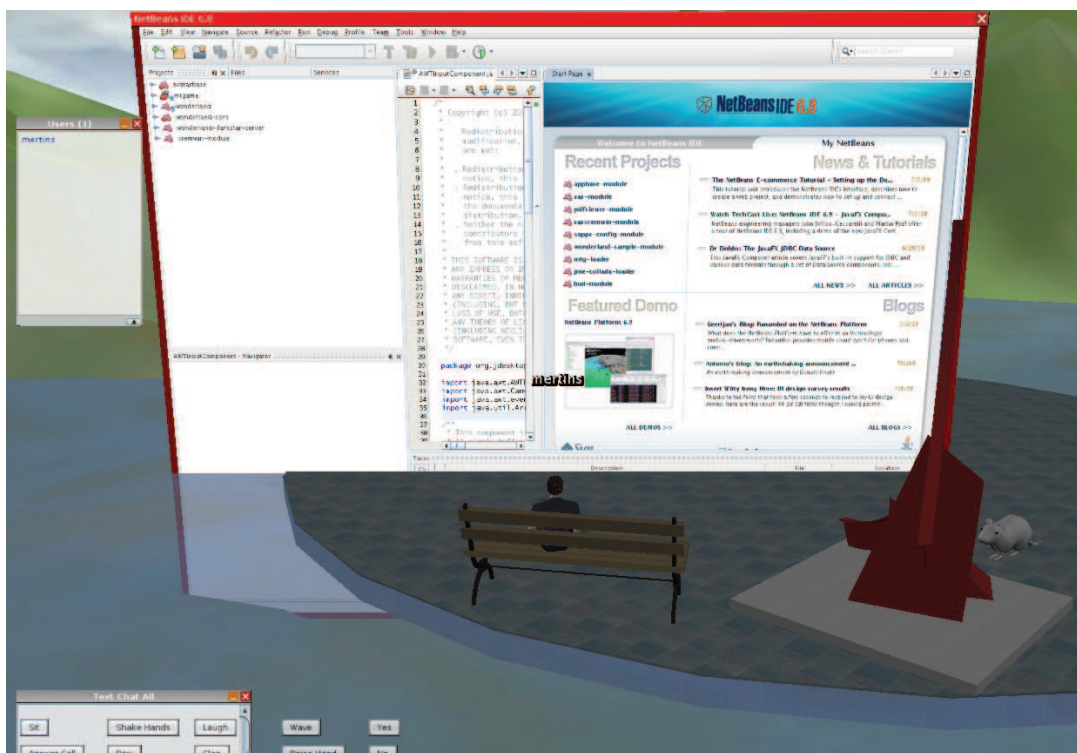


Figura 19: Uso da ferramenta de desenvolvimento Netbeans 6.8 no Mundo Virtual 3D

Desta forma ambiente de Mundo Virtual pode assessorar as reuniões e o envolvimento dos desenvolvedores, sugerindo soluções, trazendo informação já discutidas e relembrando

decisões, como se fosse a gravação de um vídeo, mas com uma capacidade maior de interpretar semanticamente o ocorrido.

6.5 Considerações

É muito importante ressaltar que os cenários anteriormente propostos não relatam preocupações com as implicações legais, morais e éticas e nem mesmo com os custos para viabilização real da aplicação. Pois, embora considera-se possível a implementação em laboratório, também considera-se necessário uma avaliação real de experimentos por parte das áreas envolvidas, para que possam validar as soluções em relação a problemas já vislumbrados e a prováveis novos problemas dentro da sua área de atuação.

7 ASPECTOS DA BASE DE CONHECIMENTO PARA MUNDOS VIRTUAIS

Se o Mundo Virtual 3D passar a ser uma enorme base de dados interligada e com semântica suficiente para que técnicas de Descoberta de Conhecimento possam ser combinadas, praticamente todas as técnicas de Mineração de Dados e Ferramentas de Descoberta de Conhecimento (Goebel and Gruenwald, 1999), podem ser aplicadas as informações existentes nas ontologias registradas e armazenadas no Mundo Virtual 3D. Sua estrutura permite o uso desde algoritmos simples como a análise de distribuição e regras de associação até classificação, indução de regras e agrupamentos.

Séries temporais (Keogh and Kasetty, 2002) podem perceber situações repetitivas, como no caso hospitalar existir pacientes com sintomas que já ocorreram em pacientes anteriores e informar médicos e enfermeiros que padrões de ocorrência estão se repetindo. Além de notificar médicos e enfermeiros, o Mundo Virtual pode, se autorizado, transformar estas informações em aprendizado de máquina e executar ações baseado no que foi realizado no passado para solucionar o problema. Desta forma o sistema passa a aprender com o passado e a ser proativo com os pacientes, médicos, enfermeiros e visitantes que em algum momento entram no Mundo Virtual e deixam registrado suas informações.

Sistemas e técnicas de recomendação podem utilizar as informações coletadas nas ontologias para sugerir opções mais interessantes e ainda não vislumbrada pelos usuários.

Técnicas de Mineração de Textos seriam necessárias para perceber informações trocadas por mensagens no Mundo Virtual 3D, mas estas técnicas poderão se beneficiar de toda a semântica existente no processo, pois todas as ações e situações estão registradas nas ontologias que realizaram a conversa textual e no ambiente que estava acolhendo os avatares. Esta é uma situação que Baeza-Yates, Mika e Zaragoza (2008b) indicam ser necessária na

evolução da *Web 2.0* e da *Web Semântica*, demonstrando que os Mundos Virtuais precisam se beneficiar dos conhecimentos já adquiridos para não incorrer nos mesmos erros e dificuldades.

Se o Mundo Virtual 3D se utilizar de recursos de reconhecimento de áudio e vídeo para identificação de pessoas através de sua voz e feições, será possível utilizar também técnicas de mineração de áudio e vídeo para perceber por exemplo, mudanças fisiológicas, mentais e comportamentais nos pacientes e tentar encontrar padrões e associações que possam explicar a situação. Tudo sempre conectado com vasta informação semântica.

A **inteligência coletiva** (Lévy, 1998) e a **sabedoria das massas** (Surowiecki, 2004) também poderão ser uma das abordagens que o Mundo Virtual utilizará, pois ao acumular diariamente informação de tudo o que acontece, permitirão aos sistemas agirem baseados no volume expressivo de avatares que efetuou um determinado procedimento sempre da mesma forma.

Propostas de adaptação e monitoramento do indivíduo baseada em trilhas (Silva et al, 2009) são perfeitamente possíveis de serem convertidas e utilizadas no Mundo Virtual, como uma forma de reconhecer os interesses e objetivos de todos os avatares. O mecanismo de trilha permite verificar o contexto do usuário e seus interesses baseando-se em *logs* de diversas aplicações, informando que tipo de equipamento, que tipo de conexão e que áreas de interesse estão sendo exploradas. Rebatendo esta perspectiva para os Mundos Virtuais 3D, tudo fica mais fácil e natural, pois o meio digital único pode permitir inclusive rever ações passadas no Mundo Virtual 3D, como se fosse um filme.

Enfim, diversas ciências e áreas poderão explorar o Mundo Virtual 3D e obter resultados positivos da massa de informação armazenada e com semântica, fazendo com que realmente o Mundo Virtual 3D seja uma extensão útil e necessária do Mundo Real.

Mas para isto é necessário a elaboração de uma base de conhecimentos que possa refletir de forma o mais fidedigna possível as entidades e o ambiente em que ela esta inserida. Esta representação é fechada em um escopo definido, pois as limitações computacionais atuais não permitem que os sistemas acompanhem o complexo mecanismo de mapeamento e relacionamento que o cérebro humano permite.

7.1 Ontologias

O modelo para representar esta base de conhecimento esta construído sobre o conceito e a padronização OWL v2.0 (OWL 2, 2010) elaborada pelo W3C (W3C, 2009) sobre ontologias, que permite agregar à representação do domínio a possibilidade de proatividade, com a construção de regras de inferência, tanto para ações diretas com o usuário, quanto para realimentar a base com novos fatos.

A proposta é que os sistemas do mundo real e os sistemas do mundo virtual contextualizariam o usuário dentro de um mesmo perfil, unificando em uma única forma as diversas possibilidades e variantes. Esta integração precisa utilizar a definição de uma ou mais ontologias, que permitiram aos sistemas efetivar as ações, percepções e experiências do usuário tanto no mundo real, como nos mundos virtuais, de forma unificada.

O uso de ontologias padronizadas favorece o reaproveitamento de modelos elaborados por outras pessoas e ou entidades. Na busca realizada por casos similares observou-se que a ontologia especificada nos termos da FOAF (FOAF, 2009) (Figura 20) pode servir de base inicial para a definição de um domínio interessado em compreender os avatares e seus usuários.

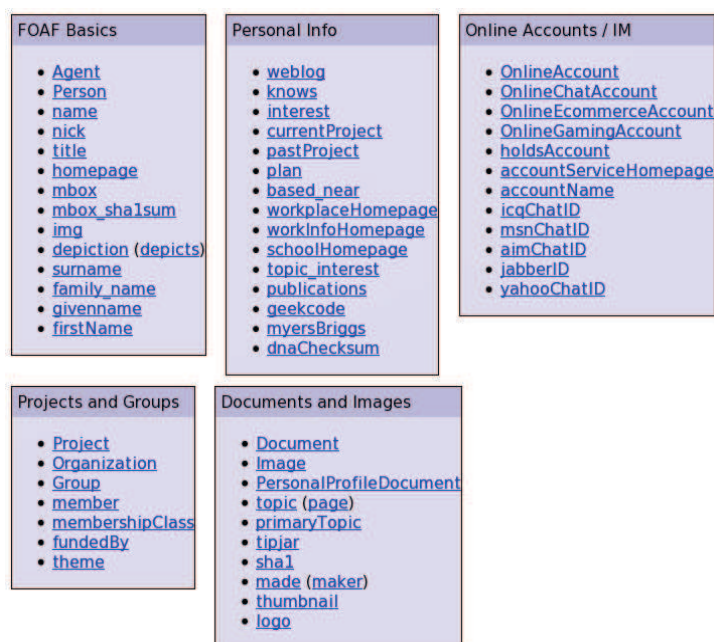


Figura 20: Termos FOAF agrupados por categorias

Ainda na busca por ontologias adequadas para o trabalho foram encontradas abordagens, que embora interessantes, não vão de encontro com os objetivos do trabalho, como por

exemplo o uso de ontologias durante a fase de modelagem na construção de um Ambiente Virtual (Vanacken et al, 2007), adicionando semântica ao processo e permitindo que ferramentas construam o Metaverso especificado, reduzindo a necessidade do desenvolvedor ser um especialista em Mundos Virtuais 3D e liberando a atenção do desenvolvedor ao escopo do ambiente em si.

Na mesma linha, diversas propostas tem sugerido o uso de ontologias na construção e manipulação de objetos no Mundo Virtual tanto na construção do ambiente (Martínez and Matta, 2006), quanto no momento de apresentação ao usuário, permitindo adaptar as formas aos dispositivos utilizados (Gutierrez et al, 2005).

A proposta de um canal padronizado de comunicação para os Ambientes Virtuais (Otto, 2005), permitindo que agentes trabalhem com informações sem a necessidade de conhecer o sistema de Mundo Virtual é a abordagem mais próxima e capaz de efetivar a expansão do Mundo Virtual sugerida neste trabalho. O processo de padronização segue a idéia de padronização semântica através de RDF e inclusive reaproveitando ontologias consolidadas, como a FOAF (Friend of a Friend).

O grupo de ontologias utilizada no *framework Seven* (Seven, 2009) apresenta a proposta de modelar os ambientes virtuais (Otto, 2005), mas avaliando as ontologias propostas, percebe-se seus diversos pontos em aberto e um foco maior apenas na representação dos objetos contidos no Mundo Virtual. Suas ontologias são descritas com RDF, uma tecnologia anterior a OWL 1 e segmentas de acordo com as classes de primeiro nível da Figura 21. Além destes fatos, é importante ressaltar que o projeto não recebe atualizações desde 2005.

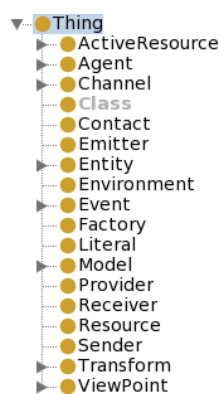


Figura 21: Classes (1º nível) utilizadas pelo Seven (Seven, 2009)

7.2 Agrupamento

Na proposta inicial do presente trabalho pretendia-se propor toda a definição de ontologias necessárias para descrever os Mundos Virtuais e o Mundo Real, mas com o transcorrer do mesmo verificou-se a enorme dificuldade em propor ontologias que definissem o Mundo como um todo, algo que diversas entidades e centros de pesquisas estão investindo esforços consideráveis, como por exemplo o DOLCE²³, OpenCyc²⁴ e SUMO²⁵. Seus trabalhos estão evoluindo e apresentando resultados, mas ainda estão longe de esgotarem as possibilidades.

