RAFAEL LIBERATO ROBERTO

UMA ABORDAGEM PARA IDENTIFICAÇÃO DE INTERESSES DOS USUÁRIOS DURANTE A NAVEGAÇÃO EM *WEBSITES* SEMÂNTICOS

MARINGÁ

2006

RAFAEL LIBERATO ROBERTO

UMA ABORDAGEM PARA IDENTIFICAÇÃO DE INTERESSES DOS USUÁRIOS DURANTE A NAVEGAÇÃO EM *WEBSITES* SEMÂNTICOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: Professor Dr. Sérgio Roberto P. da Silva

MARINGÁ
2006

Dados Internacionais de Catalogação-na-Publicação (CIP)

(Biblioteca Central - UEM, Maringá – PR., Brasil)

Roberto, Rafael Liberato

R639u Uma abordagem para identificação de interesses dos usuários durante a navegação em *websites* semânticos / Rafael Liberato Roberto. -- Maringá : [s.n.], 2006.

112 f. : il., figs.

Orientador : Prof. Dr. Sérgio Roberto P. da Silva.

Dissertação (mestrado) - Universidade Estadual de Maringá. Programa de Pós-Graduação em Ciência da Computação, 2006.

1. Web semântica. 2. Modelo de usuário. 3. Spreading activation. 4. Log semântico. 5. Ontologia. I. Universidade Estadual de Maringá. Programa de Pós-graduação em Ciência da Computação. II. Título.

CDD 21.ed. 006.331

RAFAEL LIBERATO ROBERTO

UMA ABORDAGEM PARA IDENTIFICAÇÃO DE INTERESSES DOS USUÁRIOS DURANTE A NAVEGAÇÃO EM *WEBSITES* SEMÂNTICOS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Maringá, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em

BANCA EXAMINADORA

Prof. Dr. Sérgio Roberto P. da Silva

Universidade Estadual de Maringá — UEM

Profª. Drª. Elisa Hatsue Moriya Huzita

Universidade Estadual de Maringá — UEM

Profª. Drª. Raquel Oliveira Prates

Universidade Federal de Minas Gerais — UFMG

DEDICATÓRIA

Dedico este trabalho

A **Deus** e aos meus Pais **Luis Carlos Roberto** e **Izabel de Fátima L. Roberto**, pelo incentivo, apoio, compreensão e amor que sempre me deram.

AGRADECIMENTOS

Definitivamente, não podemos dar passos tão grandes sem contar com o apoio e estímulo de pessoas maravilhosas que encontramos durante o caminho. Por isso gostaria de expressar minha gratidão a todos que, direta ou indiretamente, me ajudaram a realizar este trabalho.

Sobre tudo e todos, agradeço a **Deus**. Foi **Ele** quem me deu saúde, sabedoria, discernimento, forças e amparo necessário nos momentos que precisei. Obrigado, Senhor, pela oportunidade de concluir mais esta etapa.

Sou eternamente grato ao meu Pai, **Luis Carlos Roberto**, meu espelho de caráter, simplicidade, humildade, honestidade e seriedade. E a minha querida Mãe, **Izabel de Fátima L. Roberto**, meu espelho de dedicação, amor, carinho, doação e alegria. Pessoas fantásticas que sempre me deram amor, carinho, compreensão e apoio. Por mais que eu tente por belas palavras expressar toda minha admiração e amor, não seriam suficientes para agradecer tudo que fizeram por mim. Papai e Mamãe, muito obrigado por me ensinarem o verdadeiro sentido da vida. As minhas irmãzinhas **Camila L. Roberto** e **Bianca L. Roberto** pelo amor, carinho e cumplicidade que sempre tiveram comigo.

Agradeço ao restante de minha **família**, em especial, meus avós **João Liberato** e **Thereza Liberato** e minha Tia **Tereza Ap. Liberato** que torceram e partilharam de cada passo deste trabalho e por todo carinho e incentivo ao longo de minha vida.

Agradeço ao meu orientador **Sérgio Roberto P. da Silva**, que mais do que orientar, soube conduzir e aconselhar. Obrigado Professor pela confiança, paciência e dedicação, principalmente, na fase final desta dissertação. Meu respeito, carinho e grande admiração.

A minha noiva **Michele Mandelli**, que não só me acompanhou durante esta jornada, mas também foi minha fonte de perseverança, inspiração e muito amor. Obrigado meu amor.

Agradeço aos amigos integrantes do grupo de pesquisa GSII, **Josiane**, **Raqueline**, **Pará**, **Márcia**, **Roberto** pela amizade, incentivo e contribuições.

A todos os amigos que fiz no mestrado, os quais muitos não sabem o quanto são especiais para mim. Em especial, **Éderson**, **César**, **Thesko**, **Rogério**, **Andrezinho**, **Andrezão** e **Igor**. Amigo é um irmão que podemos escolher. Obrigado pelo companheirismo, carinho, apoio e fraternidade.

Agradeço aos funcionários do Departamento de Informática **Jayme**, **Donato** e, em especial à **Maria Inês Davanço**, o que seria de nós, pobres alunos de mestrado, sem a **Inês**. Você é nota 10.

Em especial ao **CNPQ**, pela ajuda financeira recebida durante a execução deste trabalho.

“Entrega o teu caminho ao SENHOR; confia nele, e Ele tudo fará.” (Salmos 37:5).

RESUMO

A crescente demanda pela personalização de conteúdo em *websites* tem estimulado o desenvolvimento de sistemas que buscam a identificação de padrões na navegação dos usuários. Estes padrões navegacionais podem ser obtidos por meio de um monitoramento da navegação do usuário armazenada em um arquivo de *log*. No entanto, esta solução não identifica a intenção semântica por trás da navegação do usuário. Este trabalho provê uma abordagem para incorporar conhecimento semântico a comportamentos navegacionais criando um *log* semântico que será empregada na identificação de um modelo de usuário, que expressa os possíveis interesses destes usuários ao navegar em um *website* semântico. Para isto, desenvolvemos um algoritmo probabilístico para a identificação destes interesses, que toma como base o *log* semântico, e também leva em consideração aspectos lingüísticos e cognitivos que afetam o poder de expressão dos usuários em sua navegação no *website*. O modelo do usuário obtido é representado na forma de uma lista de preferência contendo a probabilidade de interesse dos conceitos envolvidos no *website* semântico.

Palavras-Chave: *Web* semântica, Modelo de usuário, *Spreading Activation*, *Log* Semântico, Ontologia.

ABSTRACT

The growing need for content customization in websites has fostered the development of systems which try to identify the users navigation patterns. These may be identified by means of user monitoring and log file analysis. However, this solution does not identify the semantic intention behind the user navigation. This work provides an approach to incorporating semantic knowledge to navigational behaviors creating a semantic log that will be used in the identification of an user model which express the possible interests of these users when navigating in a semantic website. For this, we develop a probabilistic algorithm for the identification of user's interests, which takes as a base the semantic log and also takes into account linguistic and cognitive aspects that affect the user's power of expression in their navigation in the website. The resulting user model is represented in the form of a preference list containing the probability of interest of the involved concepts in the semantic website.

Keywords: Semantic Web , User Model, Spreading Activation, Semantic Log, Ontology.

SUMÁRIO

1 INTRODUÇÃO 14

2 FUNDAMENTAÇÃO TEÓRICA 18

1 A ADAPTAÇÃO NA *WEB* 18
2 MODELAGEM DE USUÁRIO 20
3 A *WEB* SEMÂNTICA 24
3.1 A arquitetura da Web Semântica 25
3.2 As Ontologias 34
3.3 A Linguagem OWL (Web Ontology Language) 38

3 TRABALHOS RELACIONADOS 41

1 SEWEP (*SEMANTIC ENHANCEMENT FOR WEB PERSONALIZATION*) 41
2 *INDEXFINDER* 43
3 ADAPTAÇÃO DA *WEB* SEMÂNTICA BASEADO EM DADOS DE USO 44

4 INTEGRANDO PADRÕES NAVEGACIONAIS E CONTEÚDO SEMÂNTICO 47

1 A ESTRUTURAÇÃO DE UM *WEBSITE* SEMÂNTICO 48
1.1 A modelagem da Ontologia do Domínio 49
1.1.1 A Metodologia de desenvolvimento 49
1.1.2 A Linguagem e Ferramenta adotadas na concepção da ontologia 52
1.1.3 O Desenvolvimento da Ontologia 54
1.2 Desenvolvimento do Website Semântico 58
2 A CONSTRUÇÃO DO *LOG* SEMÂNTICO 62

5 MODELAGEM DO USUÁRIO 65

1 OS ASPECTOS LINGÜÍSTICOS QUE AFETAM O MODELO DO USUÁRIO 66
2 OS ASPECTOS COGNITIVOS QUE AFETAM O MODELO DO USUÁRIO 70
3 UM ALGORITMO DE IDENTIFICAÇÃO DE INTERESSES DE UM USUÁRIO 71

6 UMA AVALIAÇÃO EXPERIMENTAL DO ALGORITMO PROPOSTO 82

1 CARACTERÍSTICAS DOS EXPERIMENTOS 82
1.1 Um Algoritmo para classificação baseada no classificador Naïve de Bayes 84
1.2 Um Algoritmo para classificação baseada em freqüência 85
1.3 O modelo de Aplicação utilizado 86
1.4 Os valores adotados para os parâmetros nos experimentos 87
2 ANÁLISE DOS RESULTADOS DOS EXPERIMENTOS 88

7 CONCLUSÃO 99

REFERÊNCIAS 104

APÊNDICE A - CLASSES IMPLEMENTADAS NO CONTROLE LÓGICO DA APLICAÇÃO 109

LISTA DE FIGURAS

Figura 1 - As camadas da Web Semântica. 26
Figura 2. Representação de uma sentença em RDF na forma de grafo 30
Figura 3. As camadas RDF e RDFS 31
Figura 4 - Tipos de ontologias de acordo com seu nível de dependência. 37
Figura 5. O *plugin* OWL é uma extensão do Protege. 53
Figura 6. Estrutura dos Conceitos da Ontologia 55
Figura 7. Propriedades da Ontologia 56
Figura 8. Restrições da Classe "Grupo de Pesquisa". 57
Figura 9. Mapeamento da ontologia para o *website*. 58
Figura 10. Classes desenvolvidas no módulo de controle lógico da aplicação. 60
Figura 11. Mapeamento dos dados da ontologia para a página *web* 61
Figura 12. Página *web* que lista os docentes do DIN. 62
Figura 13. Segmento da ontologia do domínio para o modelo de aplicação 67
Figura 14. Segmento da ontologia do domínio para o modelo de apresentação 68
Figura 15. Modelos de Conhecimento. 69
Figura 16. Os componentes de uma página *web*. 69
Figura 17. Processo de Ativação 72
Figura 18. Parâmetros no processo de propagação. 73
Figura 19. Conceitos considerados na rede semântica 74
Figura 20. Módulos de acesso do algoritmo 75
Figura 21. Lista de preferência com a porcentagem de interesse dos conceitos presentes nos modelos de conhecimento 75
Figura 22. Um algoritmo para enriquecimento semântico do arquivo de *log*. 78
Figura 23. Um algoritmo de classificação bayesiana adaptado. 85
Figura 24. Um algoritmo de classificação baseado em freqüência. 86
Figura 25. Modelo de Aplicação adotado para os experimentos 87
Figura 26. Progresso das percentagens de interesses na navegação para o algoritmo de classificação baseada em freqüência. 89
Figura 27. Progresso das percentagens de interesses na navegação para o algoritmo de classificação bayesiana. 89
Figura 28. Progresso das percentagens de interesses na navegação quando aplicado o algoritmo proposto. 91
Figura 29. Progresso das percentagens de interesses na navegação para o algoritmo de classificação baseada em freqüência. 94
Figura 30. Progresso das percentagens de interesses na navegação para o algoritmo de classificação bayesiana. 94

Figura 31. Progresso das percentagens de interesses na navegação quando aplicado o algoritmo proposto.....96

LISTA DE TABELAS

Tabela 1. Comparação de um trecho de código escrito em HTML e XML 27
Tabela 2. Representação de uma sentença RDF na forma de tripla 29
Tabela 3. Lista de preferência resultante da classificação bayseana e baseada em freqüência. 90
Tabela 4. Lista de preferência resultante da classificação baseada no algoritmo proposto..... 92
Tabela 5. Lista de preferência resultante da classificação bayseana e baseada em freqüência. 95
Tabela 6. Lista de preferência resultante da classificação baseada no algoritmo proposto..... 97

LISTA DE ABREVIATURAS

DAML	*DARPA Agente Markup Language*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
OIL	*Ontology Inference Layer*
ORL	*OWL Rule Language*
OWL	*Web Ontology Language*
PAL	*Proof Markup Language*
RDF	*Resource Description Framework*
RDFS	RDF *Schema*
RuleML	*Rule Markup Language*
SWRL	*Semantic Web Rule Language*
URI	*Uniform Resource Indicator*
URL	*Uniform Resource Locators*
W3C	*World Wide Web Consortium*
XML	*Extensible Markup Language*
XMLS	XML *Schema*

Capítulo 1

INTRODUÇÃO

A navegação de um usuário em um *website* está fortemente relacionada às suas necessidades e interesses. No entanto, a maioria dos *websites* atuais não leva isto em consideração, tendo a estrutura de suas páginas definida de forma estática. Uma solução para este problema é o emprego de mecanismos de personalização ou adaptação de *websites*. Um *website* personalizado pode incluir novas páginas *index*, fornecer resultados de buscas personalizados, criar dinamicamente recomendações (como *links* de páginas) para o usuário, ou mesmo definir novos *layouts* para uma página.

Os mecanismos de personalização atuais, em geral, utilizam uma análise de comportamento navegacional para criar um modelo do usuário [EIRINAKI e VAZIRGIANNIS, 2003] [BRUSILOVSKY, 2001] [LEI *et al*, 2004], extraindo padrões comportamentais por meio de técnicas de aprendizagem de máquina. Estes modelos podem ser representados na forma de uma ontologia [TANUDJAJA e MUI, 2002], de uma lista de páginas *web* [LIEBERMAN, 1995], de

uma lista de conceitos de interesse [MIDDLETON *et al*, 2003], entre outras. É interessante notar que estes modelos são construídos usando uma grande variedade de técnicas de aprendizagem como, por exemplo, modelos de espaço vetorial [CHEN e SYCARA, 1998], modelos probabilísticos [MLADENIC, 1999], ou clusterização [MOBASHER *et al*, 2000].

Mladenic (1999) cita que o uso exclusivo de dados de navegação pode ser problemático, pois podem surgir dificuldades quando não há dados suficientes para extrair padrões relacionados a determinadas categorias do domínio, ou quando são adicionadas novas páginas ao *website* que ainda não foram visitadas pelos usuários. A incorporação de informações relacionadas ao conteúdo das páginas, ou a estrutura do *website*, ou seja, dados referentes à sua semântica, fornecem um modo de superar estes problemas, melhorando todo o processo de personalização [DAI e MOBASHER, 2003] [EIRINAKI e VAZIRGIANNIS, 2003]. Uma abordagem comum neste contexto é a integração de características dos conteúdos das páginas com classificações definidas pelos usuários [CLAYPOOL *et al*, 1999]. Geralmente, nesta abordagem, palavras chaves são extraídas do conteúdo do *website* e são utilizadas para indexar ou classificar as páginas em várias categorias de conteúdo. Assim, estas abordagens permitiriam que máquinas de recomendação indicassem páginas a um usuário baseando-se não somente na semelhança entre navegações dos usuários, mas também (ou alternativamente) na semelhança do conteúdo das páginas.

No entanto, mesmo estes sistemas podem não ser capazes de capturar relações mais complexas entre as informações como, por exemplo, relações provenientes de um nível semântico mais profundo, que se baseiam em atributos e propriedades dos conceitos envolvidos [DAI e MOBASHER, 2003] [EIRINAKI e VAZIRGIANNIS, 2003]. Para isto, é preciso um modelo que seja capaz de representar as relações das informações com maior riqueza semântica como, por exemplo, por meio de uma ontologia do domínio.

A *web* semântica foi proposta como uma extensão da *web* atual na qual o significado, ou seja, a semântica das informações presentes nas páginas de um *website* deve estar formalmente definida por uma marcação presente no *website*. Neste novo contexto, as páginas *web* deverão ser compreendidas não somente por pessoas, mas também pelas máquinas, na forma de agentes computacionais. Uma forma de viabilizar esta idéia é a utilização de ontologias associadas aos *websites*. Estas ontologias permitiriam aos agentes de software raciocinar sobre as relações entre os conteúdos do *website* e, com isto, melhor atender as necessidades dos usuários [BERNERS-LEE et al, 2001].

A proposta deste trabalho é estudar a viabilidade de associar os benefícios disponibilizados pela estruturação semântica oferecida pela *web* semântica com a análise dos comportamentos navegacionais dos usuários para criar um modelo de usuário que expressa os possíveis interesses destes usuários ao navegar em um *website*. Para tal, propomos um processo que induz à identificação da intenção dos usuários. Este processo se baseia na análise de um *log* semântico, desenvolvido utilizando-se de ontologias de domínio disponibilizada pelos *websites* com base semântica.

Na análise do *log* semântico, consideramos alguns fatores lingüísticos e cognitivos que podem afetar a expressão dos interesses dos usuários. Assim, do ponto de vista lingüístico, levamos em conta as formas de restrição do vocabulário disponível que o usuário tem sobre sua forma de expressão. Este fator restringe o poder de expressão do usuário e procura contemplar o fato de que uma pessoa só pode expressar algo para o qual ela tem um vocabulário. Do ponto de vista cognitivo, consideramos a idéia de força cognitiva, derivada da teoria de propagação de ativação (*spreading activation*) [ANDERSON, 1983], desenvolvida na psicologia cognitiva para explicar como funciona a recuperação de informações na memória humana. Desta forma, quando um usuário escolher um *link* em uma página de um *website* semântico, ele estará ativando um conceito na rede semântica que compõe a ontologia

do *website* e a força cognitiva deste conceito, e dos conceitos a ele relacionados, será considerada.

Para avaliar o processo proposto desenvolvemos uma simulação do algoritmo probabilístico de identificação de intenção do usuário proposto. Este algoritmo toma como base que todos os conceitos envolvidos no *website* são candidatos a serem o interesse do usuário. Para determinar a relevância real de cada conceito da ontologia, definimos um conjunto de parâmetros baseados nos fatores lingüísticos e cognitivos identificados, os quais ponderam, probabilisticamente, a influência de cada conceito sobre as possíveis intenções do usuário.

No Capítulo 2 deste trabalho, apresentamos os conceitos fundamentais que provêem a base para o desenvolvimento do trabalho. No Capítulo 3, discutimos alguns trabalhos correlatos, apontando algumas contribuições e algumas diferenças em relação ao trabalho proposto. No Capítulo 4, propomos uma forma de integrar padrões navegacionais e conteúdo semântico e apresentamos o processo de desenvolvimento da ontologia do domínio e seu mapeamento para um *website*. E, por fim, mostramos uma forma de construir um *log* semântico baseado nas interações do usuário com o *website* semântico. No Capítulo 5, discutimos os aspectos lingüísticos e cognitivos que podem afetar o usuário na expressão de seus interesses e propomos um algoritmo probabilístico que leva em consideração estes aspectos na construção do modelo do usuário. No Capítulo 6, fazemos uma análise comparativa realizada por meio de simulação do algoritmo proposto com um algoritmo baseado em freqüências e com o algoritmo clássico do classificador *Naïve* de Bayes. Por último, no Capítulo 7 apresentamos uma discussão dos resultados obtidos e alguns possíveis trabalhos futuros desta dissertação.

Capítulo 2

FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda as principais características de vários temas relevantes a este trabalho, tais como a adaptação na *web*, a modelagem do usuário, a *web* semântica, as ontologias e a linguagem OWL. Estes conceitos fornecem o embasamento necessário para o desenvolvimento do trabalho proposto.

1 A Adaptação na *Web*

Com o crescimento contínuo da Internet e o aumento da quantidade de informações disponibilizadas ao usuário, surge a necessidade da adoção de novos métodos para a criação e estruturação de *websites*. A maioria dos *websites* possui uma vasta quantidade de informação e os usuários, freqüentemente, perdem o foco de sua busca, ou recebem resultados ambíguos. Assim, o desenvolvimento de um *website* se tornou uma tarefa complexa e difícil, pois as informações devem ser estruturadas e apresentadas aos usuários de uma forma clara e

objetiva, ou estes usuários podem ser desestimulados na busca de informação no *website*. Uma forma de resolver este problema é obter informações sobre como os usuários interagem com o *website* e identificar suas necessidades. Estas informações podem ser utilizadas para melhorar o *website* adaptando seu conteúdo as essas necessidades.

Segundo Perkowitz e Etzioni [2000], os *websites* podem ser adaptados de duas formas, por: customização ou otimização (ou transformação). A customização é a adaptação da apresentação do *website* conforme as necessidades individuais dos visitantes, baseada em informações relativas a esses visitantes. Qualquer alteração influenciada somente por um único usuário criará, conseqüentemente, diversas versões do *website*, uma por usuário. Já a otimização ou transformação, é o aprimoramento da estrutura do *website* baseados em informações relativas as interações de todos os visitantes.

Um exemplo do processo de customização é a personalização de *websites*. Eirinaki e Vazirgiannis [2003] definem personalização da *web* como qualquer ação que adapta informações, ou serviços, providas por um *website* às necessidades de um usuário específico, ou um grupo de usuários, tirando proveito do conhecimento adquirido por meio dos comportamentos navegacionais e dos interesses dos usuários em combinação com o conteúdo e a estrutura do *website*. Portanto, um sistema de personalização da *web* deve antecipar as necessidades do usuário sem que haja um pedido explícito.

Uma das principais técnicas para obter informações relevantes sobre os usuários é a mineração de uso na *web* (*Web Usage Mining*). Esta técnica explora as informações armazenadas em arquivos de *log* de servidores *web*, sendo capaz de extrair informações estatísticas e descobrindo interessantes padrões de uso, agrupando os usuários em grupos de acordo com seu comportamento navegacional, e descobrindo potenciais relações entre as páginas *web* e grupos de usuários [EIRINAKI e VAZIRGIANNIS, 2003]. A mineração de uso da *web* tem se tornado um tópico muito pesquisado nas áreas de mineração de dados (*Data Web*

Mining) e personalização na *web* (*Web Personalization*), as quais vêem obtendo resultados muito eficientes como, por exemplo, os trabalhos de [EIRINAKI e VAZIRGIANNIS, 2003], [DUA *et al*., 2000], [SRIVASTAVA *et al*., 2000], entre outros.

Na área de adaptação de sistemas de software existe uma diferença entre sistemas adaptativos e sistemas adaptáveis. Para Opperman [1994] um sistema adaptável é aquele em que o controle flexível da informação, ou a automação da atuação do sistema, está nas mãos do usuário, este controle deve ser obtido explicitamente através de comandos explícitos por parte do usuário, ocorrendo, normalmente, em tempo de execução. Por outro lado, em sistemas adaptativos a flexibilidade da informação, ou o comportamento automático, é controlado de forma independente pelo sistema, por meio de agentes que monitoram o comportamento do usuário e alteram o sistema em resposta a este comportamento.

Nosso trabalho se enquadra no paradigma de sistemas adaptativos, mais especificamente em *websites* adaptativos, nos quais as informações apresentadas nas páginas do *website* são adaptadas conforme as necessidades dos usuários por meio de agentes que monitoram seu comportamento. Nosso objetivo é, justamente, melhorar esta adaptação levando em consideração não somente as interações do usuário ao navegar no *website*, mas também a semântica envolvida nos conteúdos acessados.

2 Modelagem de Usuário

Em sistemas adaptativos há uma necessidade de se obter informações sobre o usuário tais como seus interesses, suas preferências ou até mesmo suas necessidades, para que o sistema possa adaptar seu conteúdo a essas necessidades e tornar sua utilização mais eficiente. Para que um sistema adapte as informações às necessidades de diferentes usuários, cada vez mais se faz necessário a utilização dos modelos de usuário [KOBSA, 1993] [KOBSA, 2001]. Nos primeiros sistemas que utilizavam modelo de usuário, a modelagem era feita de forma

integrada aos demais componentes do sistema, não havendo um módulo único específico para modelagem de usuário. Esta separação entre os processos de adaptação e a modelagem de usuário foi estabelecida somente a partir de 1985. Assim, o termo "Sistema de Modelagem de Usuário" surgiu somente em 1990, inspirado nos sistemas especialistas da inteligência artificial [PALAZZO, 2000].

O processo de modelagem de usuário envolve inferir informações sobre um determinado usuário a partir de informações que podemos observar sobre ele como, por exemplo, suas ações ou comportamento [ZUKERMAN e LITMAN, 2001]. A busca por informações sobre o usuário pode ser feita explicitamente, por meio de questionamentos do sistema (necessitando de uma intervenção do usuário), ou implicitamente, caso em que o sistema captura informações do usuário por meio de monitoramento de sua interação.

Segundo Middleton *et al* [2003], a modelagem do usuário pode ser classificada em, modelagem baseada em conhecimento ou baseada em comportamento do usuário. Na modelagem baseada em conhecimento, cria-se estereótipos fixos de tipos de usuários e compara-se cada usuário com o estereótipo mais próximo. Na modelagem baseada no comportamento, utiliza-se as interações do usuário, muitas vezes empregando técnicas de aprendizado de máquina, para descobrir padrões de comportamento.

Já Palazzo [2000] cita que, há pelo menos 5 (cinco) características associadas a um usuário que podem ser levadas em consideração na modelagem, são elas:

- **Conhecimento**: é o conhecimento que o usuário possui sobre o domínio da aplicação. Este conhecimento pode modificar durante o tempo, sendo necessário reconhecer estas modificações para refleti-las na aplicação;
- **Objetivos**: está relacionado com os objetivos do usuário. Dependendo do sistema os objetivos podem estar relacionados a objetivos de trabalho (em sistemas

aplicativos), objetivos de pesquisa (em sistemas de recuperação de informações) e a solução de problemas ou metas de aprendizado (em sistemas educacionais);

- **História e Experiência**: a história significa toda informação relacionada a experiência anterior do usuário fora do assunto abordado no domínio da aplicação, mas que seja relevante para ser considerada. Experiência é a familiaridade do usuário com a estrutura da aplicação. A experiência se difere do conhecimento, pois um usuário com pleno conhecimento sobre o domínio pode não possuir instrução alguma quanto à estrutura da aplicação, ou vice-versa;
- **Preferências**: está relacionada com as preferências do usuário, que por sua vez podem ser relativas ou absolutas, dependendo do objetivo e contexto geral. Por diversas razões certas associações de *links* e/ou conteúdos podem chamar mais a atenção do usuário.

Estas características são dinâmicas, tornando necessário ajustá-las, constantemente, para manter o modelo atualizado.

Para Kobsa [1993], a modelagem das informações referente ao usuário pode ser abordada por três visões, são elas:

- **O conhecimento do usuário**: esta visão utiliza estereótipo, sendo necessário desempenhar três tarefas: identificar subgrupos de usuários, identificar características chave e representar os estereótipos. Ela é recomendada quando não é necessária uma precisão muito alta na avaliação da modelagem do usuário;
- **Os planos do usuário**: esta visão considera as seqüências de ações do usuário dentro do sistema, na busca de um determinado objetivo. Ela tem o intuito de observar todas as ações do usuário no sistema, tentando antecipar possíveis planos e com isso complementar as ações do usuário;

- **As preferências do usuário**: esta visão baseia-se nas preferências do usuário, sendo utilizada em sistemas que necessitam de uma descrição das preferências do usuário antes da interação com o sistema.

Para o nosso trabalho, utilizamos a modelagem baseada em comportamento, citada por Middleton *et al* [2003], observando as interações do usuário com o sistema para criar o modelo do usuário. Adotamos uma forma implícita de coletar informações sobre o usuário, monitorando suas interações com o *website*. Dentre as características levantadas por Palazzo [2000] e Kobsa [1993] trabalhamos com os objetivos e planos do usuário, monitorando sua navegação em busca de um objetivo, na tentativa de antecipar possíveis intenções ou interesses.

Como citado por Middleton *et al* [2003], na modelagem baseada em comportamento, para a descoberta de padrões, normalmente, são utilizadas técnicas de aprendizado de máquina. Em nosso trabalho, também adotamos um algoritmo de aprendizado de máquina, no entanto, consideramos as informações semânticas relacionadas ao conteúdo das páginas acessadas pelo usuário como, por exemplo, quais são os conceitos do domínio que estão envolvidos no conteúdo da página, como estes conceitos se relacionam com o restante do domínio, quais são os atributos destes conceitos, entre outros. Desta forma, integramos as informações sobre o comportamento navegacional dos usuários, ou seja, a forma como eles interagem com o *website*, com as informações semânticas associadas aos conteúdos por eles acessados.

Além disto, também consideramos alguns aspectos cognitivos para a descoberta das possíveis intenções ou interesses dos usuários no processo de modelagem do usuário. Para isto, utilizamos a teoria de propagação de ativação (*spreading activation*) que procura explicar como as informações são organizadas e recuperadas na memória humana [QUILLIAN, 1968] [ANDERSON, 1983]. Esta teoria considera que a memória humana organiza as informações na

forma de uma rede semântica e que quando um conceito se torna o foco de nossa atenção ele ativa todos os conceitos a ele relacionados. Deste modo, adotamos o desenvolvimento de um *website* com uma base semântica, no qual a representação semântica das informações contidas em suas páginas está associada a um modelo conceitual, uma ontologia do domínio. O conceito de ontologia será abordado com maiores detalhes na Seção 3.2 deste capítulo.

3 A *Web* Semântica

Em 2001, Tim Berners-Lee [BERNERS-LEE *et al*., 2001], anunciou a criação do projeto *Web* Semântica com objetivo de estabelecer uma nova filosofia para o desenvolvimento e utilização da *web* tradicional. A *Web* Semântica foi proposta como uma extensão da w*eb* atual, na qual os significados das informações presentes nas páginas de um *website* são definidos formalmente. Neste novo contexto, a *web* será capaz de representar as relações semânticas das informações de maneira que haja compreensão não somente por pessoas, mas também pelas máquinas na forma de agentes computacionais. Estes agentes serão capazes de trabalhar eficientemente sobre estas informações, podendo entender seus significados e auxiliar os usuários em operações na *web*.

Desta maneira, será possível que as máquinas sejam capazes de dar apoio a tarefas como integração de dados, navegação e automatização. Com a *Web* Semântica será possível não somente obter mais exatidão em resultados de busca de uma informação, como também integrar informações de fontes diferentes, provendo todos os tipos de serviços automatizados em domínios distintos [HENDLER *et al*, 2002].

Segundo Berners-Lee *et al*. (2001), para o sucesso da *Web* Semântica os computadores necessitam ter acesso a uma coleção estruturada de informações e a um conjunto de regras de inferência que auxiliem no processo de dedução automática para conduzir o raciocínio automatizado. Porém, para que seja possível alcançar todas essas funcionalidades é necessária

a utilização de algumas tecnologias. Segundo Antoniou & Van Harmelen (2004), essas tecnologias são:

- Os metadados: para identificar e extrair recursos da *web*, capturando o significado dos dados. Os metadados são dados que descrevem dados usados para ajudar na identificação, descrição, localização e gerenciamento de recursos da *web*;
- As ontologias: para modelar o conhecimento do domínio, auxiliando buscas com interpretações das informações retornadas e provendo comunicação entre agentes;
- A lógica: para que seja possível o uso de raciocinadores, processando informações e obtendo conclusões.
- Os agentes: para fazer uso de todas as tecnologias supracitadas para auxiliar os usuários na realização de tarefas.

Hendler (2002) descreve um cenário futuro considerando a *Web* Semântica como um grande número de pequenos componentes ontológicos que apontam entre si. Desta forma, companhias, universidades, agências governamentais e grupos de interesses específicos procurarão ter seus recursos *web* ligados a um conteúdo ontológico, visto que, poderosas ferramentas serão disponibilizadas para intercambiar e processar essas informações entre aplicações *web*.

Em uma tentativa de padronizar o desenvolvimento da *Web* Semântica, a W3C (*World Wide Web Consortium*) apresentou uma arquitetura composta por camadas, visando estabelecer as tecnologias apropriadas para a construção de um *website* semântico.

3.1 A arquitetura da *Web* Semântica

A W3C tem se esforçado em padronizar novas linguagens para a criação de *websites* semânticos. Em 2000, apresentou uma proposta definindo várias camadas para a *Web*

Semântica sugerindo linguagens e padrões para cada camada. Estas camadas podem ser vistas na Figura 1 [KOIVUNEN e MILLER, 2001].

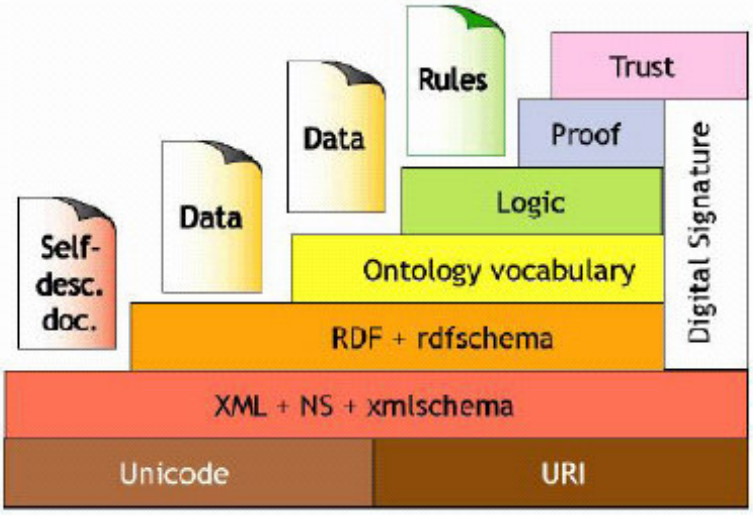

Figura 1 - As camadas da Web Semântica.
Fonte: [KOIVUNEN e MILLER, 2001].

A Camada *Unicode* / URI (*Uniform Resource Indicator*).

A primeira camada garante o uso padronizado do mesmo conjunto de caracteres (*Unicode*) e uma forma única para a identificação e localização de páginas por meio do *Uniform Resource Indicator* (URI) [KOIVUNEN e MILLER, 2001]. O URI é um identificador de recursos universal que serve para identificar unicamente determinado recurso na *web*. Um recurso, assim como na linguagem natural, pode assumir vários significados causando ambigüidade no sistema, e o URI vem como uma solução para este problema [BERNERS-LEE et al, 2001].

A Camada XML + XML *Schema.*

No modelo da *Web* atual, o fator limitante para que a estrutura utilizada não contribua com a *Web* Semântica é a ausência, quase que completa, de metadados. Podemos observar

este fato em páginas desenvolvidas na linguagem HTML (*HyperText Markup Language*), a qual não proporciona um modelo para a utilização de metadados. A linguagem XML (*Extensible Markup Language*) [FALLSIDE e WALMSLEY, 2004] apresenta uma evolução para este quadro, superando as deficiências da linguagem HTML e permitindo que as informações presentes em um documento sejam estruturadas. Cada unidade de informação é acompanhada de uma *tag* que a qualifica. Estas *tags* oferecem um ganho em termos de representação devido à sua liberdade de nomeação. Infelizmente, representação, neste contexto, diz respeito somente à facilidade de inferência do significado por um ser humano. Para agentes de software, tais como ferramentas de busca, estas informações não são mais representativas do que um conjunto de caracteres. Segue um exemplo de informações representadas na linguagem HTML e em XML:

Tabela 1. Comparação de um trecho de código escrito em HTML e XML

HTML	**XML**
<HTML> *<BODY>* *<H3>Identificação do Paciente</H3>* *<UL>* *<LI>Nome: José Santos<LI>* *<LI>Sexo: Feminino<LI>* *<LI>Nascimento: 20/08/1980<LI>* *<LI>Endereço: Rua Paraná,15<LI>* *<LI>Médico: Dr. João Santos<LI>* *</UL>* *</BODY>* </HTML>	*< Identificação>* *< Nome>José Santos< /Nome>* *< Sexo>Feminino</ Sexo>* *< Nascimento>20/08/1980</ Nascimento>* *< Endereço>Rua Paraná, 15</ Endereço>* *< Médico>Dr. João Santos</ Médico>* *< Identificação>*

A liberdade estabelecida pela linguagem XML, para nomeação das *tags*, opõe-se a associação destas a conceitos do domínio devido a problemas de diferenças de vocabulário. No processo de modelagem de arquivos XML, diferentes autores podem referenciar um mesmo conceito, atribuindo nomes distintos às *tags*. Logo, muito provavelmente, serão criados “dialetos” XML para usos específicos. Surge, então, a necessidade de se padronizar e

validar um documento XML, definindo um alinhamento léxico e a ordem dos dados no documento XML [FALLSIDE e WALMSLEY, 2004].

Procurando resolver o problema anterior, a linguagem XML *Schema* [FALLSIDE e WALMSLEY, 2004] foi desenvolvida, definindo uma gramática para os documentos XML. Um documento XML *Schema* pode definir que uma determinada unidade de informação, ou atributo, possa somente receber valores de um determinado tipo de dado. Além dos tipos de dados primitivos da linguagem, o XML *Schema* também permite a declaração de novos tipos, derivados dos tipos primitivos. Esta definição para um documento XML, fornecida pela linguagem XML *Schema*, é armazenada separadamente de sua estrutura, resolvendo, assim, um dos principais problemas encontrados na linguagem HTML.

Portanto, a evolução provida pela utilização da linguagem XML é limitada, fato que muitas vezes não é compreendido, o que faz com que a linguagem XML seja considerada erroneamente uma solução definitiva para todos os problemas de interoperabilidade semântica. Essa camada tem a função de descrever a estrutura de documentos, ou seja, fornecer uma estrutura hierárquica, deixando para as camadas superiores o trabalho de definir as relações semânticas entre as informações [BRAY *et al*., 2004].

A Camada RDF + RDF *Schema*

O RDF (*Resource Description Framework*) [LASSILA e SWICK, 1999] é um modelo para definição e uso de metadados na *web*, recomendado pelo W3C. O RDF tem por objetivo definir um mecanismo de representação de metadados para descrever recursos não vinculados a um domínio específico, provendo interoperabilidade entre as aplicações. Diferentemente do XML *Schema*, que é voltado para tipagem e definição estrutural, o RDF envolve metadados conceituais, objetivando descrever a semântica dos dados. O RDF é estruturado em um modelo de informações baseado em grafos rotulados e direcionados.

Um modelo RDF básico é composto pelos seguintes objetos:

- Recurso (*Resource*): é qualquer objeto representado em expressões RDF, por exemplo: um *website* (http://www.din.uem.br), uma parte de um *website* (http://www.din.uem.br/webmail), uma coleção de páginas, uma revista, um livro, etc., desde que esteja representado por um URI;
- Propriedade (*Property*): é uma característica, atributo ou uma relação usada para descrever um recurso;
- Literal (*Literal*): é o valor que o objeto pode assumir de acordo com a propriedade;
- Sentença (*Statements*): é a declaração de um recurso mais as propriedades que o recurso possui, e o valor da propriedade. Uma sentença é representada, por uma tripla: <sujeito, predicado e objeto>;

Como o recurso representa o sujeito em uma sentença em RDF, o significado de uma relação pode ser resumido como: "o recurso (sujeito) que possui a propriedade (predicado) com determinado valor (objeto)". O valor ou objeto pode ser tanto um tipo primitivo, definido pela XML, quanto um outro recurso [LASSILA e SWICK, 1999].

Todo recurso é identificado por um URI e por meio das sentenças possibilitam a construção básica de um modelo em RDF. Como mencionado anteriormente, uma sentença é representada na forma linear de uma tripla, composta por: sujeito (Recurso), predicado (Propriedade) e objeto (Literal). Desta forma, a sentença "José é aluno da UEM" é representada na forma de tripla como ilustra a Tabela 2.

Tabela 2. Representação de uma sentença RDF na forma de tripla

Sujeito (Recurso)	Predicado (Propriedade)	Objeto (Literal)
http://www.din.uem.br	aluno	José

A tabela acima ilustra a criação de uma sentença que descreve um recurso com a URI "http://www.din.uem.br", cuja propriedade é "Aluno" e o literal (valor) é "José".

Além de ser representado por triplas, o modelo RDF permite uma visualização na forma de grafos dirigidos e rotulados, que possibilitam representar recursos por meio de nós, propriedades por meio de arcos e literais por retângulos. Logo, a mesma sentença pode ser representado por grafos, como ilustra a Figura 2.

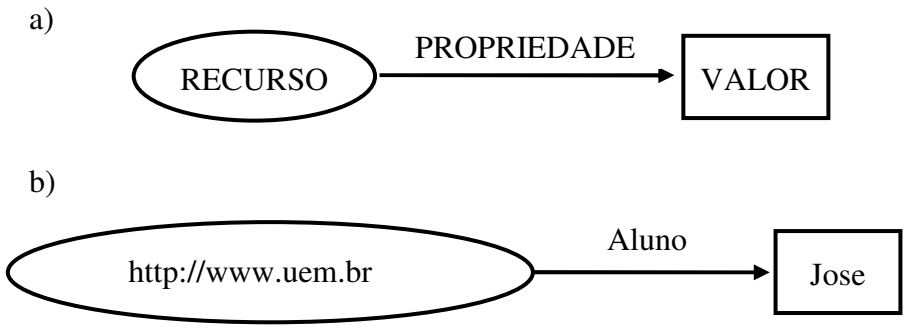

Figura 2. Representação de uma sentença em RDF na forma de grafo

Como pode ser visto na Figura 2, o nó que possui o URI indica o recurso que está sendo descrito (http://www.din.uem.br), "Aluno" indica a relação e "José" o valor da propriedade.

A representação gráfica do modelo RDF é facilmente entendida por pessoas, entretanto não é compreensível por computadores. Portanto, para que as sentenças RDF sejam compreendidas pelas máquinas é necessário que sejam são serializadas, utilizando, por exemplo, uma sintaxe XML.

Da mesma forma com que o XML requer uma linguagem para definição de como um documento deve ser estruturado, o RDF requer uma linguagem para a descrição de vocabulários, originando a linguagem RDF *Schema* (RDFS). O RDF *Schema* [BRICKLEY e GUHA, 2004] provê um sistema de tipos para RDF em que recursos podem ser definidos como instâncias de uma ou mais classes, e as classes podem ser definidas de maneira hierárquica,

com classes inferiores herdando propriedades de classes superiores. Por meio do RDF *Schema* é possível criar vocabulários representados por classes e propriedades com características restritas, visando serem reaproveitadas em outros modelos. Com intuito de ilustrar a relação entre RDF e RDFS, Antoniou e Van Harmelen [2004] mostram na Figura 3 as camadas envolvidas na seguinte sentença em RDF: Matemática discreta é ensinada por David Billington.

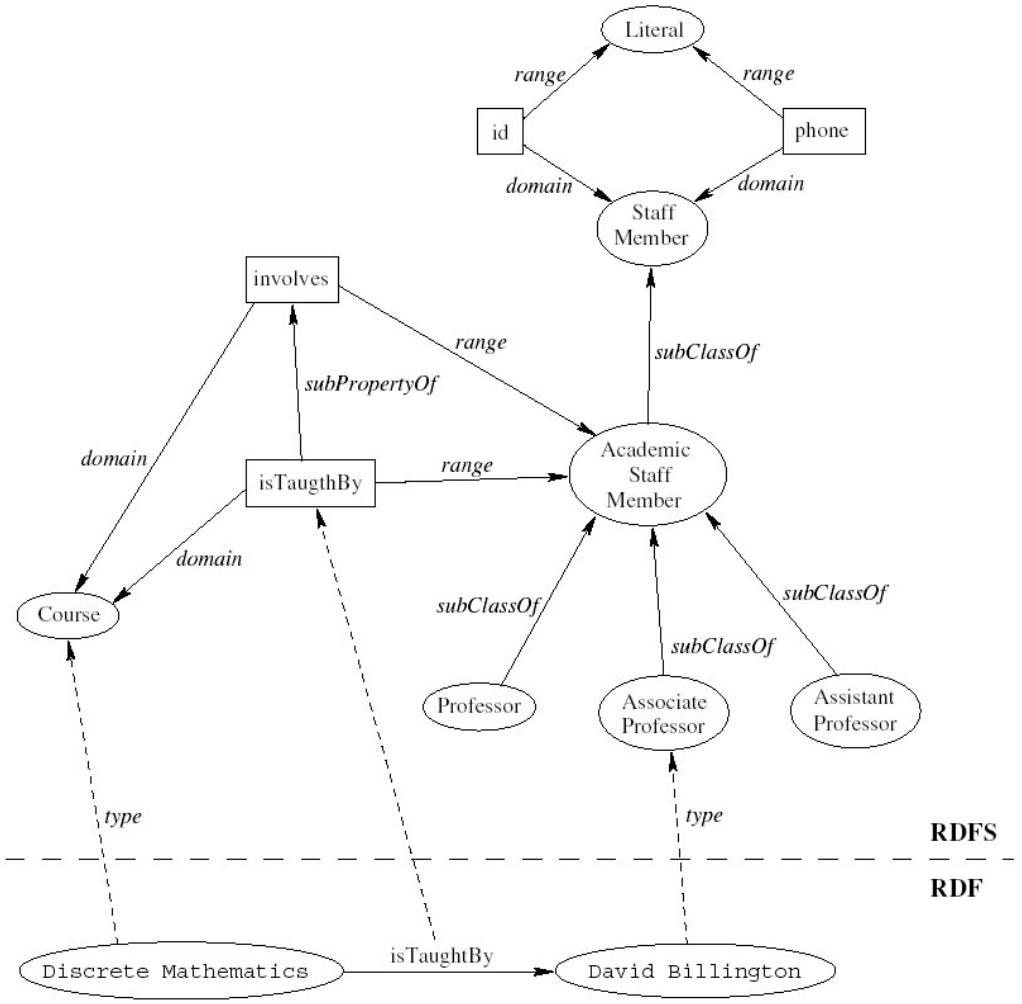

Figura 3. As camadas RDF e RDFS. Fonte: [ANTONIOU e VAN HARMELEN, 2004].

Apesar da expressividade contida no RDF e RDFS esses dois padrões permitem uma representação restrita de conhecimento ontológico. As principais primitivas de modelagem de RDF/RDFS estão concentradas na organização de vocabulários em hierarquias tipadas, como

as relações de subclasses e subpropriedades, as restrições de domínio e a faixa de valores destino, e instâncias de classes. A falta de algumas características é determinante na restrição de sua expressividade semântica, conforme apontam Antoniou e Van Harmelen [2004]:

- O escopo local de propriedades: quando definimos um escopo para uma propriedade, este é aplicado a todas as classes. Em RDFS não é possível declarar um escopo local para ser utilizado por apenas algumas classes, sendo necessário criar uma outra propriedade. Por exemplo, não podemos dizer que girafas comem somente plantas, enquanto que outros animais podem comer carne também;
- A disjunção exclusiva de classes: Algumas vezes desejamos dizer que duas classes são disjuntas e mutuamente exclusivas. Por exemplo, “Masculino” e “Feminino” são classes disjuntas que se excluem mutuamente. Porém, em RDFS podemos somente definir relações de subclasses como, por exemplo, “Masculino” é uma subclasse de “Pessoa”;
- A combinação booleana de classes: Em determinadas situações desejamos construir novas classes pela combinação de outras classes, usando união, intersecção e complementação. Por exemplo, poderíamos definir a classe “Pessoa” como uma união disjuntiva das classes “Masculino” e “Feminino”. Em RDFS esta definição não é possível;
- As restrições de cardinalidade: Muitas vezes gostaríamos de restringir os valores que uma propriedade pode receber. Por exemplo, podemos desejar que um curso seja ministrado por pelo menos um professor. Tais restrições são impossíveis de se expressar em RDFS;
- As características especiais de propriedades: Em alguns momentos é de grande utilidade dizer que uma propriedade é transitiva, única, ou inversa de outra

propriedade. Em RDFS estas características também não são possíveis de ser expressas.

Devido a ausência destas características, o RDF e RDFS possuem limitações na expressividade na modelagem de ontologias originando, assim, as linguagens de ontologia [ANTONIOU e VAN HARMELEN, 2004].

A Camada de Ontologia

As linguagens de ontologia são em sua maioria, extensões de RDF *Schema* com primitivas de modelagem mais ricas. As duas principais frentes de desenvolvimento de linguagens para definição de ontologias foram as OIL (*Ontology Inference Layer*) e a DAML (DARPA *Agent Markup Language*). Estes dois esforços se uniram criando o que se chamou DAML+OIL [CONNOLLY *et al*., 2001], que mais recentemente foi tomada como base para o desenvolvimento da linguagem OWL (*Web Ontology Language*) (McGuinness & Harmelen, 2004). A OWL é a mais recente recomendação da W3C para desenvolvimento da *Web* Semântica, no se refere à definição de ontologias [MCGUINNESS e HARMELEN, 2004], e será tratada mais detalhadamente na Seção 3.3.

As Camadas de Lógica, de Prova e de Confiança

As camadas mais altas da proposta de *Web* Semântica ainda não tomaram consistência. Entretanto, estão surgindo diversas propostas de linguagens para atender os propósitos de cada camada. A camada lógica é usada para enriquecer a linguagem de ontologia, a qual permite a especificação de regras que atuam sobre instâncias e recursos. Para esta camada algumas linguagens de regras foram propostas com o objetivo de descrever condições. Entre as linguagens propostas destacam-se a RuleML (*Rule Markup Language*) [RULEML, 2004], ORL (*OWL Rule Language*) [HORROCKS e PATEL-SCHNEIDER, 2004] e a

SWRL (*Semantic Web Rule Language*) [HORROCKS *et al*., 2004] que foi submetida ao W3C como uma recomendação (W3C *Recommendation*) [NEWTON *et al*., 2004] para criação de linguagens de regras na *Web* semântica.