Desta forma, optou-se por migrar a idéia inicial de construir toda a especificação para a ideia de agrupar as ontologias de nível superior em quatro grandes áreas, de forma a nortear futuros desenvolvimentos e implementações. As quatro grandes áreas ou ontologias de nível superior foram denominadas de: a) coisa (*thing*), b) ambiente (*space*), c) ator (*actor*) e d) ação (*action*). Portanto, o objetivo destas ontologias de nível superior é o de permitir o acoplamento de ontologias já consolidadas como a FOAF e outras, de forma organizada, além de definir claramente a área onde é necessário a construção de ontologias para atender tanto o estudo de caso apresentado, quanto futuras situações de extensão do Mundo Real através de Mundos Virtuais 3D.

7.2.1 Ontologia de Ambiente

A ontologia do ambiente registrará informações do Mundo Virtual em si, seus lugares, atmosfera, portais de acesso e seu relacionamento com as coisas.

7.2.2 Ontologia de Coisa

A ontologia que registra as coisas armazenará informações dos objetos e recursos existentes dentro do Mundo Virtual, como texturas, tamanhos, relacionamentos com outros objetos e avatares, comportamento dentro do ambiente, etc.

²³ <http://www.loa-cnr.it/DOLCE.html>

²⁴ <http://www.opencyc.org/>

²⁵ <http://www.ontologyportal.org>

7.2.3 Ontologia de Ator

A ontologia dos atores interligará usuários e avatares em um processo similar ao que ocorre com as redes sociais, inclusive permitindo que ontologias destes sistemas possam ser reaproveitadas. Jardim (2010b) apresenta em seu trabalho uma ontologia no domínio das Redes Sociais que pode ser acoplada a esta área para garantir interpretação semântica de redes sociais aos Mundos Virtuais integrando este a outras redes sociais, como Facebook²⁶ e Orkut²⁷.

7.2.4 Ontologia de Ação

Finalmente, a ontologia das ações irá registrar as ações que avatares realizam, tanto pessoais, como interpessoais (entre avatares) e entre objetos. As ações também poderão representar ações que objetos realizam por conta própria, seja através de agentes e ou mecanismos de Inteligência Artificial.

7.3 Raciocinadores

O modelo da plataforma proposta no próximo capítulo favorece o acoplamento de diversos raciocinadores capazes de interpretar a OWL 2. No protótipo em questão, foram testados o Hermit e o Pellet, onde, como já salientado, o primeiro atende a toda a expressividade da OWL 2, mas ainda não implementa os recursos de *build-ins* da SWR e o segundo implementa todos os recursos de *built-ins*, mas não toda a expressividade da OWL 2. Ambos prometem evoluir suas carências em futuras versões, que poderão facilmente serem acopladas ao atual sistema.

7.4 Otimizações na Plataforma para Mundos Virtuais 3D

Por fim, a utilização do *Open Wonderland* como a plataforma de Mundo Virtual 3D que irá suportar o experimento é a escolha deste trabalho. Dotada de recursos avançados, apesar de ainda estar em desenvolvimento, aliado a sua estrutura modular e a disponibilização de seus fontes através de um licença de *software* livre irão permitir o desenrolar deste trabalho

²⁶ <http://www.facebook.com>

²⁷ <http://www.orkut.com>

com as agregações dos componentes necessários. Além disto, esperasse que o resultado final possa ser incorporado aos fontes da plataforma em si, para que seu uso se propague a comunidade em geral.

7.5 Considerações

A situação idealizada inicialmente para este trabalho almejava a modelagem com suficiente precisão para definir os domínios de Mundos Virtuais 3D integrado-os ao Mundo Real, entretanto, no decorrer do estudo, verificou-se a impossibilidade deste feito no escopo no trabalho, por exigir entre outros fatores a) casos reais, b) um maior número de saberes, c) tempo de desenvolvimento e d) tempo de averiguação.

Sendo assim, o presente trabalho propõem a segmentação das ontologias nas 4 grandes áreas (coisa, ambiente, ação, ator) para que futuros experimentos e utilizações ligadas a aplicações práticas possam enquadrar suas necessidades de modelagem de domínios nestas subdivisões. Ou seja, a modelagem dos domínios a serem estendidos com os Mundos Virtuais é direcionada a seguir uma padronização conceitual que, na percepção deste autor, favorecerá o reaproveitamento e a reutilização de ontologias previamente construídas para estender o Mundo Real através dos Mundos Virtuais, porém em outros domínios e situações.

Dentro deste enfoque, o protótipo desenvolvido para provar que os Mundos Virtuais podem se integrar ao Mundo Real, implementa na Plataforma ORPlat uma ontologia na subárea de ator que se apresenta como suficiente para cumprir com os objetivos deste trabalho.

8 PLATAFORMA ORPLAT

Com o intuito de comprovar a viabilidade de integrar semanticamente os Mundos Virtuais com o Mundo Real o presente trabalho propõe uma plataforma modular, denominado ORPlat (Ontologia – Raciocínio - Plataforma), capaz de isolar através de interfaces o seu núcleo em relação aos Mundos Virtuais e ao Mundo Real. Tal separação permite inclusive a utilização do ORPlat para outros tipos de aplicações que não as plataformas de Mundos Virtuais 3D, como por exemplo sistemas *Web*, aplicações *desktop* e aplicações móveis.

8.1 Modelo

O planejamento e construção da plataforma ORPlat foi norteado pelo princípio de reduzir ao mínimo a necessidade de alterações na base dos sistemas de Mundos Virtuais. Ou seja, buscou-se uma clara separação e um bom nível de interface para que a plataforma seja o mais genérica possível no atendimento de sistemas de Mundos Virtuais 3D.

Percebeu-se também a necessidade de permitir que aplicações diferentes dos sistemas de Mundos Virtuais 3D consigam de alguma forma estabelecer comunicação com a plataforma ORPlat, favorecendo desta forma a integração entre o Mundo Real e os atuais sistemas que os usuários estão acostumados a utilizar, como interfaces *web* e móveis.

Assim a plataforma ORPlat foi planejada para ser o único sistema que tem acesso a Base de Conhecimento, servindo de intermediador para os sistemas que necessitarem registrar ou obter informações da base de conhecimento.

A comunicação entre os sistemas de Mundos Virtuais e a ORPlat ocorre através do fornecimento de serviços, conforme representado na Figura 22.

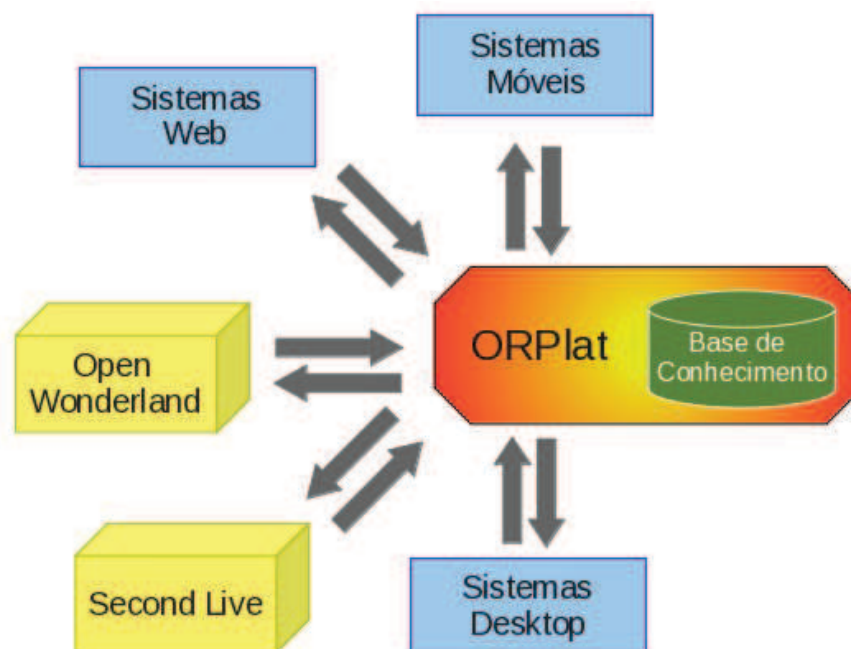


Figura 22: Modelo de integração entre aplicações, ORPlat e Bases de Conhecimento.

Os serviços fazem parte de um módulo específico na plataforma e podem ser customizados de acordo com a necessidade previa dos clientes.

8.2 Arquitetura

A arquitetura da plataforma foi projetada em quatro módulos, de forma a segmentar as necessidades do sistema, conforma a Figura 23.



Figura 23: Módulos da plataforma ORPlat

A segmentação em módulos ocorreu no intuito de permitir: a) a utilização de formas concorrentes de tecnologia e ou b) substituição por futuras abordagens que possam surgir. No

item a) temos por exemplo a possibilidade de utilizar mais de um raciocinador no módulo Raciocinadores e mais de uma forma de acessar a plataforma no módulo Serviços. Na opção b) podemos supor que a SWRL irá sofrer incrementos e ou testar outras abordagens de raciocínio sobre regras.

Com esta segmentação, o módulo de Persistência é atualmente o mais simples, pois utilizou uma abordagem tradicional, armazenando a ontologia em um Sistema Gerenciador de Banco de Dados como um único registro por área da ontologia. Fica em aberto para futuros trabalhos a sua exploração, na busca por melhores mecanismos de armazenamento e tratamento de ontologias.

O módulo de Ontologias, por sua vez, está adequado para receber as definições de ontologias de acordo com as quatro grandes áreas especificadas no capítulo anterior, a lembrar, ator, ação, ambiente e coisa. Atualmente este módulo está implementado para suportar apenas ontologias descritas com a OWL 2 e regras no modelo homogêneo com o uso da SWRL, mas sua expansão neste sentido pode ser viável e interessante.

Preparar o sistema para utilizar os raciocinadores existentes e estudados no capítulo 5 é função do módulo de Raciocinadores, que atualmente suporta a versão 2.2.2 do Pellet e a 1.3.2 do Hermit.

Por fim, o módulo de Serviços realiza a comunicação do sistema com outros sistemas. Sua responsabilidade está em definir as interfaces e seus requisitos. Este módulo, no atual nível de desenvolvimento da plataforma, suporta apenas comunicação através de RESTful *Web Services* (RESTful, 2011). O RESTful permite que praticamente qualquer tipo de aplicação moderna possa ser um cliente do ORPlat, bastando possuir conexão com a Internet.

8.2.1 Tecnologias

Além das tecnologias já mencionadas no trabalho e detalhadas nos primeiros capítulos por serem de importância relevante ao trabalho, algumas outras tecnologias precisaram ser incluídas na implementação da plataforma ORPlat, mas não mereceram a importância de serem citadas até este momento.

A primeira delas é a interface de programação de aplicações OWLAPI²⁸. Construída em Java, permite a manipulação completa da OWL 2 e realiza o interfaceamento entre diversos raciocinadores e em específico o Pellet e o Hermit. É um código de fontes livres que responde pela licença LGPL²⁹.

Para suportar a plataforma ORPlat foi adotado a tecnologia Java EE³⁰ versão 6.0, pois esta tecnologia fornece uma série de serviços e bibliotecas prontas, tudo de forma padronizada e uniforme. A Figura 24 demonstra o ORPlat instalado em um servidor de aplicações Java EE previamente instalado e configurado. O ganho com o uso deste tipo de servidor é a garantia de escalabilidade e independência, pois fornece serviços de *cluster* e balanceamento de carga, ao mesmo tempo que permite a escolha de vários fornecedores do sistema de servidor em si, inclusive o uso de servidores de código livre.

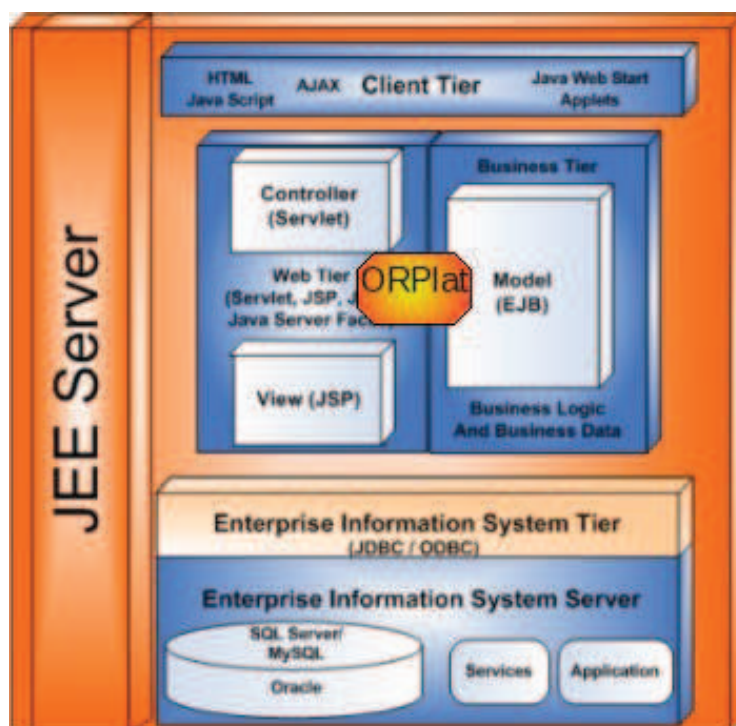


Figura 24: ORPlat implementado em um Servidor JEE. Imagem adaptada do site http://www.e-zest.net/distributed_application.html

Por fim o armazenamento em Sistemas Gerenciadores de Bancos de Dados está garantido através do uso da biblioteca Java *Persistence API*³¹ que abstrai o mapeamento

²⁸ <http://owlapi.sourceforge.net/>

²⁹ <http://www.gnu.org/licenses/lgpl.html>

³⁰ http://en.wikipedia.org/wiki/Java_EE

³¹ http://en.wikipedia.org/wiki/Java_Persistence_API

Objeto x Relacional permitindo independência, inclusive de fabricantes de Sistemas Gerenciadores de Bancos de Dados.