A camada de prova envolve o processo dedutivo de representação e execução de provas sobre as linguagens que estão em níveis inferiores. Algumas linguagens de prova começam a surgir como, por exemplo, a PML (*Proof Markup Language*) [PINHEIRO DA SILVA *et al*., 2005].

A camada de confiança tem o objetivo de avaliar se a prova está correta ou não. Esta camada se encontra em processo de pesquisa, como pode ser visto em trabalhos como [MCGUINNESS e PINHEIRO DA SILVA, 2004] e [ZAIHRAYEU *et al*., 2005].

Um dos principais objetivos deste trabalho é utilizar a semântica das informações envolvidas nas páginas de um *website* para obter informações mais precisas sobre a intenção do usuário em sua navegação. Acreditamos que, no contexto de sistemas adaptativos, o conhecimento das páginas acessadas por um determinado usuário, juntamente com o conhecimento da semântica de seus conteúdos, podem tornar mais preciso o processo de identificação dos interesses de um usuário. Esta integração procura contemplar o fato de que para toda interação existe uma intenção, ou preferência, referente ao domínio da aplicação. Porém, a extração desta semântica não é uma tarefa trivial. Sendo assim, optamos por modelar a representação da semântica das informações na fase de desenvolvimento do *website*, utilizando o paradigma da *web* semântica.

3.2 As Ontologias

A pesquisa em ontologias na Ciência da Computação teve um notável crescimento nas últimas duas décadas. Um dos principais motivos deste crescimento foi seu importante papel

em áreas como Inteligência Artificial, Lingüística Computacional e Teoria de Banco de Dados.

Devido ao grande interesse atual por ontologias, diversas interpretações surgiram em busca de uma definição abrangente. Em uma tentativa de esclarecimento destas interpretações, Guarino e Giaretta [1995] propuseram uma diferenciação entre a ontologia estudada atualmente pela comunidade de Inteligência Artificial e a ontologia filosófica de Aristóteles e a possibilidade de fornecer uma interpretação formal para as definições mais citadas na área. Segundo Guarino e Giaretta [1995], uma ontologia (com o artigo indefinido e inicial minúscula) diz respeito a um determinado objeto em particular, enquanto Ontologia (sem o artigo indefinido e com a inicial maiúscula) refere-se à disciplina filosófica que lida com a natureza e a organização da realidade. Ontologia é freqüentemente contrastada com Epistemologia. A diferença entre epistemologia e ontologia na representação de conhecimento é discutida em [GUARINO e POLI, 1995], na qual epistemologia é analisada como sendo o campo da filosofia que lida com a natureza e as fontes de conhecimento, enquanto que ontologia é vista como o estudo da organização e da natureza do mundo independentemente da forma de nosso conhecimento a respeito dele. A definição de ontologia encontrada com mais freqüência na literatura da área de computação é a proposta por Grubber [1993]:

> "Ontologia é uma especificação formal e explicita de uma conceitualização compartilhada" [GRUBBER,1993].

Na qual conceitualização representa um modelo abstrato de como os elementos são interpretados no mundo real, normalmente restrito por um domínio particular. Uma especificação explícita significa que os conceitos e relacionamentos do modelo abstrato são determinados por termos explícitos e definições [GRUBBER, 1993]. Desta forma, duas ontologias podem ser distintas quanto ao vocabulário mas, mesmo assim, compartilhar da mesma conceitualização. Guarino e Giaretta [1995] estendem a definição de Gruber dizendo:

> "Uma ontologia é uma descrição parcial e explícita de uma conceitualização." [GUARINO e GIARETTA, 1995].

Portanto, uma ontologia tem o compromisso apenas com a consistência do modelo de uma parte de um determinado domínio, e não com a completude. Esse compromisso é definido como um compromisso ontológico, o qual tem por objetivo filtrar somente o conteúdo relevante sobre um determinado domínio. Segundo Davis *et al*. [1993] esta função é inevitável devido às imperfeições naturais da representação da realidade, como também é parte essencial do que uma representação pode oferecer, devido a grande complexidade do mundo natural.

É evidente o benefício fornecido pela utilização de ontologias como pode ser visto em [GRUBER, 1993], [GRUBER, 1995], [GUARINO,1997a] entre outros. Entretanto, muitos destes benefícios também podem ser obtidos sem o uso de ontologias, talvez com menor eficiência, mas com mesma eficácia. Deste modo, com o intuito de esclarecer as verdadeiras vantagens da utilização de ontologias, Uschold e Gruninger [1996] e Noy e McGuiness [2001] levantam algumas razões que justificam seu uso. Sob um ponto de vista mais específico, Uschold e Gruninger [1996] subdividem o espaço do uso de ontologias em três categorias principais: da comunicação, da interoperabilidade e de sistemas de informação. Com uma visão mais abrangente Noy e McGuiness [2001] apresentam alguns motivos da utilização de ontologias:

- Compartilhar a mesma estrutura de informação entre pessoas ou agentes de software;
- Permitir o reuso do conhecimento do domínio;
- Separar o conhecimento do domínio do conhecimento operacional; e
- Analisar o conhecimento do domínio.

Ainda, segundo Guarino [1997a] as ontologias podem ser classificadas quanto ao seu nível de generalidade. Ele identifica quatro tipos de ontologias cujas dependências estão

ilustradas na Figura 3. A definição destes tipos foi resultado de uma análise realizada em [GUARINO, 1997b], na qual ele aborda a classificação das ontologias em relação ao tipo de estrutura de sua conceitualização.

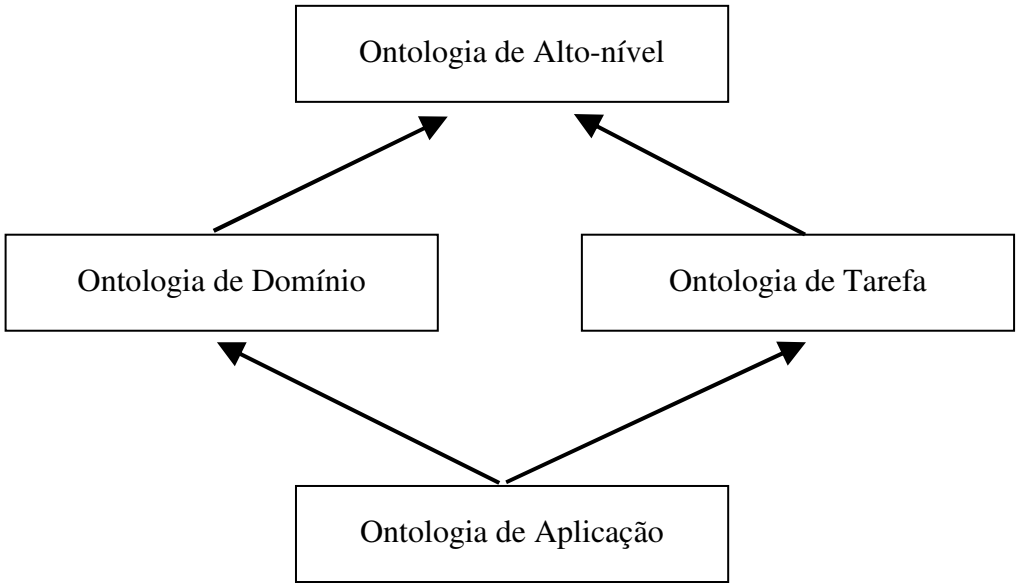

Figura 4 - Tipos de ontologias de acordo com seu nível de dependência.
As setas representam relações de especialização.
Fonte (Guarino, 1997a).

- Ontologias de alto-nível: são compartilhadas por uma grande comunidade e definem apenas termos muito gerais, como espaço, tempo, matéria, objeto, evento, ação, etc., os quais são representados por conceitos que não dependem de um problema ou domínio específico;
- Ontologias de domínio: destinam-se a um determinado domínio de conhecimento, por exemplo, um setor como informática ou medicina;
- Ontologias de tarefas: destinam-se a uma determinada tarefa, por exemplo, diagnóstico médico ou especificação de requisitos. As ontologias de domínio e de tarefa especializam os termos introduzidos na ontologia de alto-nível.
- Ontologias de aplicação: destinam-se a uma determinada aplicação, descrevendo conceitos especializados de uma ontologia de domínio e de tarefa. Estes conceitos

correspondem, freqüentemente, a funções exercidas por entidades de domínio enquanto executam uma determinada atividade.

A ontologia desenvolvida em nosso trabalho se encaixa nas ontologias de domínio. Porém, como o processo de construção de uma ontologia é um processo criativo, não existindo uma única forma de modelagem, a utilização de uma ontologia de alto-nível, seguida de uma especialização para uma ontologia de domínio, seria uma boa alternativa para tentar resolver a falta de padrão entre as ontologias. Esta classificação também nos mostra que, quando desenvolvemos aplicações baseadas no conhecimento do domínio, ou seja, sobre a ontologia do domínio, não utilizamos o conhecimento por completo, utilizamos apenas a parte referente à aplicação. Desta forma, segmentamos a ontologia de domínio, resultando na ontologia de aplicação, que possui somente os conceitos utilizados na aplicação. Esta diferença entre ontologias de domínio e aplicação, apontada nesta classificação, contribuiu para definirmos alguns parâmetros que influenciam na expressão dos interesses dos usuários, pois ela mostra uma restrição no vocabulário que o usuário pode utilizar sobre o domínio. Estes parâmetros serão tratados mais detalhadamente no Capítulo 5.

Desta forma, fica evidente que o uso de ontologias tem papel fundamental tanto para aquisição de conhecimento quanto para a criação e o desenvolvimento de software. Ela será usada neste trabalho para a modelagem do conhecimento do domínio, que servirá de base para o desenvolvimento do *website* semântico.

3.3 A Linguagem OWL (*Web Ontology Language*)

A proposta da *Web* Semântica envolverá a capacidade da linguagem XML em definir esquemas (estruturas) e a flexibilidade da linguagem RDF em representar dados, porém, sobre a RDF é necessário uma linguagem para a descrição de ontologias para que se possa descrever formalmente os significados das terminologias utilizadas em documentos *web*.

A linguagem OWL [MCGUINNESS e HARMELEN, 2004] é uma linguagem de definição semântica recomendada pela W3C para a criação, publicação e compartilhamento de ontologias na *Web* Semântica. Ela é desenvolvida como uma extensão do vocabulário RDF e é derivada da DAML+OIL [CONNOLLY *et al*., 2001]. Ela possui maior poder para expressar a semântica do que a RDF e RDFS, superando, assim, estas linguagens na definição semântica do conteúdo na *web*. Dessa forma, é possível que programas (ou agentes de software) capazes de executar raciocínio lógico, chamados de raciocinadores, se beneficiem do formalismo semântico desta linguagem para que possam raciocinar e inferir informações sobre a ontologia. A linguagem OWL é subdividida em três sublinguagens [MCGUINNESS e HARMELEN, 2004]:

- A OWL *Lite* é a sublinguagem menos expressiva, que oferece um apoio básico aos usuários que necessitam apenas classificar uma hierarquia de classes e definições simples de propriedade. Sua vantagem é que o raciocínio sobre ela fica ainda mais simples e a desvantagem é a expressividade limitada;
- A OWL DL fornece apoio aos usuários que necessitam de uma boa expressividade sem perder processamento computacional e o auxílio de raciocinadores para realizar inferências. A OWL DL é nomeada assim devido a sua correspondência com lógicas descritivas (***D**escription **L**ogic*. [BAADER *et al*., 2002]. Ela foi projetada para prover alta expressividade, decidibilidade e suporte a um raciocínio eficiente. Uma desvantagem desta sublinguagem é a perda de total compatibilidade com a RDF. Desta forma todo documento em OWL DL é um documento em RDF, mas nem todo documento em RDF é um documento em OWL DL;
- A OWL *Full* é a mais expressiva entre as sublinguagens. Ela oferece apoio aos usuários que necessitam de expressividade máxima e a liberdade sintática do RDF, porém, sem garantias computacionais. Sua vantagem é a completa compatibilidade

com RDF, tanto sintática como semanticamente. Toda essa expressividade, entretanto, causa algumas desvantagens como a indecidibilidade e a ausência de um apoio de raciocínio completo e eficiente.

A OWL *Lite* é um subconjunto da OWL DL, que, por sua vez, é um subconjunto da OWL *Full*. A escolha entre as sublinguagens depende da expressividade requerida pelo usuário. Em nosso trabalho escolhemos a linguagem OWL DL para modelar o conhecimento do domínio. Primeiramente, porque existem ótimos raciocinadores genéricos que oferecem apoio que, certamente, facilitarão nosso trabalho, como o RACER [HAARSLEV, 2005] e o PELLET [PARCIA e SIRIN, 2003]. E, segundo, porque a linguagem OWL é a linguagem recomendada pela W3C para *Web* Semântica, no que se refere a definição de ontologias.

Neste capítulo mostramos o contexto para o qual estamos desenvolvendo este trabalho. Apresentamos o conceito de sistemas adaptativos e sua necessidade de informações sobre os usuários, quais as características associadas aos usuários que podem ser consideradas no modelo do usuário e como capturar essas informações. Além disso, apresentamos uma breve definição da *web* semântica, os padrões para o desenvolvimento de um *website* semântico, o conceito de ontologia, e as principais linguagens utilizadas para a construção de ontologias. No próximo capítulo, vamos apresentar alguns trabalhos correlatos que nos permitiram a identificação de características importantes para o desenvolvimento de nosso trabalho.

Capítulo 3

TRABALHOS RELACIONADOS

Este capítulo apresenta alguns trabalhos relacionados encontrados na literatura que foram tomados como base para o desenvolvimento do trabalho aqui proposto. A análise destes trabalhos permitem a identificação de características importantes sobre como integrar padrões navegacionais e conteúdo semântico, mostrado no SEWeP; a forma de como os padrões de visitas são extraídos, mostrado no *IndexFinder*; e as características consideradas na adaptação de um *website* semântico, mostrado no trabalho do Mikroyannidis e Theodoulidis (2005).

1 SEWeP (*Semantic Enhancement for Web Personalization*)

O projeto SEWeP é um sistema de personalização *web* que utiliza *logs* de uso e a semântica do conteúdo de um *website* para personalizá-lo. O processo de personalização se baseia em um conjunto de recomendações ao usuário com possíveis adaptações. Estas recomendações são apresentadas em forma de *links* que poderão ser inseridos dinamicamente

na página que o usuário está visitando. A principal característica do SEWeP é a criação de C-*logs* (*concept-logs*) a partir de arquivos de *log* comuns. Os C-*logs* são *logs* enriquecidos semanticamente baseados em uma taxonomia criada para modelar conceitos do domínio. Cada registro do arquivo de *log* é aprimorado com palavras-chaves representando a semântica do respectivo URI. A extração destas palavras-chave é realizada por meio de técnicas de mineração de dados que separam a estrutura da página e analisam somente seu conteúdo. Encontrada as palavras-chave, é realizado um mapeamento para os conceitos da taxonomia, e para cada página é associada um ou mais conceitos. Desta forma, a cada página do *website* visitada são agregados um ou mais conceitos do domínio, com base nas palavras-chave encontradas no conteúdo das páginas.

Os C-*logs* são usados como entrada para algoritmos de mineração de dados, resultando em padrões de comportamento navegacional do usuário na forma de regras de associação. Estas regras de associação estão fundamentadas nos conceitos do domínio, os quais foram agregados no processo de enriquecimento do *log*. A partir destas regras de associação a máquina de adaptação pode fazer recomendações baseadas nos padrões encontrados [EIRINAKI e VAZIRGIANNIS, 2003].

Em nosso trabalho, utilizamos uma das principais características do SEWeP, a idéia agregação da semântica das informações a comportamentos navegacionais dos usuários. Porém, o processo de extração de palavras-chave para valoração semântica das páginas não se enquadra no escopo de nosso trabalho, pois optamos pelo desenvolvimento de um *website* com apoio semântico. Assim, todo conteúdo do *website* está diretamente relacionada a uma ontologia construída especificamente para modelar o conhecimento do domínio em questão. O conteúdo das páginas *web* em nosso trabalho serão instâncias desta ontologia. Desta forma, devido ao fato de que as ontologias detêm um grande poder de expressividade semântica, como descrito no Capítulo 2, não será necessária a agregação de mais semântica às páginas.

E, por reduzirmos o escopo de nosso trabalho a *websites* do tipo portais de *intranet*, optamos por criar um *log* semântico personalizado e não enriquecer semanticamente o arquivo de *log* comum.

2 *IndexFinder*

O *IndexFinder* é um sistema que adapta *websites* a partir da criação semi-automática de páginas *index* baseado em padrões de acesso dos visitantes. Este processo semi-automático busca, principalmente, solucionar o problema da criação de novas páginas *index* com base em informações obtidas pela freqüência dos acessos das páginas. Assume-se que se um grande número de visitantes, freqüentemente, visita um conjunto de páginas, existem fortes evidências de que estas páginas estejam relacionadas. Desta forma, o *IndexFinder* sugere páginas candidatas para a página *index*. Quando é gerada uma página candidata, o conteúdo é apresentado ao projetista, que pode aceitá-la ou rejeitá-la [PERKOWITZ e ETZIONI,2000].

O *IndexFinder* tem como entrada os arquivos *log* de acessos e uma descrição conceitual de cada página, criada pelo seu projetista por meio da extração de dados sobre a freqüência com que as páginas são visitadas. A partir destes dados, é aplicada uma técnica de mineração de dados para produzir grupos de páginas, possivelmente relacionadas, baseadas na similaridade encontrada nas visitas. Posteriormente, o *IndexFinder* utiliza a descrição conceitual das páginas do *website*, fornecida pelo seu projetista, para transformar esses grupos em conceitos coerentes. Por último, esses conceitos são mapeados em forma de páginas candidatas à página *index* [PERKOWITZ e ETZIONI,2000].

Este projeto realiza grandes esforços para identificar a semântica que está implícita entre os conteúdos das páginas. A semântica é identificada por meio da freqüência com que essas páginas são visitadas, ou seja, acredita-se que se existe um grande número de acessos freqüentes a um determinado grupo de páginas, possivelmente elas possuem alguma relação

semântica entre elas. Este esforço para explicitação da semântica não será necessário para o nosso trabalho, devido à adoção de ontologias para modelar o domínio. Entretanto, as técnicas utilizadas para a extração de informações, como a freqüência em que as páginas são visitadas e os padrões de visita, são de extrema importância para o nosso trabalho.

3 Adaptação da *Web* Semântica baseado em dados de uso

Mikroyannidis e Theodoulidis (2005) propõem em seu trabalho, uma forma de adaptação da *web* semântica utilizando dados de uso. Eles exploram o aspecto semântico do *website* para tentar melhorar a reorganização das informações. Uma ontologia do *website* foi construída considerando a organização definida pela topologia do *website*, definindo as categorias temáticas cobertas pelas páginas do *website*. Estas categorias representam os conceitos na ontologia. Assim, cada página *web*, dependendo de seu conteúdo, é uma instância de um ou mais conceitos da ontologia. Os conceitos podem ser organizados em uma hierarquia, representando uma relação "*is a*", de forma que uma classe seja subclasse de outra classe [MIKROYANNIDIS e THEODOULIDIS, 2005].

O processo de adaptação neste sistema leva em consideração os arquivos de *log* do servidor *web* e a topologia e a ontologia do *website*. Os arquivos de *log* são pré-processados, quando são identificadas as sessões de visitas e os *pagesets* (conjunto de páginas que são freqüentemente acessadas juntas na mesma sessão). A partir dos *pagesets*, dois critérios de adaptação são considerados: as ligações dos *links* das páginas contidas no *pageset* e o conteúdo destas páginas.

O primeiro critério refere-se à conexão que as páginas de cada *pageset* possuem de acordo com a estrutura do site. A principal característica deste critério é se as páginas contidas em um *pageset* estão diretamente ligadas a outras ou não. Os *pageset* de páginas não ligadas poderiam sugerir a inserção de *links* de atalhos entre essas páginas para alcançar caminhos de

navegação mais curtos. Já os *pageset* de páginas ligadas poderiam sugerir mudanças na aparência de *links* existentes. Por exemplo, se uma página *index* e algum de seus *links* estão inclusos em um ou mais *pagesets*, então realçando estes *links* na página *index*, na primeira visita dos visitantes, a navegação no *website* poderia ser mais fácil [MIKROYANNIDIS e THEODOULIDIS, 2005].

O segundo critério de classificação refere-se ao conteúdo das páginas contidas em cada *pageset*. As páginas do *pagesets* são classificadas para descobrir novas associações entre os conceitos da ontologia do *website*. Particularmente, se um *pageset* incluir páginas que pertencem a conceitos que previamente não foram ligados, a ontologia deveria ser modificada para refletir a relevância que estes conceitos têm, de acordo com as preferências dos usuários [MIKROYANNIDIS e THEODOULIDIS, 2005].

O aspecto semântico explorado neste trabalho está relacionado mais a estrutura do *website* do que a semântica presente nas informações. A ontologia utilizada como base de conhecimento para o *website* é organizada como uma taxonomia conforme a estrutura do *website*. A ontologia do *website* utilizada modela o mapa do *website*, ou seja, os conceitos presentes no primeiro nível da ontologia correspondem aos menus do menu da página *index* e assim por diante. Quando um conceito é identificado com um alto nível de relevância, o processo sugere a transferência deste conceito para o primeiro nível da ontologia, facilitando, assim, seu acesso. Esta ontologia do *website* difere da ontologia do domínio, uma vez que a ontologia do domínio modela o conhecimento do domínio, independentemente de como essas informações são estruturadas ou utilizadas por uma determinada aplicação.

Deste modo, o processo de integração de semântica aos dados de uso dos usuários utiliza a semântica presente na estrutura do *website* e nas preferências do usuário, não abordando a semântica presente nas informações. Porém, a forma de extrair os dados de uso do usuário, por meio do arquivo de *log*, e a forma com que a ontologia se relaciona com o

website contribuíram para a definição de algumas características de nosso trabalho. No entanto, o foco de nosso trabalho é um pouco diferente, nos propomos a trabalhar com *websites* do tipo portais de *intranet*. Nós decidimos restringir o tipo de *website*, para podermos identificar os usuários de forma individual. A semântica abordada em nosso trabalho se refere à semântica das informações, ou seja, como essas informações se relacionam com o restante do domínio. Para isto, desenvolvemos uma ontologia do domínio como base para o *website* semântico. Entretanto, a forma com que este trabalho de Adaptação da *Web* Semântica se preocupa com a estrutura do *website*, nos chamou a atenção a considerar que a forma com que as informações estão estruturadas no *website* pode fazer diferença na modelagem das intenções do usuário.

Neste capítulo apresentamos um levantamento de características importantes para identificação de interesses de usuários que encontramos em trabalhos correlatos, quais destas características adotamos para nosso trabalho e quais as diferenças de nossa abordagem para os trabalhos analisados. No próximo capítulo, vamos mostrar como integramos as informações semânticas do conteúdo acessado a comportamentos navegacionais do usuário.

Capítulo 4

INTEGRANDO PADRÕES NAVEGACIONAIS E CONTEÚDO SEMÂNTICO

A navegação de um usuário em um *website* está fortemente relacionada às suas necessidades, interesses e conhecimentos. Deste modo, diversos pesquisadores têm explorado os dados de navegação presentes em arquivos de *log* para obter informações sobre a forma com que os usuários navegam em um *website* [EIRINAKI e VAZIRGIANNIS, 2003] [BRUSILOVSKY, 2001] [LEI *et al*, 2004]. Tais arquivos possuem valiosas informações referentes ao comportamento navegacional dos usuários, entretanto, eles não possuem informações semânticas que descrevam as intenções destes usuários quando realizaram sua navegação no *website*. Neste trabalho, tentamos amenizar este problema criando um *log* semântico que agrega informações semânticas do conteúdo do *website* as interações do usuário.

É importante ressaltar que, no escopo deste trabalho, estaremos limitando nossa discussão a *websites* constituídos de portais do tipo *intranets* que tenham sido construídos com base semântica desde o princípio. Sabemos que o tratamento dos *websites* construídos com tecnologia atual é muito relevante, mas ele não será alvo deste trabalho.

Assim, antes de tratarmos do problema de identificação das intenções dos usuários é necessário definirmos como se pode incorporar uma base semântica a um *website* e como se pode gerar um *log* semântico baseado nas interações do usuário e nas informações semânticas do *website*.

1 A Estruturação de um *website* semântico

Para a construção de um *website* com suporte semântico se faz necessário adicionar uma representação da semântica das informações contidas em suas páginas. Esta representação da semântica precisa ser disponibilizada por meio de um modelo conceitual que possa ser processado por agentes de software. Em um nível conceitual, pode haver vários tipos diferentes de objetos contidos em um *website* que são acessíveis aos usuários. Em um nível físico, estes objetos podem ser representados por uma ou mais páginas *web* ou por fragmentos destas.

Como estudo de caso empregaremos o *website* do Departamento de Informática (DIN) da Universidade Estadual de Maringá. Assim, desenvolvemos um novo *website* baseado no *website* atual do DIN, porém, com uma representação da semântica das informações nele contidas. O *website* do DIN possui páginas relacionadas a professores, projetos de pesquisa, publicações, disciplinas, etc. Conceitualmente, cada uma destas entidades representa um tipo diferente de objeto semântico. Durante uma visita a este *website*, um usuário pode acessar vários destes objetos juntos em uma única página. Com a representação da semântica da

informação contida no *website* é possível ter uma arquitetura uniforme para modelar tais objetos, suas propriedades e suas relações.