8.3 Considerações

A plataforma ORPlat foi projetada para permitir a interface entre os Sistemas de Mundos Virtuais, as Ontologias e os Raciocinadores. Sua abordagem procura evitar alterações drásticas e complexas nos códigos dos Sistemas de Mundos Virtuais, o que inviabilizaria na prática a proposta do trabalho.

Como os Sistemas de Mundos Virtuais não precisam ser desenvolvidos com a mesma linguagem de programação que a plataforma ORPlat, bastando apenas construir interfaces de comunicação comuns a ambos, estes não tem suas performances e seus requisitos de hardware afetados pela proposta. O ORPlat pode ser escalonado independentemente dos Sistemas de Mundos Virtuais e inclusive compartilhado por mais de uma aplicação, seja de sistemas de Mundos Virtuais, seja de outros tipos de sistemas. Um bônus a ser explorado no futuro.

O ORPlat precisa evoluir nos mecanismos de segurança de acesso e testado em relação ao consumo de recursos de *hardware* quando utilizado com ontologias maiores e mais complexas, mas para o presente trabalho a plataforma ORPlat apresenta-se adequada à execução do experimento, detalhado no próximo capítulo.

9 EXPERIMENTO

O experimento utilizado para comprovar a viabilidade de integrar semanticamente os Mundos Virtuais 3D com o Mundo Real e garantir mecanismos de inferência consiste na definição de uma ontologia simples de parentescos e disponibilizada sobre a ontologia de Ator.

Cada Avatar, no Mundo Virtual 3D, pode especificar outros Avatares como seus Pais (mãe e pai) e seu Cônjuge (marido/esposa). Estas manipulações na ontologia podem ocorrer através do próprio Mundo Virtual, em uma tela específica, ou através da comunicação direta ao ORPlat, o que permite o uso de praticamente qualquer sistema conectado em rede.

A ontologia possui regras de parentescos (item 9.2), como o que é um tio, um sobrinho e um avô, onde o ORPlat pode acionar seus raciocinadores para definir o parentesco entre os indivíduos.

Para o *Open Wonderland* foi desenvolvido um módulo que desenha uma base ou plataforma no solo (área com o logo da UCPel em frente ao avatar na Figura 25), sendo que o

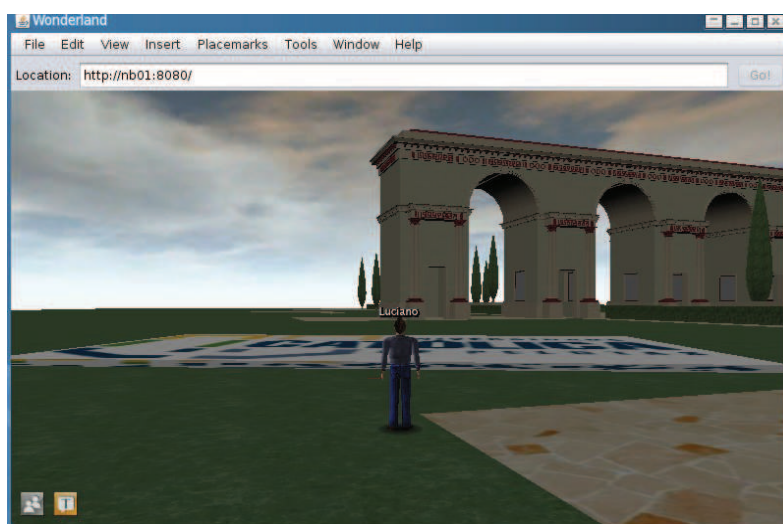


Figura 25: Base capaz de comunicar-se com o ORPlat

avatar que adicioná-la ao Mundo Virtual é automaticamente considerado o seu proprietário e diversas bases podem estar presentes no ambiente 3D.

Qualquer avatar que subir sobre a base, dispara internamente chamadas ao ORPlat que informarão os parentescos dos Avatares existentes sobre a base em relação ao proprietário da mesma, descartando informações consideradas óbvias, como a ligações diretas entre o visitante e o proprietário. A informação é apresentada especificamente para cada usuário mantendo uma visão distinta entre eles.

Portanto o experimento exigiu a construção de vários artefatos, que parte foram implementados no ORPlat e parte implementados no *Open Wonderland*, como descrito nos próximos itens.

9.1 Ontologia

Utilizando o Protégé 4, construiu-se a ontologia demonstrada na Figura 26, não sendo de forma alguma uma ontologia complexa, pois possui caráter específico para o experimento. Sua implementação se dá sobre a tutela da Ontologia dos Atores e pode ser combinada com outras ontologias através de associações, como por exemplo a de definir que Pessoa e Person da FOAF são conceitos equivalentes.

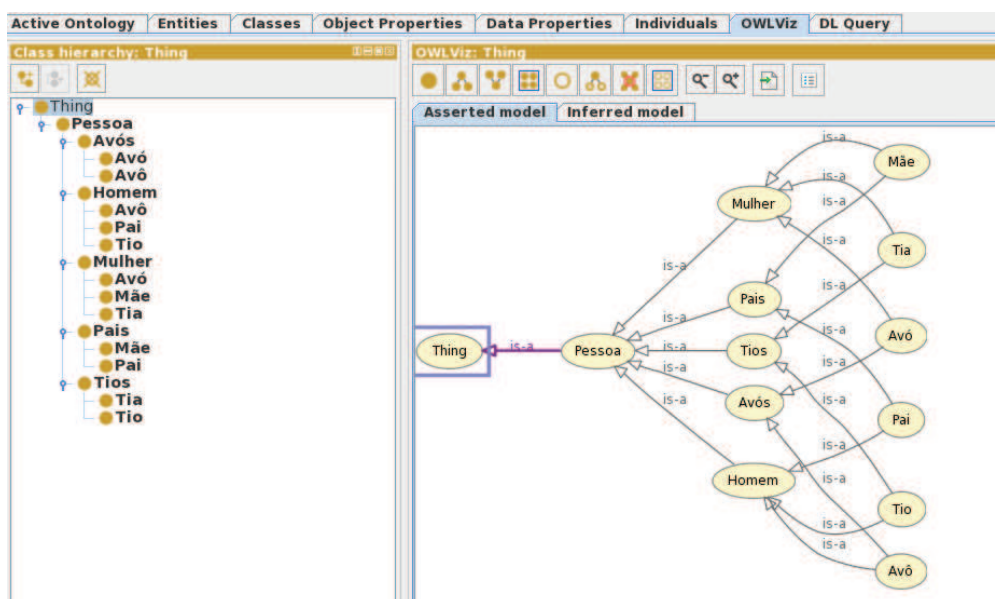


Figura 26: Ontologia de Ator definindo parentescos

Os conceitos pretendem ser auto explicativos por fazerem parte do cotidiano familiar e a ontologia completa com seus conceitos, propriedades e regras no formato XML encontra-se no anexo A.

Cada conceito de Pessoa na ontologia precisou receber a propriedade de dados de um “id” numérico e único, como demonstrado na Figura 27. Este identificador foi necessário para ser utilizado em conjunto com uma cláusula *Built-Ins* da SWRL, detalhada a frente.

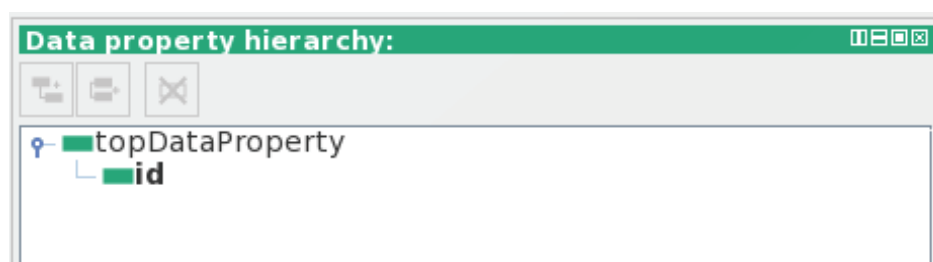


Figura 27: Propriedade de Dado

As propriedades de objetos são detalhadas na Figura 28 e definem as associações possíveis entre conceitos. Estas propriedades serão parte alimentadas pelas informações fornecidas pelos usuários, como “temMãe”, “temPai” e “temMatrimonio” e o restante sendo interligado pelos raciocinadores através de regras estabelecidas e detalhadas no próximo sub tópico.

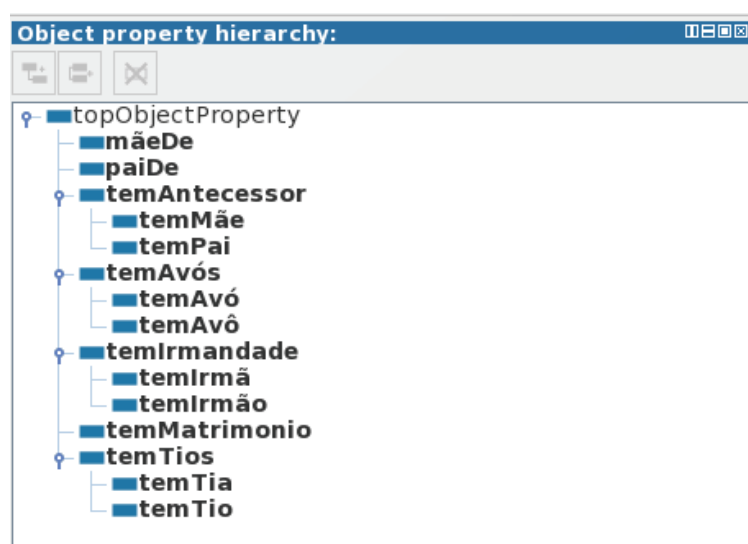


Figura 28: Propriedade de Objeto

9.2 Regras

As regras foram todas especificadas dentro da recomendação SWRL conforme Tabela 6.

Tabela 6: Cláusulas SWRL

temPai(?x, ?y) -> paiDe(?y, ?x)
Tia(?x) -> Tios(?x)
Avó(?x) -> Avós(?x)
Mulher(?w), Pessoa(?x), Pessoa(?y), temAntecessor(?x, ?y), temIrmã(?y, ?z), temMatrimonio(?z, ?w) -> temTia(?x, ?w)
temAvó(?x, ?y) -> Avó(?y)
Mulher(?y), Pessoa(?x), Pessoa(?z), temAntecessor(?x, ?z), temAntecessor(?y, ?z), id(?x, ?idx), id(?y, ?idy), notEqual(?idx, ?idy) -> temIrmã(?x, ?y)
Mulher(?x), temAntecessor(?y, ?x) -> Mãe(?x)
temTia(?x, ?y) -> Tia(?y)
temMãe(?x, ?y) -> mãeDe(?y, ?x)
Avó(?x) -> Avós(?x)
temTio(?x, ?y) -> Tio(?y)
Homem(?z), Pessoa(?x), Pessoa(?y), temAntecessor(?x, ?y), temIrmã(?y, ?z) -> temTio(?x, ?z)
Pai(?x) -> Pais(?x)
Mãe(?x) -> Pais(?x)
Homem(?y), Pessoa(?x), Pessoa(?z), temAntecessor(?x, ?z), temAntecessor(?y, ?z), id(?x, ?idx), id(?y, ?idy), notEqual(?idx, ?idy) -> temIrmão(?x, ?y)
Tio(?x) -> Tios(?x)
Mulher(?z), Pessoa(?x), Pessoa(?y), temAntecessor(?x, ?y), temIrmã(?y, ?z) -> temTia(?x, ?z)
Mulher(?z), temAntecessor(?x, ?y), temAntecessor(?y, ?z) -> temAvó(?x, ?z)
Homem(?x), temAntecessor(?y, ?x) -> Pai(?x)
Homem(?w), Pessoa(?x), Pessoa(?y), temAntecessor(?x, ?y), temIrmã(?y, ?z), temMatrimonio(?z, ?w) -> temTio(?x, ?w)
temAvó(?x, ?y) -> Avó(?y)
Homem(?z), temAntecessor(?x, ?y), temAntecessor(?y, ?z) -> temAvó(?x, ?z)

São regras simples, baseadas nos parentescos familiares utilizando apenas um *Built-Ins* da SWRL (*notEqual*), necessário para evitar a que o raciocinador acredite que um indivíduo pode ser irmão/irmã dele mesmo. O Anexo A apresenta as regras SWRL junto as definições OWL.

9.3 Serviços do ORPlat

Os atuais serviços disponibilizados pelo ORPlat para atender a necessidade do experimento utilizam *webservices* RESTful onde todas as respostas e chamadas são tratadas através de arquivos XML que respondem ao esquema XSD definido no Anexo B. Todos os serviços suportam o método GET, sendo que alguns também atendem ao método PUT, permitindo alterações na ontologia diretamente.

No Anexo C está a definição completa em WADL³² do serviços suportados pelo ORPlat e na Figura 29 encontra-se a visão geral dos serviços presentes no mesmo, sendo possível testa-los através da interface *web* disponível pela ferramenta de desenvolvimento Netbeans³³.

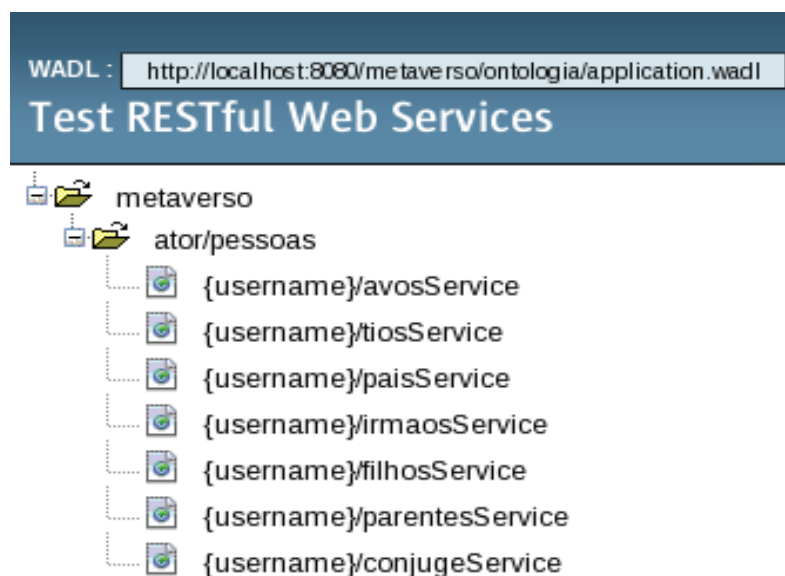


Figura 29: serviços RESTful

Lembrando que para os testes via navegador *web* e posterior integração com o Sistema de Mundo Virtual *Open Wonderland* foi necessário instalar o ORPlat em um servidor Glassfish³⁴ v3, que atende a especificação JEE 6.

Em relação aos serviços, para invocá-los, são necessárias a combinação de três componentes: a) recurso, b) método de chamada e c) conteúdo XML. O recurso é um *link* http, que por sua vez indica o servidor, a porta e a URL do serviço. O método de chamada, no ORPlat, aceita “GET”, para requisitar informações, ou “PUT”, para inserir informações. Por fim, o conteúdo xml, necessário apenas quando o método for “PUT”, especifica as informações a serem transmitidas através de *tags* XML

As alterações na Ontologia através da ação humana são realizadas apenas nos serviços que contempla o método “PUT”, sendo eles: ator/pessoas, ator/pessoas/{username}/paisService e ator/pessoas/{username}/conjugueService, onde {username} precisa ser substituído pela identificação do indivíduo no sistema de Mundo Virtual 3D. As

³² http://en.wikipedia.org/wiki/Web_Application_Description_Language

³³ [Http://www.netbeans.org](http://www.netbeans.org)

³⁴ <http://glassfish.java.net/>

demais informações que os serviços fornecem, através do “GET”, são processadas pelos raciocinadores.

9.3.1 Funcionamento dos *Webservices* RESTful

Dois serviços serão detalhados para garantir o entendimento geral do funcionamento da plataforma ORPlat e sua integração com os sistemas de Mundos Virtuais. Os demais serviços seguem a mesma linha de funcionamento destes dois serviços abordados.

9.3.1.1 Serviço ator/pessoas

O serviço “ator/pessoas” permite através do método GET verificar todas as pessoas que a ontologia conhece, bastando para isto invocar a URL adequada, como demonstrado na Figura 30 através de um navegador *web*.

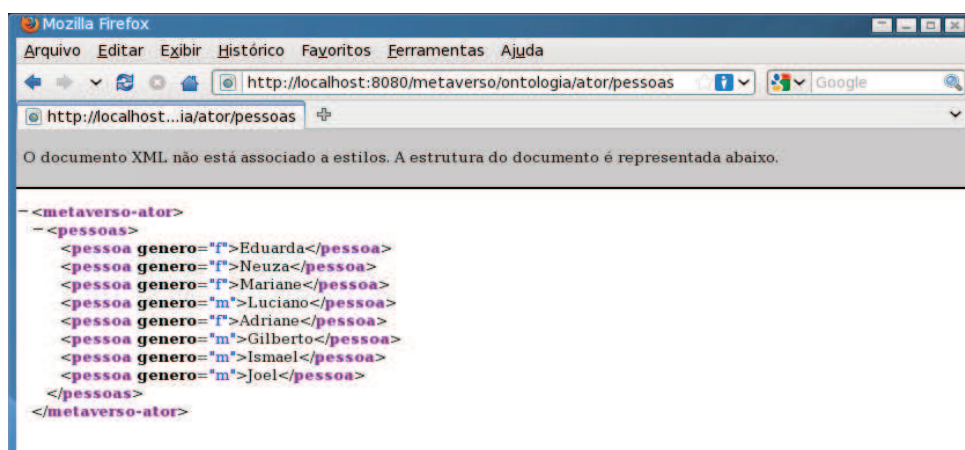


Figura 30: Requisição de Pessoas na Ontologia

O método “PUT”, por sua vez, permite adicionar instâncias ou de Homem ou de Mulher, conforme Figura 31, onde um indivíduo Homem é adicionado a ontologia através da *tag* <homem>. A definição de uma Mulher segue a mesma abordagem, apenas substituindo a *tag* <homem> pela *tag* <mulher> e seu conteúdo de acordo.

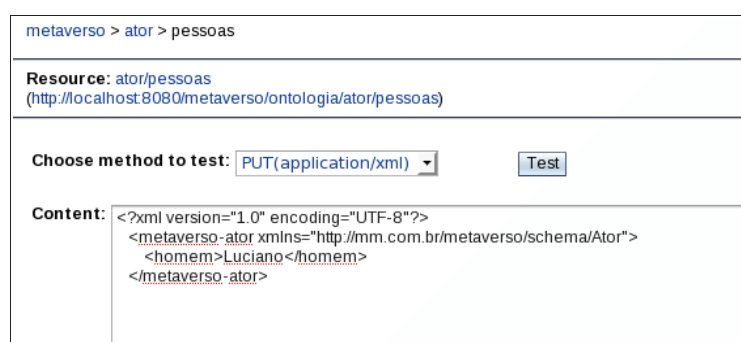


Figura 31: Adicionando uma instância ao conceito Homem

9.3.1.2 Serviço ator/pessoal/{username}/paisService

Os demais serviços atualmente presentes na plataforma ORPlat exigem, na definição do recurso, a especificação do indivíduo que servirá de referência a obtenção da informação ou a associação da mesma, substituindo a variável {username} e colocando a identificação diretamente no *link* do serviço. No caso da interfaces de apoio demonstrada (Figura 32), a informação digitada na caixa de texto *username* é incorporada ao *link* no *Resource*.

Figura 32: Associando Pais a um indivíduo

Portanto o serviço *paisService*, por ser um representante desta linha de serviços, precisa ter incorporado ao seu *link* de recurso a identificação do usuário, indiferente se o método a ser utilizado for o “PUT” ou “GET”.

No caso de uma chamada “PUT” para associar os Pais de um indivíduo a ontologia é necessário especificar um conteúdo XML que informará as instâncias correspondentes, conforme já apresentado na Figura 32, onde o XML especifica duas instâncias já existentes, “Edio” e “Naira”, para associar ao indivíduo “Luciano”. A interface *web* realizará a ação no momento que o usuário clicar no botão “Test”. A partir de então, uma chamada através de um navegador *web* para o *link* do serviço é capaz de informar os pais do indivíduo Luciano, sendo que esta informação deve constar na URL (Figura 33).

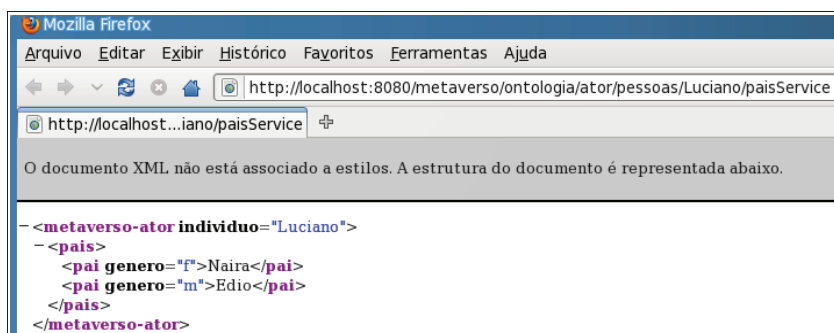


Figura 33: Pais de um indivíduo

9.3.2 Ontologia e Persistência

Todas as chamadas a qualquer serviço da plataforma ORPlat irão automaticamente controlar a persistência da ontologia carregando a ontologia do Gerenciador de Banco de Dados, alterando-a e gravando-a novamente.

Além disto, a representação da ontologia completa pode ser a qualquer momento invocada pela *web*, conforme o *link* <http://servidor:porta/metaverso/ontologia/owl?onto=ator>, onde o servidor e porta devem representar respectivamente, o *ip* e a porta do servidor onde esta executando o servidor JEE que contempla a plataforma ORPlat.

9.4 Módulo do *Open Wonderland*

O *Open Wonderland* é um sistema de Mundo Virtual 3D que prima pela modularidade de seus componentes e serviços. Esta frase ficou confirmada pelo fato que o desenvolvimento das funcionalidades que permitem acesso a plataforma ORPlat não exigiram em nenhum momento a modificação dos códigos principais do *Open Wonderland*, sendo tudo desenvolvido dentro de um módulo do mesmo e sua instalação ocorrendo de forma transparente ao sistema. Isto permite que qualquer Mundo Virtual 3D configurado em um servidor *Open Wonderland* aceite o módulo e o utilize adequadamente.

9.4.1 Instalação e configuração

A instalação do módulo segue o procedimento comum ao *Open Wonderland* de utilizar a sua interface *web* (maiores informações no site de documentação do sistema³⁵). Após a instalação do módulo é necessário a criação de um arquivo texto no sistema operacional do servidor *Open Wonderland*, contendo uma propriedade com a localização do ORPlat (Figura 34).

```
#URL Base para localizar os RestFul Webservice de Ontologias  
ontology.webservice.baseuri=http://10.1.1.50:8080/metaverso/ontologia
```

Figura 34: Arquivo de Propriedades do Módulo *Ontology*

³⁵ <http://code.google.com/p/openwonderland/wiki/OpenWonderland>

Em seguida uma variável (*ontology.webserver.fileconfig*) (Figura 35) no servidor *Darkstar* precisa indicar a localização do arquivo mencionado. Importante salientar a necessidade do Sistema Operacional permitir o acesso por parte do *Open Wonderland* ao referido arquivo.

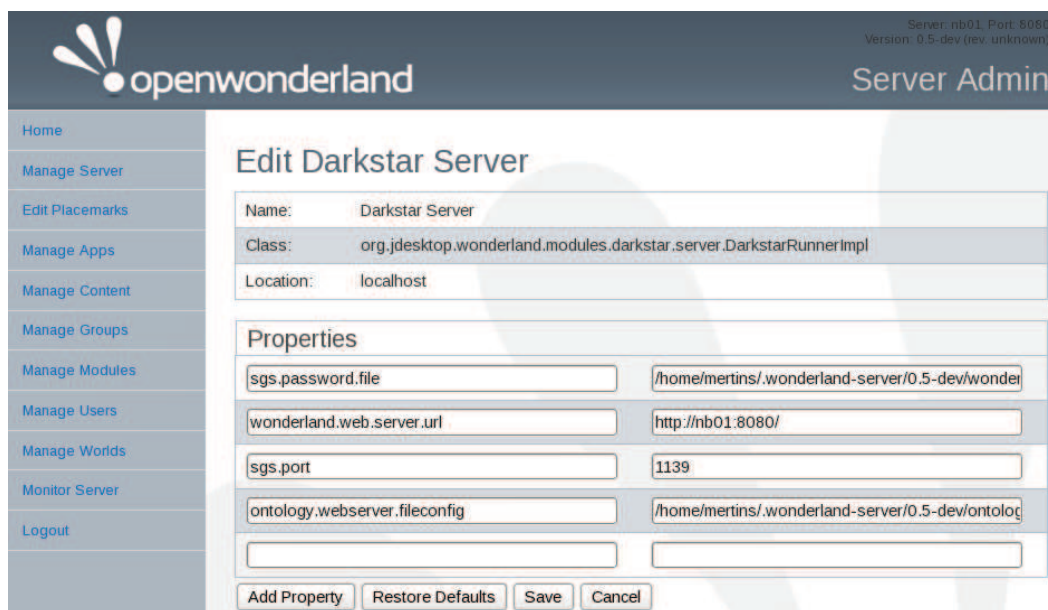


Figura 35: Tela de configuração do *Darkstar* no *Open Wonderland*

Após estas configurações, qualquer avatar no Mundo Virtual pode selecionar o objeto *ORPlat server* no seu menu (Figura 36) e inserir uma plataforma no ambiente. Automaticamente o sistema irá considerar aquele avatar como proprietário da base que se comunica com o *ORPlat*.