Para modelar as relações semânticas de um *website*, utilizamos as tecnologias definidas pelo W3C para o desenvolvimento da *web* semântica, discutidas no Capítulo 2, e passamos a denominar nosso *website* de ***website* semântico**.

Dentre as camadas definidas para a *web* semântica [KOIVUNEN e MILLER, 2001], é na camada de ontologia que construímos um modelo conceitual do domínio, no qual definimos formalmente os conceitos e relações existentes no domínio e, consequentemente, possibilitamos o compartilhamento das informações.

1.1 A modelagem da Ontologia do Domínio

Nesta seção, apresentamos a metodologia empregada na construção de uma ontologia para um *website*, empregando como linguagem de representação de conhecimento a linguagem OWL, seu desenvolvimento e sua aplicação sobre um domínio.

1.1.1 A Metodologia de desenvolvimento

O desenvolvimento de uma ontologia é um processo criativo, sendo que raramente duas ontologias para um mesmo domínio serão iguais quando modeladas por pessoas diferentes. Os prováveis usos da ontologia e, a visão e a compreensão, que o projetista tem do domínio afetam as decisões do projeto. Deste modo, a avaliação da qualidade de uma ontologia só será possível por meio de seu uso em aplicações para as quais ela foi desenvolvida [NOY e MCGUINESS, 2001].

Segundo Noy e McGuinness [2001], não se deve imaginar que exista apenas uma maneira correta para se definir uma ontologia, assim como não existe apenas um caminho correto para se modelar um domínio: sempre existirão várias alternativas viáveis. A melhor

solução, freqüentemente, depende da aplicação que se tem em mente e das extensões que foram previstas para o modelo. Além disso, o desenvolvimento de ontologias é, necessariamente, um processo iterativo, o que, freqüentemente, resulta no desenvolvimento de uma primeira versão com melhoramentos sucessivos, baseados na experiência de utilização da versão anterior. Desta forma, nos baseamos no processo e diretrizes apontadas por Noy e McGuinness [2001] para a etapa de definição da Ontologia do domínio do DIN. Um breve resumo dos passos empregados é apresentado a seguir:

1) **Determinar o escopo e o domínio da ontologia**: uma das formas de determinar o escopo de uma ontologia é a elaboração de um questionário de competência, o qual a ontologia deve estar apta a responder.
2) **Avaliar o reuso de ontologias existentes**: o reuso de ontologias existentes pode ser um requisito quando o sistema precisar interagir com outras aplicações que já tenham um contrato estabelecido com uma ontologia em particular ou um vocabulário controlado.
3) **Relacionar os termos importantes da ontologia:** pode ser útil escrever uma lista com todos os termos cuja definição ou explicação ao usuário é desejável? Quais são os termos com que se gostaria de trabalhar? Quais propriedades eles têm? O que se deseja dizer sobre esses termos?

Os próximos dois passos — definir a hierarquia de classes e definir as propriedades ou os conceitos — estão inter-relacionados. É difícil fazer apenas um e depois fazer o outro. Geralmente, são criadas algumas definições para os conceitos na hierarquia e, então, são descritas as propriedades desses conceitos, e assim por diante. Esses dois passos também são os mais importantes do processo de projeto de uma ontologia.

4) **Definir as classes e a hierarquia de classes**: existem várias abordagens para o desenvolvimento de uma hierarquia de classes, tais como o processo de

desenvolvimento *top-down*, *bottom-up* e uma combinação dos dois. Porém, independente da abordagem escolhida, o processo começa com a definição de classes. A partir da lista criada no terceiro passo, os termos que descrevem os objetos serão classes na ontologia e devem se tornar âncoras na hierarquia de classes. Para organizar as classes em uma taxonomia hierárquica, é necessário perguntar se todas as instâncias de uma subclasse também devem ser instâncias da superclasse. Se A é uma superclasse de B, então cada instância de B é também uma instância de A. Em outras palavras, a classe B é um conceito que é um tipo do conceito A.

5) **Definir as propriedades das classes**: as classes por si só não fornecerão informações suficientes para responder às questões de competência do primeiro passo. Uma vez que algumas classes tenham sido definidas, deve-se descrever a estrutura interna dos conceitos. Na lista de termos criada no terceiro passo, alguns elementos tornaram-se classes. Muitos dos termos restantes, provavelmente, serão propriedades das classes. Para cada propriedade na lista deve-se determinar qual classe que ela descreve. Essas propriedades tornam-se relacionamentos anexos às classes.
6) **Definir as características das propriedades:** as propriedades podem ter características diferentes, sendo mais comuns: a cardinalidade (define quantos valores serão válidos para a propriedade), o tipo de dado (descreve qual o tipo de dado válido para a propriedade), o domínio e a faixa (o conjunto de classes que contêm certa propriedade do tipo instância é chamado de **domínio** da propriedade e o conjunto de classes permitidas para uma propriedade do tipo instância é chamado de **faixa** (*Range*) da propriedade).

Esta metodologia foi adaptada na definição da ontologia do DIN e foi implementada na linguagem OWL para modelar o conhecimento que será empregado na construção do *website* semântico.

1.1.2 A Linguagem e Ferramenta adotadas na concepção da ontologia

A ontologia do domínio do DIN foi desenvolvida na linguagem OWL (*Web Ontology Language*), a qual é aceita como um padrão de linguagem, para construção de ontologias na *web* semântica [W3C, 2004]. A OWL é dividida em três sublinguagens (a OWL *Lite*, a OWL DL e a OWL *Full*), distintas pelo nível de formalidade exigido e a liberdade dada ao usuário para a definição de ontologias. Para nossa ontologia, optamos pelo uso da sublinguagem OWL-DL, por esta possuir raciocinadores que facilitam a verificação de inconsistências na definição da ontologia e pela possibilidade da descrição de relacionamentos entre as classes utilizando expressões da lógica descritiva.

Para a construção e edição da ontologia do DIN utilizamos a ferramenta Protégé [PROTÉGÉ, 2006], que é uma plataforma de código-aberto extensível e customizável por meio de *plugins*, desenvolvida para modelar ontologias e para aquisição de conhecimento. Em particular, foi utilizado o *plugin* Protégé OWL, que é uma extensão complexa do Protégé para a manipulação de arquivos na linguagem OWL e para a criação de classes, propriedades e instâncias da ontologia [KNUBLAUCH *et al*, 2004]. Uma das principais vantagens oferecidas pelo Protégé, juntamente com o *plugin* Protégé OWL, é a possibilidade de descrever classes complexas baseadas na lógica descritiva, expressando condições e restrições de propriedade que as instâncias da classe devem obedecer.

O *plugin* Protégé OWL provê um acesso direto a raciocinadores, tais como o *Racer* [HAARSLEV, 2005] e o *Pellet* [PARCIA e SIRIN, 2003]. A definição formal de primitivas em OWL pode ser explorada por estes raciocinadores, realizando tarefas como a detecção de

inconsistências lógicas na ontologia e a classificação automática das classes de acordo com suas definições lógicas. O raciocinador *Racer* [HAARSLEV, 2005] foi empregado no processo de desenvolvimento da ontologia do DIN para verificar se as definições lógicas das classes estavam de acordo com o esperado, e para ter certeza de que não havia nenhuma inconsistência na ontologia. Este recurso também será de grande utilidade em tarefas futuras como na manutenção, atualização e ampliação da ontologia.

Para entender melhor a relação entre o Protégé e o *plugin* Protégé OWL, a Figura 5 ilustra como é feita a extensão do *plugin* sobre o núcleo do sistema Protégé. O *plugin* Protégé OWL é composto, basicamente, por dois módulos: o Protégé OWL GUI e o Protégé OWL API. O primeiro é responsável pela implementação das interfaces de interação com o usuário como, por exemplo, o editor de expressões de lógica descritiva em OWL. Estas interfaces atuam sobre o módulo Protégé OWL API que refletem as ações do usuário na ontologia. O segundo é uma API na linguagem Java que possui classes e métodos específicos para acessar e manipular ontologias em OWL. Entretanto, ela utiliza uma outra API, chamada Jena, para serializar os objetos em um arquivo. A API Jena [JENA, 2006] foi desenvolvida pela *Hewlett-Packard* e será discutida com maiores detalhes nas próximas seções.

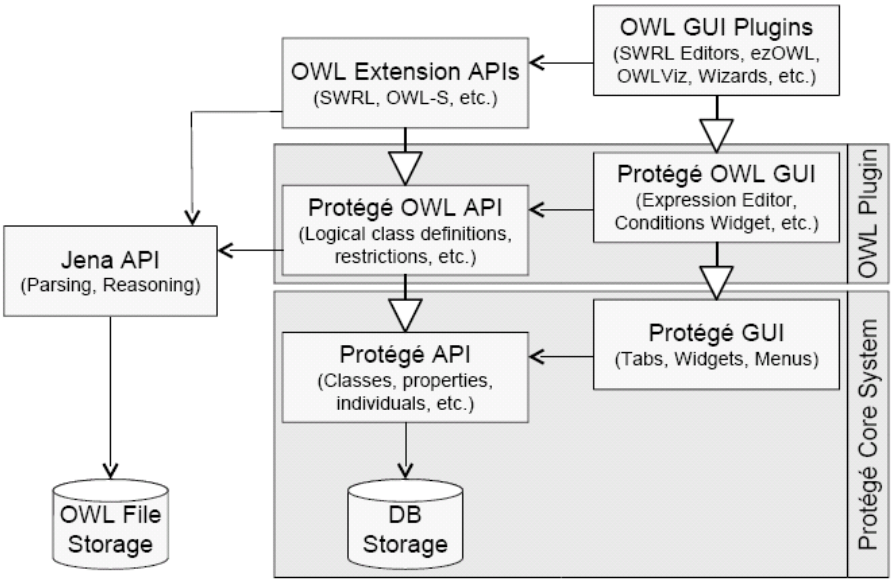

Figura 5. O *plugin* OWL é uma extensão do Protege.

1.1.3 O Desenvolvimento da Ontologia

Visando a descrição de características do Departamento de Informática da Universidade Estadual de Maringá, primeiramente, realizamos um levantamento a respeito dos conceitos que deveriam ser obtidos para a descrição do departamento, ou seja, o vocabulário utilizado e quais características e serviços são geralmente disponibilizados. Para isso, tomamos como base o *website* atual do DIN definindo os conceitos presentes na ontologia e sua hierarquia. Em seguida, definimos as propriedades, e suas características, para cada classe e, consequentemente, as relações existentes entre os conceitos. Por último, realizamos a instanciação dos conceitos da ontologia, tornando-a apta para ser utilizada. A estrutura dos conceitos foi definida com um total de 57 conceitos, entre classes e subclasses, as quais estão ilustradas na Figura 6.

Para definir uma propriedade precisamos determinar um domínio, que é a classe a que ela pertence, e uma faixa (*range*) que é o tipo da propriedade, ou seja, a faixa de valores que a propriedade aceitará. As propriedades OWL são divididas em dois tipos: propriedades de objetos e propriedades de dados. As propriedades de dados definem um tipo para a propriedade em questão. Esses tipos podem ser booleano, *float*, *string*, inteiro, entre outros. Podemos descrever mais características das propriedades de dados como, por exemplo, definir se a propriedade é funcional e inversamente funcional. Uma propriedade é funcional quando é permitida a instanciação de somente um valor para ela e é inversamente funcional quando o valor instanciado pode pertencer somente a uma instância da classe. Por exemplo, em nossa ontologia a classe Pessoa possui uma propriedade de dado do tipo *string* para armazenar o CPF. Cada instância da classe pessoa pode ter somente um CPF, e um CPF pode pertencer somente a uma instância da classe Pessoa. Assim, definimos a propriedade CPF sendo funcional e inversamente funcional.

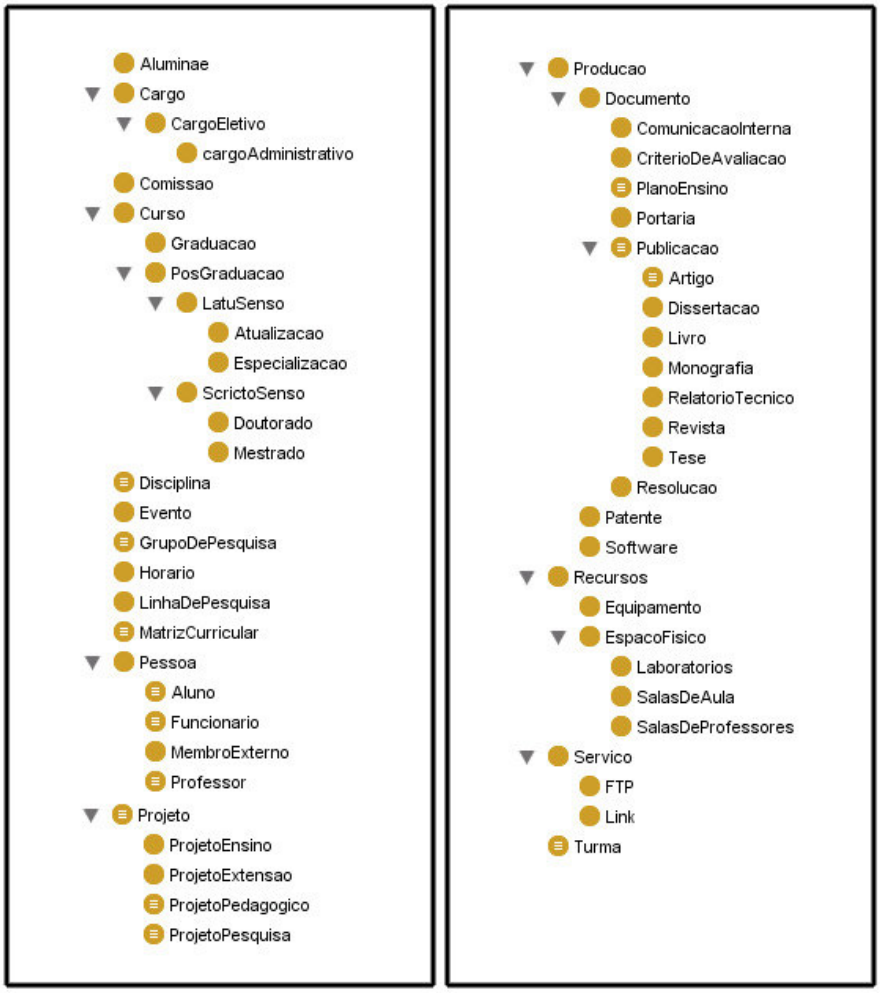

Figura 6. Estrutura dos Conceitos da Ontologia

Nas propriedades de objeto definimos a faixa da propriedade com uma classe, ou seja, as instâncias de uma classe. Desta forma, conseguimos definir uma relação entre a classe definida no domínio e a classe definida na faixa da propriedade. Por exemplo, a classe “Professor” possui uma propriedade de objeto “ministra”, essa propriedade possui como faixa as instâncias da classe “Turma”. Assim, temos o seguinte relacionamento: “Professor” “ministra” uma “Turma”. Podemos inferir, ainda, que uma “Turma” é “ministradaPor” um “Professor” se definirmos a propriedade como inversa, ou seja, a propriedade “ministra” é inversa a “ministradaPor”.

Outros tipos de restrição foram feitos para as propriedades de objeto tais como: cardinalidade, simetria e transitividade. Entre as propriedades de dados e objetos definimos no

total 59 propriedades na ontologia, como mostra a Figura 7. Para cada propriedade foi analisada a necessidade de se adicionar mais características, restringindo sua faixa de valores.

Figura 7. Propriedades da Ontologia

Além da descrição das propriedades, também descrevemos as classes inserindo restrições a elas. Para que uma instância pertença a uma classe, ela deve satisfazer as restrições impostas pela classe. As restrições são criadas por meio da lógica descritiva e das propriedades definidas. As restrições em OWL são divididas em três categorias: Restrições de Quantificadores, Restrições de Cardinalidade e Restrições de Valor. A categoria mais utilizada na descrição das classes foram as Restrições de Quantificadores. Por exemplo, restringimos a classe “Grupo de Pesquisa” com as seguintes restrições ilustradas na Figura 8.

Desta forma, estamos dizendo que a classe “Grupo de Pesquisa” possui ao menos uma relação “possuiCoordenador” com a classe “Professor” e que todas essas relações são, obrigatoriamente, com a classe “Professor”. Estamos dizendo também, que ela possui ao menos uma relação “possuiIntegrantes” com a união das classes “Aluno”, “Professor” e “Membro Externo” e que todas as relações “possuiIntegrantes” podem ser somente com a união destas classes. Resumindo, dizemos que um grupo de Pesquisa deve ter um ou mais coordenadores e integrantes; o coordenador deve obrigatoriamente ser um professor; e o integrante obrigatoriamente um aluno, um professor ou um membro externo.

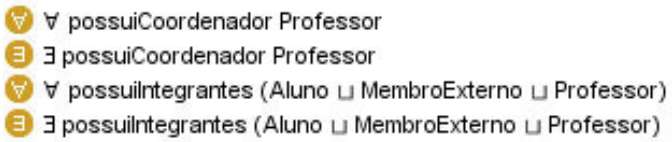

Figura 8. Restrições da Classe “Grupo de Pesquisa”.

A instanciação da ontologia foi feita por meio da inserção das instâncias utilizando o próprio Protégé. Todos os objetos na ontologia sejam eles uma classe, propriedade ou instância possuem um identificador de recursos universal, chamado *Uniform Resource Indicator* (URI) que o identifica de forma única. O URI é composto pelo *namespace* da ontologia mais o objeto. Por exemplo, "http://www.din.uem.br/ontologia/Professor", é o URI que identifica a classe Professor, na qual "http://www.din.uem.br/ontologia/” é o *namespace* atribuído à ontologia e “Professor” o nome da classe. Assim com o URI do objeto podemos identificá-lo na ontologia, independentemente de que tipo ele seja.

As definições desenvolvidas na ontologia do DIN darão embasamento semântico necessário para o desenvolvimento do *website* semântico. É importante ressaltar que a modelagem de uma ontologia requer um processo extenso de iterações e dada à complexidade do domínio, pode ser que a mesma não aborde o domínio em sua completude. Entretanto, ainda que a ontologia do DIN não tenha sido desenvolvida abrangendo toda extensão do

domínio, a definição das características e restrições para as propriedades e relacionamentos entre as classes modeladas é de extrema relevância, sendo o ponto inicial para que aplicativos possam realizar inferências sem o auxilio humano.

1.2 Desenvolvimento do *Website* Semântico

Com o desenvolvimento e instanciação da ontologia do DIN, o passo seguinte é exibir essas informações no *website* tornando-o, assim, um *website* semântico. O *website* semântico do DIN foi desenvolvido baseado na arquitetura proposta por Holger Knublauch [KNUBLAUCH *et al*, 2004] com algumas adaptações para atender as necessidades de nossa aplicação. O esquema final é ilustrado pela Figura 9.

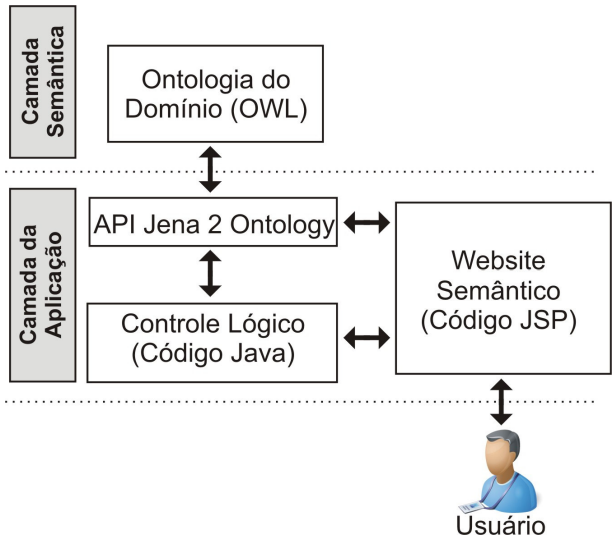

Figura 9. Mapeamento da ontologia para o *website*.

No esquema da Figura 9, consideramos uma camada semântica e uma camada de aplicação. A camada semântica armazena a estrutura conceitual dos dados na forma de uma ontologia em OWL. A camada de aplicação é composta por três módulos: o controle lógico da aplicação, a API para manipulação de ontologia e o *website* semântico.

Como tecnologia base para o desenvolvimento da aplicação, escolhemos a linguagem de programação orientada a objetos Java [JAVA, 2006], devido às suas características de

orientação a objeto, portabilidade, e uma grande quantidade de bibliotecas (API) de alta qualidade. Para a dinâmica geração das páginas do *website* semântico utilizamos a linguagem *Java Server Pages* (JSP) [JSP, 2006]. Esta linguagem permite embutir ou acessar códigos na linguagem Java, facilitando a comunicação entre os demais módulos da aplicação.

Como pode ser observado na Figura 9, todas as informações contidas nas páginas são retiradas da ontologia do domínio, sendo, normalmente instâncias dos conceitos modelados. Para isto utilizamos uma API de manipulação de ontologia para que o *website* semântico tenha acesso à ontologia modelada em OWL. Nesta camada poderíamos utilizar tanto a API Protégé OWL, quanto a API Jena 2 *Ontology*, pois ambas possuem a mesma função. Escolhemos a API Jena 2 *Ontology* devido a suas características, tais como: a serialização dos objetos da ontologia em arquivos, a possibilidade de persistência da ontologia em um banco de dados e a existência de uma linguagem de consulta para OWL (chamada SPARQL).

A API Jena 2 *Ontology* faz parte do *framework* Java para desenvolvimento de aplicações para *Web* Semântica chamado Jena [JENA, 2006]. O *framework* Jena fornece um ambiente de programação para as linguagens RDF, RDFS e OWL, além da linguagem de consulta SPARQL e uma máquina de inferência baseada em regras. Assim, esta API acessa as informações da ontologia fazendo um mapeamento das instâncias, propriedades ou classes (descrita na linguagem OWL) para objetos na linguagem Java, sem a perda das características semânticas da ontologia. Tanto o módulo de controle lógico da aplicação, quanto o *website* utilizam a API Jena 2 *Ontology* para acessar e manipular os dados da ontologia.

No módulo de controle lógico da aplicação, desenvolvemos classes e métodos na linguagem Java para padronizar as formas de manipulação da ontologia. Desta forma, conseguimos centralizar todo o controle da manipulação neste módulo, visando facilitar posteriores modificações e atualizações. Assim, para alterar alguma característica na forma de acesso à ontologia não será necessário alterar as páginas do *website*, mas sim os métodos no

módulo de controle lógico uma única vez. A Figura 10 mostra as classes implementadas no módulo de controle lógico.

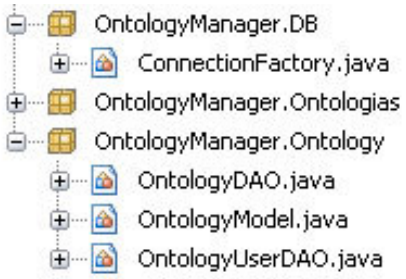

Figura 10. Classes desenvolvidas no módulo de controle lógico da aplicação.

As classes do módulo de controle estão divididas em dois pacotes, um para classes responsáveis pelo acesso ao banco de dados e um para classes responsáveis pela manipulação dos dados. O pacote *OntologyManager.DB* contém uma classe *ConnectionFactory* que possui métodos para conectar à ontologias, sejam elas, armazenadas em um banco de dados ou em um arquivo OWL. O pacote *OntologyManager.Ontology* possui classes responsáveis pela manipulação e o mapeamento dos objetos da ontologia em objetos Java. As principais funcionalidades providas por estas classes são: acessar qualquer objeto da ontologia (classe, propriedade ou instância) por meio de seu URI e retornar o objeto correspondente em Java; executar consultas na ontologia por meio da linguagem SPARQL; retornar a lista de instâncias de uma determinada classe; e retornar a lista de propriedades de uma determinada classe ou instância. O código que implementa estas funcionalidades encontra-se no Apêndice A.