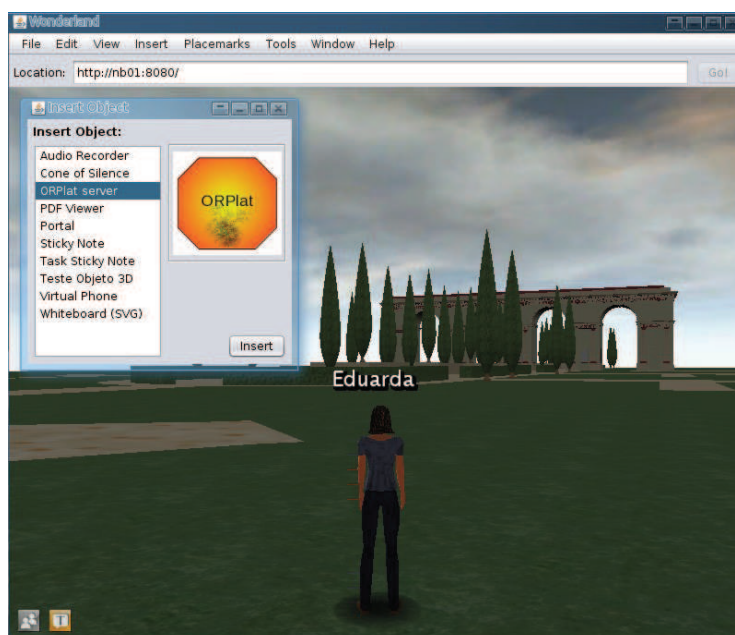


Figura 36: Adicionando a base que utilizará os recursos *ORPlat*

9.4.2 Comportamento

O módulo ORPlat para o *Open Wonderland* contempla códigos que realizam ações na camada cliente e na camada servidor. Esta abordagem busca interpretar as ações e movimentos no cliente e transmiti-las ao servidor que as recebe e as encaminha a plataforma ORPlat para avaliação e inferência.

Ao desenvolver um módulo no *Open Wonderland* é necessário ter a compreensão que diversas áreas são combinadas, pois em um módulo ficam armazenados os arquivos com as artes gráficas, os códigos que devem ser baixados no cliente, os códigos do servidor e os códigos comuns a ambos os lados. O controle sobre a troca de mensagens e os mecanismos de serviços precisam ser codificados e planejados para que todos os clientes tenham uma visão concisa e adequada ao seu posicionamento dentro do Mundo Virtual.

A Figura 37 representa como o módulo ORPlat atua em relação ao *Open Wonderland* e ao ORPlat. O componente *OntologyCellMO* é a representação de tudo que é necessário para a base existir no Mundo Virtual, sendo sua representação em cada cliente responsabilidade do componente *OntologyCell*. Para manter o sincronismo, ambos trocam mensagens de eventos que ocorrem sobre suas responsabilidades. Por exemplo, se um Avatar subir sobre a base, é responsabilidade da *OntologyCell* perceber e notificar de forma assíncrona a *OntologyCellMO*. Por sua vez este deve fazer uma requisição ao ORPlat e aguardar a resposta. Ao receber a resposta, deve enviar mensagens síncronas para seus clientes *OntologyCell* que precisaram se adequar a situação, provavelmente mostrando informações pertinentes.

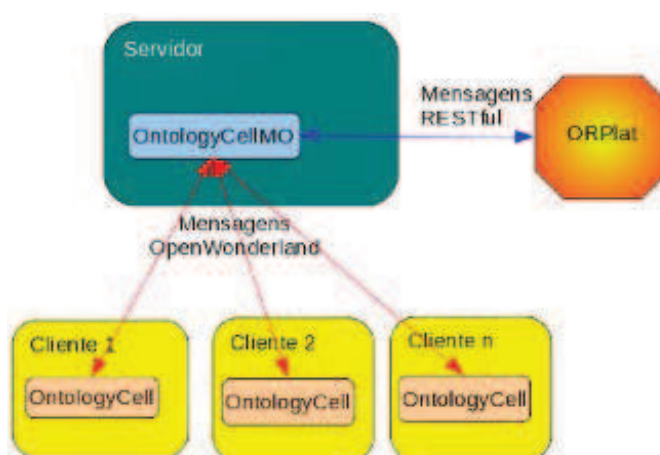


Figura 37: Arquitetura de comunicação entre o Módulo instalado no *Open Wonderland* e o ORPlat

9.5 Base de Conhecimento

Para a realização do experimento foi necessário a instanciação da ontologia previamente abordada, tendo como objetivo fornecer manualmente o menor conjunto de informações por indivíduo. Optou-se por fornecer apenas as informações de identificação no sistema, “id”, associação entre cônjuges, associação do Pai e associação da Mãe, inserido-as na ontologia através dos serviços do ORPlat de forma a compor o esquema demonstrado na Figura 38.

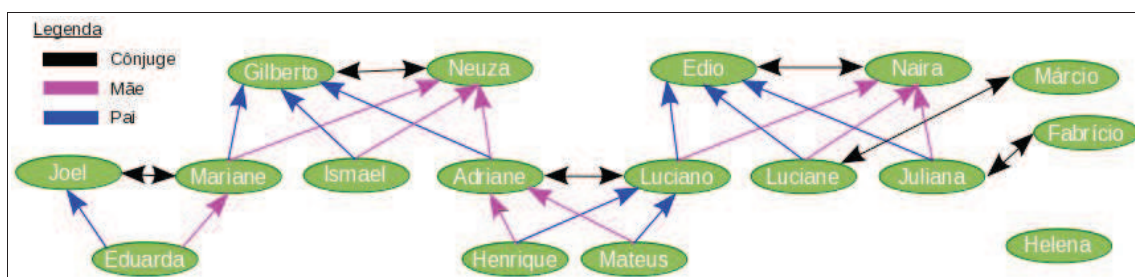


Figura 38: Instâncias na Ontologia

9.6 Resultados

O resultado visual do experimento pode ser conferido na Figura 39 que apresenta a tela de dois usuários distintos, lado a lado.

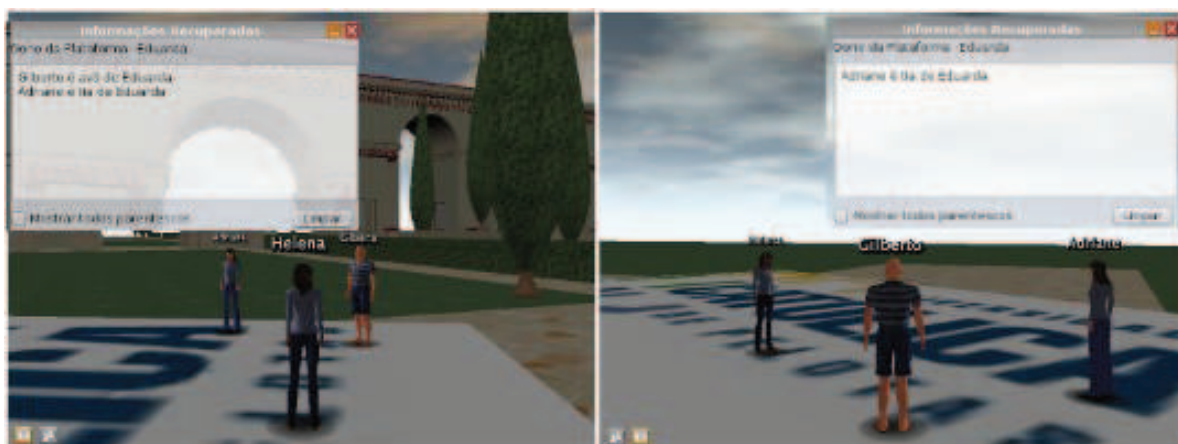


Figura 39: Avatares sobre a mesma base

A situação da Figura 39 representa o encontro dos avatares “Adriane”, “Helena” e “Gilberto” sobre a base de propriedade do avatar “Eduarda”. O cliente da esquerda é a tela do usuário que manipula “Helena” e o Mundo Virtual informa automaticamente que “Adriane” é tia de “Eduarda” e que “Gilberto” é avô da “Eduarda”.

Pelo lado direito da mesma Figura 39 tem-se a imagem apresentada ao usuário que utiliza o avatar “Gilberto” e nela constam apenas informações sobre “Adriane”, visto que o sistema não possui informações semânticas sobre “Helena”.

Desta forma o resultado final do experimento é a capacidade que o sistema de Mundo Virtual apresenta de identificar os avatares sobre a base e informar a cada usuário os graus de parentescos existentes entre eles e o proprietário da base. Mas sobre um contexto geral, viabilizou-se a possibilidade de Mundos Virtuais consultarem Bases de Conhecimento que inferiram novas informações, baseadas em regras previamente fornecidas. Estas novas informações foram apresentadas aos usuários, que decidiram sobre sua utilidade. Ressalta-se também que a plataforma ORPlat capacita os Mundos Virtuais a coletarem informações e as transportarem para as Bases de Conhecimento como sendo fatos, permitindo o crescimento da mesma.

9.7 Considerações

O experimento utilizou uma base no solo do Mundo Virtual para testar a comunicação com os serviços do ORPlat permitindo modificar a base de conhecimento e solicitar a inferência de informações que não foram inseridas manualmente na base. Mas verificou-se que é possível disparar este tipo de comunicação em diversos outros objetos e componentes do *Open Wonderland*, como por exemplo na entrada do Avatar no Mundo Virtual ou em um aceno para outros Avatares.

A abordagem de transferir a comunicação com o ORPlat do cliente para o servidor do *Open Wonderland* foi válida e adequada para este sistema de Mundo Virtual e parece ser possível de adapta-la a outros sistemas de Mundo Virtual como o *Second Life*, por exemplo, onde dois componentes da linguagem de script apresentam suporte a serviços RESTful (*LSL HTTP server*³⁶, *LIHTTPRequest*³⁷). Entretanto testes precisam ser realizados e não foram incorporados ao escopo deste trabalho.

A plataforma ORPlat permite evolução, seja em seus componentes internos com a adequação de novas tecnologias, seja pelos sistemas que a utilizam através do incremento da própria ontologia. Este incremento não se dá apenas com novas instâncias, como demonstrado no experimento, mas com novas definições de classes, associações e regras, bastando apenas o desenvolvimento de novos serviços na plataforma ORPlat.

³⁶ http://wiki.secondlife.com/wiki/LSL_HTTP_server

³⁷ <http://wiki.secondlife.com/wiki/LIHTTPRequest>

Portanto o experimento permitiu visualizar na prática a capacidade de comunicação entre os Mundos Virtuais e o Mundo Real através da plataforma ORPlat, com condições totais de manipular a Base de Conhecimento, inferir novas informações e transmiti-las aos avatares para que estes possam se beneficiar delas.

10 CONSIDERAÇÕES FINAIS

10.1 Conclusões

Este trabalho propôs e realizou adequadamente a integração semântica entre o Mundo Real e os Mundos Virtuais, permitindo inferência sobre a base de conhecimento construída, pois o experimento demonstrou a capacidade agregada ao Mundo Virtual de apresentar ao usuário informações obtidas através da aplicação de raciocinadores sobre uma base de conhecimento externa. O mesmo experimento permitiu a atribuição de informações sobre a base de conhecimento através de outros sistemas, o que permite diversas formas de comunicação.

Tal feito exigiu abordar e estudar duas áreas distintas do conhecimento, uma sobre sistemas de Mundos Virtuais 3D e outra sobre recursos da Inteligência Artificial para representar Bases de Conhecimento e mecanismos para inferir sobre as mesmas.

E para combinar estas duas áreas, foi proposto e desenvolvido uma plataforma, o ORPlat, que permite aos Mundos Virtuais 3D e outros sistemas computacionais trocarem informações com significado semântico. Esta plataforma permitiu atingir os objetivos propostos para este trabalho, combinando em um único sistema:

- tecnologias para receber e manipular ontologias especificadas com a OWL 2;
- raciocinadores capazes de atuarem sobre as referidas ontologias;
- integração com a Programação Lógica através da inclusão de regras SWRL;
- controle de persistência das modificações sobre as ontologias;
- interfaces abertas de comunicação para com outros sistemas.

O experimento demonstrou a facilidade de combinar os Sistemas de Mundos Virtuais 3D, representado no protótipo pelo *Open Wonderland*, com a plataforma ORPlat, agregando aos Mundos Virtuais 3D a capacidade de receberem e transmitirem informações com semântica. Esta semântica permitiu a inferência de novas informações, o que favoreceu aos Sistemas de Mundos Virtuais 3D atuarem de forma proativa, fornecendo aos usuários e seus avatares novas funcionalidades.

10.2 Contribuições

A proposta de integrar os Mundos Virtuais com o Mundo Real demonstra à comunidade científica a importância de agregar semântica ao processo evolutivo dos sistemas de Mundos Virtuais 3D. Estes sistemas tem grande potencial de uso em situações que exijam integração entre usuários de forma remota e síncrona, porém, ainda não utilizam todo o seu potencial computacional, por não compreenderem exatamente quais são os interesses dos avatares e seus usuários. Agregar esta compreensão por parte dos sistemas pode auxiliar as atividades de Ensino a Distância, os processos de simulação, o entendimento de atividades colaborativas e facilitar a exploração na área da descoberta do conhecimento, por exemplo.

O desenvolvimento, construção e implantação da plataforma ORPlat permite o estudo de diversas tecnologias em conjunto e que anteriormente estavam isoladas. Assim as ontologias, suas instâncias e regras de Programação em Lógica podem ser combinadas para serem utilizadas, não apenas em Sistemas de Mundos Virtuais 3D, mas também em Sistemas *Web* comuns, como redes sociais, *sites* de compras e ou ferramentas de ensino. Espera-se que a utilização da plataforma permita avaliações reais dos mecanismos atuais e que novas técnicas possam ser propostas e incorporadas a plataforma.

Além da integração semântica já citada acima, este trabalho também aponta para os benefícios que outras áreas de conhecimento terão, caso utilizem os sistemas de Mundos Virtuais 3D como ferramentas para atingir seus objetivos, explorando novas interfaces para atividades tradicionais. E neste contexto o *Open Wonderland* apresentou recursos e capacidade para atender estudos nas mais diversas áreas, seja em simulações de ambientes, auxílio no estudo a distância ou na integração de novas interfaces com o ser humano.

10.3 Trabalhos Futuros

O trabalho apresentou e executou uma proposta que permite a sua continuidade pela exploração de diversos outros ângulos e abordagens, onde em um primeiro momento, seria interessante comprovar a integração do *Second Life* e de outras plataformas de Mundos Virtuais 3D com a plataforma ORPlat, para certificar a generalização do trabalho. Tal tarefa, na opinião do autor, é de razoável facilidade.

Na área da Inteligência Artificial, aponta-se possibilidades de experimentos mais complexos que combinem as Lógica de Descrição e a Programação em Lógica, averiguando outras abordagens e comparando-as. Seria a combinação híbrida mais adequada do que a homogênea? Outros modelos de regras atenderiam satisfatoriamente? Quais tecnologias permitiriam transpor as limitações da SWRL? É possível evolui-la?

Sobre a ótica das ontologias para representar os domínios dos Mundos Virtuais, estas devem ser expandidas sobre as 4 áreas básicas propostas (ator, ambiente, ação, coisa) e integradas com outros modelos já disponíveis. O autor acredita que esta extensão deve estar vinculada ao uso do sistema de Mundo Virtual 3D como ferramenta para atender necessidades de outras áreas do conhecimento, distintas da Computação, que se beneficiariam do ambiente colaborativo e da capacidade de inferência incorporada.

A própria plataforma ORPlat foi desenvolvida com a pretensão de permitir diversos tipos de alterações, mas precisa comprovar sua suposta versalidade, onde novas ontologias poderão ser armazenadas, novas formas de tratar o armazenamento poderão ser estudadas, novos raciocinadores poderão ser avaliados e principalmente, outros sistemas poderão realizar experimentos sobre sua arquitetura.

Na linha dos Sistemas de Mundos Virtuais 3D, estes permitem uma variedade extensa de estudos como ferramenta a ser utilizada em diversas ciências como a da Saúde e a do Ensino a Distância (EAD), por exemplo. Na Saúde, o uso dos Mundos Virtuais 3D poderiam auxiliar pessoas com dificuldades motoras, integrando o uso de novas interfaces entre o computador e o ser humano. Por sua vez, a EAD talvez seja a área não ligada a computação que mais esteja utilizando os Mundos Virtuais como ferramenta para aprimorar suas atividades síncronas e assíncronas, focando bastante no uso do *Second Life*. Neste aspecto o autor considera que o

uso de plataformas abertas como o *Open Wonderland* expandem o domínio das propostas, pois removem qualquer tipo de restrição e controle por terceiros.

Enfim, a simples apresentação do *Open Wonderland* como uma competente plataforma de sistemas de Mundos Virtuais 3D pode instigar pesquisadores de outras áreas a avaliarem seu potencial como ferramenta para a resolução seus problemas. E o acréscimo da capacidade semântica à estes Mundos Virtuais proposta por este trabalho potencializa os resultados.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALLEN, J.F.; FRISCH, A. M., **What's in a semantic network?** Proceedings of the 20th annual meeting on Association for Computational Linguistics, June 16-18, 1982, Toronto, Ontario, Canada, 1982.
- ALSAÇ, G.; BARAL, C., **Reasoning in description logics using declarative logic programming.** Tech. Rep. CS&E Dept, Arizona State University, 2001.
- ANTONIOU, G.; DAMÁSIO, C. V.; GROSOFF, B.; HORROCKS, I.; KIFER, M.; MALUSZYSKI, J. and PETER, **Combining Rules and Ontologies. A survey.** Technical Report IST506779/Linköping/I3-D3/D/PU/a1, Linköping University, February 2005. IST-2004-506779 REWERSE Deliverable I3-D3. <http://rewerse.net/publications/>, 2005.
- ANTONIOU, G.; WAGNER, G., **Rules and Defeasible Reasoning on the Semantic Web.** Second International Workshop, pg 111-120, RuleML 2003, Sanibel Island, FL, USA, October 20, 2003.
- BAADER, F.; CALVANESE, D.; MCGUINNESS, D.; NARDI, D.; PATEL-SCHNEIDER, P., **The Description Logic Handbook: Theory, Implementation, Applications.** Cambridge University Press, Cambridge, UK, 2003.
- BAADER, F.; HORROCKS, I.; SATTLER, U., **Description Logics.** In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, Handbook of Knowledge Representation. Elsevier, 2007.
- BAEZA-YATES, R.; MIKA, P.; ZARAGOZA, H., **Search, Web 2.0, and the Semantic Web.** IEEE Intelligent Systems, Vol 23, Issue 1, p. 80-82, 2008.
- BALKIN, J. M.; NOVECK, B. S. (Edited), **The State of Play: Law, Games, And Virtual Worlds.** New York University Press, Nova York, 2006.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O., **The Semantic Web.** Scientific American, [S.1.], v.5, n.284, p.34-43, 2001.
- BRY, F.; SCHAFFERT, S., **An Entailment Relation for Reasoning on the Web.** Second International Workshop, pg 17-34, RuleML 2003, Sanibel Island, FL, USA, October 20, 2003.

CALVANESE, D.; De GIACOMO, G., **Description Logics for Conceptual Data Modeling in UML**. Disponível em: <<http://ftp.dis.uniroma1.it/~degiacom/didattica/esslli03/>>. Acesso em: 20/05/2010. 2003.

CARDOZO, M. L., **Propaganda Pessoal: Redes Sociais na Internet**. Intercom - Sociedade Brasileira de Estudos Interdisciplinares da Comunicação, p 1-13. XXXI Congresso Brasileiro de Ciências da Comunicação - Natal, RN, 2008.

DAYANTIS, G., **Logic program derivation for a class of first order logic relations**. In Proceedings of the 10th international joint conference on Artificial intelligence - Volume 1 (IJCAI'87), Vol. 1. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 9-14., 1987.

DIG 2, **The new DIG interface standard (DIG 2.0)** Disponível em: <<http://dl.kr.org/dig/interface.html>>. Acesso em: 18/06/2010. 2010.

DRABENT, W., **Hybrid Reasoning with Non-monotonic Rules**. Reasoning Web. Semantic Technologies for Software Engineering, 6th International Summer School 2010, Dresden, Germany, August 30 - September 3, 2010. Tutorial Lectures, 2010.

FACT++, **FaCT++**. Disponível em: <<http://owl.man.ac.uk/factplusplus/>>. Acesso em: 10/06/2010. 2010.

FACT++1.4.1, **Summary: FaCT++ 1.4.1** Disponível em: <<http://code.google.com/p/factplusplus/>>. Acesso em: 18/06/2010. 2010.

FENSEL, D.; KERRIGAN, M.; ZAREMBA, M., **Implementing Semantic Web Services**. The SESA Framework, Springer, 2008.

FOAF, **The Friend of a Friend project**. Disponível em: <<http://www.foaf-project.org>>. Acesso em: 29/06/2009. 2009.

GARTNER, **Gartner Consulting**. Disponível em: <<http://www.gartner.com/it/page.jsp?id=503861>>. Acesso em: 10/10/2009. 2007.

GOEBEL, M.; GRUENWALD, L., **A Survey of Data Mining and Knowledge Discovery Software Tools**. ACM SIGKDD, Junho 1999, 1999.

GOOGLE HEALTH, **Google Health**. Disponível em: <<http://www.google.com/intl/pt-BR/health/tour/>>. Acesso em: 27/11/2009. 2009.

GRAU, B.C.; HORROCKS, I.; MOTIK, B.; PARSIA, B.; PATEL-SCHNEIDER, P.; SATTTLER, U., **OWL 2: The next step for OWL**. Web Semantics: Science, Services and Agents on the World Wide Web, Volume 6, Issue 4, Semantic Web Challenge 2006/2007, November 2008, Pages 309-322, ISSN 1570-8268, DOI: 10.1016., 2008.

GUTIERREZ, M.; VEXO, F.; THALMANN, D., **Semantics-based representation of virtual environments**. In: International Journal of Computer Applications in Technology, vol. 23, Nos. 2/3/4, 2005.

HAARSLEV, V.; MÖLLER, R., **RACER System Description**. In. Proc. of the 1999 Description Logic Workshop (DL'99), pg 130-132. CEUR Eletronic Workshop Proceedings, <http://ceur-ws.org/Vol-22>, 2001.

HEBELER, J.; FISCHER, M.; BLACE, R.; PEREZ-LOPES, A., **Semantic Web Programming**. Wiley Publishing. Capítulo 4, pg 103, Wiley Publishing, 2009.

HERMIT, **Hermit OWL Reasoner**. Disponível em: <<http://hermit-reasoner.com/>>. Acesso em: 18/06/2010. 2010.

HORN, A. **On sentences which are true of direct unions of algebras**. Journal of Symbolic Logic, 16:14-21, 1951.

HUANG, Z.; BISHOP, B.; BRAGA, D.; FISCHER, F.; VAN HARMELEN, F.; ZENG, Y.; ZHOU, H.; QUESADA, J.; RIESKAMP, J.; SHOOLER, L., **A Survey of Web Scale Reasoning**. The Large Knowledge Collider LarKC a platform for large scale integrated reasoning and Websearch, 2009.

JARDIM, A. D., **Aplicações de Modelos Semânticos em Redes Sociais**. Dissertação de Mestrado em Ciência da Computação. Programa de Pós-Graduação em Informática - CPOLI/UCPel, 2010.

JENA, **Jena – A Semantic Web Framework for Java**. Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: 18/06/2010. 2010.

KARRAY, F.; ALEMZADEH, M.; SALEH, J. A.; ARAB, M. N., **Human-Computer Interaction: Overview on State of the Art**. In: International Journal on Smart Sensing and Intelligent Systems, Vol. 1, No. 1, March, 2008.

KEOGH, E.; KASSETTY, S., **On the need for time series data mining benchmarks: a survey and empirical demonstration**. In: ACM SIGKDD, Edmonton, Canada, 2002, p.102-111, 2002.

KNORR, M.; ALFERES, J. J.; HITZLER, P., **A Well-founded Semantics for Hybrid MKNF Knowledge Bases**. In: Diego Calvanese, Enrico Franconi, Volker Haarslev, Domenico Lembo, Boris Motik, Anni-Yasmin Turhan, Sergio Tessaris, Proceedings of the 2007 International Workshop on Description Logics (DL-2007), Brixen-Bressanone, Italy, June 2007. CEUR Workshop Proceedings, Vol-250, 2007.

LEVY, A. Y.; ROUSSET, M.-C., **Combining Horn Rules and Description Logic in CARIN**. Artificial Intelligence, 104(1-2):741-843, 1998.

LÉVY, P., **A inteligência coletiva: por uma antropologia do ciberespaço**. São Paulo: ed. Loyola, 1998.

LINDEN LAB, **Second Life Economy Hits New All-Time High in Q1 2010**. Disponível em: <<http://blogs.secondlife.com/community/features/blog/2010/04/28/second-life-economy-hits-new-all-time-high-in-q1-2010>>. Acesso em: 08/05/2010. 2010.

MANCHESTER, **The University of Manchester and FaCT++**. Disponível em: <<http://owl.cs.manchester.ac.uk/fact++/>>. Acesso em: 16/06/2010. 2010.

MARTÍNEZ, J. I.; MATTA, C. D., **A Basic Semantic Common level for Virtual Environments**. In: International Journal of Virtual Reality, 5(3):25-32, 2006.

MEDITSKOS, G.; BASSILIADES, N.; **Combining a DL Reasoner and a Rule Engine for Improving Entailment-based OWL Reasoning.** In: Proc. 7th International Semantic Web Conference (ISWC-2008), 26-30 Oct 2008, Karlsruhe, Germany, Springer, LNCS, Vol. 5318, pp. 277-292 , 2008.