As classes e métodos implementados no módulo de controle lógico são utilizados pelas páginas do *website* para publicar as informações contidas na ontologia. Por exemplo, na Figura 11, mostramos um trecho do código que exibe informações como o nome, o e-mail e a *homepage* de todos os professores do departamento, ou seja, de todas as instâncias da classe "Professor" na ontologia e, na Figura 12, mostramos a página *web* resultante. Na linha dois

(2) mapeamos a classe “Professor” da ontologia para um objeto Java chamado “prof”, por meio do método *getOntClass()* implementado no módulo de controle lógico. A estrutura de repetição na linha três (3) garante o acesso a todas as instâncias da classe com o método *listInstances()*, e para cada instância é criado um objeto Java “Individual”. Assim, nas linhas oito (8), onze (11) e quatorze (14) são exibidas as propriedades “nome”, “email” e “*homepage*” respectivamente de cada instância.

```
1.       <%
2.    OntClass prof= m.getOntClass(request.getParameter("Professor"));
3.    for (Iterator i= prof.listInstances(); i.hasNext();){
4.      Individual ind = (Individual)i.next();
5.     %>
6.      <tr>
7.        <td width="108" bordercolor="1">
8.          <%=ind.getProperty(m.getDatatypeProperty(dao.getURL()+"nome")).getString()
9.        </td>
10.       <td width="108" bordercolor="1">
11.         <%=ind.getProperty(m.getDatatypeProperty(dao.getURL()+"email")).getString()%>
12.       </td>
13.       <td width="108" bordercolor="1">
14.          <a
  href="<%=ind.getProperty(m.getDatatypeProperty(dao.getURL()+"homepage")).getString()%>"
  >Homepage</a>
15.       </td>
16.     </tr>
17.   <%}
18.   }%>
```

Figura 11. Mapeamento dos dados da ontologia para a página *web*

É importante ressaltar que o *website* não fica restrito somente aos métodos implementados no módulo de controle lógico para acessar as informações na ontologia, ele possui a liberdade de acessar diretamente os dados utilizando a API Jena 2 *Ontology*. Desenvolvemos o módulo de controle lógico com o objetivo de centralizar o controle, facilitar alterações e automatizar as funções mais utilizadas na construção das páginas.

Figura 12. Página *web* que lista os docentes do DIN.

Na construção do *website* semântico adotamos algumas estratégias para atender algumas de nossas necessidades específicas. Por exemplo, decidimos restringir nosso *website* semântico a um portal do tipo *intranet* que identifica os usuários por meio de *login* e senha. Este foi um condicionamento que adotamos para conseguir identificar as interações dos usuários com o *website* de forma individual. Outra decisão de projeto relevante foi a de que criar nosso próprio arquivo de *log*, ou seja, o *website* semântico realizará um monitoramento das interações dos usuários criando um *log* para armazenar essas informações.

2 A Construção do *Log* Semântico

Com objetivo de auxiliar a modelagem das intenções dos usuários, no desenvolvimento das páginas do *website* semântico adotamos uma estratégia para monitorar de forma transparente as interações do usuário com o *website*, criando um arquivo de *log* próprio. Assim, cada página do *website* visitada insere um registro neste arquivo armazenando a data e hora do acesso, e seu endereço. Porém, no contexto de personalização de *websites*,

como visto em [MLADENIC, 1999], a utilização única de dados provenientes da navegação do usuário pode proporcionar o surgimento de dificuldades quando não há dados suficientes para extrair padrões relacionados a determinadas categorias do domínio, ou quando são adicionadas novas páginas ao *website* que ainda não foram visitadas pelos usuários. Desta forma, devido ao fato de que todas as informações das páginas do *website* são baseadas na ontologia do domínio, os conceitos envolvidos nas informações das páginas visitadas também são inseridos no arquivo de *log*, criando o que denominamos de *log* semântico. Deste modo, a construção do *log* semântico agrega as informações semânticas do conteúdo do *website* com os comportamentos navegacionais do usuário.

Em uma abordagem formal, uma visita feita a uma página do *website* gera uma entrada no *log* semântico representado por $v = \{d, t, p, < c_1, c_2, \ldots, c_{nv} >\}$ onde d representa a data do acesso, t a hora do acesso, p o endereço da página e c o(s) conceito(s) envolvidos na página. Desta forma, quando um usuário navega no *website*, conforme vai acessando as páginas são geradas n entradas v no *log* semântico representado por $pv = \{v_1, v_2, \ldots, v_n\}$. O conjunto de todos os usuários que acessaram o *website* é representado por $U = \{u_1, u_2, \ldots, u_{nu}\}$, na qual cada usuário está devidamente cadastrado por meio de *login* e senha. Assim, o *log* semântico pode ser definido como $L_{sem} = \{(u_1, pv_1), (u_2, pv_2), \ldots, (u_n, pv_n)\}$ onde u_i representa o usuário e pv_i o conjunto de páginas visitadas por ele.

Outra possibilidade de monitorar as interações do usuário no *website* é utilizar os arquivos de *log* criados pelo servidor *web*. No entanto, seria necessário minerar este arquivo para identificar as interações realizadas por um determinado usuário e daí fazer seu enriquecimento semântico. Como em nosso trabalho o *website* possui uma autenticação de usuário via *login* e senha, possibilitando a identificação individualizada das interações do usuário, decidimos criar nosso próprio *log*.

Neste capítulo, mostramos como integramos as informações semânticas do conteúdo do *website* a padrões navegacionais. Apresentamos a forma com que incorporamos uma base semântica ao *website*, como modelamos a ontologia do domínio e quais foram as tecnologias utilizadas. Além disso, apresentamos a forma com que criamos o *log* semântico. No próximo capítulo, vamos discutir uma forma de modelagem do usuário, quais são os parâmetros considerados na criação do modelo e a técnica de aprendizagem empregada. Em seguida, vamos apresentar um algoritmo de identificação de interesses de um usuário.

Capítulo 5

MODELAGEM DO USUÁRIO

A criação de um modelo de intenções de um usuário é sempre muito complexa, pois o conjunto de parâmetros a se considerar é sempre muito alto. A escolha destes parâmetros, juntamente com a técnica de aprendizagem de máquina empregada, faz a real diferença na qualidade do modelo.

Para definirmos que parâmetros deveriam ser empregados na modelagem das intenções de um usuário durante sua navegação em um *website* semântico foram levados em conta aspectos lingüísticos e cognitivos que afetam o usuário na expressão de seus interesses. Deste modo, inicialmente vamos discutir o efeito que o vocabulário disponível ao usuário tem sobre sua forma de expressão. Em seguida, vamos discutir o efeito que o relacionamento semântico entre os conceitos tem sobre a memória humana e como isto pode afetar a expressão do usuário e a identificação de sua intenção, induzida pelo modelo de

conhecimento, ao navegar um *website*. E, por fim, vamos discutir a utilização destes parâmetros e de uma técnica de aprendizagem de máquina no desenvolvimento de um algoritmo de identificação de interesses de um usuário.

1 Os Aspectos Lingüísticos que Afetam o Modelo do usuário

Como descrevemos no Capítulo 4, como forma de representar a semântica da informação presente em um *website*, utilizamos uma ontologia para modelar o conhecimento presente no domínio. Uma ontologia do domínio, basicamente, consiste de um conjunto de conceitos e relacionamentos que descrevem o conhecimento sobre um domínio de interesse. No entanto, ela também pode ser vista como definidora de um vocabulário para uma linguagem controlada que permite aos usuários se expressarem sobre o domínio. Olhando por este ponto de vista, identificamos para o problema de navegação na *web* duas formas em que o vocabulário do usuário é restringido as quais denominamos de **modelo da aplicação** e **modelo de apresentação**. Ambos os modelos representam um segmento do conhecimento modelado na ontologia do domínio.

É importante ressaltar que o usuário somente pode expressar interesse sobre algo que esteja modelado no modelo de conhecimento, ou seja, sobre o domínio no qual o *website* foi construído. Deste modo, quando nos referimos na identificação da intenção ou interesse do usuário, significa a intenção ou interesse induzido pelo modelo de conhecimento.

Assim, a introdução destes dois modelos procura contemplar o fato de que uma pessoa só pode expressar algo para o qual ela tem um vocabulário disponível, o qual tem influência indireta na criação de um modelo dos interesses dos usuários.

O Modelo da Aplicação

Quando desenvolvemos uma aplicação de software baseada em conhecimento, seja ela para *desktop* ou para *web*, nem todo conhecimento modelado na ontologia do domínio é utilizado, ou seja, nem todos os conceitos presentes na ontologia do domínio precisam ser envolvidos no desenvolvimento. Desta forma, indiretamente, estamos segmentando a ontologia do domínio, selecionando apenas os conceitos e relações que estão envolvidos diretamente com a aplicação, como ilustrado na Figura 13.

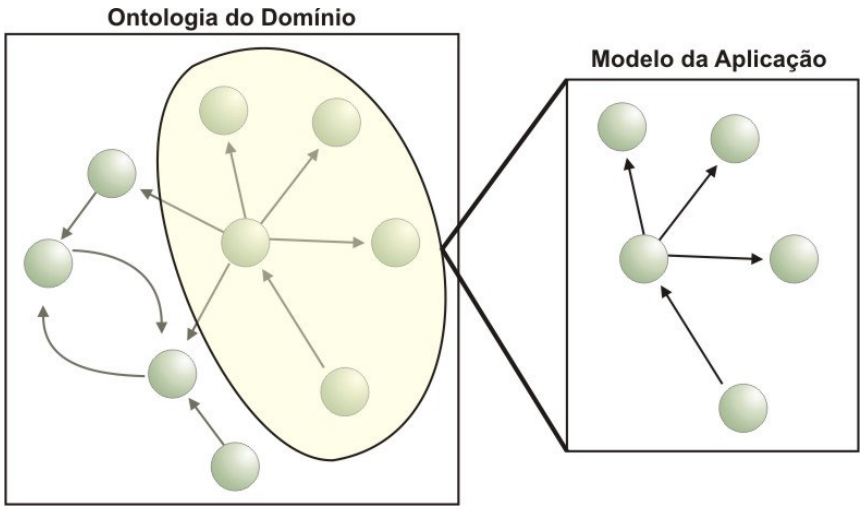

Figura 13. Segmento da ontologia do domínio para o modelo de aplicação

Este segmento da ontologia do domínio, que denominamos de **modelo da aplicação**, tem influência direta sobre a expressão dos interesses dos usuários, pois ele restringe o vocabulário que o usuário pode usar para expressar seus interesses sobre o domínio para os conceitos e relações presentes neste modelo.

O Modelo de Apresentação

No caso do desenvolvimento de um *website* o problema de restrição do vocabulário vai mais longe, pois cada página *web* emprega apenas uma parte do conhecimento contido no modelo da aplicação, limitando ainda mais o poder de expressão dos usuários. Para

refletirmos esta nova redução no vocabulário do usuário empregamos um **modelo de apresentação**, que representa um segmento do modelo da aplicação que é apresentado em cada página *web*, como ilustrado na Figura 14. Deste modo, a partição do conhecimento para um *website* semântico fica estruturada como ilustra a Figura 15.

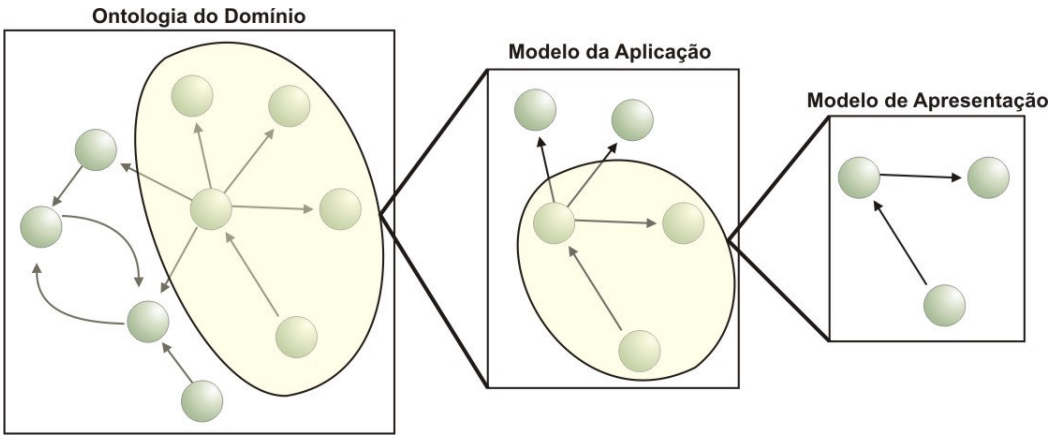

Figura 14. Segmento da ontologia do domínio para o modelo de apresentação

Assim, a introdução do modelo de apresentação se faz necessária para modelar as limitações impostas sobre a linguagem de expressão do usuário. Para levar isto em conta foi definido um parâmetro no algoritmo de identificação de interesses do usuário, denominado ***status*****,** que contabilizará a proeminência do conceito na página e, portanto, sua influência na expressão de intenção do usuário.

É importante salientar que, neste trabalho, o modelo de apresentação está relacionado, principalmente, ao conteúdo da informação presente nas páginas *web*. Ele não leva em conta o efeito que a apresentação estética tem sobre a atenção do usuário e que também pode direcionar suas intenções.

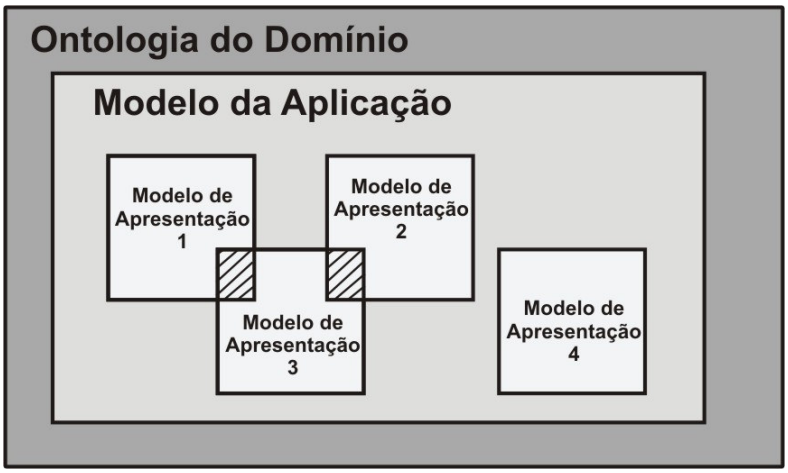

Figura 15. Modelos de Conhecimento.

Apesar de não considerar o fator estético neste trabalho, não é possível descartar o fator *layout* quando falamos em apresentação de uma página *web*. Como uma forma de considerar, parcialmente, o efeito que o *layout* tem sobre o modelo de apresentação, destacamos três principais componentes em uma página *web* atribuindo a eles diferentes pesos, baseados em seu nível de proeminência na página, são eles: o *Menu* Principal, o *Menu* Secundário e o Corpo da Página, ilustrado pela Figura 16. Os diferentes pesos atribuídos a estes componentes são refletidos como um parâmetro no algoritmo de identificação de intenções do usuário, chamado *Status.* Este parâmetro representa o nível de destaque que o conceito possui na página. Por exemplo, para os conceitos presentes no componente do *Menu* Principal, é definido um *status* com peso maior por eles possuem maior destaque na página.

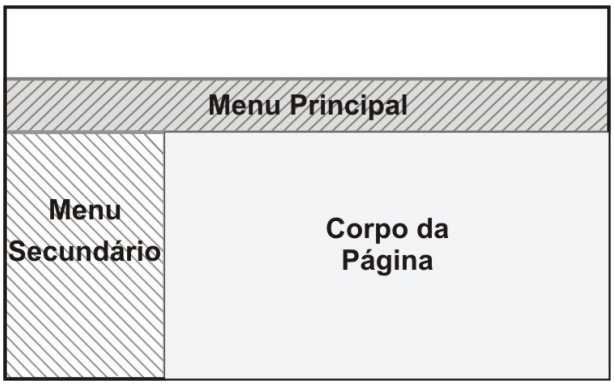

Figura 16. Os componentes de uma página *web*.

2 Os Aspectos Cognitivos que Afetam o Modelo do usuário

Uma das teorias dominantes no processamento semântico na psicologia cognitiva é conhecida como **propagação de ativação** (*spreading activation*) [ANDERSON, 1983] [ROELOFS, 1992] [COLLINS e LOFTUS, 1975] [RUMELHART e MCCLELLAND, 1982]. Esta teoria procura explicar como funciona a recuperação de informações da memória humana [QUILLIAN, 1968] [ANDERSON, 1983]. Ela considera que a memória humana é organizada na forma de uma rede semântica propondo que quando um conceito se torna o foco de nossa atenção todos os conceitos a ele associados também são ativados, ou seja, a ativação de um conceito se propaga para todos os conceitos diretamente associados a ele. Esta propagação da ativação ajuda a explicar como a lembrança de um tópico pode trazer à mente tópicos relacionados [QUILLIAN, 1968]. Segundo Collins e Loftus [1975], para melhor explicar o processo cognitivo de propagação de ativação também é necessário considerar uma **força de ativação** para cada associação existente com o conceito foco. Desta forma, é possível ampliar ou reduzir a **força cognitiva** no processo de propagação conforme o número de conceitos relacionados ao conceito em questão.

Neste trabalho, a ontologia de domínio, o modelo da aplicação e de apresentação, são representados por redes semânticas expressas na linguagem OWL. Se considerarmos que, ao escolher um *link* em uma página de um *website* semântico, o usuário estará ativando um conceito (denominado **conceito foco**) nas redes semânticas que compõem o *website*, de forma semelhante a lembrança de um conceito na rede de conceitos mentais que temos, é bastante razoável empregar o conceito de força cognitiva para avaliar o real interesse de um usuário.

3 Um Algoritmo de Identificação de Interesses de um Usuário

O algoritmo proposto neste trabalho é um algoritmo probabilístico que leva em consideração tanto os aspectos lingüísticos quanto os cognitivos que influenciam o processo de construção do modelo do usuário. Assim, ele emprega o modelo de apresentação e a teoria de propagação da ativação, considerando a força cognitiva de cada conceito, na seleção dos conceitos sobre o modelo de conhecimento que expressam o atual interesse de um usuário do *website*.

Em nosso algoritmo, o modelo de apresentação tem duas funções. A primeira é a levar em consideração a limitação lingüística imposta pelo conteúdo apresentado na página *web*. Como os conceitos presentes neste modelo estão mais fortemente ativados na memória do usuário, focando seu escopo de expressão, eles devem ser ponderados mais fortemente que os demais conceitos do modelo da aplicação. Assim, levamos em consideração que o usuário não pode expressar interesse em algo que não está ao seu alcance. Desta forma, quando o usuário está visualizando uma página *web*, o conhecimento (a rede semântica das informações relevantes ao que ele pode encontrar na página) é representado no modelo de apresentação e cada conceito terá seu *status* baseado no seu nível de destaque na página, como visto na Seção 1 deste Capítulo.

A segunda função do modelo de apresentação é levar em consideração o aspecto cognitivo de propagação da ativação na memória humana. Como o modelo de apresentação é representado por uma rede semântica, organizando as informações de forma similar à mente humana [QUILLIAN, 1968], tentamos tirar proveito desta similaridade na identificação de interesses do usuário. Para isso, utilizamos a teoria de propagação de ativação para simular o funcionamento da recuperação de informação na mente humana em nosso algoritmo. Assim, quando o usuário acessa uma página, identificamos o conceito foco e o ativamos na rede semântica. Quando este conceito é ativado, ele também ativará todos os outros que estejam

relacionados a ele, como ilustra a Figura 17. Entretanto, não podemos ponderar os conceitos com pesos iguais, precisamos levar em consideração a força cognitiva que cada conceito exerce dentro da rede semântica. A força cognitiva aumenta ou diminui a força de uma ativação do conceito no processo de propagação.

Para compor a força cognitiva de cada conceito, primeiramente consideramos o número de relações que ele possui. Assim quanto maior o número de relações maior sua força cognitiva, pois quanto mais relações um conceito possui mais a possibilidade dele ser lembrado, ou seja, ser ativado a partir de outro conceito. Outro fator que consideramos para a força cognitiva é o nível de destaque que o conceito ocupa na página, ou seja, quanto maior o destaque, mais força cognitiva ele possui.

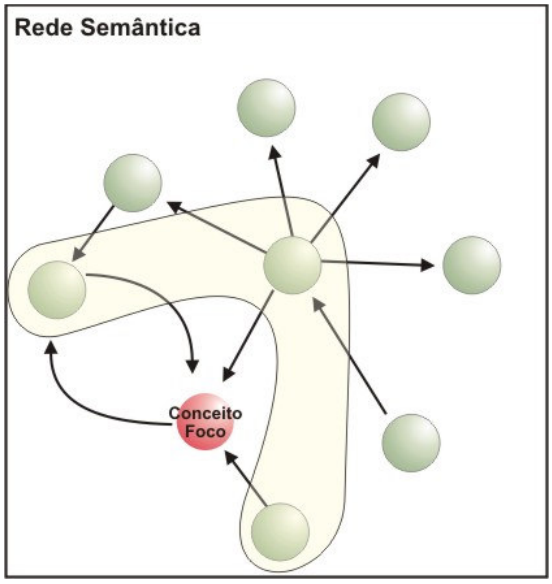

Figura 17. Processo de Ativação

Sendo assim, a força cognitiva de um conceito é definida pela fórmula $\boldsymbol{F = S * R}$, onde R é o número de relações que o conceito em questão possui e S o *status* do conceito no modelo de apresentação. Para definirmos R precisamos levar em conta que o modelo de apresentação é um segmento do modelo da aplicação, que por sua vez é um segmento da ontologia do domínio. Assim, algumas relações do conceito presente no modelo de apresentação podem ter sido eliminadas no segmento do modelo da aplicação. Estas relações

eliminadas não são descartadas na contabilização de *R*, mas serão ponderadas com um menor valor. Desta forma, classificamos as relações que um conceito possui em dois grupos: $\#R_{aplic}$ fornece o número de relações com outros conceitos no modelo da aplicação (conceitos estes que não aparecem no modelo de apresentação), $\#R_{apres}$ fornece o número de relações com outros conceitos do modelo de apresentação. Logo, o parâmetro *R* é definido pela fórmula $R = p_{aplic} \cdot \#(R_{aplic}) .+ p_{apres} \cdot \#(R_{apres})$, onde p_x é um parâmetro do algoritmo que determina um peso específico, atribuído empiricamente, para cada tipo de relação, respeitando a primeira função do modelo de apresentação. O parâmetro *S* é definido pela posição em que o conceito ocupa no *layout* da página *web*, como visto na Figura 18. A Figura 18 ilustra os parâmetros que precisam ser determinados para qualquer conceito do modelo de apresentação durante o processo de propagação de ativação.

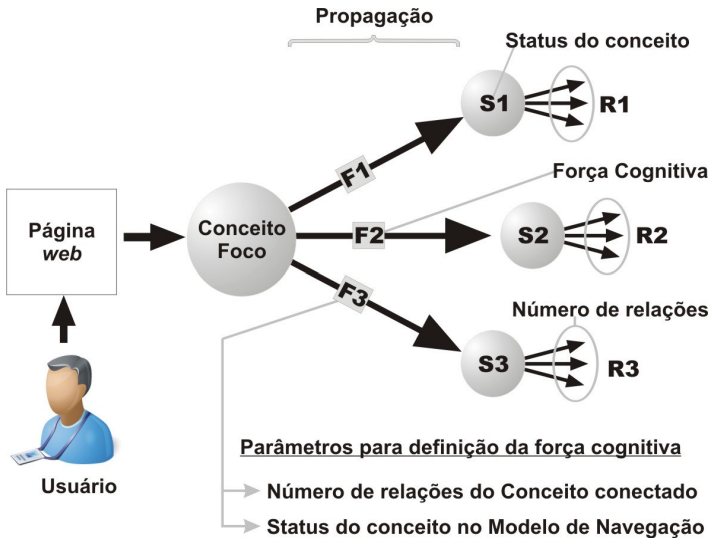

Figura 18. Parâmetros no processo de propagação.

A força cognitiva é utilizada no processo de propagação de ativação para aumentar ou diminuir a força de ativação de um conceito quando ele for ativado, ou seja, é a forma com que podemos dizer que um conceito possui mais relevância do que outro na rede. Entretanto, também precisamos diferenciar quando o conceito é ativado como foco e quando ele é ativado pela relação que possui com o conceito foco, chamamos este parâmetro de $\boldsymbol{T_{conc}}$. Assim,

definimos pesos diferenciados para três diferentes tipos de conceitos: **o conceito foco** — associado ao *link* que o usuário escolheu e que, portanto, tem uma probabilidade alta de ser o seu real interesse; — recebe um valor mais alto, o qual também depende de sua força cognitiva; os **conceitos diretamente conectados** ao conceito foco — os que têm uma boa probabilidade de serem o real interesse do usuário — recebem um valor dependente somente de sua força cognitiva, como definido anteriormente; e os **conceitos indiretamente conectados** do conceito foco — os que não estão envolvidos no processo de propagação e são considerados como **ruído** por não compartilharem a atenção do usuário, no entanto, eles não devem ser desprezados como poderá ser visto mais adiante e, portanto, recebem um valor residual. A Figura 19 ilustra os tipos de conceito considerados na rede semântica.