MEECH, A., **Business Rules Using OWL and SWRL.** TMRF e-Book. Advances in Semantic Computing,(Eds. Joshi, Boley & Akerkar), Vol. 2, pp 23 - 31, 2010.

MESSINGER, P. R.; STROULIA, E.; LYONS, K., **A Typology of Virtual Worlds: Historical Overview and Future Directions.** Historical overview and future directions, Journal of Virtual Worlds Research, vol. 1, no. 1, 2008.

MESSINGER, P. R.; STROULIA, E.; LYONS, K.; BONE, M.; NIU, R. H.; SMIRNOV, K.; PERELGUT, S., **Virtual worlds - past, present, and future: New directions in social computing.** Decision Support Systems, Elsevier B.V., n. 47, págs 204-228, 2009.

MICROSOFT HEALTHVAULT, **Microsoft HealthVault.** Disponível em: <<http://www.healthvault.com/>>. Acesso em: 27/11/2009. 2009.

MINH, D. T.; **Default Reasoning on Top of Ontologies with dl-Programs.** Master Thesis. Vienna University of Technology, 2008.

MINSKY, M. A., **A Framework for Representing Knowledge.** McGraw-Hill, New York, 1975.

MOTIK, B.; HORROCKS, I.; ROSATI, R.; SATTLER, U., **Can OWL and Logic Programming Live Together Happily Ever After?** In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, Proc. of the 5th Int. Semantic Web Conference (ISWC 2006), volume 4273 of LNCS, pages 501–514, Springer, Athens, GA, USA, November 5–9, 2006.

MOTIK, B.; PATEL-SCHNEIDER, P.F.; PARSIA, B., **OWL web ontology language semantics and abstract syntax.** Disponível em: <<http://www.w3.org/TR/owl2-overview/>>. Acesso em: 03/04/2010. 2009.

MOTIK, B.; SHEARER, R.; HORROCKS, I., **Hypertableau Reasoning for Description Logics.** Journal of Artificial Intelligence Research 36 (2009) 165–228, 2010.

NARDI, D.; BRACHMAN, R. J., **An Introduction to Description Logics.** In The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University, p. 1-40, 2003.

NEBOLSKY, C.; YEE, N. K.; PETRUSHIN, V. A.; GERSHMAN, A. V., **Using Virtual Worlds for Corporate Training.** icalt, pp.412, Third IEEE International Conference on Advanced Learning Technologies (ICALT'03), 2003.

O'CONNOR, M. J.; DAS, A.; **A Mechanism to Define and Execute SWRL Built-ins in Protégé-OWL.** In: 9th Int. Protégé Conference, Stanford, California, July 23-26 , 2006.

OPENWONDERLAND, **Open Wonderland.** Disponível em: <<http://www.openwonderland.org/>>. Acesso em: 16/11/2010. 2010.

OTTO, K. A., **The Semantics of Multi-user Virtual Environments**. In Workshop towards SVE, pages 35--39, 2005.

OWL, **Web Ontology Language**. Disponível em: <<http://www.w3.org/2004/OWL/>>. Acesso em: 29/06/2009. 2009.

OWL 2, **OWL 2 Web Ontology Language Document Overview**. Disponível em: <<http://www.w3.org/TR/owl2-overview/>>. Acesso em: 08/06/2010. 2010.

OWL 2 Profile, **OWL 2 Web Ontology Language Profiles**. Disponível em: <<http://www.w3.org/TR/owl2-profiles/>>. Acesso em: 20/05/2010. 2010.

OWL API, **The OWL API** Disponível em: <<http://owlapi.sourceforge.net/>>. Acesso em: 18/06/2010. 2010.

PALAZZO, L. A. M., **Introdução à Programação PROLOG**. Editora EDUCAT. Pelotas, 1997.

PAN, J. Z.; ZHAO, Y.; REN, Y., **Towards Soundness Preserving Approximation for TBox Reasoning in OWL 2**. In.: The 8th International Semantic Web Conference (ISWC 2009) at the Westfields Conference Center near Washington, DC, 2009.

PARSIA, B.; SIRIN, E.; GRAU, B. C.; RUCKHAUS, E.; HEWLETT, D., **Cautiously approaching SWRL**. Preprint submitted to Elsevier Science, 2005.

PELLET, **Pellet**. Disponível em: <<http://clarkparsia.com/pellet/>>. Acesso em: 10/06/2010. 2010.

PEQUENO, M. C.; VERAS, R. M. S.; TAVARES, W. A., **Lógica não monotônica com prioridade às exceções**. Encontro Nacional de Inteligência Artificial 2007 - ENIA 2007, 30 de junho à 06 de julho, 2007.

PETRY, L. C., **Estruturas cognitivo-ontológicas dos Metaversos**. In: SLACTIONS 2009 International Conference: Life, imagination, and work using metaverse platforms, 24 e 25 de Setembro de 2009, Vila Real: UTAD, 2009.

PROTÉGÉ, **Protégé**. Disponível em: <<http://protege.stanford.edu/>>. Acesso em: 10/06/2010. 2010.

RESTful, **Representational State Transfer**. Disponível em: <http://en.wikipedia.org/wiki/Representational_State_Transfer>. Acesso em: 26/02/2011. 2011.

RICESU, **Ilha RICESU – Second Life**. Disponível em: <<http://gpedunisinos.wordpress.com/ambientes/ilha-ricesu-second-life/>>. Acesso em: 28/05/2010. 2010.

SANTOS, R. P., **Second Life physics: Virtual, real or surreal?** Journal of Virtual Worlds Research, v. 2, n. 1, (special issue: Pedagogy, Education and Innovation in Virtual Worlds), Apr., 2009.

SCHLEMMER, E.; BAKES, L., **Metaversos: novos espaços para construção do conhecimento**. Revista Diálogo Educacional (PUCPR), v. 24, p. 10-20, 2008.

SCHLEMMER, E.; TREIN, D.; OLIVEIRA, C., **Metaverso: a telepresença em Mundos Digitais Virtuais 3D por meio do uso de avatares**. In. XIX Simpósio Brasileiro de Informática Educativa – SBIE 2008. Fortaleza – CE, 2008.

SCHLEMMER, E.; TREIN, D.; SOARES, H., **Espaço de convivência Digital Virtual (ECODI) RICESU: Uma experiência em rede com a Tecnologia de Metaverso Second Life**. In: 15ª Edição do Congresso Internacional ABED de Educação a Distância, Fortaleza, 2010.

SECOND LIFE, **Second Life**. Disponível em: <<http://secondlife.com>>. Acesso em: 26/10/2009. 2003.

SENSOY, M.; MEL, G.; VASCONCELOS, W. W.; NORMAN, T. J., **Position Paper: Ontological Logic Programming**. Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10), pages to appear, 2010.

SEVEN, **Seven - Semantic Virtual Environments**. Disponível em: <<http://page.mi.fu-berlin.de/otto/sve/>>. Acesso em: 12/12/2009. 2009.

SHEARER, R.; MOTIK, B.; HORROCKS, I., **HermiT: A Highly-Efficient OWL Reasoner**. In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, OWLED 2008, volume 432 of CEUR Workshop Proceedings. CEUR-WS.org, 2008.

SILVA, J.; ROSA, J.; BARBOSA, J.; BARBOSA, D; PALAZZO, L., **Distribuição de Conteúdo em Ambientes Cientes de Trilhas**. In: WebMedia. Fortaleza, Outubro, 2009.

SIRIN, E.; PARSIA, B.; GRAU, B.C.; KALYANPUR, A.; KATZ, Y., **Pellet: A practical OWL-DL reasoner**. Web Semantics: Science, Services and Agents on the World Wide Web, vol. 5, no. 2, pp. 51-53, June, 2007.

SMITH, M.; WELTY, C.; MCGUINNES, D., **Web Ontology Language (OWL) Guide Version 1.0**. <http://www.w3.org/TR/owl-guide/>, 2004.

SPARQL, **Sparql Query Language for RDF**. Disponível em: <<http://www.w3.org/TR/rdf-sparql-query/>>. Acesso em: 02/02/2010. 2010.

SUROWIECKI, J., **The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdow Shapes Business, Economies, Societies and Nations**. Little, Brown, 2004.

SWRL, **SWRL: A Semantic Web Rule Language Combining OWL and RuleML**. Disponível em: <<http://www.w3.org/Submission/SWRL/>>. Acesso em: 20/02/2011. 2011.

TOBIES, S., **Complexity Results and Practical Algorithms for Logics in Knowledge Representation**. PhD thesis, RWTH Aachen, Germany, 2001.

TSARKOV, D.; HORROCKS, I., **FaCT++ Description Logic Reasoner: System Description**. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006). Volume 4130 of Lecture Notes in Artificial Intelligence., Springer (2006) 292–297, 2006.

VANACKEN, L.; RAYMAEKERS, C.; CONINX, K., **Introducing semantic information during conceptual modelling of interaction for virtual environments.** In Proceedings of WMISI'07, pages 17–24, 2007.

VIEIRA, R.; SANTOS, D. A.; SILVA, D. M.; SANTANA, M. R., **Web semântica: ontologias, lógica de descrição e inferências.** In: Cesar Teixeira; Eduardo Barrere; Iran Abraão. (Org.). Web e Multimídia: Desafios e Soluções (WebMedia 2005 - Minicursos). 1 ed. Porto Alegre: SBC,2005, v. 1, p. 127-167., 2005.

W3C, **The World Wide Web Consortium.** Disponível em: <<http://www.w3.org/>>. Acesso em: 29/06/2009. 2009.

WIKIHORN, **Cláusula de Horn.** Disponível em: <http://pt.wikipedia.org/wiki/Cláusula_de_Horn>. Acesso em: 04/12/2010. 2010.

ANEXO A

Ontologia de Ator para definir regras de parentesco:

```
<?xml version="1.0"?>
```

```
<!DOCTYPE Ontology [
```

```
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
```

```
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
```

```
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
```

```
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
```

```
]>
```

```
<Ontology xmlns="http://www.w3.org/2002/07/owl#">
```

```
  xml:base="http://www.mm.com.br/ontologies/metaversoAtor.owl"
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#">
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
  xmlns:xml="http://www.w3.org/XML/1998/namespace">
```

```
  ontologyIRI="http://www.mm.com.br/ontologies/metaversoAtor.owl">
```

```
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
```

```
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
```

```
  <Prefix name="" IRI="http://www.w3.org/2002/07/owl#"/>
```

```
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
```

```
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
```

```
  <Declaration>
```

```
    <Class IRI="#Avó"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Avós"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Avô"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Homem"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Mulher"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Mãe"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
    <Class IRI="#Pai"/>
```

```
  </Declaration>
```

```
  <Declaration>
```

```
<Class IRI="#Pais"/>
</Declaration>
<Declaration>
  <Class IRI="#Pessoa"/>
</Declaration>
<Declaration>
  <Class IRI="#Tia"/>
</Declaration>
<Declaration>
  <Class IRI="#Tio"/>
</Declaration>
<Declaration>
  <Class IRI="#Tios"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#mãeDe"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#paiDe"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temAntecessor"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temAvó"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temAvós"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temAvô"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temIrmandade"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temIrmã"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temIrmão"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temMatrimonio"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temMãe"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temPai"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temTia"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temTio"/>
</Declaration>
<Declaration>
  <ObjectProperty IRI="#temTios"/>
</Declaration>
<Declaration>
```

```

    <DataProperty IRI="#id"/>
</Declaration>
<SubClassOf>
  <Class IRI="#Avó"/>
  <Class IRI="#Avós"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Avó"/>
  <Class IRI="#Mulher"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Avós"/>
  <Class IRI="#Pessoa"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Avô"/>
  <Class IRI="#Avós"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Avô"/>
  <Class IRI="#Homem"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Homem"/>
  <Class IRI="#Pessoa"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Mulher"/>
  <Class IRI="#Pessoa"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Mãe"/>
  <Class IRI="#Mulher"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Mãe"/>
  <Class IRI="#Pais"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Pai"/>
  <Class IRI="#Homem"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Pai"/>
  <Class IRI="#Pais"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Pais"/>
  <Class IRI="#Pessoa"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Tia"/>
  <Class IRI="#Mulher"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Tia"/>
  <Class IRI="#Tios"/>
</SubClassOf>
<SubClassOf>
  <Class IRI="#Tio"/>

```

```

    <Class IRI="#Homem"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#Tio"/>
    <Class IRI="#Tios"/>
  </SubClassOf>
  <SubClassOf>
    <Class IRI="#Tios"/>
    <Class IRI="#Pessoa"/>
  </SubClassOf>
  <DisjointClasses>
    <Class IRI="#Homem"/>
    <Class IRI="#Mulher"/>
  </DisjointClasses>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#mãeDe"/>
    <ObjectProperty abbreviatedIRI=":topObjectProperty"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temAvó"/>
    <ObjectProperty IRI="#temAvós"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temAvô"/>
    <ObjectProperty IRI="#temAvós"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temIrmandade"/>
    <ObjectProperty abbreviatedIRI=":topObjectProperty"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temIrmã"/>
    <ObjectProperty IRI="#temIrmandade"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temIrmão"/>
    <ObjectProperty IRI="#temIrmandade"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temMãe"/>
    <ObjectProperty IRI="#temAntecessor"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temPai"/>
    <ObjectProperty IRI="#temAntecessor"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temTia"/>
    <ObjectProperty IRI="#temTios"/>
  </SubObjectPropertyOf>
  <SubObjectPropertyOf>
    <ObjectProperty IRI="#temTio"/>
    <ObjectProperty IRI="#temTios"/>
  </SubObjectPropertyOf>
  <SymmetricObjectProperty>
    <ObjectProperty IRI="#temIrmandade"/>
  </SymmetricObjectProperty>
  <SymmetricObjectProperty>
    <ObjectProperty IRI="#temMatrimonio"/>
  </SymmetricObjectProperty>

```

```

<ObjectPropertyDomain>
  <ObjectProperty IRI="#temAntecessor"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#temIrmã"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#temIrmão"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#temMatrimonio"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#temTio"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyDomain>
  <ObjectProperty IRI="#temTios"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyDomain>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temAntecessor"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temIrmã"/>
  <Class IRI="#Mulher"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temIrmão"/>
  <Class IRI="#Homem"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temTia"/>
  <Class IRI="#Mulher"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temTio"/>
  <Class IRI="#Homem"/>
</ObjectPropertyRange>
<ObjectPropertyRange>
  <ObjectProperty IRI="#temTios"/>
  <Class IRI="#Pessoa"/>
</ObjectPropertyRange>
<DataPropertyDomain>
  <DataProperty IRI="#id"/>
  <Class IRI="#Pessoa"/>
</DataPropertyDomain>
<DataPropertyRange>
  <DataProperty IRI="#id"/>
  <Datatype abbreviatedIRI="xsd:integer"/>
</DataPropertyRange>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Avó"/>

```

```

    <Variable IRI="urn:swrl#x"/>
  </ClassAtom>
</Body>
<Head>
  <ClassAtom>
    <Class IRI="#Avós"/>
    <Variable IRI="urn:swrl#x"/>
  </ClassAtom>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Avô"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Avós"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Homem"/>
      <Variable IRI="urn:swrl#w"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temIrmandade"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temMatrimonio"/>
      <Variable IRI="urn:swrl#z"/>
      <Variable IRI="urn:swrl#w"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temTio"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#w"/>
    </ObjectPropertyAtom>

```

```

</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Homem"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#x"/>
    </ObjectPropertyAtom>
  </Body>
</Head>
<DLSafeRule>
  <ClassAtom>
    <Class IRI="#Pai"/>
    <Variable IRI="urn:swrl#x"/>
  </ClassAtom>
</Head>
<DLSafeRule>
  <DLSafeRule>
    <Body>
      <ClassAtom>
        <Class IRI="#Homem"/>
        <Variable IRI="urn:swrl#y"/>
      </ClassAtom>
      <ClassAtom>
        <Class IRI="#Pessoa"/>
        <Variable IRI="urn:swrl#x"/>
      </ClassAtom>
      <ClassAtom>
        <Class IRI="#Pessoa"/>
        <Variable IRI="urn:swrl#z"/>
      </ClassAtom>
      <ObjectPropertyAtom>
        <ObjectProperty IRI="#temAntecessor"/>
        <Variable IRI="urn:swrl#x"/>
        <Variable IRI="urn:swrl#z"/>
      </ObjectPropertyAtom>
      <ObjectPropertyAtom>
        <ObjectProperty IRI="#temAntecessor"/>
        <Variable IRI="urn:swrl#y"/>
        <Variable IRI="urn:swrl#z"/>
      </ObjectPropertyAtom>
      <DataPropertyAtom>
        <DataProperty IRI="#id"/>
        <Variable IRI="urn:swrl#x"/>
        <Variable IRI="urn:swrl#idx"/>
      </DataPropertyAtom>
      <DataPropertyAtom>
        <DataProperty IRI="#id"/>
        <Variable IRI="urn:swrl#y"/>
        <Variable IRI="urn:swrl#idy"/>
      </DataPropertyAtom>
      <BuiltInAtom IRI="http://www.w3.org/2003/11/swrlb#notEqual">
        <Variable IRI="urn:swrl#idx"/>
        <Variable IRI="urn:swrl#idy"/>
      </BuiltInAtom>
    </Body>
  </DLSafeRule>
</DLSafeRule>

```

```

<Head>
  <ObjectPropertyAtom>
    <ObjectProperty IRI="#temIrmão"/>
    <Variable IRI="urn:swrl#x"/>
    <Variable IRI="urn:swrl#y"/>
  </ObjectPropertyAtom>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Homem"/>
      <Variable IRI="urn:swrl#z"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temIrmãmandade"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
  </Body>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Homem"/>
      <Variable IRI="urn:swrl#z"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
  </Body>
</DLSafeRule>
  <ObjectPropertyAtom>

```

```

    <ObjectProperty IRI="#temAvô"/>
    <Variable IRI="urn:swrl#x"/>
    <Variable IRI="urn:swrl#z"/>
  </ObjectPropertyAtom>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mulher"/>
      <Variable IRI="urn:swrl#w"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temIrmandade"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temMatrimonio"/>
      <Variable IRI="urn:swrl#z"/>
      <Variable IRI="urn:swrl#w"/>
    </ObjectPropertyAtom>
  </Body>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mulher"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#x"/>
    </ObjectPropertyAtom>
  </Body>
</Head>
  <ClassAtom>
    <Class IRI="#Mãe"/>
    <Variable IRI="urn:swrl#x"/>
  </ClassAtom>

```

```

    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mulher"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#z"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
    <DataPropertyAtom>
      <DataProperty IRI="#id"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#idx"/>
    </DataPropertyAtom>
    <DataPropertyAtom>
      <DataProperty IRI="#id"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#idy"/>
    </DataPropertyAtom>
    <BuiltInAtom IRI="http://www.w3.org/2003/11/swrlb#notEqual">
      <Variable IRI="urn:swrl#idx"/>
      <Variable IRI="urn:swrl#idy"/>
    </BuiltInAtom>
  </Body>
  <Head>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temIrmã"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mulher"/>
      <Variable IRI="urn:swrl#z"/>
    </ClassAtom>
    <ClassAtom>
      <Class IRI="#Pessoa"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>

```

```

<ClassAtom>
  <Class IRI="#Pessoa"/>
  <Variable IRI="urn:swrl#y"/>
</ClassAtom>
<ObjectPropertyAtom>
  <ObjectProperty IRI="#temAntecessor"/>
  <Variable IRI="urn:swrl#x"/>
  <Variable IRI="urn:swrl#y"/>
</ObjectPropertyAtom>
<ObjectPropertyAtom>
  <ObjectProperty IRI="#temIrmandade"/>
  <Variable IRI="urn:swrl#y"/>
  <Variable IRI="urn:swrl#z"/>
</ObjectPropertyAtom>
</Body>
<Head>
  <ObjectPropertyAtom>
    <ObjectProperty IRI="#temTia"/>
    <Variable IRI="urn:swrl#x"/>
    <Variable IRI="urn:swrl#z"/>
  </ObjectPropertyAtom>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mulher"/>
      <Variable IRI="urn:swrl#z"/>
    </ClassAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAntecessor"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAvó"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#z"/>
    </ObjectPropertyAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Mãe"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Pais"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>

```

```

</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Pai"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Pais"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Tia"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Tios"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ClassAtom>
      <Class IRI="#Tio"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Tios"/>
      <Variable IRI="urn:swrl#x"/>
    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temAvó"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Avó"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>

```

```

<Body>
  <ObjectPropertyAtom>
    <ObjectProperty IRI="#temAvô"/>
    <Variable IRI="urn:swrl#x"/>
    <Variable IRI="urn:swrl#y"/>
  </ObjectPropertyAtom>
</Body>
<Head>
  <ClassAtom>
    <Class IRI="#Avô"/>
    <Variable IRI="urn:swrl#y"/>
  </ClassAtom>
</Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temMãe"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#mãeDe"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#x"/>
    </ObjectPropertyAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temPai"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#paiDe"/>
      <Variable IRI="urn:swrl#y"/>
      <Variable IRI="urn:swrl#x"/>
    </ObjectPropertyAtom>
  </Head>
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temTia"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Body>
  <Head>
    <ClassAtom>
      <Class IRI="#Tia"/>
      <Variable IRI="urn:swrl#y"/>
    </ClassAtom>
  </Head>

```

```
</DLSafeRule>
<DLSafeRule>
  <Body>
    <ObjectPropertyAtom>
      <ObjectProperty IRI="#temTio"/>
      <Variable IRI="urn:swrl#x"/>
      <Variable IRI="urn:swrl#y"/>
    </ObjectPropertyAtom>
  </Body>
</Head>
  <ClassAtom>
    <Class IRI="#Tio"/>
    <Variable IRI="urn:swrl#y"/>
  </ClassAtom>
</Head>
</DLSafeRule>
</Ontology>
```

<!-- Generated by the OWL API (version 3.1.0.20069) <http://owlapi.sourceforge.net> -->

ANEXO B

Esquema XSD utilizado para transmitir informações pelos serviços RESTful na plataforma ORPlat:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://mm.com.br/metaverso/schema/Ator"
  elementFormDefault="qualified" xmlns:tns="http://mm.com.br/metaverso/schema/Ator">
  <xsd:element name="metaverso-ator">
    <xsd:complexType>
      <xsd:all>
        <xsd:element name="erro" type="tns:Error" maxOccurs="1" minOccurs="0" form="qualified"
          xmlns:tns="http://mm.com.br/metaverso/schema/Ator"/></xsd:element>
        <xsd:element name="pessoas" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="pessoa" type="tns:Pessoa" maxOccurs="unbounded"
                minOccurs="0" form="qualified"/></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="tios" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="tio" type="tns:Tio" maxOccurs="unbounded"
                minOccurs="0"/></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="avos" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="avo" type="tns:Avo" maxOccurs="unbounded"
                minOccurs="0"/></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="pais" minOccurs="0">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="pai" type="tns:Pai" maxOccurs="unbounded"
                minOccurs="0"/></xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:all>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```

</xsd:element>
<xsd:element name="conjuge" type="tns:Conjuge" minOccurs="0"></xsd:element>
<xsd:element name="irmaos" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="irmao" type="tns:Irmao" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="filhos" minOccurs="0">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="filho" type="tns:Filho" minOccurs="0"
maxOccurs="unbounded"></xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="homem" type="tns:Homem" minOccurs="0"></xsd:element>
<xsd:element name="mulher" type="tns:Mulher" minOccurs="0"></xsd:element>
</xsd:all>
<xsd:attribute name="individuo" type="xsd:string"/>
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="Error">
  <xsd:restriction base="xsd:string"/>
</xsd:simpleType>
<xsd:complexType name="Pessoa">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="genero" type="xsd:string"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Avo">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Tio">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Pai">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Conjuge">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Irmao">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Filho">

```

```
<xsd:simpleContent>
  <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
</xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Homem">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="Mulher">
  <xsd:simpleContent>
    <xsd:extension xmlns:tns="http://mm.com.br/metaverso/schema/Ator" base="tns:Pessoa"/>
  </xsd:simpleContent>
</xsd:complexType>
</xsd:schema>
```

ANEXO C

Descrição WADL para os serviços RESTful fornecidos pela plataforma ORPlat.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<application xmlns="http://research.sun.com/wadl/2006/10">
  <doc xmlns:jersey="http://jersey.dev.java.net/" jersey:generatedBy="Jersey: 1.1.5 01/20/2010 04:04
  PM"/>
  <resources base="http://localhost:8080/metaverso/ontologia">
    <resource path="/ator/pessoas">
      <method id="getXml" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
      <method id="putXml" name="PUT">
        <request>
          <representation mediaType="application/xml"/>
        </request>
      </method>
      <resource path="{username}/avosService">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
        style="template" type="xs:string"/>
        <method id="getXml" name="GET">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{username}/tiosService">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
        style="template" type="xs:string"/>
        <method id="getXml" name="GET">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
      </resource>
      <resource path="{username}/paisService">
        <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
        style="template" type="xs:string"/>
        <method id="getXml" name="GET">
          <response>
            <representation mediaType="application/xml"/>
          </response>
        </method>
        <method id="putXml" name="PUT">

```

```

        <request>
          <representation mediaType="application/xml"/>
        </request>
      </method>
    </resource>
    <resource path="{username}/irmaosService">
      <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
style="template" type="xs:string"/>
      <method id="getXml" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
    </resource>
    <resource path="{username}/filhosService">
      <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
style="template" type="xs:string"/>
      <method id="getXml" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
    </resource>
    <resource path="{username}/conjugeService">
      <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
style="template" type="xs:string"/>
      <method id="getXml" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
      <method id="putXml" name="PUT">
        <request>
          <representation mediaType="application/xml"/>
        </request>
      </method>
    </resource>
    <resource path="{username}/parentesService">
      <param xmlns:xs="http://www.w3.org/2001/XMLSchema" name="username"
style="template" type="xs:string"/>
      <method id="getXml" name="GET">
        <response>
          <representation mediaType="application/xml"/>
        </response>
      </method>
    </resource>
  </resources>
</application>

```