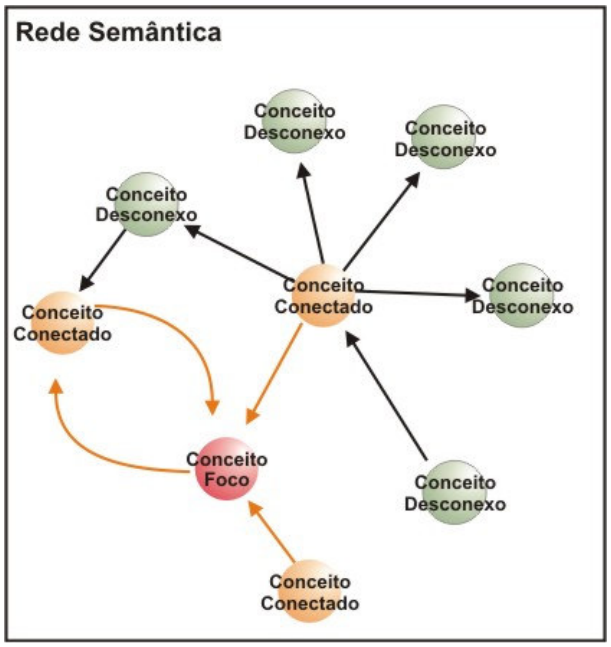

Figura 19. Conceitos considerados na rede semântica

Após definirmos os fatores lingüísticos e cognitivos que afetam a modelagem do usuário e os parâmetros a serem considerados, desenvolvemos um algoritmo para identificação de interesses do usuário. O algoritmo proposto tem como entrada o *log* semântico com informações sobre as interações realizadas pelos usuários e quais foram os conceitos envolvidos nestas interações. Além do *log* semântico o algoritmo também tem

acesso aos modelos de apresentação e de aplicação empregados na determinação dos valores dos parâmetros para a identificação dos interesses do usuário, como mostra o esquema da Figura 20.

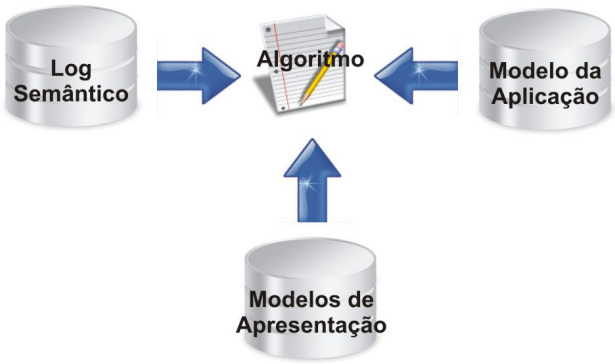

Figura 20. Módulos de acesso do algoritmo

Em nosso algoritmo, os interesses do usuário são identificados por meio de uma lista de preferência com a probabilidade de interesse nos conceitos presentes nos modelos de conhecimento, como ilustra a Figura 21, dada as interações com o *website* semântico. Esta probabilidade pode ser utilizada por uma máquina de recomendação para recomendar informações pertinentes aos conceitos de maior interesse do usuário. Ela também pode auxiliar uma busca semântica de informações com maior precisão, retornando nas primeiras posições os assuntos relacionados com os conceitos de maior interesse do usuário.

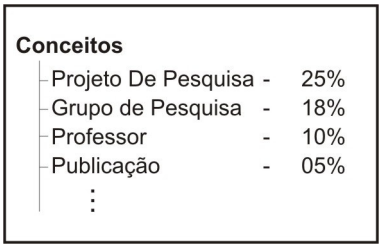

Figura 21. Lista de preferência com a porcentagem de interesse dos conceitos presentes nos modelos de conhecimento

Assim, a cada interação do usuário, representada por um registro do *log* semântico, o algoritmo identifica o conceito foco, os conceitos diretamente conectados e os conceitos

indiretamente conectados ao foco no modelo de apresentação e aplicação e atribuí os pesos correspondentes a cada um deles. Ao final de todas as interações são definidas as probabilidades de interesse sobre cada conceito. Estas probabilidades são definidas pela fórmula:

$$P(C_i \mid I_1 ... I_j) = \sum_{1}^{j} Tc_i + F_i$$

onde:

- P = probabilidade de interesse do usuário a um determinado conceito
- C = o conceito em questão
- I = as interações presentes no *log* semântico
- Tc = o valor correspondente ao tipo de conceito (Conceito Foco, Conceito Diretamente Conectado e Conceito Indiretamente Conectado)
- F = a força cognitiva do conceito (F = 0, se o tipo do conceito for um Conceito Indiretamente Conectado)

Para obter o conceito que representa o maior interesse do usuário é utilizada a fórmula:

$$\arg\max_i (P(C_i \mid I_1 ... I_j))$$

onde i varia para todos os conceitos e j é o total de interações. Desta forma, o algoritmo mantém uma lista atualizada de todos os conceitos com suas respectivas probabilidades de interesse, como ilustra a Figura 21. Esta lista é armazenada como um modelo das intenções do usuário.

A Figura 22 mostra o algoritmo desenvolvido para identificar os interesses de um usuário. No algoritmo consideramos que:

- M_{apres} é o modelo de apresentação;
- M_{aplic} é o modelo de aplicação;

- v é um acesso a uma página do *website* semântico de um determinado usuário, onde $v = \{d, t, p, < c_1, c_2, \ldots, c_{nv} >\}$;
- *Conceito* representa o URI de um conceito pertencente ao modelo da aplicação, $Conceito \in M_{aplic}$;
- $Conceito_{foco}$ é o conceito foco da pagina visitada;
- $Conceitos_{conec}$ representa um conjunto de conceitos;
- R_{apres} é o número de relações que o conceito possui no modelo de apresentação;
- R_{aplic} é o número de relações que o conceito possui no modelo da aplicação;
- P_{apres} é o peso aplicado a R_{apres}, relações que o conceito possui no modelo de apresentação ;
- P_{aplic} é o peso aplicado a R_{aplic}, relações que o conceito possui no modelo de apresentação ;
- R é o parâmetro que contabiliza o numero de relações na determinação da força cognitiva de um conceito;
- S é o parâmetro que identifica o nível de destaque na página que um conceito possui;
- F é a força cognitiva de um determinado conceito;
- Tc_{foco} é o peso atribuído a um conceito foco na propagação de ativação;
- Tc_{conec} é o peso atribuído aos conceitos conectados diretamente ao conceito foco na propagação de ativação;
- $Tc_{ruído}$ é o peso atribuído aos conceitos indiretamente conectados ao conceito foco na propagação de ativação;
- *Lista* é a lista atualizada contendo as porcentagens de interesse para cada conceito do modelo de aplicação.

Procedimento Identifica_Interesse(M_{apres}, M_{aplic}, v)

$conceito_{foco} \leftarrow c \in v$

$conceitos_{conec} \leftarrow identifica\_conectados(conceito_{foco}, M_{apres})$

Para todo $conceito_i \in M_{aplic}$ **faça**

Se $(conceito_i = conceito_{foco})$ **ou** $(conceito_i \subset conceitos_{conec})$ **então**

$R_{apres} \leftarrow identifica\_relacoes(conceito_i, M_{apres})$

$R_{aplic} \leftarrow identifica\_relacoes(conceito_i, M_{aplic})$

$R = (R_{apres} * p_{apres}) + (R_{aplic} * p_{aplic})$

$S \leftarrow identifica\_status(conceito_i, M_{aplic})$

$F \leftarrow S * R$

Se $conceito_i = conceito_{foco}$ **então**

$Lista[conceito_i] \leftarrow Lista[conceito_i] + Tc_{foco} + F$

Senão Se $conceito_i \subset conceitos_{conec}$ **então**

$Lista[conceito_i] \leftarrow Lista[conceito_i] + Tc_{conec} + F$

Fim

Senão

$Lista[conceito_i] \leftarrow Lista[conceito_i] + Tc_{ruido}$

Fim

Fim

Retorna(*Lista*)

Fim

Figura 22. Um algoritmo para enriquecimento semântico do arquivo de *log*.

Antes de apresentarmos detalhes do algoritmo, é importante ressaltar que todas as menções feitas a um conceito estão referenciando ao URI do conceito. É por meio deste URI que conseguimos localizar e acessar qualquer objeto na ontologia. Do mesmo modo, as menções feitas a um modelo de conhecimento estão referenciando uma ontologia expressa na linguagem OWL.

O algoritmo tem como entrada o modelo da aplicação, os modelos de apresentação e uma entrada do *log* semântico. A partir da entrada do *log* semântico identificamos os conceitos envolvidos no processo de propagação de ativação: o conceito foco – armazenado em $conceito_{foco}$; e os conceitos diretamente conectados a ele – armazenados em $conceitos_{conec}$. Estes conceitos são identificados para que possam ser ponderados posteriormente. A função responsável pela identificação é a *identifica_conectados( )* a qual, dado um conceito e um modelo de conhecimento, retorna todos os conceitos diretamente relacionados a ele no modelo de conhecimento.

Depois de identificados os conceitos envolvidos na propagação de ativação é preciso determinar a força cognitiva de cada um, então, todos os conceitos envolvidos na aplicação são percorridos. Caso o conceito em questão seja o foco ou um conceito diretamente conectado é calculada sua força cognitiva, caso contrário, ou seja, quando for um conceito indiretamente conectado do foco, desconsideramos sua força cognitiva *(F=0)*. Como visto anteriormente, para calcular a força cognitiva de um conceito é preciso definir alguns parâmetros, tais como *R* e *S*.

Para a definição de *R* são identificados: o número de relações que o conceito possui no modelo da aplicação R_{aplic} e no modelo de apresentação R_{apres}. Estas relações são identificadas por meio da função *identifica_relações( )* a qual, dado um conceito e um modelo de conhecimento, retorna o número de relações que este possui no modelo de conhecimento. Para as relações presentes em cada modelo foram definidos pesos diferenciados, p_{aplic} para as relações presentes no modelo da aplicação e p_{apres} para as relações presentes no modelo de apresentação. Assim, o parâmetro *R* é definido como $R = (R_{apres} * p_{apres}) + (R_{aplic} * p_{aplic})$.

Outro parâmetro definido para compor a força cognitiva do conceito no algoritmo é o S, o qual representa o nível que o conceito ocupa na página. A função responsável por buscar esta informação é a *identifica_status( )* a qual, dado um conceito e o modelo de apresentação,

retorna o *status* correspondente. Definidos os parâmetros a força cognitiva é calculada e representada por *F*.

O algoritmo mantém uma lista com todos os conceitos do modelo da aplicação com os valores atribuídos a cada conceito. Estes valores, normalizados, representam o interesse do usuário pelo conceito. Para ponderar determinado conceito, primeiramente, precisamos identificar qual é seu envolvimento no processo de propagação de ativação, se ele é o conceito foco, um conceito conectado ao foco ou um conceito indiretamente conectado. Em cada tipo de conceito identificado na propagação de ativação foi definido um determinado peso, Tc_{foco} para conceitos foco, Tc_{conec} para conceitos conectados e T_{ruido} para conceitos indiretamente conectados. Desta forma, quando o conceito é identificado como foco recebe o valor de sua força cognitiva (*F*) mais o valor correspondente ao seu tipo, no caso Tc_{foco}. O mesmo acontece quando o conceito é identificado como conectado, ele recebe o valor de sua força cognitiva mais Tc_{conec}. No caso em que o conceito é identificado como ruído, recebe apenas o Tc_{ruido}, pois não consideramos a força cognitiva para conceitos indiretamente conectados ao foco.

O algoritmo, acima descrito, foi simulado no software Microsoft® Excel utilizando programação de macros na linguagem denominada VBA – *Visual Basic for Applications*. Escolhemos esta ferramenta por facilitar posteriores avaliações dos resultados do algoritmo na forma de gráficos.

Um ponto importante a ressaltar é que há uma limitação inerente aos parâmetros (p_{aplic}, p_{apres}, S, Tc_{foco}, Tc_{conec} e Tc_{ruido}) utilizados no algoritmo, pois atribuímos os valores destes parâmetros de uma forma empírica. Consideramos que a melhor forma de determinar estes parâmetros seja por meio do desenvolvimento de um algoritmo de aprendizagem dos parâmetros, porém, devido à limitação de tempo, não foi possível realizar esta tarefa. Pretendemos desenvolver esta tarefa futuramente, pois acreditamos que com este algoritmo de

estimação de parâmetros o desempenho do algoritmo de identificação de interesses irá melhorar consideravelmente.

Neste capítulo, apresentamos os aspectos lingüísticos e cognitivos que podem influenciar na modelagem do usuário. Apresentamos, também, um algoritmo para identificação de interesses de um usuário, bem como os parâmetros que levamos em consideração em sua construção. No próximo capítulo, vamos apresentar e analisar alguns testes empíricos de simulações realizadas com o algoritmo.

Capítulo 6

UMA AVALIAÇÃO EXPERIMENTAL DO ALGORITMO PROPOSTO

Este capítulo descreve um estudo experimental, baseado em testes empíricos, no qual simulamos o algoritmo proposto no Capítulo 5 objetivando verificar a validade da abordagem de integração de conhecimento semântico e de dados de navegação no processo de identificação de possíveis interesses do usuário, bem como a influência exercida pelos aspectos lingüísticos e cognitivos identificados na criação de modelo de usuário. A seção seguinte descreve as características do experimento e a Seção 2 apresenta os resultados dos testes realizados no experimento.

1 Características dos Experimentos

Para que tivéssemos um ponto de referência para comparação da qualidade de nosso algoritmo, adotamos outros dois algoritmos conhecidos para aplicar o mesmo conjunto de dados. Os algoritmos adotados seguiram uma abordagem de classificação baseada em

freqüência e uma abordagem de classificação bayesiana. No algoritmo que utiliza a abordagem bayesiana, utilizamos uma adaptação do algoritmo do classificador *Naïve de* Bayes. Para utilizar o algoritmo de Bayes, adotamos uma heurística que considera somente os exemplos positivos como atributo de classificação, visto que, neste caso, não é possível obter exemplos negativos sem uma intervenção explícita do usuário. Os exemplos positivos foram extraídos por meio do *log* semântico simulado, no qual cada entrada foi considerada como um exemplo positivo. Por outro lado, no algoritmo que utiliza uma abordagem baseada em freqüência, consideramos simplesmente a freqüência de visitas as paginas *web*. Entretanto, como a freqüência é obtida por meio do *log* semântico, foi considerada de fato a freqüência dos conceitos envolvidos na página *web*.

Para a execução dos testes desenvolvemos um *website* semântico para o Departamento de Informática da Universidade Estadual de Maringá, cujo processo de desenvolvimento foi descrito no Capítulo 4. Definimos algumas navegações simulando situações específicas de uso, na qual os interesses da navegação estivessem predefinidos. As navegações foram induzidas a determinados interesses, ou seja, elas foram definidas a fim de caracterizar estes interesses, que são expressos por meio de conceitos do modelo da aplicação. Assim, quando dizemos “interesses do usuário”, queremos dizer conceitos do modelo de conhecimento que o usuário está interessado.

Para cada navegação realizamos 100 (cem) interações com o *website* e definimos os conceitos do modelo de aplicação que caracterizam os interesses da navegação. E como resultado de cada navegação no *website* semântico um arquivo de *log* semântico é gerado. A partir deste *log* semântico, juntamente com a ontologia de domínio e os modelos de conhecimento, aplicamos as duas abordagens mencionadas anteriormente e nosso algoritmo probabilístico a fim de comparar seus resultados.

1.1 Um Algoritmo para classificação baseada no classificador *Naïve de* Bayes

O algoritmo *Naïve de* Bayes é um algoritmo de classificação probabilística, baseado no teorema de Bayes, que assume independência dos atributos. Dado um conjunto de atributos $X_1,...,X_n$, todos são considerados condicionalmente independentes um dos outros. Esta independência minimiza a complexidade do classificador, pois reduz o número de parâmetros a serem estimados para $P(H \mid E)$. A regra de Bayes é representada pela fórmula

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)}$$

onde:

- H é a hipótese e E a evidência;
- $P(H \mid E)$ é a probabilidade a posteriori da hipótese dada a evidência;
- $P(H)$ é a probabilidade a priori da hipótese;
- $P(E \mid H)$ é a probabilidade de uma evidência dado uma hipótese. Esse parâmetro precisa ser estimado;
- $P(E)$ é o fator para normalização da probabilidade, ou seja, a probabilidade a priori da evidência.

Desta forma, podemos calcular a probabilidade relativa de uma hipótese, dado uma ou mais evidências, pela fórmula $P(H \mid E_1,\ldots,E_n) = P(H) * \prod_n P(E_n \mid H)$, na qual o denominador $P(E)$ foi eliminado por se tratar de um fator de normalização da probabilidade.

Em nosso algoritmo aplicamos a fórmula $P(H \mid E)$ para cada conceito do modelo de aplicação a fim de verificar a probabilidade de todos os conceitos, sendo H o conceito e E um registro no *log* semântico, ou seja, dado um registro do *log* semântico identificamos qual é a probabilidade de um determinado conceito dado aquele registro. Como analisamos o *log* semântico linha a linha adotamos uma aprendizagem incremental, considerando que ao final da análise de cada linha, ela se torna parte do conhecimento a priori, ou seja, na próxima linha

a ser analisada este conhecimento é utilizado no cálculo da probabilidade a priori. Deste modo, o conhecimento prévio vai sendo incrementado a cada análise. Para evitar problemas com relação aos conceitos que ainda não foram acessados utilizamos a suavização de *Laplace*, como pode ser visto com maiores detalhes em [MITCHELL, 2005]. Como atributo de classificação, consideramos o próprio conceito envolvido na página acessada como sendo o conceito de interesse naquele registro.

A Figura 23 mostra o algoritmo de classificação de *Naïve de* Bayes adaptado. O algoritmo recebe como entrada um registro do *log* semântico, onde o conceito *c* é a evidência. Dado a evidência, é calculada a probabilidade a posteriori *P(H | E)* para todos os conceitos. A função *calcula_priori( )* calcula a probabilidade a priori para um determinado conceito. Da mesma forma, a função *calcula_likelihood( )* estima o parâmetro dada a evidência e o conceito.

```
Procedimento algoritmo_bayesiano(<u,v>)
    evidencia ← c ∈ v
    Para todo conceito_i ∈ M_aplic faça
        priori ← calcula_priori(conceito_i)
        likelihood ← calcula_likelihood(evidencia,conceito_i)
        posteriori ← priori * likelihood
        Lista[conceito_i] ← posteriori
    Fim
    Retorna( Lista )
Fim
```

Figura 23. Um algoritmo de classificação bayesiana adaptado.

1.2 Um Algoritmo para classificação baseada em freqüência

O algoritmo baseado em freqüência é uma implementação simples que considera somente os conceitos envolvidos nas páginas acessadas. Os conceitos são ponderados com

apenas dois valores 0 (zero) e 1 (um). Quando um conceito é identificado como acessado atribuí-se 1 (um) e 0 (zero) aos demais. Estes valores vão sendo somados e normalizados a cada entrada do *log* semântico. Desta forma, o algoritmo recebe um registro do *log* semântico, soma o valor 1 (um) aos conceitos envolvidos na página acessada, os quais são armazenadas em uma lista de preferência. A Figura 24 ilustra o algoritmo.

```
Procedimento algoritmo_frequencia(<u,v>)
    conceitos ← c∈ v
    Para todo conceito_i∈ conceitos faça
        Lista[conceito_i] ← Lista[conceito_i] + 1
    Fim
    Retorna( Lista )
Fim
```

Figura 24. Um algoritmo de classificação baseado em freqüência.

1.3 O modelo de Aplicação utilizado

Para realizar nossos experimentos restringimos nossas navegações no *website* semântico de forma que sejam feitos somente acessos a páginas *web* que envolvam um conjunto de apenas 10 (dez) conceitos da ontologia do domínio, ou seja, definimos o modelo da aplicação com 10 (dez) conceitos. Esta restrição foi realizada com o objetivo de facilitar a visualização dos resultados na forma de gráficos. A Figura 25 ilustra o modelo da aplicação considerado nos experimentos.

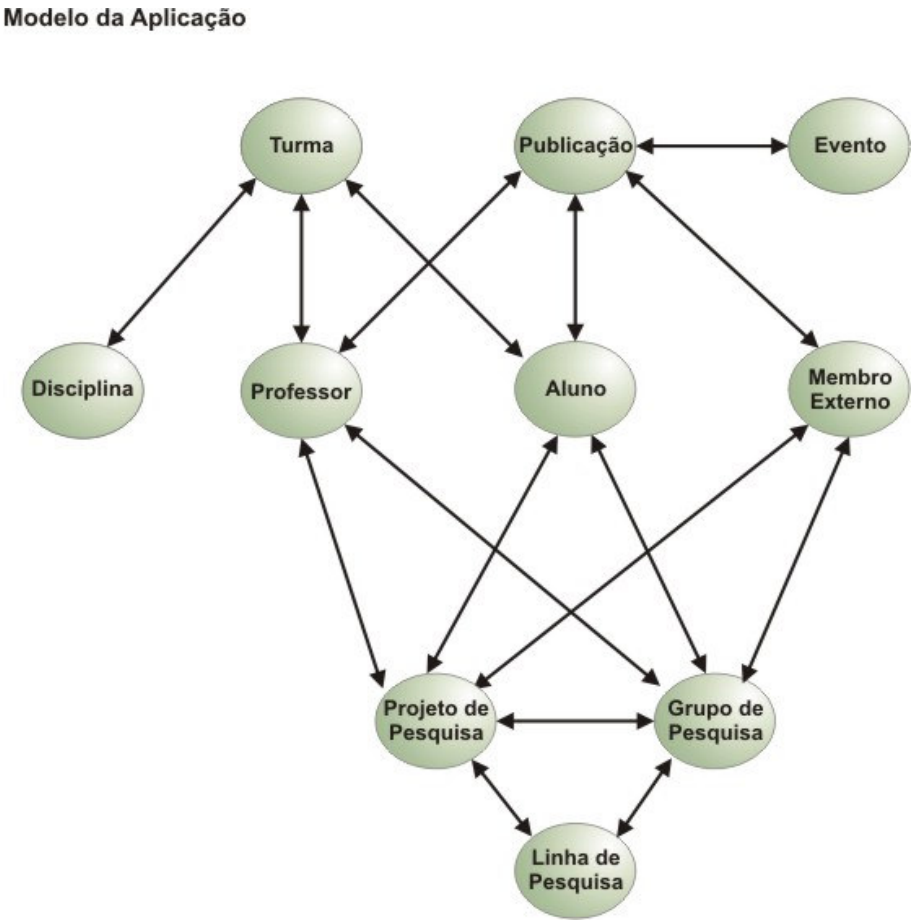

Figura 25. Modelo de Aplicação adotado para os experimentos

1.4 Os valores adotados para os parâmetros nos experimentos

Como havíamos dito anteriormente, os parâmetros do algoritmo probabilístico proposto foram definidos de forma empírica, baseado em sucessivas tentativas para encontrar os valores que obtivessem um melhor resultado. Estes parâmetros foram identificados a partir dos modelos de conhecimento e do processo de propagação. Para realizarmos os experimentos definimos os parâmetros como segue:

- $p_{aplic} = 0{,}05$;
- $p_{apres} = 0{,}1$;
- $S = 1{,}03$ para “menu principal”;
- $S = 1{,}02$ para “menu secundário”;

- S = 1,01 para “corpo da página”;
- Tc_{foco} = 1;
- Tc_{conec} = 0,1;
- Tc_{ruido} = 0,01.

2 Análise dos Resultados dos Experimentos

Nesta seção vamos relatar os resultados de dois experimentos, dentre vários realizados, para demonstrar a validade de nossa abordagem.

Experimento 1

Em um primeiro experimento, definimos uma navegação com o objetivo de buscar por “Publicações” e ”Eventos” em que alguns ”Professores” estão envolvidos. Para encontrar o professor desejado acessamos os projetos de pesquisa aos quais ele participa. Quando o professor desejado é encontrado suas publicações, bem como os eventos da publicação, podem ser visualizadas. Nesta navegação, a maior concentração de acessos se encontra no processo de busca do professor, ou seja, a maior parte das páginas acessadas é de professores. Desta forma, os conceitos predefinidos como de maior interesse na navegação foram “Professor”, “Publicação” e “Evento”.

A abordagem de classificação bayesiana e a abordagem baseada em freqüência apresentaram resultados semelhantes. Na classificação baseada em freqüência, os conceitos identificados como de maior interesse são os conceitos envolvidos nas páginas *web* mais acessadas. Assim, inicialmente tivemos um valor muito alto (100%) para o conceito “Projeto de Pesquisa” e um valor nulo para os demais, com uma convergência melhor para o valor real com o aumento do número de interações, como ilustra a Figura 26.

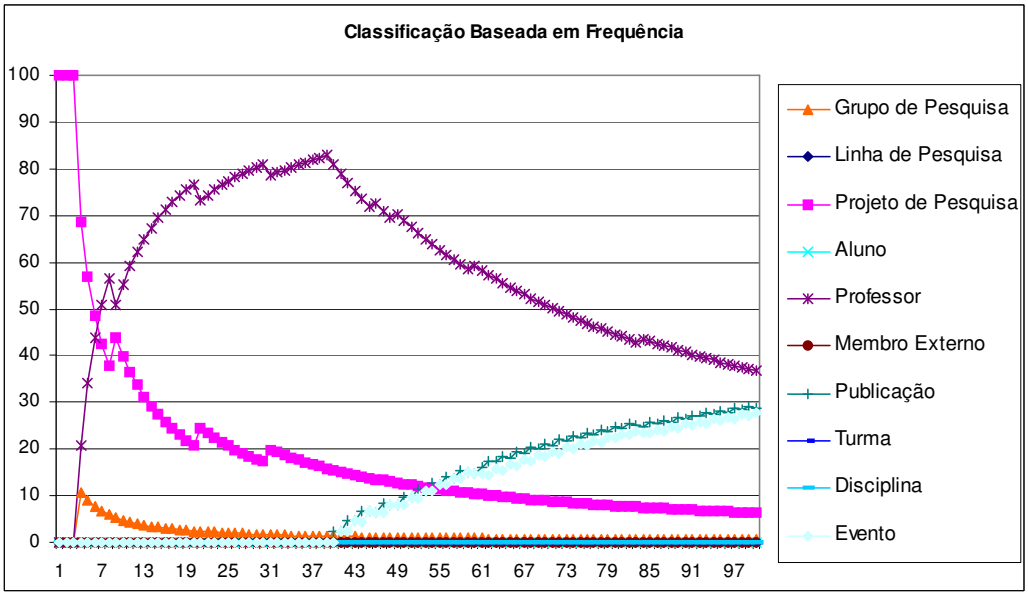

Figura 26. Progresso das percentagens de interesses na navegação para o algoritmo de classificação baseada em freqüência.

Na classificação bayesiana esta total preferência por um único conceito é corrigida, como ilustra a Figura 27, pois o classificador *Naïve* de Bayes considera uma probabilidade igual para todos os conceitos no início, mas com o aumento do número de interações este classificador converge para o mesmo resultado.

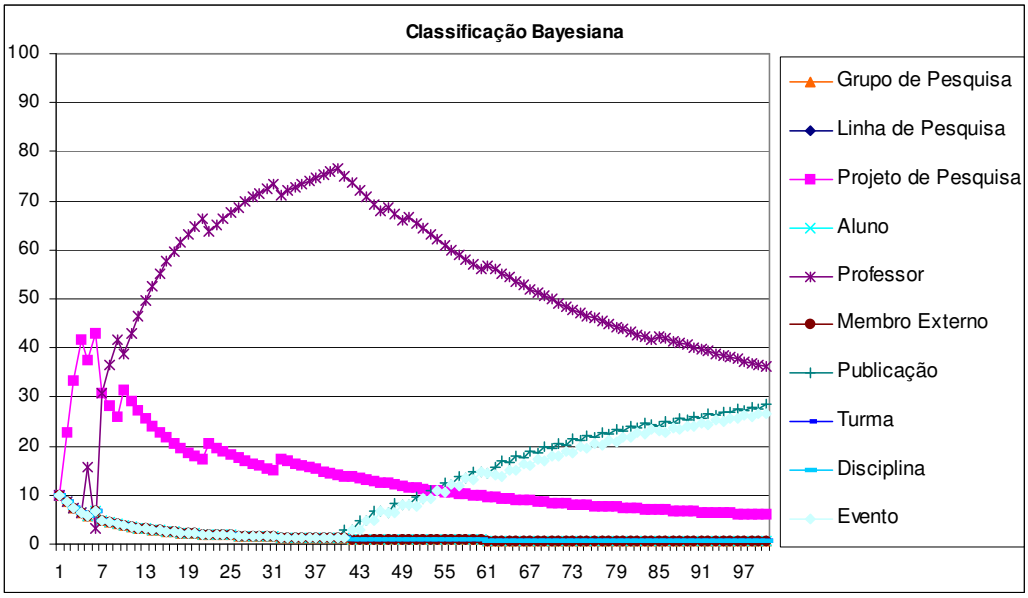

Figura 27. Progresso das percentagens de interesses na navegação para o algoritmo de classificação bayesiana.

Assim, as duas abordagens identificaram os mesmos conceitos predefinidos na navegação, a dizer: “Professor”, “Publicação” e “Evento” respectivamente, como os de maiores interesses do usuário. Pode-se perceber que na classificação baseada em freqüência somente os conceitos envolvidos nas páginas *web* acessadas são ponderados, e todos os outros conceitos que não foram acessados recebem um valor nulo para sua porcentagem de interesse. O mesmo acontece na classificação bayesiana, porém, ela atribui um valor irrisório para a porcentagem de interesse, como ilustra a Tabela 3.

Tabela 3. Lista de preferência resultante da classificação bayseana e baseada em freqüência.

Classificação Bayesiana		**Classificação Baseada em Freqüência**	
Conceitos	*Percentagem de Interesse*	*Conceitos*	*Porcentagem de Interesse*
Professor	36,22%	Professor	37%
Publicação	28,40%	Publicação	29%
Evento	26,45%	Evento	28%
Projeto de Pesquisa	5,99%	Projeto de Pesquisa	6%
Grupo de Pesquisa	0,49%	Grupo de Pesquisa	0%
Aluno	0,49%	Aluno	0%
Turma	0,49%	Turma	0%
Disciplina	0,49%	Disciplina	0%
Linha de Pesquisa	0,49%	Linha de Pesquisa	0%
Membro Externo	0,49%	Membro Externo	0%

Uma característica que consideramos negativa nestas duas abordagens é a desconsideração de conceitos envolvidos nas páginas *web* que ainda não foram acessadas. Estes conceitos podem possuir grande relação semântica com os conceitos de interesse atual do usuário e, devido a essa proximidade semântica, podem ser bons candidatos a se tornarem o foco do interesse do usuário em uma navegação posterior.

Nosso algoritmo também identificou como principais interesses do usuário os conceitos que foram predefinidos antes da navegação, como mostra a Figura 28, porém apontamos algumas diferenças.

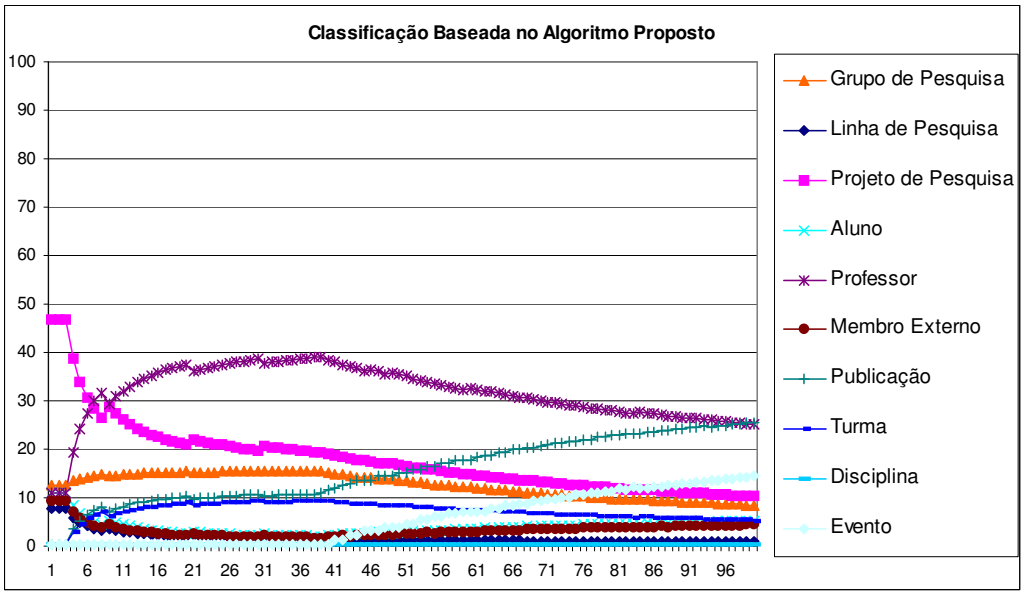

Figura 28. Progresso das percentagens de interesses na navegação quando aplicado o algoritmo proposto.

Primeiramente, o conceito "Publicação" foi identificado como o de maior interesse do usuário, mesmo tendo menor número de acessos na navegação. Este fato é bastante relevante, considerando-se que durante a navegação o processo de busca do professor é secundário, pois ele é somente um caminho para encontrar as publicações e os eventos do professor desejado. Deste modo, levando em conta os aspectos lingüísticos e cognitivos de processo, o algoritmo proposto foi capaz de identificar essa relação semântica e ponderar o conceito "Publicação" (de acordo com sua força cognitiva) cada vez que o conceito "Professor" foi acessado. Como pode ser visto na Tabela 4, a diferença entre a percentagem de interesse dos conceitos "Publicação" e "Professor" ficou pequena, o que confirma um interesse equivalente entre os conceitos. O mesmo não acontece nas outras abordagens, na qual existe uma diferença considerável entre os conceitos "Publicação" e "Professor".

Outra característica que avaliamos como relevante foi o fato do conceito "Grupo de Pesquisa", mesmo não recebendo acesso algum, foi ponderado com um valor relativamente

alto, próximo do conceito “Projeto de Pesquisa” que foi acessado algumas vezes. Isto aconteceu devido a relação direta que possui com os conceitos “Projetos de Pesquisa” e “Professor” no modelo da aplicação (como pode ser visto na Figura 25), ou seja, há uma proximidade semântica considerável entre estes conceitos, uma vez que o usuário seleciona em “Projeto de Pesquisa” e logo após o “Professor” do projeto, isto, provavelmente, provoca uma lembrança do conceito “Grupo de Pesquisa”. Fato este considerado pela teoria de propagação de ativação adotada em nosso algoritmo.

Tabela 4. Lista de preferência resultante da classificação baseada no algoritmo proposto.

Classificação baseada no Algoritmo Proposto	
Conceitos	*Percentagem de Interesse*
Publicação	25,51%
Professor	25,15%
Evento	14,47%
Projeto de Pesquisa	10,25%
Grupo de Pesquisa	8,48%
Turma	5,31%
Aluno	5,13%
Membro Externo	4,36%
Linha de Pesquisa	0,95%
Disciplina	0,39%

Podemos observar também, que o gráfico das percentagens sofreu um achatamento. Este fenômeno pode ser explicado pelo fato de que em nosso algoritmo os conceitos envolvidos nas páginas não acessadas também foram ponderados, isto ocorre devido as suas relações semânticas com os conceitos de maior interesse e sua força cognitiva. Esta ponderação dos conceitos que não aparecem diretamente nas páginas aproxima o percentual de interesse nestes conceitos e permite uma resposta mais rápida às mudanças de interesse futuras do usuário.

Experimento 2

Neste experimento definimos uma navegação com o objetivo de buscar por turmas de alunos. Uma turma é ministrada por um professor e pertence a uma disciplina, entretanto, uma disciplina pode possuir várias turmas. A navegação foi induzida a priorizar os acessos às páginas relacionadas aos conceitos "Disciplina", "Turma" e "Professor", predefinidos como os conceitos de maior interesse. A maior concentração de acessos foi às páginas referentes às disciplinas e turmas respectivamente, sendo o conceito "Disciplina" o mais acessado. As páginas referentes ao conceito "Professor" foram utilizados como ponte para o acesso as páginas das turmas ministradas por ele.

Assim como no Experimento 1, o algoritmo de classificação bayesiana e o baseado em freqüência apresentaram resultados similares. Na classificação baseada em freqüência, os conceitos identificados como de maior interesse são os conceitos envolvidos nas páginas *web* mais acessadas. Como característico deste algoritmo, inicialmente um valor muito alto (100%) foi atribuído para o conceito "Disciplina" e um valor nulo para os demais. O conceito "disciplina" continuou até 1/3 (um terço) da navegação com um valor muito alto em relação aos outros dois conceitos de interesse predefinidos. Após este período a porcentagem de interesse do conceito "Professor" chegou a ser o conceito de maior interesse por volta 50ª interação. No entanto, este logo perdeu força e os conceitos "Disciplina" e "Turma" assumiram as posições de maior interesse, como ilustra a Figura 29.

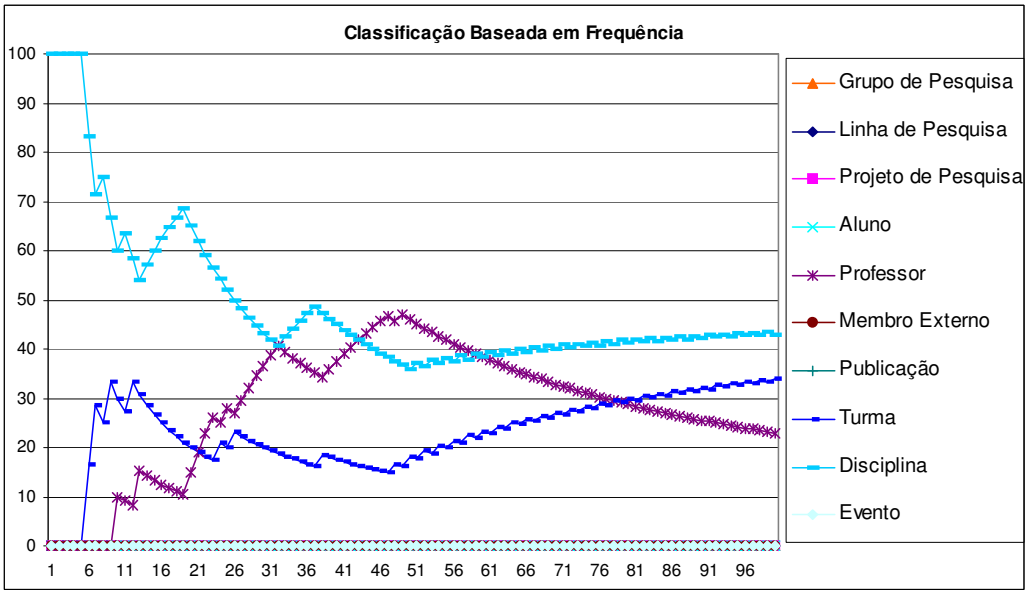

Figura 29. Progresso das percentagens de interesses na navegação para o algoritmo de classificação baseada em freqüência.

Na classificação bayesiana esta total preferência por um único conceito inicialmente é corrigida, como ilustra a Figura 30, pois o classificador *Naïve* de Bayes considera uma probabilidade igual para todos os conceitos no início, mas com o aumento do número de interações este classificador converge para um resultado parecido.

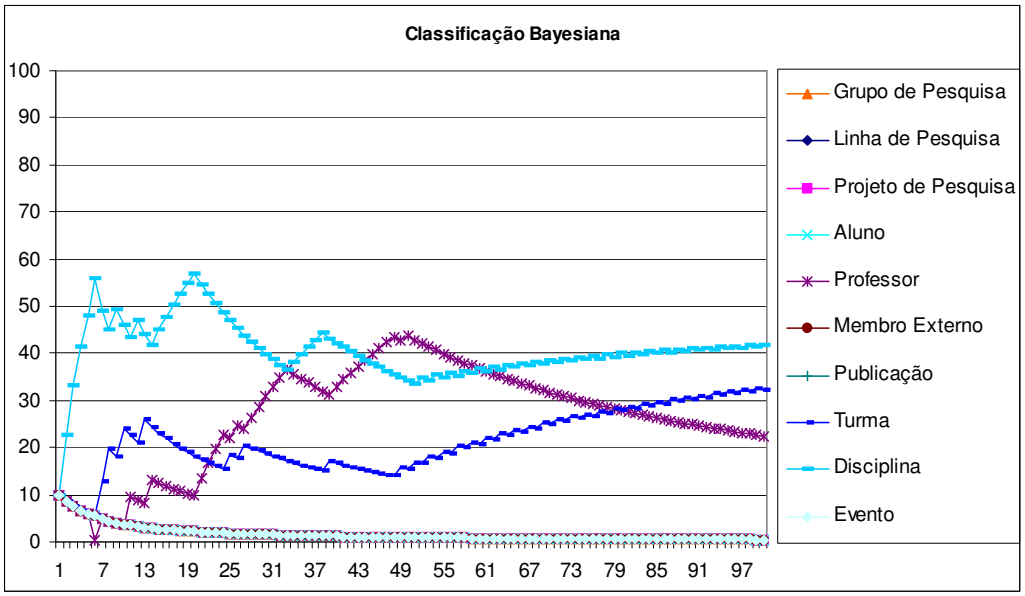

Figura 30. Progresso das percentagens de interesses na navegação para o algoritmo de classificação bayesiana.

Assim, as duas abordagens identificaram os mesmos conceitos predefinidos na navegação, “Disciplina”, “Turma” e “Professor”, como os de maiores interesses do usuário, ficando o conceito “Disciplina” como o de maior interesse do usuário. Isto novamente aconteceu devido ao maior número de acesso às páginas de disciplinas. A classificação baseada em freqüência anula todos os conceitos envolvidos nas páginas *web* que não foram acessadas, atribuindo um valor nulo para sua percentagem de interesse. O mesmo acontece na classificação bayesiana, porém, ela atribui um valor irrisório para a percentagem de interesse, como ilustra a Tabela 5.

Tabela 5. Lista de preferência resultante da classificação bayseana e baseada em freqüência.

Classificação Bayesiana		**Classificação Baseada em Freqüência**	
Conceitos	*Percentagem de Interesse*	*Conceitos*	*Porcentagem de Interesse*
Disciplina	41,94%	Disciplina	43%
Turma	32,19%	Turma	34%
Professor	22,46%	Professor	23%
Aluno	0,49%	Aluno	0%
Projeto de Pesquisa	0,49%	Projeto de Pesquisa	0%
Grupo de Pesquisa	0,49%	Grupo de Pesquisa	0%
Publicação	0,49%	Publicação	0%
Evento	0,49%	Evento	0%
Membro Externo	0,49%	Membro Externo	0%
Linha de Pesquisa	0,49%	Linha de Pesquisa	0%

Como no Experimento 1, os dois algoritmos desconsideram os conceitos envolvidos nas páginas *web* que ainda não foram acessadas. E como mencionamos, consideramos este fato como negativo por motivos já apresentados. Outro ponto considerado negativo foi a distância das porcentagens de interesse entre os conceitos “Disciplina” e “Turma” durante a navegação. Esta distância não reflete o fato de que o usuário somente acessa uma turma

vinculada a uma disciplina, ou seja, a porcentagem de interesse destes dois conceitos deveria ser mais próxima durante a navegação.

O nosso algoritmo também apresentou como resultado final os conceitos predefinidos antes da navegação como os de maior interesse, como mostra a Figura 31, porém destacamos algumas diferenças com relação aos outros dois algoritmos.

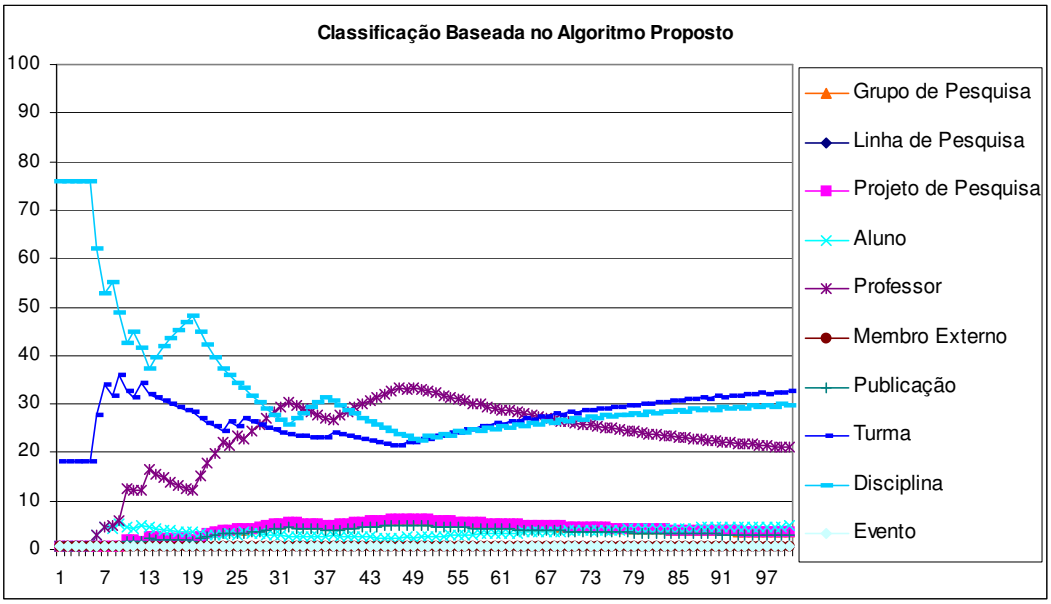

Figura 31. Progresso das percentagens de interesses na navegação quando aplicado o algoritmo proposto.

A ordem de interesse dos três principais conceitos não foi a mesma, nosso algoritmo identificou o conceito “Turma” como o de maior interesse, logo após o conceito “Disciplina” e “Professor”, mesmo o conceito “Turma” tendo menor número de acessos que “Disciplina”. Porém, a diferença entre eles é pequena, em torno de 3%. Esta mudança ocorreu devido à força cognitiva presente no conceito “Turma” e sua relação semântica com o conceito “Disciplina”. Por conseqüência desta proximidade semântica entre os dois conceitos, a todo acesso ao conceito “Disciplina” o conceito “Turma” é ativado, e como ele possui uma força cognitiva considerável ele acabou superando o conceito “Disciplina”. Da mesma forma, quando o conceito “Professor” é acessado, o conceito “Turma” também é reforçado devido à sua relação semântica com o conceito “Professor”. Consideramos este fato como positivo

porque o acesso a turmas depende da escolha de uma disciplina a qual ela pertence e esta dependência foi refletida pela aproximação das porcentagens de interesse entre os conceitos. Desta forma, vimos esta aproximação como um ponto coerente.

Como pode ser visto na Tabela 6, a diferença entre a percentagem de interesse dos conceitos "Turma" e "Disciplina" ficou pequena, o que confirma um interesse equivalente entre os conceitos. O mesmo não acontece nas outras abordagens, nas quais existem uma disparidade considerável entre os conceitos "Turma" e "Disciplina".

Tabela 6. Lista de preferência resultante da classificação baseada no algoritmo proposto.

Classificação baseada no Algoritmo Proposto	
Conceitos	*Percentagem de Interesse*
Disciplina	32,72%
Turma	29,66%
Professor	21,00%
Aluno	4,80%
Projeto de Pesquisa	3,59%
Grupo de Pesquisa	3,57%
Publicação	2,90%
Evento	0,59%
Membro Externo	0,59%
Linha de Pesquisa	0,59%

Da mesma forma como no Experimento 1, o gráfico das percentagens do nosso algoritmo sofreu um achatamento, pelo fato de que os conceitos envolvidos nas páginas não acessadas também foram ponderados, isto ocorre devido as suas relações semânticas com os conceitos de maior interesse e sua força cognitiva.

Neste capítulo relatamos dois dos experimentos realizados, por meio de testes empíricos simulando situações específicas de uso em nosso algoritmo. Como um ponto de referência para comparação de qualidade, aplicamos o mesmo conjunto de dados a outros dois

algoritmos conhecidos: o primeiro baseado em uma abordagem de classificação freqüentista e o segundo baseado em uma abordagem de classificação bayesiana, e mostramos os resultados da comparação. No próximo capítulo discutiremos os resultados obtidos nas simulações, as limitações de nossa abordagem e as propostas para trabalhos futuros.

Capítulo 7

CONCLUSÃO

Na tentativa de melhorar, e facilitar, a interação dos usuários com os *websites* surgiram os mecanismos de personalização de *websites* provendo um conteúdo customizado ao usuário. Estes mecanismos, em sua maioria, analisam a seqüência de interações dos usuários extraindo padrões comportamentais e utilizam estes padrões na construção de um modelo do usuário que servirá como base para a personalização do *website*. Porém, a utilização exclusiva dos dados de navegação pode gerar algumas incoerências como, por exemplo, a ausência de recomendação de uma nova página que é adicionada ao *website*, mesmo ela possuindo informações com fortes relações semânticas ao interesse do usuário [MLADENIC, 1999].

Por outro lado, a incorporação de informações relacionadas à semântica do conteúdo das páginas e/ou a estrutura do *website* melhora todo o processo de personalização [EIRINAKI e VAZIRGIANNIS, 2003] [DAI e MOBASHER, 2003]. Entretanto, quando não se tem uma estrutura formal do conhecimento do domínio vinculado ao *website* há um grande esforço para se extrair características úteis dos conteúdos das páginas. Por esta razão, neste trabalho adotamos uma abordagem baseada na *web* semântica, na tentativa de integrar os benefícios disponibilizados pela estruturação semântica por ela oferecida juntamente com a análise dos comportamentos navegacionais dos usuários. Como resultado desta integração, definimos um modelo de usuário na forma de uma lista de preferências dos conceitos envolvidos no *website*.

No capítulo 4, apresentamos um processo para incorporação de uma base semântica a um *website*, baseado em uma adaptação da arquitetura proposta por Holger Knublauch [KNUBLAUCH *et al*, 2004]. Para criar um modelo conceitual do domínio, definindo formalmente os conceitos e relações existentes neste domínio que é a base para a construção do *website* semântico, empregamos os passos e diretrizes apontadas por Noy e McGuinness [2001]. Estas diretrizes fornecem passos básicos para a definição de uma ontologia de qualidade, porém, acreditamos que para obtermos um modelo mais conciso do conhecimento do domínio ainda são necessários melhoramentos sucessivos, que somente podem ser obtidos com a experiência de utilização de versões anteriores da ontologia.

Mostramos também um processo de construção de um *log* semântico, integrando a semântica das informações contidas nas páginas do *website* semântico a comportamentos navegacionais do usuário. Com o *log* semântico proposto, é possível identificar as páginas acessadas pelo usuário e a semântica das informações nelas contidas. Assim, conseguimos verificar indiretamente quais foram os conceitos acessados pelo usuário em sua navegação. É importante salientar que, limitamos nossa discussão a *websites* constituídos de portais do tipo

intranets, que tenham sido construídos com tecnologia que disponibiliza uma base semântica desde o princípio.

No Capítulo 5, propusemos um algoritmo probabilístico para identificação dos interesses do usuário. Este algoritmo leva em consideração, além das informações semânticas obtidas no *log* semântico, alguns aspectos lingüísticos e cognitivos que afetam o usuário na expressão de seus interesses. Do ponto de vista lingüístico, consideramos o efeito restritivo que o vocabulário disponível ao usuário tem sobre suas formas de expressão. Do ponto de vista cognitivo, consideramos a teoria da propagação de ativação (*spreading activation*), e da força cognitiva, sobre a recuperação de conceitos na memória humana [ANDERSON, 1983] [ROELOFS, 1992] [COLLINS e LOFTUS, 1975] [RUMELHART e MCCLELLAND, 1982].

Para avaliar a qualidade de nosso algoritmo, realizamos alguns testes empíricos nos quais simulamos situações específicas de uso. Como um ponto de referência para comparação de qualidade, aplicamos o mesmo conjunto de dados a outros dois algoritmos conhecidos: o primeiro baseado em uma abordagem de classificação freqüentista e o segundo baseado em uma abordagem de classificação bayesiana.

Os resultados obtidos pelos algoritmos, nos experimentos realizados, baseados na abordagem de classificação freqüentista e na de classificação bayesiana foram bem parecidos. Estes algoritmos identificaram os conceitos que foram predefinidos como de maior interesse do usuário na navegação, porém, apresentaram um valor muito alto para as porcentagens de interesse em relação aos demais conceitos. Isto é explicado pelo fato de que os conceitos que não obtiveram acesso foram desconsiderados pela abordagem baseada na classificação frequentista e receberam um valor irrisório pela abordagem bayesiana. Já nos resultados das simulações com nosso algoritmo, observamos que a diferença entre as porcentagens de interesses diminuiu e o gráfico sofreu um achatamento. Este fenômeno pode ser explicado pelo o fato de que, em nosso algoritmo, os conceitos envolvidos nas páginas não acessadas

também foram ponderados, isto ocorre devido às suas relações semânticas com os conceitos de maior interesse e pela consideração de sua força cognitiva. Esta ponderação dos conceitos que não aparecem diretamente nas páginas aproxima o percentual de interesse nos demais conceitos e permite uma resposta mais rápida às mudanças de interesse futuras do usuário.

Como demonstrado no Capítulo 6, os resultados destes experimentos foram satisfatórios, confirmando nossa expectativa de que a utilização de aspectos lingüísticos e cognitivos faria diferença na determinação das intenções do usuário. A utilização do modelo da aplicação e do modelo de apresentação na determinação das ponderações do nosso algoritmo nos permitiu levar em conta a restrição de vocabulário que o usuário percebe ao utilizar o *website*, pois um usuário só pode expressar seu interesse em algo para o qual ele possui um vocabulário. O emprego da teoria de propagação de ativação se mostrou satisfatória na determinação das ponderações que cada conceito da ontologia deveria receber, devido, principalmente, ao fato de considerar os conceitos adjacentes ao conceito foco. Outro fato que vimos como positivo na utilização da idéia de propagação de ativação foi a possibilidade de atribuir uma força cognitiva diferente para os conceitos, visto que, conforme o contexto e a proximidade semântica, realmente alguns conceitos são lembrados com mais facilidade do que outros.

O algoritmo desenvolvido depende de um conjunto de parâmetros que foram identificados empiricamente. A forma adotada para atribuir valores a estes parâmetros não é a ideal, entretanto foi a alternativa mais viável encontrada dentro das nossas restrições de tempo. Acreditamos que a melhor forma seja construir um algoritmo de aprendizagem para estimar os valores ideais destes parâmetros.

Entretanto, é importante salientar que existem outros fatores que influenciam nos resultados do algoritmo proposto. Os aspectos lingüísticos e cognitivos considerados pelo algoritmo são altamente influenciados pelo modelo da *website* semântico e da ontologia.

Como o modelo da aplicação e de apresentação segmentam a ontologia do domínio, restringindo o vocabulário da linguagem que permite aos usuários se expressarem sobre o domínio nas páginas do *website*, a forma como o *website* é modelado, ou seja, quais, e como, os conceitos são envolvidos e apresentados nas páginas *web* tem influência direta sobre a expressão da intenção do usuário e influenciarão no desempenho do algoritmo proposto. Do mesmo modo, a forma com que o conhecimento é estruturado na ontologia do domínio é determinante para o desempenho do algoritmo. Pois, como o algoritmo considera a estrutura do modelo na determinação da força cognitiva, a coerência das relações semânticas que os conceitos possuem afetam diretamente as ponderações que cada conceito recebe alterando sua força na indicação da intenção do usuário. Com isso acreditamos que além da valoração dos parâmetros, deve haver uma coerência entre a modelagem do conhecimento, bem como sua apresentação no *website* semântico para obtenção de um resultado satisfatório.

Podemos apontar alguns problemas de pesquisa interessantes para discussão e trabalhos futuros. Como próximo trabalho, pretendemos implementar o algoritmo proposto na linguagem Java e realizar os testes com os usuários em situações reais de uso para melhor validar nosso trabalho. Além disso, como já dissemos, é interessante desenvolver um algoritmo de aprendizagem que identifique valores ideais para os parâmetros para a ponderação da força cognitiva e para os valores individuais dos diferentes tipos de conceitos.

REFERÊNCIAS

ANDERSON, J. R. **A spreading activation theory of memory.** *Journal of Verbal Learning and Verbal Behavior* (1983), 22.

ANTONIOU, G.; VAN HARMELEN, F. **A semantic Web primer. Cambridge**, MA: MIT Press, 2004. 238 pp. ISBN 0-262-01210-3.

BAADER, F.; CALVANESE, D.; MCGUINNESS, D. L.; NARDI, D.; PATEL-SCHNEIDER, P. F. **The Description Logic Handbook: Theory, Implementation and Application**. Cambridge University Press, 2002.

BERNERS – LEE, T.; HENDLER, J., LASSILA, O. **The Semantic Web.** Scientifc American, May 2001.

BULTERMAN , D.; RUTLEDGE, L.; HARDMAN, L.; VAN OSSENBRUGGEN, J. **Supporting Adaptive and Adaptable Hypermedia Presentation Semantics**,' in *The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems*, (Rotorua, New Zealand, 5-8 January 1999), 1999.

BRAY, T.; PAOLI, J.; SPELBERG-MCQUEEN, C.; YEGEAU, F. ***Extensible Markup Language (XML) 1.0 (Third Edition)***. W3C Recommendation 04 February 2004. Disponível em: <http://www.w3.org/TR/REC-xml/ >. Acesso em 15 jan 2005.

BRICKLEY, D; GUHA, R. V. ***RDF Vocabulary Description Language 1.0: RDF Schema***. W3C Recommendation 10 February 2004. Disponível em: <http://www.w3.org/TR/rdf-schema>. Acesso em: 10 jan. 2005.

BRUSILOVSKY, P., (2001). **Adaptive Hypermedia, User Modeling and User-Adapted Interaction**, Kluwer Academic Publishers, Netherlands, 2001. pp. 87-110

CHEN L.;SYCARA K. **Web Mate: A Personal Agent for Browsing and Searching**. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multi Agent Systems, AGENTS '98*, ACM, pp. 132 – 139, 1998.

CLAYPOOL, M., GOKHALE, A., MIRANDA, T., MURNIKOV, P., NETES, D. and SARTIN, M.. **Combining Content-based and Collaborative Filters in an Online Newspaper**. In *Proceedings of the ACM SIGIR '99 Workshop on Recommender Systems: Algorithms and Evaluation*. University of California, Berkeley, Aug. 1999

COLLINS, A. M., & LOFTUS, E. F. (1975). **A spreading activation theory of semantic priming**. *Psychological Review*, 82, 407-428.

CONNOLLY, D.; VAN HARMELEN, F.; HORROCKS, L.; MCGUINNESS, D.L.; PATEL-SCHNEIDER, P. F.; STEIN, L. A. **DAML+OIL (March 2001) Reference Description**, The World Wide Web Consortium 2001. Disponível em: < http://www.w3.org/TR/daml+oil-reference >. Acesso em: 25/01/2005.

COOLEY, R., MOBAASHER, B., SRIVASTAVA, J. **Data preparation for mining World Wide Web browsing patterns**. *Knowledge and Information Systems*, February 1999/Vol.1, No. 1

DAI, H. and MOBASHER, B. **A Road map to More Effective Web Personalization: Integrating Domain Knowledge with Web Usage Mining**. *Proc.of the International Conference on Internet Computing 2003 (IC'03)*, Las Vegas, Nevada, June 2003.

DAVIS, R.; SHROBE H.; SZOLOVITS, P. **What is Knowledge Representation.** *AI Magazin*, 14(1):17-33, 1993.

DUA, S.; CHO, E.; IYENGAR, S. S. **Discovery of Web Frequent Patterns and User Characteristics from Web Access Logs – A Framework for Dynamic Web Personalization**. Proc. Int. Conf. on Application-Specific ystems and Software Engineering Technology, pp. 3-8, 2000.

EIRINAKI, M.; VAZIRGIANNIS, M. **Web Mining for Web Personalization**, *ACM Transactions on Internet Technology (TOIT)*, February 2003/ Vol.3, No.1, 1-27.

FALLSIDE, D. C.; WALMSLEY, P. ***XML Schema Part 0: Primer Second Edition***. W3C Recommendation 28 October 2004. Disponível em: <http://www.w3.org/TR/xmlschema-0>. Acesso em 13 mar. 2005

GUARINO, N. and POLI, R. **Formal Ontology in Conceptual Analysis and Knowledge Representation.** Special issue of the International Journal of Human and Computer Studies, vol. 43 n. 5/6, Academic Press, 1995.

GUARINO, Nicola; GIARETTA, Perdaniele. **Ontologies and knowledge bases – towards a terminological clarification.** In: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. Amsterdam: IOS Press, 1995. p. 25-32.

GUARINO, N. **Semantic Matching: Formal ontológica, distinctions for information organization, extraction and integration. SCIE**, 1997a. p. 139-170. Disponível em: <http://citeseer.nj.nec.com/guarino97 semantic. html>. Acesso em: 15 jun. 2005.

GUARINO, N. **Understanding, Building, and Using Ontologies: A Commentary to "Using Explicit Ontologies in KBS Development"**, by van Heijst, Schreiber, and Wielinga. *International Journal of Human and Computer Studies*(46), 1997b 293-310.

GUARINO, N. **Formal Ontology in Information Systems.** Proceedings of FOIS'98, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press, pp. 3-15.

GRUBER, T. R. **A translation approach to portable ontology specification**. Knowledge Acquisition**,** v. 5, p. 199-220, 1993.

GRUBER, T. R. Toward **principles for the design of ontologies used forknowledge sharing**. International Journal of Human and Computer Studies, n. 43, p. 907-928, 1995.

HAARSLEV, V. **Racer – An Inference Engine for Semantic Web**. Concordia: Department of computer science and software engineering, University of Concordia, 2005. Disponível em: <http://www.franz.com/products/racer/Racer_presentation.pdf>. Acesso em: 10 mai. 2006.

HAIGH, S.; MEGARITY, J. **Measuring Web Site Usage: Log File Analysis**. in *Network Notes #57.* National Library of Canada, August 1998.

HENDLER, J., BERNERS-LEE, T. and MILLER, E. "**Integrating Applications on the Semantic Web**," Journal of the Institute of Electrical Engineers of Japan, Vol 122(10), October, 2002, p. 676-680. http://www.w3.org/2002/07/swint.

HORROCKS, I. and PATEL-SCHNEIDER, P. F. **A Proposal for an OWL Rules Language**. WWW2004, May 17-22, 2004, New York, NY USA.

HORROCKS, I.; PATEL-SCHNEIDER, P. F.; BOLEY, H.; TABET, S.; GROSOF, B.; DEAN, M. SWRL: **A Semantic Web Rule Language Combining OWL and RuleML**. 2004. Disponível em: <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>.

JSP. **The JavaServer Pages Technology**, available in: http://java.sun.com/products/jsp/, access in Jan, 2006

JAVA. **The Java Technology**, available in: http://java.sun.com, access in Jan, 2006.

JENA. **The Semantic Web Framework**, available in: http://jena.sourceforge.net/, access in Jan, 2006

KNUBLAUCH, H., FERGERSON, R., NOY, N., MUSEN, M.: **The Protégé-OWL Plugin: An Open Development Environment for Semantic Web Applications**. Third International Semantic Web Conference, Hiroshima, Japan (2004)

KOBSA, Alfred.**Generic User Modeling Systems**. In: User Modeling and User Adapted Interaction v.11 n.1 pp.49-63. Kluwer. Amsterdam, 2001.

KOBSA, Alfred. **User Modeling: Recent Work, Prospects and Hazards**. In: M.Schneider-Hufschmidt, T. Kühme, U. Malinowski, eds. (1993): Adaptive User Interfaces: Principles and Practise. Amsterdam: North Holland Elsevier.

KOIVUNEN, M; MILLER. E. *W3C **Semantic Web Activity***. 2001. Disponível em: http://www.w3.org/2001/12/semweb-fin/w3csw>. Acesso em: 15 jun. 2005.

LASSILA, O.; SWICK, R. R. ***Resource Description Framework (RDF) Model and Syntax Specification*.** W3C Recommendation 10 February 2004. Disponível em: <http://www.w3.org/TR/REC-rdf-syntax>. Acesso em: 13 jun. 2005.

LEI, Y.; MOTTA, E.; DOMINGUE, J. Modelling **Data- Intensive Web Sites with OntoWeaver**, in International Workshop on Web Information Systems Modelling (WISM 2004), Riga, Latvia, 2004.

LIEBERMAN, H. Letizia: An Agent That Assists Web Browsing. *In Proceedings of the International Joint Conference on Artificial Intelligence*, Montreal, CA, 1995

MCGUINNESS, D. L.; VAN HARMELEN, F. **OWL Web Ontology Language Overview**. W3C Recommendation 10 February 2004. Disponível em: <http://www.w3.org/TR/2004/REC-owl-features-20040210/>. Acesso em: 20 jan. 2005.

MACGUINNESS, D. L.; PINHEIRO DA SILVA, P. **Trusting Answers from Web Applications**. In Mark T. Maybury, editor, New Directions in Question Answering. Chapter 21, AAAI/MIT Press, October 2004.

MITCHELL, Tom M. Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression. In http://www.cs.cmu.edu/%7Etom/mlbook/NBayesLogReg.pdf. November, 2005

MIDDLETON, S. E.; SHADBOLT, N. R.; ROURE, D. C. D. **Capturing interest through inference and visualization: ontological user profiling in recommender systems**. In: KCAP '03: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON KNOWLEDGE CAPTURE, 2003, New York, NY, USA. **Anais. . .** ACM Press, 2003. p.62–69.

MIKROYANNIDIS, A., THEODOULIDIS, B. **Web Usage Driven Adaptation of the Semantic Web**, in Proceedings of the End User Aspects of the Semantic Web Workshop, 2nd European Semantic Web Conference, May 29, 2005, Heraklion, Greece.

MLADENIC, D. **Text-learning and related intelligent agents**. Revised version In *IEEE Expert*, special issue on Applications of Intelligent Information Retrieval, 1999

MOBASHER, B., DAI, H., LUO, T., SUNG, Y. and ZHU, J. **Integrating Web Usage and Content Mining for More Effective Personalization**, in *Proc. of the International Conference on E-Commerce and Web Technologies (ECWeb2000)*, Greenwich, UK, September 2000.

MORANDINI, M. Ergo-Monitor: Monitoramento da Usabilidade Em Ambiente Web Por Meio Da Análise De Arquivos De Log. Florianópolis: UFSC, 2003. Tese (Doutorado em Engenharia de Produção – Programa de Pós Graduação em Engenharia de Produção), Universidade Federal de Santa Catarina, Florianópolis, 2003.

NEWTON, G.; POLLOCK, J.; MCGUINNESS, D. L.; PATEL-SCHNEIDER, P. F.; TABET, S.; GROSOF, B.; DEAN, M. **Semantic web rule language (SWRL)**. 2004 Disponível em: <http://www.w3.org/Submission/2004/03/>

NOY, F. N.; McGUINESS, D. L. **Ontology development 101: a guide to creating your first ontology**. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Mar. 2001 Disponível em: <http://protégé.stanford.edu/publications/ontology_development/ontology101-noy-mcguiness.html>. Acesso em: 10 mar. 2006.

OPPERMAN, R. **Adaptive User Support**. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1994.

PALAZZO, L. A. M. **Modelos Proativos para Hipermídia Adaptativa**. 2000. Tese de Doutorado— UFRGS, Porto Alegre.

PARCIA, B; SIRIN, E. **Pellet: An OWL DL Reasoner**, MINDSWAP Research Group, Maryland: University of Maryland, 2003. Disponível em: <http://iswc2004.semanticweb.org/posters/PID-ZWSCSLQK-1090286232.pdf>. Acesso em: 10 maio. 2006.

PERKOWITZ, M.; ETZIONI, O. **Towards adaptive web sites: Conceptual framework and case study**. Artificial Intelligence, 2000.

PINHEIRO DA SILVA, P.P.; MCGUINNESS, D. L.; FIKES, R. **A Proof Markup Language for Semantic Web Services**. Information Systems 2005. Disponível em : http://xml.coverpages.org/PML-StanfordKSL-04-01.pdf

PROTÉGÉ. Disponível em: <http://protege.stanford.edu>. Acesso em 20/06/2006.

QUILLIAN, M. R. (1968). **Semantic memory**. In M. L. Minsky (Ed.), *Semantic Information Processing*. Cambridge, MA: MIT Press

ROELOFS, A. (1992). **A spreading-activation theory of lemma retrieval in speaking**. *Cognition*, 42, 107-142.

RULEML. **The Rule Markup Initiative**. 2004. Disponível em: <http://www.ruleml.org/>.

RUMELHART, D. E., and MCCLELLAND, J. L. (1982). **An interactive activation model of context effects in letter perception: Part 2**. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, 89, 60-94.

SRIVASTAVA, J.; COOLEY, R.; DESHPANDE, M.; TAN, P. (2000). **Web usage mining: Discovery and applications of usage patterns from Web data. ACM SIGKDD Explorations**, v.1, n.2, Janeiro de 2000.

TANUDJAJA, F., MUI, L. Persona: A contextualized and personalized Web search. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)*, Big Island, Hawaii, 2002, volume3 pp. 53

UsableWeb – **Links About Usability**, disponível em http://www.usableweb.com, acesso em: 15 jun 2005.

USCHOLD, M; GRUNINGER, M.. ***Ontologies: Principles*, Methods and Applications**, In: Knowledge Engineering Review (11:2), pp. 93-136.

ZAIHRAYEU, L.; PINHEIRO DA SILVA, P.; MACGUINNESS, D. L. **IWTrust: Improving User Trust in Answers from the Web**. Proceedings of 3rd International Conference on Trust Management (iTrust2005), Springer, Rocquencourt, France, 2005.

ZUKERMAN, I.; LITMAN, D. Natural Language Processing and User Modeling: Synergies and Limitations. **User Modeling and User-Adapted Interaction**, Hingham, MA, USA, v.11, n.1-2, p.129–158, 2001.

Apêndice A

CLASSES IMPLEMENTADAS NO CONTROLE LÓGICO DA APLICAÇÃO

Este Apêndice mostra as classes implementadas no módulo de controle lógico da aplicação apresentado no Capítulo 4. A classe classe OntologyModel implementa os métodos para a criação e armazenamento da ontologia na forma de objeto. A classe OntologyDAO implementa os métodos de acesso ao objeto que representa a ontologia.

```
package OntologyManager.Ontology;

import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.rdf.model.*;
import com.hp.hpl.jena.shared.PrefixMapping;
import com.hp.hpl.jena.db.*;
import javax.swing.*;
import java.io.PrintStream;
import java.util.*;

public class OntologyModel{

        private static String dbURL,dbUser,dbPassword,DB;
        public OntModel model;
        public ModelMaker maker;
        public OntModelSpec spec;

        public OntologyModel(){
           spec = new OntModelSpec( OntModelSpec.OWL_MEM );
        }

        public OntModel criarOntologia(String ontologia){

           model = ModelFactory.createOntologyModel(spec,
           maker.openModel(ontologia));
        return model;
        }

        public OntModel getOntologyModel(){
           return model;
        }

        protected String getDbURL(){
           return this.dbURL;
        }
        protected void setDbURL(String dbURL){
           this.dbURL = dbURL;
        }

        protected String getDbUser(){
           return this.dbUser;
        }
        protected void setDbUser(String dbUser){
           this.dbUser = dbUser;
        }

        protected String getDbPassword(){
           return this.dbPassword;
        }
        protected void setDbPassword(String dbPassword){
           this.dbPassword = dbPassword;
        }

        protected String getDB(){
           return this.DB;
        }
        protected void setDB(String db){
           this.DB = db;
        }

        protected ModelMaker getMaker(){
```

```
            return maker;
        }

        protected OntModelSpec getSpec(){
            return spec;
        }

        protected ModelMaker setMaker() {
            try {
                String className = "com.mysql.jdbc.Driver";
                Class.forName(className);
                IDBConnection conn = new DBConnection ( dbURL, dbUser,
                dbPassword, DB );
                maker = ModelFactory.createModelRDBMaker(conn);
                spec.setModelMaker( maker );
                return maker;
            }
            catch (Exception e) {
                e.printStackTrace();
                System.exit( 1 );
            }
            return null;
        }
}

package OntologyManager.Ontology;

import com.hp.hpl.jena.util.ModelLoader;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;
import OntologyManager.DB.ConnectionFactory;
import com.hp.hpl.jena.rdf.model.ModelMaker;
import com.hp.hpl.jena.rdf.model.ModelFactory;
import com.hp.hpl.jena.ontology.*;
import com.hp.hpl.jena.query.*;
import com.hp.hpl.jena.rdf.model.*;

public class OntologyDAO {

    private ModelMaker maker;
    private String uri;
    private OntModel m;
    private Model depData;
    private OntModel depSchema;

        public OntologyDAO()
        {
            depSchema=ModelFactory.createOntologyModel(OntModelSpec.OWL_MEM,
            null );
            depSchema.getDocumentManager().addAltEntry(
            "http://www.w3.org","file:c://ontologia//Departamento.owl");
            depSchema.read("file:c://ontologia//Departamento.owl");
            this.uri = "http://www.owl-ontologies.com/unnamed.owl/";
        }

        public ResultSet retornaQuery(String queryString)
        {
            Query query = QueryFactory.create(queryString);
            QueryExecution qe = QueryExecutionFactory.create(query,
            depSchema);
```

```
            ResultSet results = qe.execSelect();
            return results;
        }

        public OntModel getModelSchema()
        {
            return this.depSchema;
        }

        public Model getModelData()
        {
            return this.depData;
        }

        public String getURL()
        {
            return this.uri;
        }

        public OntModel createOntologyModel(String ontology)
        {
            OntModelSpec spec = new OntModelSpec( OntModelSpec.OWL_MEM );
            m = ModelFactory.createOntologyModel(spec,maker.openModel(
            ontology));
            return m;
        }

        public OntModel getOntology()
        {
            return this.depSchema;
        }

        public OntClass getOntClass(String className)
        {
            OntClass c = m.getOntClass(className);
            return c;
        }

        public Iterator getRootClasses( OntModel m )
        {
            List roots = new ArrayList();
            for (Iterator i = m.listClasses();  i.hasNext(); ){
                OntClass c = (OntClass) i.next();
                if (c.isAnon()){
                    continue;
                }
                if (c.hasSuperClass( m.getProfile().THING(), true ) ){
                    roots.add( c );
                }
                else if (c.getCardinality(m.getProfile().SUB_CLASS_OF())==0)
                {
                    roots.add( c );
                }
            }
            return roots.iterator();
        }
}
```

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Administração
Baixar livros de Agronomia
Baixar livros de Arquitetura
Baixar livros de Artes
Baixar livros de Astronomia
Baixar livros de Biologia Geral
Baixar livros de Ciência da Computação
Baixar livros de Ciência da Informação
Baixar livros de Ciência Política
Baixar livros de Ciências da Saúde
Baixar livros de Comunicação
Baixar livros do Conselho Nacional de Educação - CNE
Baixar livros de Defesa civil
Baixar livros de Direito
Baixar livros de Direitos humanos
Baixar livros de Economia
Baixar livros de Economia Doméstica
Baixar livros de Educação
Baixar livros de Educação - Trânsito
Baixar livros de Educação Física
Baixar livros de Engenharia Aeroespacial
Baixar livros de Farmácia
Baixar livros de Filosofia
Baixar livros de Física
Baixar livros de Geociências
Baixar livros de Geografia
Baixar livros de História
Baixar livros de Línguas

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo