



Pós-Graduação em Ciência da Computação

***PSemRef: Personalização de Consultas em Ambientes
Distribuídos e Dinâmicos***

Por

Thiago Arruda Neves

Dissertação de Mestrado



Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, JULHO/2010



Universidade Federal de Pernambuco

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Thiago Arruda Neves

*PSemRef: Personalização de Consultas em Ambientes
Distribuídos e Dinâmicos*

*ESTE TRABALHO FOI APRESENTADO A
PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO DO CENTRO DE
INFORMÁTICA DA UNIVERSIDADE FEDERAL
DE PERNAMBUCO COMO REQUISITO
PARCIAL PARA OBTENÇÃO DO GRAU DE
MESTRE EM CIÊNCIA DA COMPUTAÇÃO.*

ORIENTADORA: **ANA CAROLINA SALGADO**

CO-ORIENTADORA: **DAMIRES YLUSKA DE
SOUZA FERNANDES**

RECIFE, JULHO/2010

Neves, Thiago Arruda

PSemRef: personalização de consultas em ambientes distribuídos e dinâmicos / Thiago Arruda Neves. - Recife: O Autor, 2010.

xii, 101 folhas: il., fig., tab.

Dissertação (mestrado) – Universidade Federal de Pernambuco. Cln. Ciência da Computação, 2010.

Inclui bibliografia e apêndice.

1. Banco de dados. 2. Personalização de consulta. 3. Semântica. 4. Ambientes distribuídos e dinâmicos. I. Título.

025.04

CDD (22. ed.)

MEI2010 – 0109

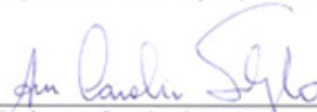
Dissertação de Mestrado apresentada por **Thiago Arruda Neves** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "**PSemRef: Personalização de Consultas em Ambientes Distribuídos e Dinâmicos**", orientada pela **Profa. Ana Carolina Brandão Salgado** e aprovada pela Banca Examinadora formada pelos professores:



Profa. Patricia Cabral de Azevedo Restelli Tedesco
Centro de Informática / UFPE



Profa. Bernadette Farias Lóscio
Departamento de Computação / UFC



Profa. Ana Carolina Salgado
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 30 de julho de 2010.



Prof. Nelson Souto Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

A **Deus**, que sempre me guiou nessa jornada,

Aos meus pais, **Manuel e Neide**,

Ao meu irmão, **Felipe**,

Minha gratidão pelo apoio, incentivo,
paciência e compreensão que tornaram

esta dissertação possível.

*Faço cada noite o balanço do meu dia:
Se ele foi estéril como educação pessoal,
Ou impiedoso com os que me fizeram perdê-lo...
A vida corriqueira tem tão pouca importância, e se parece tanto;
A vida interior é difícil de dizer, há uma espécie de pudor,
É tão pretensioso falar a respeito.
Você não pode imaginar como é a única coisa que importa para mim,
O que modifica todos os valores,
Até meus julgamentos sobre os outros...
Antes de tudo, sou duro comigo mesmo,
E tenho todo o direito de renegar nos outros
O que renego ou corrijo em mim.*

Antoine de Saint-Exupéry

Agradecimentos

Gostaria de agradecer primeiramente às pessoas que, independentemente de qualquer situação, sempre estiveram do meu lado desde o início da minha graduação e nesse período de mestrado: minha família, em especial meus pais e meu irmão. Obrigado pelo apoio contínuo, pelo incentivo incansável e pela compreensão que tornaram possível que eu chegasse onde estou hoje e que me permitem almejar grandes conquistas.

Agradeço de forma muito especial minhas orientadoras Ana Carolina e Damires Souza pelos ensinamentos, amizade, paciência, sem contar a enorme contribuição e apoio durante o período de desenvolvimento deste trabalho. Gostaria de agradecer-las também pela confiança e oportunidade dada, em 2008, de ingressar no grupo SPEED da UFPE, que me permitiu não só o desenvolvimento do meu projeto de mestrado como também do meu trabalho de graduação. Aprendi muito com vocês.

Agradeço também aos demais membros do grupo SPEED por sempre terem criado um excelente ambiente de trabalho, de aprendizado e de companheirismo. Um agradecimento especial a Thiago Pachêco que, além de ter ingressado no SPEED também em 2008, é um grande amigo desde o início de nossa graduação em 2004. Meu agradecimento ao colega Gustavo Ferraz, que me auxiliou com o desenvolvimento da ferramenta deste trabalho.

Gostaria de agradecer também aos docentes do Centro de Informática da UFPE pelos ensinamentos e auxílio durante meus anos de graduação e mestrado. Um agradecimento especial aos amigos do CIn com os quais mantenho amizade desde 2004: Age, Xico, BLBS, Hugo, Leonardo, Arthur, Icamaan, Bill, Vinícius, Brayner, Renato, Menelau, Fábio (espero não ter esquecido ninguém). Obrigado por seu companheirismo, amizade e pelos momentos de descontração durante esses anos de CIn. Espero manter a amizade de vocês durante muito tempo.

Por último, mas não menos importante, agradeço a Deus pela força e saúde que me possibilitaram cumprir mais uma etapa do meu ciclo acadêmico.

Agradeço, enfim, a todos que acreditaram e participaram de alguma maneira da minha jornada na UFPE e que continuam me incentivando para os novos desafios que me aguardam. **Muito Obrigado!**

Resumo

O processamento de consultas é um tema de pesquisa muito importante em ambientes distribuídos e dinâmicos, tais como os *Peer Data Management Systems*. Uma questão crucial nesse processo é a reformulação da consulta em um ponto de origem em termos de um ponto destino, considerando as correspondências existentes entre eles.

Como parte do projeto SPEED (*Semantic Peer-to-Peer Data Management System*) uma ferramenta de reformulação de consultas, chamada *SemRef* (*Semantic Reformulator*), foi desenvolvida. Por seu intermédio, as consultas submetidas pelo usuário são reformuladas, executadas e os resultados integrados para exibição. O *SemRef* explora um conjunto de correspondências semânticas para enriquecimento da consulta na sua reformulação. A idéia é produzir um conjunto de resultados que expresse o que os usuários consideram importante no momento em que submetem suas consultas, considerando a dinamicidade do sistema.

De uma maneira geral, os resultados das consultas são gerados de forma única sem refletir as preferências e perfis dos usuários. Neste trabalho, o objetivo é estender o módulo *SemRef*, passando a ser chamado *PSemRef* (*Personalized Semantic Reformulator*), propondo uma abordagem na qual as consultas sejam personalizadas e seus resultados ordenados de acordo com as preferências dos usuários e um conjunto de informações contextuais.

Abordamos o problema da personalização de consultas no SPEED, bem como a especificação de nossa proposta para solucioná-lo, a qual envolve a obtenção de uma informação contextual estática (na forma de perfil do usuário) e dinâmica (localização do usuário). Capturamos as preferências do usuário na seleção e na priorização de um conjunto de variáveis de enriquecimento, as quais estão ligadas diretamente à reformulação da consulta. Todas essas informações citadas influenciam na geração de um conjunto ordenado de resultados das consultas. Detalhamos também os aspectos de implementação do nosso trabalho e, por fim, apresentamos a experimentação realizada envolvendo um estudo de caso com os resultados que foram obtidos.

Palavras-Chave: Personalização de Consultas, Preferências do Usuário, Contexto, *Ranking*, Ambientes Distribuídos e Dinâmicos

Abstract

Query answering is a very important research topic in distributed and dynamic environments, like *Peer Data Management Systems*. A crucial issue concerning this process is how we can reformulate a query posed at a peer into a new query expressed in terms of a target peer, considering existing correspondences between them.

As part of the SPEED project (*Semantic Peer-to-Peer Data Management System*), a reformulation query tool, called *SemRef* (*Semantic Reformulator*), was developed. Queries submitted by the users are reformulated, executed and their results are integrated for viewing. *SemRef* uses semantic correspondences for query reformulation. The idea is to produce a result set which expresses what users consider important at query submission time, considering the dynamicity of the system.

Usually, query results are generated without reflecting users' preferences and profiles. In this work, our goal is to extend the *SemRef* module, now calling it as *PSemRef* (*Personalized Semantic Reformulator*), by proposing an approach in which queries are personalized and their results are ranked according to users' preferences and some contextual information.

We address the problem of query personalization in SPEED, and the specification of our proposal to solve it by dealing with static contextual information (user profile) and dynamic (user location) contextual information. We capture user's preferences through a set of enriching variables, which are selected and prioritized, and then linked directly to the query reformulation process. All this mentioned information influences the generation of a ranked set of query results. We detail implementation aspects of our work and, finally, we present the experimentation we have done showing some obtained results.

Keywords: Query Personalization, User's Preferences, Context, *Ranking*, Dynamic and Distributed Environments

Sumário

Capítulo 1.....	1
1.1. Motivação.....	1
1.2. Objetivos	2
1.3. Estrutura da Dissertação.....	2
Capítulo 2.....	4
2.1. Personalização	4
2.2. Abordagens Existentes para Personalização	6
2.2.1. Bancos de Dados	6
2.2.2. Recomendação.....	9
2.2.3. Busca na Web.....	10
2.3. Personalização e Contexto	13
2.3.1. Contexto e Bancos de Dados.....	13
2.3.2. Contexto e Recomendação.....	16
2.4. Análise Comparativa.....	17
2.5. Considerações.....	18
Capítulo 3.....	20
3.1. Reformulação de Consultas em PDMS.....	20
3.2. Abordagens Existentes para o Processamento de Consultas em PDMS.....	23
3.2.1. HELIOS	23
3.2.2. SUNRISE.....	24
3.2.3. Humboldt Discoverer.....	25
3.2.4. SomeRDFS.....	27
3.3. Considerações.....	28
Capítulo 4.....	29
4.1. O Sistema SPEED.....	29
4.1.1. Arquitetura	30
4.1.2. Geração de Correspondências entre Esquemas	31
4.1.3. Conexão de um Novo Ponto.....	35
4.1.4. Processamento de Consultas	36
4.2. Reformulação de Consultas no SPEED	36
4.2.1. Características	37
4.2.2. Formato das Consultas.....	37
4.2.3. Reformulando Consultas.....	39
4.2.4. Arquitetura do <i>SemRef</i>	41
4.2.5. Casos de Uso	43
4.3. Um Exemplo Prático.....	44
4.4. Personalização das Consultas no <i>SemRef</i>	45
4.4.1. Arquitetura do <i>PSemRef</i>	46
4.4.2. Priorização das Variáveis de Enriquecimento	48
4.4.3. Utilização do Contexto do Usuário.....	50
4.5. Considerações.....	53
Capítulo 5.....	54
5.1. Implementação do Módulo <i>PSemRef</i>	54
5.1.1. Reformulação e Execução de Consultas	55
5.1.2. Interfaces Gráficas.....	60
5.2. Implementação da Personalização	66

5.2.1. Variáveis de Enriquecimento	67
5.2.2. Perfil e Localização do Usuário	68
5.3. Exemplo de Uso.....	72
5.4. <i>Feedback</i> de Usuários	81
5.5. Considerações.....	82
Capítulo 6.....	84
6.1. Considerações Finais	84
6.2. Contribuições.....	85
6.3. Trabalhos Futuros	85
Referências Bibliográficas.....	87
Apêndice A.....	91
Apêndice B.....	93
Apêndice C.....	95
Apêndice D.....	99

Lista de Figuras

Figura 1 – Arquitetura de um banco de dados personalizado [Koutrika et al. 2004a].....	7
Figura 2 – Esquema textual de uma base de dados [Koutrika et al. 2004a].....	8
Figura 3 – Perfil de um usuário [Koutrika et al. 2004a].....	8
Figura 4 – Esquema gráfico de uma base de dados [Stefanidis et al. 2009].....	9
Figura 5 – Arquitetura de uma busca na web personalizada [Sendhikumar et al. 2008]	11
Figura 6 – Hierarquia de parâmetros contextuais [Stefanidis et al. 2007b].....	14
Figura 7 – Esquema da base de dados [Stefanidis et al. 2005].....	15
Figura 8 – Cubos de dados para cada parâmetro contextual [Stefanidis et al. 2005].....	15
Figura 9 – Exemplos de consultas sensíveis ao contexto [Stefanidis et al. 2005]	16
Figura 10 – Arquitetura do CARS [Abbar et al. 2009]	17
Figura 11 – Exemplo de PDMS e caminhos semânticos entre os <i>peers</i> [Tatarinov, Halevy 2004]..	22
Figura 12 – HELIOS <i>toolkit</i> [Castano et al. 2003]	23
Figura 13 – Exemplo de organização de uma rede no SUNRISE [Mandreoli et al. 2008]	24
Figura 14 – Arquitetura estendida do Humboldt Discoverer [Herschel, Heese 2005].....	25
Figura 15 – Construtores utilizados no SomeRDFS [Adjiman et. al 2007]	27
Figura 16 – Arquitetura do SPEED [Pires 2009].....	30
Figura 17 – Utilizando uma OD para especificar as Correspondências Semânticas [Souza 2009]..	33
Figura 18 – Ontologia de Domínio [Souza 2009].....	34
Figura 19 – Ontologias Comparáveis [Souza 2009]	34
Figura 20 – Descoberta de <i>clusters</i> semânticos através de comparação ontológica [Pires 2009]	35
Figura 21 – Exemplo de Consulta SPARQL.....	38
Figura 22 – Cenário das consultas no <i>SemRef</i> [Souza 2009].....	41
Figura 23 – Arquitetura do <i>SemRef</i>	42
Figura 24 – Diagrama de Casos de Uso do <i>SemRef</i>	43
Figura 25 – Trecho de correspondências semânticas entre O_1 e O_2	44
Figura 26 – Arquitetura do <i>PSemRef</i>	47
Figura 27 – Diagrama de Casos de Uso do <i>PSemRef</i>	47
Figura 28 – Resumo da Especificação do <i>PSemRef</i>	48
Figura 29 – Atividades do <i>PSemRef</i>	52
Figura 30 – Correspondências Semânticas geradas entre P_1 e P_2	55
Figura 31 – Visão em alto nível do algoritmo de reformulação do <i>PSemRef</i> [Souza 2009]	56
Figura 32 – Tradução consulta DL – SPARQL (um conceito).....	59
Figura 33 – Tradução consulta DL – SPARQL (disjunção).....	59
Figura 34 – Tradução consulta DL – SPARQL (conjunção).....	59
Figura 35 – Tradução consulta DL – SPARQL (negação).....	60
Figura 36 – Consulta em SPARQL (propriedades)	60
Figura 37 – Tela Principal do <i>PSemRef</i> – Consulta em DL.....	61
Figura 38 – Tela Principal do <i>PSemRef</i> – Consulta em SPARQL	61
Figura 39 – Template SPARQL de uma consulta por Propriedades.....	63
Figura 40 – Tela de escolha de Variáveis de Enriquecimento.....	63
Figura 41 – Tela do Log de Reformulação.....	65
Figura 42 – Tela do Log de Resultados.....	66
Figura 43 – Tela de Priorização das Variáveis de Enriquecimento.	67
Figura 44 – Algoritmo de Ordenação dos Resultados Enriquecidos	68
Figura 45 – Tela de Sessão de um usuário	69
Figura 46 – Algoritmo de Ordenação dos Resultados Exatos	70
Figura 47 – Resultados Personalizados	72
Figura 48 – Ontologias do Estudo de Caso	73
Figura 49 – Consulta Q_1 submetida por Thiago.....	77
Figura 50 – Consulta Q_2 submetida por Jorge	79
Figura A.1 – Algoritmo de Reformulação [Souza 2009]	92
Figura B.1 – Método <i>Build_Exact_Reformulation</i> [Souza 2009].....	93

Figura B.2 – Método <i>Build_Enriched_Reformulation</i> [Souza 2009].....	93
Figura B.3 – Método <i>Build_Final_Exact_Reformulation</i> [Souza 2009].....	94
Figura B.4 – Método <i>Build_Final_Enriched_Reformulation</i> [Souza 2009]	94
Figura D.1 – Ontologia <i>Semiport.owl</i>	99
Figura D.2 – Ontologia <i>Univbench.owl</i>	100
Figura D.3 – Ontologia de Domínio <i>UnivCSCMO.owl</i>	101

Lista de Tabelas

Tabela 1 – Análise Comparativa de abordagens de Personalização de Consultas.....	18
Tabela 2 – Mapa de Consultas Reformuladas produzidas.....	43
Tabela 3 – Áreas de Estudo e Países.....	73
Tabela 4 – Conceitos Personalizados.....	73
Tabela 5 – Correspondências Semânticas entre O_1 e O_2	74
Tabela 6 – Perfis de Usuário cadastrados.....	75
Tabela 7 – Ordem de prioridade das variáveis de enriquecimento para Q_1	76
Tabela 8 – Resultados para a consulta Q_1	77
Tabela 9 – Ordem de prioridade das variáveis de enriquecimento para Q_2	79
Tabela 10 – Resultados para a consulta Q_2	80
Tabela C.1 – Instâncias cadastradas em <i>Semiport</i>	95
Tabela C.2 – Instâncias cadastradas em <i>Univbench</i>	97

Capítulo 1

Introdução

Neste capítulo, será feita uma breve introdução dos aspectos que motivaram e caracterizaram este trabalho. Também apresentaremos os objetivos desta pesquisa, assim como a estrutura desta dissertação.

Este capítulo está organizado como segue. A Seção 1.1 descreve a motivação para o nosso trabalho. Os objetivos que pretendemos alcançar com nossa proposta estão descritos na Seção 1.2. Finalmente, a Seção 1.3 especifica a estrutura desta dissertação.

1.1. Motivação

Os PDMS (*Peer Data Management System*) são sistemas de integração de dados distribuídos que possibilitam o acesso a diversas bases de dados heterogêneas, sem a necessidade de representação de um esquema único global e centralizado. Ao invés disso, cada *peer* dentro do sistema representa uma fonte de dados autônoma com sua própria representação de esquema de dados. O compartilhamento desses esquemas permite aos *peers* a troca de informações entre si e a possibilidade de realização de outras tarefas, como o processamento de consultas.

O processamento de consultas é um tema de pesquisa muito importante em ambientes dinâmicos tais como os *Peer Data Management Systems* [Stuckenschmidt et al. 2005; Tatarinov, Halevy 2004]. Uma tarefa importante nesse processo é a reformulação de uma consulta submetida em um *peer* (fonte de dados) em uma nova consulta expressa sob os termos de um *peer* destino – considerando correspondências semânticas existentes entre os termos dos esquemas dos *peers*.

O sistema SPEED [Pires 2009] é um PDMS baseado em semântica que possui como objetivo prover facilidades para o compartilhamento e a integração dos dados distribuídos ao longo de pontos conectados (*peers*), os quais possuem seus esquemas de dados representados por ontologias. Um dos principais serviços oferecidos pelo SPEED é o processamento de consultas. Uma consulta submetida em um *peer* pode obter uma resposta relevante de qualquer outro *peer* no PDMS, desde que estes estejam ligados por um caminho semântico através de mapeamentos [Tatarinov, Halevy 2004].

Cada *peer* do SPEED possui um módulo de consultas no qual os usuários submetem suas requisições. A consulta original é executada em um *peer* fonte e reformulada a partir de correspondências semânticas existentes entre os termos das ontologias do *peer* fonte e de *peers* destino. Após essa reformulação, as consultas reformuladas são executadas nos *peers* destino. Por fim, os resultados são integrados e apresentados ao usuário.

No SPEED, a ferramenta *SemRef* (*Semantic Reformulator*) [Neves 2008; Souza et al. 2009a; Souza et al. 2009b] foi desenvolvida, na qual o usuário submete suas consultas, as quais são reformuladas, executadas e os resultados integrados para posterior exibição. O *SemRef* explora um conjunto de correspondências semânticas para reformulação das consultas submetidas. A idéia é produzir um conjunto de resultados que expresse o que os usuários consideram importante no momento em que submetem suas consultas, considerando a dinamicidade do sistema e essas correspondências semânticas existentes.

Uma questão envolvendo o *SemRef* é que os resultados recuperados por uma mesma consulta Q são apresentados de uma mesma maneira para todos os usuários que a submetem. Além disso, estes resultados não possuem uma ordenação (*ranking*) bem definida para uma melhor visualização. Neste sentido, o principal aspecto de pesquisa e desenvolvimento deste trabalho é como tornar as consultas do SPEED personalizadas, ou em outras palavras, como podemos ordenar o conjunto de resultados recuperados de uma maneira que reflita as preferências e perfis de diferentes usuários.

1.2. Objetivos

Neste trabalho, nosso objetivo é estender o módulo *SemRef* propondo uma abordagem na qual as consultas sejam personalizadas e seus resultados ordenados de acordo com as preferências dos usuários e um conjunto de informações contextuais, ou seja, informações que caracterizam a situação do usuário na sessão de consultas. Nomeamos o nosso trabalho de *PSemRef* (*Personalized Semantic Reformulator*).

Esperamos que cada usuário, ao submeter suas consultas no SPEED, receba um conjunto de resultados relevante e ordenado de acordo com suas preferências e características, considerando não só a consulta original em si como também as reformulações que podem ser produzidas.

1.3. Estrutura da Dissertação

A organização deste trabalho está da seguinte maneira:

- O **Capítulo 1** introduz e motiva as idéias principais do trabalho, destacando os objetivos e a estrutura da dissertação;
- O **Capítulo 2** discute o que está sendo pesquisado e desenvolvido na área de personalização da informação, mais especificamente de consultas submetidas sobre dados de interesses dos usuários. A conceitualização de personalização e suas principais áreas de aplicação também são vistas neste capítulo;
- O **Capítulo 3** destaca as características e como funciona o processamento de consultas em alguns dos principais sistemas PDMS encontrados na literatura;
- O **Capítulo 4** aborda o problema da personalização de consultas no sistema SPEED, bem como a especificação de nossa proposta para solucioná-lo;
- O **Capítulo 5** aborda as questões de implementação do nosso trabalho, detalhando os aspectos do seu desenvolvimento. Fornece também exemplos de utilização do *PSemRef* envolvendo um estudo de caso e um *feedback* de usuários reais sobre nossa proposta;
- O **Capítulo 6** resume o trabalho realizado discutindo as contribuições alcançadas e indicando algumas pesquisas futuras para extensão de nossa proposta.

Finalmente, um conjunto de apêndices é fornecido da seguinte maneira:

- O **Apêndice A** ilustra a versão completa (em pseudocódigo) do algoritmo de reformulação da nossa abordagem;
- O **Apêndice B** contém os métodos complementares que auxiliam o processo de reformulação do nosso trabalho;
- O **Apêndice C** contém tabelas as quais ilustram os dados cadastrados nas ontologias que envolvem o estudo de caso da nossa abordagem;
- O **Apêndice D** ilustra a estrutura completa das ontologias envolvidas no estudo de caso do nosso trabalho.

Capítulo 2

Personalização de Consultas

Este capítulo tem como objetivo principal discutir o que está sendo pesquisado e desenvolvido na área de personalização da informação, mais especificamente da personalização aplicada a consultas submetidas sobre dados de interesse dos usuários. A definição de personalização e as suas principais áreas de aplicação são vistas neste capítulo. O conceito de personalização é importante para compreensão das características e objetivos do nosso trabalho.

Este capítulo está organizado como segue. A Seção 2.1 conceitua e lista as principais características da personalização. Algumas abordagens para o uso de personalização em consultas são apresentadas na Seção 2.2. A utilização de informação contextual como forma de melhorar a personalização é discutida na Seção 2.3. A Seção 2.4 traz uma análise comparativa entre as principais abordagens apresentadas. Finalmente, a Seção 2.5 conclui o capítulo com algumas considerações.

2.1. Personalização

Segundo o dicionário Priberam da língua portuguesa [Priberam 2009], a personalização pode ser definida como o ato ou efeito de personalizar. Ainda segundo esta fonte, “personalizar” significa tornar algo pessoal, individualizar; designar pelo nome; dar caráter original a um objeto fabricado em série; ou ainda, adaptar algo às preferências ou necessidades do utilizador.

Personalização pode envolver a utilização de tecnologia para amenizar as diferenças entre indivíduos. Possui como conceito chave a questão da preferência do usuário, ou seja, cada indivíduo é reconhecido de alguma maneira ao acessar um serviço (uma página na web, por exemplo), baseado em suas características e/ou comportamento. Sendo assim, esse serviço estará disposto para cada usuário de uma forma diferente, ou seja, estará personalizado.

As páginas web são o maior foco da utilização de personalização na atualidade. Elas podem estar personalizadas aos usuários baseando-se em dados implícitos como, por exemplo, através dos cliques dos usuários em certos *links* da página, ou ainda a partir de subpáginas visitadas; ou podem também fazerem uso de dados explícitos,

solicitando algum tipo de configuração do usuário. Neste último caso, podemos também dizer que o serviço é customizado.

Analisando os trabalhos estudados a respeito do uso de personalização [Koutrika et al. 2004a; Kostadinov et al. 2007; Stefanidis et al. 2009; Sendhikumar et al. 2008; Giannopoulos et al. 2009; Campi et al. 2009] (discutidos nas Seções 2.2 e 2.3 deste capítulo), definimos três categorias básicas de personalização:

- Baseada em Perfil: perfis são criados contendo preferências e características pessoais dos usuários;
- Baseada em Comportamento: normalmente armazenado na forma de logs que contêm as ações realizadas pelos usuários em cada sessão de utilização do serviço;
- Baseada em Colaboração: este último caso é comum em sistemas de recomendação, onde o serviço é personalizado também a partir de perfis semelhantes aos do usuário que está acessando o serviço.

Ainda a partir dos trabalhos estudados, podemos categorizar três métodos gerais à realização da personalização:

- Implícito: realizado pelo próprio serviço (página web), ou sistema;
- Explícito: realizado pelo usuário (customização);
- Híbrido: combinação das abordagens acima.

Um exemplo clássico de serviço que emprega personalização é o Google, que utiliza um histórico de cada usuário de suas buscas, e quais páginas foram mais visitadas para cada busca. Nesse caso, podemos dizer que a busca é personalizada, pois uma mesma consulta sendo submetida por usuários diferentes trará resultados diferentes.

Outro sistema personalizado é o da Amazon [Amazon 2009], que armazena histórico de compras dos usuários e trabalha com sugestão de itens relacionados às compras efetuadas pelas pessoas. Sendo assim, usuários com preferências diferentes que utilizarem o sistema, receberão sugestões de compras distintas baseadas em seus perfis.

Ao analisar o conceito de personalização e suas características, ficam claras algumas vantagens relacionadas a esse conceito. Uma delas envolve a informação em si. A informação personalizada é uma informação enxuta, mais específica, direcionada ao usuário. Outra vantagem é o ganho de tempo em tarefas diárias, pois com a personalização eliminamos atividades repetitivas, já que agora existe um

reconhecimento dos hábitos da pessoa, fazendo com que o caminho de uma tarefa se torne mais curto.

Porém, a personalização traz consigo algumas desvantagens, como no caso de algum serviço armazenar perfis com as características pessoais dos seus utilizadores. Podem não haver garantias de que esse serviço seja confiável, ou ainda não ser relevante (o usuário utilizou o serviço apenas uma vez sem interesse particular, e seu perfil foi criado).

O maior entrave ligado à personalização está no conceito de anonimato. É uma preferência de muitas pessoas não desejarem ser identificadas, seja por razões simples ou éticas e legais. Um usuário simplesmente prefere que algum comportamento passageiro que ele possuía ao utilizar um sistema não seja armazenado, nem seja criado um perfil para ele; e no final das contas, o usuário tem esse direito.

2.2. Abordagens Existentes para Personalização

Técnicas de personalização foram apresentadas em vários trabalhos com características particulares. Nesta seção, revisamos algumas dessas abordagens relacionadas ao tema de personalização de consultas em diferentes ambientes (áreas) e sob perspectivas distintas.

2.2.1. Bancos de Dados

Koutrika e Ioannidis [Koutrika et al. 2004a; Koutrika et al. 2004b] trataram a questão da personalização para consultas em bancos de dados relacionais de uma maneira muito interessante. Eles desenvolveram um *framework* de personalização para sistemas de bancos de dados baseado nos perfis dos seus usuários. A questão que é levantada é a de que um banco de dados responde sempre as mesmas consultas com os mesmos dados, sem distinguir as preferências de quem realiza a consulta.

Perfis são criados para os usuários do banco de dados e armazenados. Esses perfis podem ser criados explicitamente pelo usuário informando suas preferências ou implicitamente pelo sistema, analisando a interação (histórico) do usuário com o banco. A cada preferência é designado um escore entre 0 e 1 para auxiliar na ordenação dos resultados. A arquitetura proposta neste trabalho é apresentada na Figura 1.

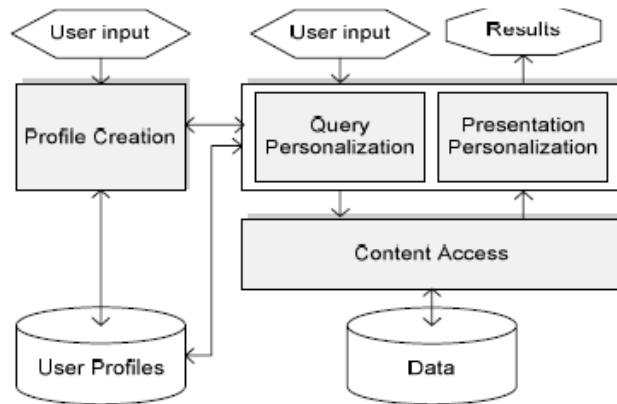


Figura 1 – Arquitetura de um banco de dados personalizado [Koutrika et al. 2004a]

A arquitetura sugere uma variedade de módulos além do componente tradicional que acessa o conteúdo nos dados. O sistema mantém um repositório de perfis dos usuários (*User Profiles*), que pode ser inserido explicitamente pelo usuário ou implicitamente pelo sistema (*Profile Creation*). Cada informação do perfil é integrada numa consulta submetida, na seleção do conteúdo (*Query Personalization*), e na apresentação do resultado (*Presentation Personalization*).

A idéia é inserir nas consultas algumas restrições que caracterizem as preferências do usuário que a está formulando. Por exemplo, podemos considerar dois usuários, Bob e Tim, que desejam saber que filmes serão apresentados num cinema dia 20 de Novembro de 2009. Tipicamente, isso é feito com uma simples consulta SQL, como:

```
SELECT TITULO FROM FILMES WHERE DATA = '20/11/2009'
```

No entanto, de acordo com os perfis, Bob gosta de comédias enquanto Tim prefere filmes de terror. Sendo assim, o sistema pode acrescentar essas preferências nas consultas originais e ordenar os resultados de acordo com o usuário. Bob então teria sua consulta original modificada para:

```
SELECT TITULO FROM FILMES WHERE DATA = '20/11/2009' AND GENERO = 'COMEDIA'
```

A consulta de Tim seria submetida assim:

```
SELECT TITULO FROM FILMES WHERE DATA = '20/11/2009' AND GENERO = 'TERROR'
```

Dependendo de cada perfil, as consultas serão modificadas de uma maneira diferente, sempre de acordo com o esquema da base de dados, já que algumas preferências podem ser expressas através de operações de junção de acordo com o esquema.

A Figura 2 ilustra um exemplo para um esquema de uma base de dados sobre filmes e a Figura 3 mostra um possível perfil de um usuário modelado considerando esse esquema, com as preferências do lado esquerdo e o escore associado do lado direito.

```

THEATRE (tid, name, phone, region),
PLAY (tid, mid, date), MOVIE (mid, title, year),
CAST (mid, aid, award, role), ACTOR (aid, name),
DIRECTED (mid, did), DIRECTOR (did, name),
GENRE (mid, genre)

```

Figura 2 – Esquema textual de uma base de dados [Koutrika et al. 2004a]

```

[ THEATRE.tid=PLAY.tid,      1 ]
[ PLAY.tid=THEATRE.tid,    1 ]
[ PLAY.mid=MOVIE.mid,      1 ]
[ MOVIE.mid=PLAY.mid,      0.8 ]
[ MOVIE.mid=GENRE.mid,     0.9 ]
[ ACTOR.name='A. Hopkins',  0.8 ]
[ GENRE.genre='comedy',     0.9 ]
[ GENRE.genre='thriller',   0.7 ]

```

Figura 3 – Perfil de um usuário [Koutrika et al. 2004a]

O perfil mostrado na Figura 3 ilustra preferências de um usuário em um ator e em gêneros de filmes. Também expressa preferências em junções (*joins*) entre as relações do esquema, para permitir que as consultas expressas em uma relação levem em consideração também suas preferências expressas em outras relações. Por exemplo, um usuário pode considerar o diretor de um filme mais importante do que o elenco [Koutrika et al. 2004a].

Outro trabalho de Koutrika [Koutrika 2006] estende a linha de seus trabalhos de personalização em bases de dados baseadas no perfil dos usuários, para a restrição das consultas, resultando em respostas personalizadas. Porém, nesse trabalho, há a noção de colaboração. É dito que a colaboração é uma opção vantajosa para o usuário, pois além de receber respostas para consultas baseadas em seu perfil, ele também pode visualizar respostas baseadas em perfis de outros usuários com preferências semelhantes, criando a noção de comunidades de usuários, o que auxilia para uma decisão mais fundamentada baseada nas respostas encontradas para cada consulta.

Kostadinov e colegas [Kostadinov et al. 2007] também propõem uma solução baseada em perfis de usuários para consultas em bases de dados, porém cada perfil deve ser independente do esquema do banco (*schema free*), pois é entendido que o perfil deva

ter como objetivo exclusivo a caracterização das preferências e domínios de interesse do usuário. O perfil é um conjunto de atributos cujos valores podem ser definidos explicitamente pelo usuário ou dinamicamente derivados do comportamento do usuário.

O objetivo de Kostadinov é obter resultados relevantes para o usuário que submete uma consulta, o que significa, neste caso, ampliar a consulta original com predicados mais restritivos, de acordo com o perfil. Como consequência, as consultas personalizadas irão obter geralmente um conjunto de resultados mais restritos se comparados com as consultas originais sem personalização.

2.2.2. Recomendação

Recomendação significa sugestão; no contexto de consultas, significa obter resultados a partir de outras pessoas que possuem preferências similares às suas. O trabalho de Stefanidis [Stefanidis et al. 2009] sugere uma recomendação personalizada, no sentido de que há uma expansão nos resultados das consultas dos usuários, baseada em suas próprias preferências.

A recomendação ocorre no formato de tuplas não retornadas na consulta original do usuário, mas que possuem um interesse em potencial. Por exemplo, ao se consultar por filmes dirigidos por Woody Allen, o sistema poderá recomendar também uma biografia de Woody Allen.

Esse sistema desenvolvido por Stefanidis e colegas [Stefanidis et al. 2009], chamado de YMAL (*You May Also Like*), pode computar os resultados das consultas para recomendação de três maneiras: utilizando o estado corrente da consulta e da base de dados; baseada em histórico; e utilizando fontes externas.

A Figura 4 ilustra um esquema para uma base de dados de filmes, utilizada como exemplo no trabalho.

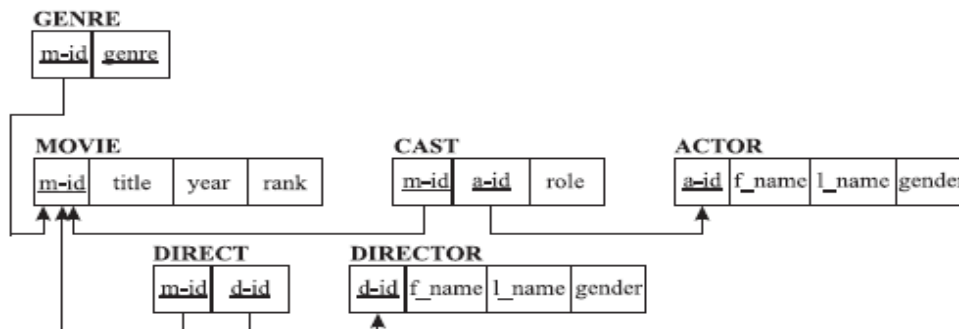


Figura 4 – Esquema gráfico de uma base de dados [Stefanidis et al. 2009]

Na abordagem utilizando o estado corrente da consulta, é assumido que não há outra fonte de informação disponível além da consulta submetida e do resultado

original. Há então uma análise local da consulta e global do esquema da base de dados, visando encontrar algum padrão no resultado que sirva de base para a sugestão de novas tuplas. Por exemplo, ao se consultar filmes onde Morgan Freeman atua, os resultados mostram diversos filmes onde o ator interpreta o papel de um detetive. Sendo assim, o sistema recomenda outras tuplas de novos filmes onde outros atores atuam como detetives.

Se houver informação disponível sobre interações passadas do usuário com o sistema, isso poderá ser utilizado para a recomendação. Neste caso, quando uma consulta é submetida, há uma comparação de similaridade entre a consulta e outras armazenadas num log (histórico). Os resultados das consultas mais similares são incluídos na resposta original do usuário. Há nesse caso uma recomendação clássica, e também personalizada.

A abordagem baseada em fontes externas tem como objetivo ir além do esquema do banco de dados para capturar informações e relacionamentos relevantes na submissão de consultas, que não estão mapeados no esquema da base. Por exemplo, a partir do esquema mostrado na Figura 4, um usuário submete uma consulta procurando por filmes de Sofia Coppola. Utilizando informação externa (da Wikipedia, por exemplo), o sistema pode recomendar alguns filmes de Francis Ford Coppola, pai de Sofia Coppola, um relacionamento não refletido no esquema do banco de dados.

2.2.3. Busca na Web

Os sistemas de busca na web mais conhecidos atualmente – Google e Yahoo – trabalham com buscas personalizadas, baseadas na interação explícita do usuário em suas consultas e na visualização dos resultados. Esses interesses, então, são salvos nos servidores de busca, para que consultas futuras reflitam de uma forma cada vez mais precisa as intenções e preferências dos usuários. Nesses casos, uma mudança de interesse do usuário não é refletida (em curto prazo) nos resultados de suas buscas, ou seja, mudanças mais dinâmicas no comportamento do usuário não são consideradas.

Os trabalhos de [Sendhikumar et al. 2008], [Campi et al. 2009] e [Giannopoulos et al. 2009] abordam o tema da personalização para as buscas na web, com o objetivo de cada vez mais melhorar a experiência e satisfação do usuário em suas buscas.

Sendhikumar [Sendhikumar et al. 2008] chama a atenção para o fato da quantidade de resultados irrelevantes que os usuários recebem em suas buscas na web. O trabalho argumenta que uma solução para esse problema é a personalização, e a

define como sendo o processo de customização do ambiente web de acordo com os interesses do usuário.

Para melhorar as buscas na web, um novo índice de busca personalizado foi desenvolvido, chamado de *User Conceptual Index* (UCI), que provê um relacionamento conceitual entre as palavras-chave da busca e as páginas, que correspondem às necessidades de um usuário. Os autores propuseram uma coleta de dados durante as buscas ocorrendo do lado do cliente, capturando as interações do usuário pelo *browser*.

A abordagem leva em consideração, para os propósitos da personalização, fatores como páginas visitadas, tempo de visita nas páginas ou ações desempenhadas na página, para desenvolvimento de uma ontologia personalizada para melhorar a busca na web. É criada então uma ontologia para a página web e outra para o perfil do usuário. A arquitetura proposta possui cinco camadas, e está ilustrada na Figura 5.

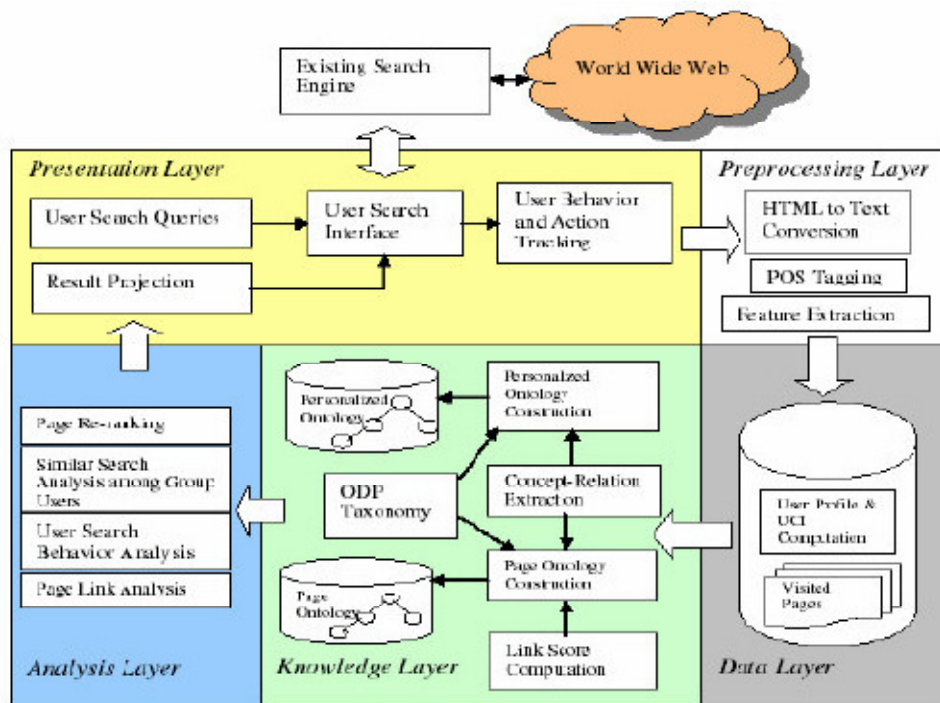


Figura 5 – Arquitetura de uma busca na web personalizada [Sendhikumar et al. 2008]

A camada de apresentação (*Presentation Layer*) é o local onde o usuário realiza suas buscas através de uma interface especial e visualiza os resultados. Essa interface é desenvolvida de tal forma que mantém um rastro de todos os dados do usuário, como as consultas submetidas, páginas visitadas, tempo gasto em cada página e ações realizadas. Essas informações são armazenadas na camada de dados (*Data Layer*) após o pré-processamento.

A camada de pré-processamento (*Preprocessing Layer*) é responsável por algumas atividades como converter a página HTML para documento de texto e extração

de nomes. No final, as palavras-chave da página são coletadas para identificação do tema da página. Os dados coletados no cliente são armazenados na camada de dados. As páginas são indexadas por suas palavras-chave, coletadas na camada de pré-processamento. A camada de dados é a fonte para a extração do conhecimento e para a camada de análise (*Analysis Layer*).

A camada de conhecimento (*Knowledge Layer*) é o local onde os conhecimentos de domínio e personalizado são gerados e representados por ontologias. A ontologia da página é incremental e é construída baseada no conjunto de páginas visitadas pelo usuário. A ontologia personalizada é construída a partir dos dados do perfil do usuário presentes na camada de dados. Pesos são associados às ações que o usuário realiza numa página e essa computação é útil para atualização da ontologia da página.

Essas ontologias são utilizadas para análises posteriores na camada de análise para classificação e recomendação das páginas. Na camada de análise, estatísticas são geradas para um usuário, assim como para grupos de usuários, para identificação de padrões para consultas similares, por exemplo.

O trabalho de Campi [Campi et al. 2009] propõe uma nova técnica para personalização que tenta agrupar os resultados das buscas na Internet em *clusters*. A abordagem se baseia numa caracterização automática do histórico de buscas de um usuário através de uma coleção de domínios semânticos e páginas web escolhidas pelo usuário. Uma pessoa submete uma consulta qualquer num engenho de busca e então seleciona grupos de páginas do resultado total.

A partir das páginas visitadas, o tópico que o usuário estava interessado pode ser inferido. Essa informação é utilizada para a construção de um perfil para o usuário, mantendo num histórico as páginas mais confiáveis para cada busca. Por exemplo, se uma pessoa busca a palavra “Java”, pode querer procurar sobre a Ilha de Java ou sobre a Linguagem de Programação Java. Dependendo das páginas que a pessoa visitou, a consulta então deixa de ser ambígua nesse perfil, e supondo que a pessoa estava buscando informações sobre a Linguagem de Programação, uma das páginas confiáveis armazenadas pode ser www.java.com.

A pesquisa de Giannopoulos [Giannopoulos et al. 2009] sugere um *framework* para melhorar o processo de ordenação de páginas nas buscas personalizadas. O método é baseado na quantidade de cliques dos usuários nas páginas, com o intuito de criar múltiplas funções de *ranking* correspondendo a diferentes áreas de pesquisa. Essas funções são combinadas, cada vez que um usuário realiza uma nova busca, para

produzirem um novo *ranking*, levando em consideração a similaridade da consulta com as áreas mapeadas anteriormente.

2.3. Personalização e Contexto

Uma definição de contexto bastante referenciada é a proposta por Dey e Abowd [Dey et al. 2000], que diz que “Contexto é qualquer informação que caracteriza a situação de uma entidade, onde uma entidade é uma pessoa, lugar ou objeto considerados relevantes para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação”.

No caso de aliar contexto à personalização, podemos caracterizar na grande maioria das vezes a informação contextual como sendo a própria essência da personalização ou, em outras palavras, o usuário, mais especificamente, o perfil do usuário e suas interações.

Alguns trabalhos interessantes procuraram aliar a técnica de personalização de consultas com a adição de informações contextuais para fornecer resultados mais completos e de maior qualidade. Neste tópico, discutiremos frentes de trabalho em duas áreas envolvendo contexto e personalização.

2.3.1. Contexto e Bancos de Dados

Stefanidis [Stefanidis et al. 2007a; Stefanidis et al. 2007b] possui abordagens muito interessantes para a personalização de consultas em bancos de dados envolvendo também informações contextuais. A motivação existente é a quantidade enorme de informação disponível para muitos usuários, necessitando então de uma personalização na busca por essa informação, dependendo de quem necessita.

Além da utilização de personalização, há também a preocupação de atribuir as preferências aos usuários, com uma dependência do contexto em que estes usuários se encontram. Surge então nos trabalhos de Stefanidis a noção de Parâmetro Contextual associado ao perfil do usuário.

O trabalho sugere uma representação relacional em um SGBD pela criação de uma relação de Lugares de Interesse como, por exemplo, museus, parques, etc. Os parâmetros contextuais associados aos usuários dessa base de dados são a sua Localização, a Temperatura corrente, e as Pessoas que acompanham o usuário. A cada preferência é atribuído um escore entre zero e um.

Um Ambiente Contextual (*Context Environment*) é composto por vários parâmetros contextuais, e um Estado Contextual (*Contextual State*) é uma instanciação possível dos parâmetros contextuais. Por exemplo, uma instância para essa base de

dados poderia ser {Atenas, Frio, Amigos} (Localização, Temperatura, Pessoas). O conjunto de todos os estados contextuais possíveis forma o Mundo da aplicação.

Os parâmetros contextuais são modelados em uma hierarquia de níveis, como ilustra a Figura 6.

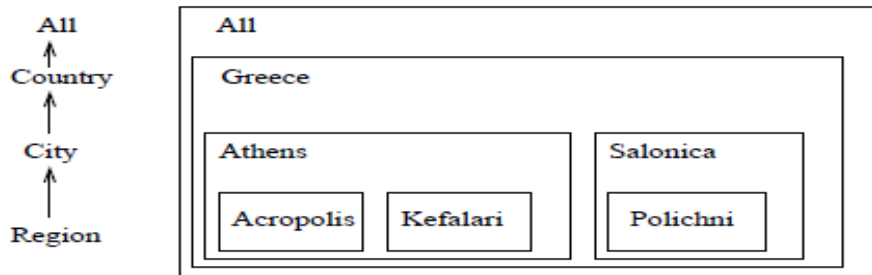


Figura 6 – Hierarquia de parâmetros contextuais [Stefanidis et al. 2007b]

Os relacionamentos entre os níveis contextuais são modelados através de funções: ancestral (subindo na hierarquia) e descendente (descendo na hierarquia). O que caracteriza (descreve) as consultas formuladas são os chamados Descritores. Por exemplo, se uma consulta possui como parâmetros contextuais a localização de Atenas, a temperatura Quente e as pessoas como sendo a Família, o Descritor associado será: (localização = Atenas ^ temperatura = Quente ^ pessoas = Família). Uma mesma consulta vai produzir resultados diferentes devido às preferências de usuários distintos.

A cada combinação de um Descritor, um escore entre zero e um é associado a um Lugar de Interesse. Por exemplo, caso o usuário esteja em Plaka e a temperatura esteja Morna, ele atribui um escore de 0.8 a um zoológico e um escore de 0.5 a um restaurante (possíveis Locais de Interesse). Então, nessas condições de local e temperatura, o usuário estaria mais interessado em ir a um zoológico do que a um restaurante. Não é necessário conferir um valor a todos os parâmetros contextuais para que seja gerado um escore. Essa atribuição de escore a um Descritor é chamada de Preferência Contextual. O perfil do usuário é composto de várias Preferências Contextuais.

Uma *Contextual Query* é uma consulta enriquecida com informação contextual. Nesse caso, o contexto associado é aquele que envolve o usuário no momento da consulta. Na submissão de uma consulta, mais de um estado contextual pode ser recuperado, então a ordenação dá preferência ao mais detalhado na hierarquia. Dependendo do usuário que formula as consultas, os resultados serão distintos baseados em seu perfil, e também mais ricos levando em consideração as informações contextuais.

Outro trabalho de Stefanidis [Stefanidis et al. 2005] considera um sistema de preferências similar que facilita a realização de consultas OLAP sensíveis ao contexto. Novamente, perfis são criados para os usuários, os quais armazenam preferências sobre parâmetros contextuais e não contextuais. O trabalho considera um esquema para uma base de dados composto apenas por duas relações: Restaurante e Usuário, ilustrado na Figura 7.

Restaurant(rid, name, phone, region, cuisine)
User(uid, name, phone, address, e-mail)

Figura 7 – Esquema da base de dados [Stefanidis et al. 2005]

Usuários possuem preferências sobre restaurantes e as expressam com um escore entre zero e um. O grau de interesse que um usuário expressa por um restaurante depende dos valores dos parâmetros contextuais, que nesse caso são a Localização do usuário e a Temperatura corrente. Por exemplo, uma pessoa pode atribuir um escore maior a um restaurante chinês quando o clima está frio.

As preferências são armazenadas em cubos de dados, seguindo o paradigma OLAP, como ilustrado na Figura 8.

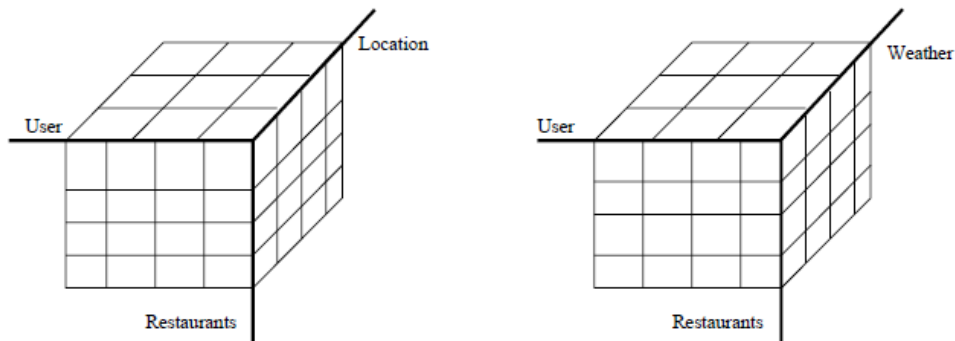


Figura 8 – Cubos de dados para cada parâmetro contextual [Stefanidis et al. 2005]

Na definição do perfil do usuário, há a atribuição de pesos (a critério do usuário) para os parâmetros contextuais. Depois disso, as preferências podem ser calculadas para montagem do perfil. Por exemplo, o usuário José atribuiu peso 0.6 para o parâmetro Localização e 0.4 para Temperatura. Supondo ainda que o usuário atribuísse o seguinte: Restaurante = ChinesX, Localização = Recife, 0.8; Restaurante = ChinesX, Temperatura = Fria, 0.9. Conseqüentemente, a preferência gerada para o perfil de José para esse caso seria:

Preferência(Recife, Fria, ChinaX, José) = 0.84, pois o cálculo teria sido (0.6 * 0.8 + 0.4 * 0.9 = 0.84).

As consultas que podem ser realizadas nessa base são personalizadas e sensíveis ao contexto, e são facilitadas em sua execução por operações OLAP. Exemplos de consulta estão ilustrados na Figura 9.

Query 1 Look for Mary's most preferable restaurants in Athens, independently of the status of weather. In SQL, the query is:

```

- SELECT R.name, FL.score
  FROM Users U, Restaurants R, Fact_Location FL, Location L
  WHERE U.uid = FL.uid AND R.rid = FL.rid AND L.lid = FL.lid AND
  U.name = 'Mary' AND L.region = 'Athens'
  ORDER BY FL.score DESC;

```

Query 2 Look for Mary's most preferable restaurants (in the current context).

```

- SELECT R.name, FL.score
  FROM Users U, Restaurants R, Fact_Location FL, Location L
  WHERE U.name = 'Mary' AND U.uid = FL.uid AND R.rid = FL.rid
  AND L.lid = FL.lid AND current_location = 'Acropolis';
and
- SELECT R.name, FW.score
  FROM Users U, Restaurants R, Fact_Weather FW
  WHERE U.name = 'Mary' AND U.uid = FW.uid AND R.rid = FW.rid
  AND current_weather = 'sunshine';

```

Figura 9 – Exemplos de consultas sensíveis ao contexto [Stefanidis et al. 2005]

Essas consultas são executadas rapidamente com o auxílio dos operadores OLAP *Slice-and-dice*, para manipulação dos cubos de dados, e *Roll-up* e *Drill-down*, para navegação na hierarquia dos dados (por exemplo, ao se encontrar uma cidade, desejar visualizar os bairros da cidade que podem responder à consulta).

2.3.2. Contexto e Recomendação

Sofiane Abbar [Abbar et al. 2009] propõe um sistema de recomendação sensível ao contexto (CARS), baseado também no perfil do usuário. A informação contextual é representada na forma de logs da interação do usuário com o sistema de recomendação.

Esses logs contêm o IP do usuário, a data e hora de acesso ao CARS. Com isso, algumas preferências podem ser geradas. Por exemplo, pode se inferir que o usuário João sempre que acessa o CARS do trabalho (um determinado IP), normalmente durante o dia, procura por revistas científicas. Já quando acessa o sistema de sua casa (outro IP), durante a noite, procura por filmes.

A arquitetura do CARS, apresentada na Figura 10, compreende dois blocos principais:

- Processo de Aquisição do Conhecimento: esse bloco é responsável pela aquisição e gerenciamento do conhecimento que o sistema necessita para operar. Esse conhecimento é dividido em três entidades: perfis dos usuários, descritores de conteúdos e contexto.
- Recomendação Personalizada: esse bloco envolve as operações de recomendação propriamente ditas. Um engenho de recomendação possui como entradas (i) perfis de contextos vindo dos arquivos de *log* e (ii) o contexto

corrente do usuário que requer uma recomendação. O comportamento desse usuário quando interagir com o que foi recomendado é armazenado nos arquivos de *log*, nos quais os processos de aquisição do conhecimento são baseados para atualizar os perfis, contextos e seus mapeamentos. Assim, o ciclo entre aquisição do conhecimento e sua utilização é fechado.

A arquitetura descrita está ilustrada na Figura 10.

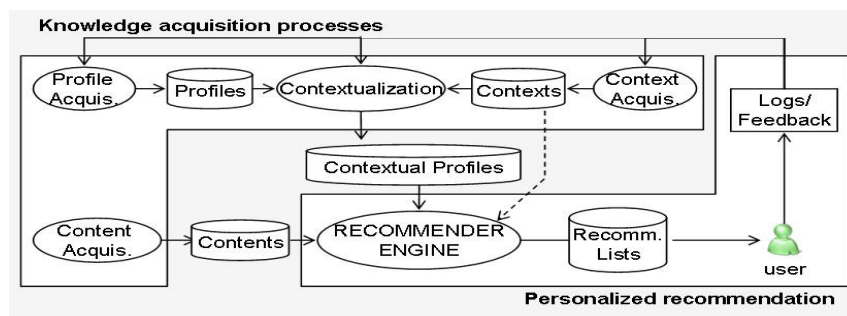


Figura 10 – Arquitetura do CARS [Abbar et al. 2009]

Os exemplos de abordagens de bancos de dados e recomendação para personalização aliados ao contexto compartilham a mesma essência: incrementar o perfil do usuário (personalizar) com informações contextuais (contexto), enriquecendo mais ainda a interação das pessoas nos momentos de busca por determinadas informações de seu interesse, seja na web, num banco de dados ou num sistema de recomendação.

2.4. Análise Comparativa

A Tabela 1 resume as principais características das abordagens de personalização de consultas abordadas neste capítulo. Essas abordagens foram desenvolvidas em diferentes ambientes, dividindo-se em três grupos principais: Banco de Dados (OLAP), Sistemas de Recomendação e Web.

Os trabalhos de Koutrika e Ioannidis [2004a] [2004b] se assemelham bastante ao de Kostadinov [2007], porém, neste último, o perfil dos usuários é independente do esquema do banco, e o usuário possui maior liberdade de escolha de suas características que serão importantes nas consultas.

O método de personalização predominante é a montagem de perfis dos usuários, podendo ser implícitos, explícitos ou híbridos. Alguns trabalhos, como os de Stefanidis [2009] e de Sendhikumar e Geetha [2008] levam em consideração os termos-chave da consulta, assim como as interações passadas dos usuários com os sistemas, na montagem de um histórico.

O efeito da personalização caracteriza-se basicamente na ordenação dos resultados das consultas de acordo com os usuários que as submetem. O trabalho de Stefanidis [2009] se preocupa também em expandir os resultados baseando-se no padrão das respostas capturadas pelas consultas, e Koutrika [Koutrika et al. 2004] acrescenta restrições nas consultas submetidas, produzindo assim um conjunto menor de resultados do que a consulta original.

Por último, é interessante notar alguns trabalhos que, além da personalização das consultas baseada em informações dos usuários, também agregam contexto em seus sistemas. O contexto pode ser incorporado na forma de informações do ambiente no qual o usuário se encontra, como nos trabalhos de Stefanidis [2005] [2007a] [2007b] (o primeiro trabalho [2005] possui uma modelagem diferente, incluindo conceitos e operadores OLAP), como também na forma de logs da interação do usuário com o sistema, como no CARS [Abbar et al. 2009].

Tabela 1 – Análise Comparativa de abordagens de Personalização de Consultas

Abordagem	Ambiente	Método	Montagem do Perfil	Resultados	Uso de Contexto
[Koutrika et al. 2004a]	Banco de Dados	Perfil	Híbrida	<i>Ranking</i> Restrição	Não
[Kostadinov et al. 2007]	Banco de Dados	Perfil	Híbrida	<i>Ranking</i> Restrição	Não
[Stefanidis et al. 2009]	Recomendação	Consulta, Histórico	_____	<i>Ranking</i> , Expansão	Não
[Sendhikumar et al. 2008]	Web	Histórico	_____	<i>Ranking</i>	Não
[Campi et al. 2009]	Web	Perfil	Implícita	<i>Ranking</i>	Não
[Stefanidis et al. 2007b]	Banco de Dados	Perfil	Explícita	<i>Ranking</i>	Sim
[Stefanidis et al. 2005]	OLAP	Perfil	Explícita	<i>Ranking</i>	Sim
[Abbar et al. 2009]	Recomendação	Perfil	Implícita	<i>Ranking</i>	Sim

2.5. Considerações

Neste capítulo, fornecemos uma visão geral sobre o que está sendo pesquisado na área de personalização de consultas em diferentes ambientes. Também apresentamos trabalhos que aliam informação contextual à personalização, como uma forma de dar mais qualidade e precisão aos resultados das consultas dos usuários.

A grande maioria das abordagens de personalização de consultas foca na construção de um perfil para os usuários dos sistemas. Mesmo as abordagens que enriquecem a personalização com informações contextuais, as utilizam como um adendo aos perfis dos usuários. Esses perfis podem ser montados basicamente de duas

maneiras: explicitamente pelo usuário (processo conhecido como customização), ou implicitamente pelo sistema, baseando-se no histórico das interações da pessoa com o sistema.

As abordagens híbridas para montagem do perfil se mostram mais interessantes, pois dá a liberdade ao usuário de fornecer as informações desejadas, ao mesmo tempo em que uma possível mudança de comportamento em suas ações seja capturada pelo sistema para atualização de suas preferências. O uso de contexto é abordado em alguns trabalhos, sendo incorporado aos sistemas como informações do ambiente no qual o usuário está inserido, sendo algo vantajoso para lidar com o dinamismo de algumas aplicações.

Capítulo 3

Consultas em PDMS

Este capítulo visa discutir o processamento de consultas em alguns dos principais sistemas PDMS encontrados na literatura. A utilização de semântica no acesso aos dados e no processamento das consultas foi o critério escolhido para a seleção e estudo dos PDMS que serão apresentados aqui, assemelhando-se dessa forma ao SPEED, um PDMS baseado em semântica e ambiente no qual este trabalho foi desenvolvido, e que será apresentado no Capítulo 4. Em especial, a questão da reformulação de consultas, assim como a personalização, é bastante relevante para a compreensão deste trabalho.

O capítulo está organizado como segue. A Seção 3.1 traz uma breve discussão sobre os sistemas PDMS e descreve os principais objetivos da reformulação de consultas nesses ambientes. A Seção 3.2 descreve resumidamente as abordagens para submissão de consultas propostas por alguns sistemas PDMS baseados em semântica. Finalmente, a Seção 3.3 conclui o capítulo com algumas considerações.

3.1. Reformulação de Consultas em PDMS

Os sistemas P2P se popularizaram entre os usuários devido à funcionalidade de troca de arquivos (mais especificamente de músicas no início de seu aparecimento) [Napster 2007; Kazaa 2007]. Atualmente já são uma realidade, e suas características principais, como flexibilidade e descentralização, são bem conhecidas.

Embora difundidos, os sistemas P2P existentes funcionam oferecendo suporte a apenas algumas funções limitadas e em casos simples de compartilhamento de dados. Uma consulta nesses sistemas é realizada por meio do uso de palavras-chave. Neste sentido, inicialmente desenvolvidos para suportar o compartilhamento de dados não estruturados, os sistemas P2P vêm evoluindo para sistemas de gerenciamento de dados estruturados e/ou semi-estruturados por meio dos PDMS – *Peer Data Management Systems*.

Como um típico sistema P2P, um PDMS herda todas as suas características como, por exemplo, o fato de cada ponto poder entrar e sair da rede a qualquer

momento e a autonomia que cada um possui. Do ponto de vista de gerenciamento de dados, um PDMS deve lidar com questões como [Sung et al. 2005]:

- Localização dos Dados: pontos devem ser capazes de referenciar e localizar dados armazenados em outros pontos;
- Processamento de Consultas: dada uma consulta, o sistema deve ser capaz de descobrir os pontos que podem contribuir com dados relevantes à sua execução e processá-la eficientemente;
- Integração de Dados: mesmo quando fontes de dados compartilhadas no sistema seguem diferentes esquemas, o sistema deve ainda ser capaz de acessar os dados, integrá-los e retornar os resultados;
- Consistência dos Dados: a consistência deve ser mantida em caso de replicação e uso de *cache*.

Um dos serviços mais importantes em um PDMS é o processamento de consultas nos pontos (utilizamos também o termo *peer* como sinônimo). Quando uma consulta é submetida em um ponto, o sistema que possui a rede consegue obter dados relevantes não só do ponto que recebeu a consulta, mas de qualquer outro conectado a este por algum caminho semântico (correspondências semânticas definidas entre os pontos para os termos da consulta). Ao contrário de uma hierarquia estrita de integração de dados, um PDMS permite qualquer rede de relacionamentos entre os *peers* [Tatarinov, Halevy 2004].

Uma grande vantagem dos sistemas PDMS é que qualquer ponto pode submeter uma consulta na rede utilizando apenas seu próprio esquema de dados, sem precisar conhecer esquemas de outros pontos. Com isso, com base nos caminhos semânticos existentes de que o ponto da submissão da consulta participa, a consulta original pode ser reformulada (adequada) aos outros pontos do caminho semântico, de acordo com seus respectivos esquemas de dados.

Por exemplo, considerando a Figura 11, se uma consulta *Q* é submetida no *peer* Stanford, o PDMS encaminha a consulta a outros *peers* (Paris e *DB-Projects*) por meio dos caminhos semânticos estabelecidos, e então integra todos os resultados de todos os *peers* para retorná-los de uma vez. A consulta *Q* é reformulada para os vizinhos semânticos de Stanford e outros resultados são retornados baseados em outros esquemas de dados. Isso favorece a obtenção de uma resposta mais completa, dado um certo domínio de aplicação.

Na Figura 11, uma consulta submetida no ponto Roma pode obter dados do ponto Paris, seguindo o caminho por *DB-Projects* e Stanford.

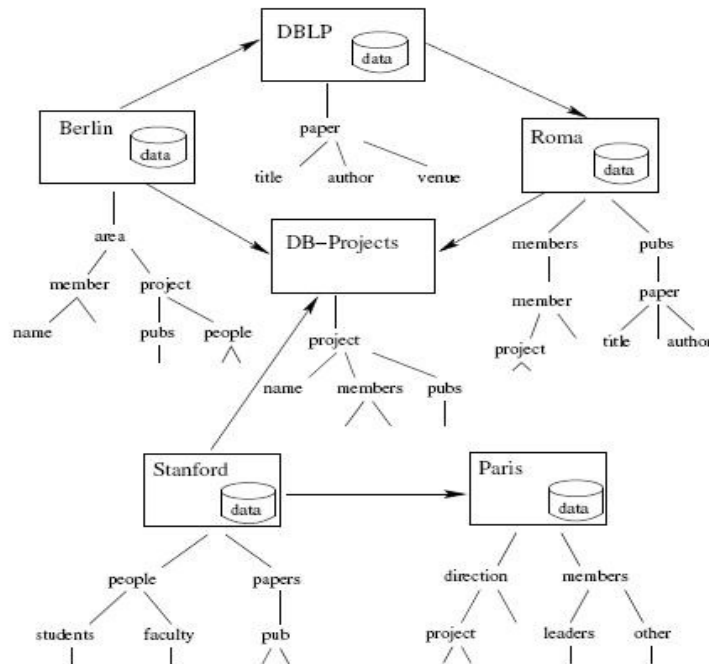


Figura 11 – Exemplo de PDMS e caminhos semânticos entre os *peers* [Tatarinov, Halevy 2004]

Dependendo do caminho semântico seguido pela consulta, novas respostas podem ser obtidas. A escolha do caminho semântico a ser seguido é muito importante e pode resultar em alguns problemas caso seja seguido um caminho incorreto. Alguns exemplos desses problemas citados em [Tatarinov, Halevy 2004] são:

- Surgimento de respostas redundantes. Cada reformulação desnecessária degrada a performance do sistema;
- Caminho encontrando um final muito cedo, passando por poucos pontos (ou, em outras palavras, podado precocemente);
- Reformulações ineficientes, ou seja, consultas em pontos que poderiam ser mais bem aproveitadas (otimizadas, ou enriquecidas) antes de serem executadas.

Esses problemas constituem um impedimento para o processamento de consultas mais eficientes e completas em PDMS.

Os pontos de uma rede em um PDMS não possuem a informação completa de um domínio, por isso há uma grande importância em se escolher um bom caminho semântico quando uma consulta é processada, a fim de se extrair mais informações.

A importância da escolha correta do caminho semântico está no fato de se conseguir uma maior exploração de *peers* que contenham mais informações relevantes

àquela consulta em seu esquema de dados. Com essa exploração eficiente da rede P2P, um resultado mais rico (com mais dados) é alcançado.

3.2. Abordagens Existentes para o Processamento de Consultas em PDMS

Esta seção ilustra de uma forma resumida abordagens para o processamento de consultas em alguns PDMS existentes. Todos os PDMS que serão apresentados utilizam semântica para representação dos dados e no tratamento das consultas.

3.2.1. HELIOS

O HELIOS [Castano et al. 2003; Castano et al. 2005] é definido como um sistema de compartilhamento de conhecimento em sistemas distribuídos abertos P2P. Este sistema compartilha conhecimento a partir do uso de ontologias (cada *peer* possui sua ontologia) e utiliza uma operação de *matching* semântico para identificar a afinidade semântica entre conceitos armazenados nas ontologias de diferentes *peers* [Castano et al. 2005].

A Figura 12 ilustra os componentes principais do *toolkit* do HELIOS com os quais um *peer* é equipado para submeter consultas e processar consultas de outros *peers* [Castano et al. 2003]:

- *Query processing manager*: realiza a composição das consultas de acordo com os modelos de representação de dados;
- *Matching manager*: realiza a comparação (*matching*) de um conceito de uma consulta com a ontologia de um *peer*, para encontrar possíveis resultados;
- *Ontology manager*: responsável pela definição e manutenção da ontologia do *peer*.

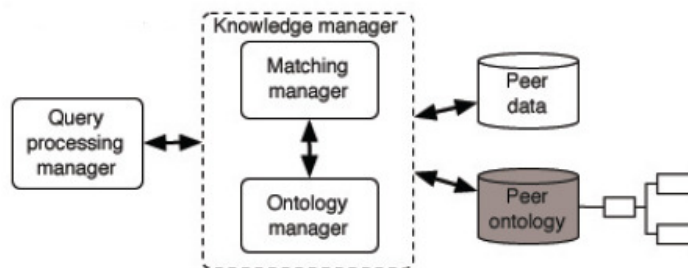


Figura 12 – HELIOS *toolkit* [Castano et al. 2003]

São oferecidos três tipos de modelos de consultas no HELIOS. O primeiro é chamado *search model* e tem como objetivo encontrar conteúdos relacionados a um ou mais conceitos; neste caso, os resultados retornados são dados. O segundo modelo é

chamado *probe model* e visa estender o conhecimento de um ponto. Nesse último caso, os resultados retornados são metadados. Há ainda um modelo híbrido chamado *probe/search model* que retorna tanto dados quanto metadados sobre um conceito.

Quando uma consulta *Q* é submetida, ela é enviada ao componente de roteamento semântico que verifica os pontos aptos a responderem à consulta. Após isso, a consulta é passada para o gerenciador de processamento de consultas (*query processing manager*), que reescreve *Q* para os outros pontos selecionados responderem à consulta. Por fim, as respostas são enviadas de volta para o ponto de origem, que apresenta os resultados.

3.2.2. SUNRISE

O SUNRISE [Mandreoli et al. 2008] é um PDMS desenvolvido pela Universidade de Modena e Reggio Emilia para tratar com fontes de dados XML. O SUNRISE possui uma infraestrutura completa que estende cada *peer* com funcionalidades para capturar a aproximação semântica originada da heterogeneidade de esquemas e explorar essa aproximação para a organização de uma rede semântica e para propósitos de roteamento das consultas.

Na formação de sua rede, são geradas as chamadas SONS (*Semantic Overlay Networks*), que agrupam pontos com conteúdos semanticamente similares em subredes lógicas. Dentro de uma SON, cada ponto tem seu próprio grupo de pontos vizinhos pertencentes àquela SON. Uma SON também possui um ponto responsável por armazenar os termos mais relevantes da SON, o chamado APS (*Access Point Structure*).

Quando um ponto se junta ao sistema, os termos do esquema do ponto são comparados com os termos encontrados no APS que representa cada SON. Dependendo do quão semelhante são os conceitos, o ponto entrante pode fazer parte de uma ou mais SONS. A Figura 13 ilustra um exemplo de uma rede formada por duas SONS referentes a dados relacionados a cinema.

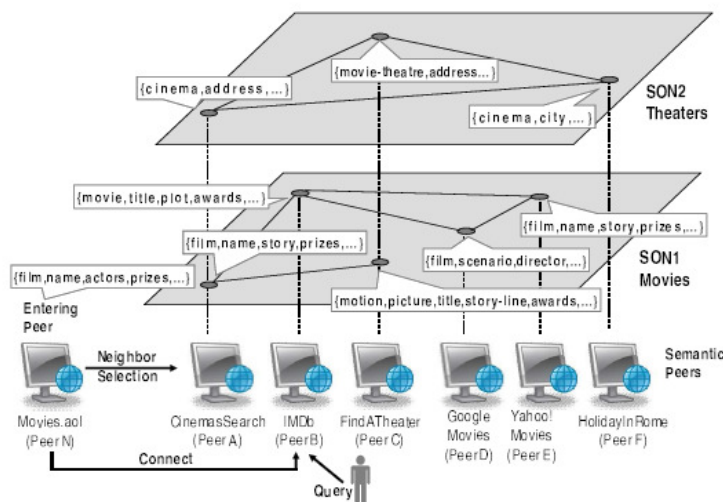


Figura 13 – Exemplo de organização de uma rede no SUNRISE [Mandreoli et al. 2008]

O processamento de consultas é feito considerando apenas os pontos vizinhos do ponto onde foi submetida a consulta. Dentre os pontos vizinhos, são encontrados aqueles aptos a responder a consulta. A consulta é então reformulada e enviada a cada um dos pontos aptos. Os resultados são recebidos no ponto origem, integrados e apresentados ao usuário.

O principal objetivo da SON é aumentar a eficiência do processamento de consultas limitando o número de contatos apenas para *peers* relevantes [Mandreoli et al. 2008].

3.2.3. Humboldt Discoverer

Um PDMS, de uma maneira geral, consiste num conjunto de *peers*, cada um atuando como um componente de integração de informação. Consultas submetidas num *peer* são respondidas tanto com dados locais como com dados de outros *peers* da rede em que este se encontra. Esses outros *peers* são alcançados por meio de caminhos e mapeamentos definidos entre os *peers* na rede. Esses mapeamentos são formulados manualmente. O Humboldt Discoverer (ou apenas Humboldt) [Herschel, Heese 2005] supera esta restrição provendo um índice semântico para a arquitetura do PDMS. Utilizando esse índice, um *peer* localiza fontes de informação úteis que não são alcançáveis por meio de um caminho de mapeamentos convencional.

O Humboldt pode ser entendido como um PDMS estendido com a adição de algumas dimensões, permitindo uma pesquisa de fontes de informação mais eficiente e uma melhoria no roteamento de consultas. A Figura 14 ilustra essa arquitetura.

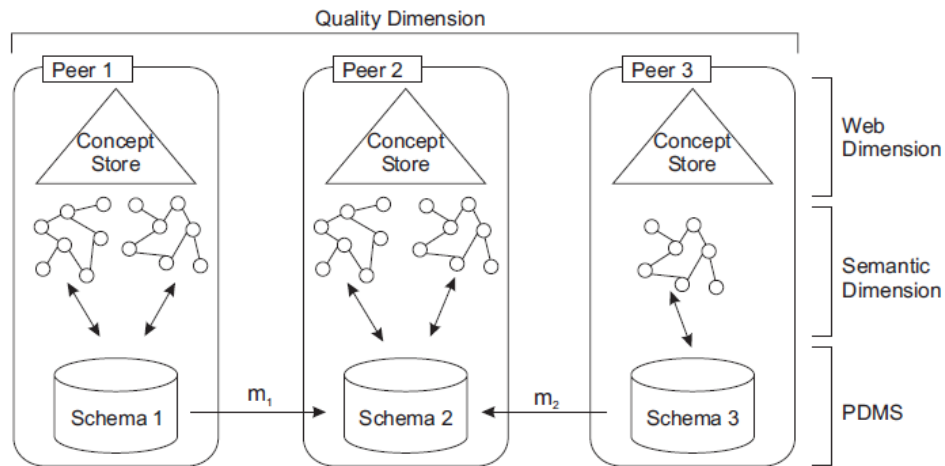


Figura 14 – Arquitetura estendida do Humboldt Discoverer [Herschel, Heese 2005]

As dimensões que estendem um PDMS tradicional e que fazem parte do Humboldt são [Herschel, Heese 2005]:

- Dimensão semântica: provê um caminho alternativo para os mapeamentos semânticos entre os esquemas dos *peers*. Essa camada consiste de ontologias e mapeamentos entre elas. Para se ter vantagem da dimensão semântica para o processamento de consultas, um *peer* pode ter mapeamentos do seu esquema local para uma ou mais ontologias;
- Dimensão web: um índice P2P é introduzido, que auxilia a resolver o problema de localização de fontes relevantes para uma consulta que não foi mapeada anteriormente. Cada *peer* mantém uma base de conceitos que indexa os esquemas dos seus *peers* vizinhos;
- Dimensão de qualidade: influencia o processamento da consulta em todas as suas dimensões. No nível do PDMS e na dimensão semântica, a qualidade das respostas da consulta depende do mapeamento entre os *peers*. À medida que uma consulta é encaminhada para outros *peers*, a qualidade dos mapeamentos é registrada. Na dimensão web uma medida é definida para localizar e classificar os pontos que são relevantes para a consulta.

O processamento de consultas ocorre em duas fases. Na primeira fase, a consulta é processada na camada PDMS, utilizando o mapeamento relacional entre os *peers*. Se o resultado dessa primeira fase for insuficiente, a segunda fase é iniciada e a consulta é processada ao longo dos mapeamentos das ontologias.

Nessa segunda fase, a consulta é traduzida para uma representação gráfica e a base de conceitos é consultada para obtenção de fontes de informação que possam responder a consulta.

3.2.4. SomeRDFS

O SomeRDFS [Adjiman et al. 2007] é um PDMS que utiliza um modelo de dados baseado em RDF desenvolvido sobre a infraestrutura do SomeWhere, um PDMS onde não existem *super-peers* nem um servidor centralizado [Adjiman et al. 2006]. Os esquemas são representados através de ontologias, os mapeamentos em RDFS e os dados representados em RDF. Utilizando o núcleo do RDFS, as ontologias e os mapeamentos podem ser representados, permitindo a representação de classes, propriedades, tipo do domínio e limites dos valores das propriedades.

Os mapeamentos são declarações RDFS envolvendo vocabulários de diferentes *peers*, estabelecendo assim correspondências semânticas entre eles [Adjiman et al. 2007]. Podem ser de dois tipos:

- Uma declaração de inclusão entre classes ou propriedades de dois *peers* distintos;
- Uma declaração de uma propriedade de um *peer* com uma classe de outro *peer*.

Para tornar a semântica clara, as ontologias, dados e mapeamentos são representados em Lógica de Primeira Ordem (FOL, de *First Order Logic*), como um conjunto de fórmulas. A Figura 15 ilustra os construtores utilizados na semântica do SomeRDFS e suas traduções para a Lógica de Primeira Ordem.

Constructor	FOL translation
Class inclusion	$\forall X (C_1(X) \Rightarrow C_2(X))$
Property inclusion	$\forall X \forall Y (P_1(X, Y) \Rightarrow P_2(X, Y))$
Domain typing of a property	$\forall X \forall Y (P(X, Y) \Rightarrow C(X))$
Range typing of a property	$\forall X \forall Y (P(X, Y) \Rightarrow C(Y))$

Figura 15 – Construtores utilizados no SomeRDFS [Adjiman et al. 2007]

As consultas no SomeRDFS são conjunções que podem envolver os vocabulários de vários *peers*. A reformulação é reduzida para um problema de encontrar consequências sobre teorias da Lógica Proposicional resolvidas pelo DECA (*Decentralized Consequence Finding Algorithm*) [Adjiman et al. 2006], o algoritmo utilizado no SomeWhere, onde os modelos de dados dos *peers* são representados como um conjunto de cláusulas proposicionais.

A estratégia do algoritmo é reescrever os trechos da consulta do usuário independentemente com o DECA e então combinar os resultados dessas reescritas para a geração de conjunções da consulta com relação ao SomeRDFS [Adjiman et al. 2007].

3.3. Considerações

Neste capítulo mostramos de forma resumida as características de alguns sistemas PDMS baseados em semântica e como funciona o processamento de consultas nesses ambientes.

Alguns sistemas PDMS requerem a definição de uma semântica mais formal entre suas fontes de dados. Esse fato faz com que haja um aumento do poder de entendimento a respeito de uma rede P2P por parte dos usuários, permitindo assim que esses usuários elaborem consultas mais adequadas às suas necessidades.

No entanto, mesmo com a adição de uma semântica mais formal, os resultados das consultas podem ser massivos e alguns desses resultados serem irrelevantes para o usuário.

Uma solução para tornar resultados de consultas mais relevantes é a utilização de personalização (discutida no Capítulo 2), que já é empregada em bancos de dados, buscas na web, sistemas de recomendação, mas não é difundida em sistemas PDMS. No Capítulo 4, nós propomos uma nova abordagem para a reformulação e execução de consultas em um PDMS utilizando personalização.

Capítulo 4

Personalização de Consultas no SPEED

Um PDMS é um sistema de gerenciamento de dados em ambientes *peer-to-peer* (P2P) e, como tal, possui como principal serviço o processamento de consultas. Um PDMS deve ser capaz de descobrir quais pontos numa rede P2P podem contribuir com dados relevantes às consultas e processá-las eficientemente. Um grande desafio é fazer com que as consultas submetidas em um PDMS retornem um conjunto de resultados que, da melhor forma possível, expressem as intenções do usuário, considerando a dinamicidade do sistema.

O sistema SPEED utiliza informações semânticas tais como conhecimento de uma ontologia de domínio e informação contextual (do usuário, da consulta, de correspondências semânticas e do ambiente) como forma de enriquecer a etapa de reformulação de consultas. A proposta descrita neste trabalho é estender o módulo de reformulação de consultas do sistema SPEED (chamado *SemRef*) [Souza 2009; Neves 2008; Souza et al. 2009a; Souza et al. 2009b; Souza et al. 2009c] com o uso de personalização a partir do perfil dos usuários e de suas localizações no momento da submissão das consultas. Sendo assim, este capítulo tem como objetivo principal abordar o problema da personalização de consultas no SPEED, bem como a especificação de nossa proposta para solucioná-lo.

Este capítulo está organizado como segue. A Seção 4.1 apresenta em detalhes o SPEED. A Seção 4.2 descreve as características e funcionalidades do módulo de reformulação de consultas do SPEED, o *SemRef* e a Seção 4.3 provê um exemplo mostrando o *SemRef* na prática, sem a extensão desenvolvida neste trabalho. A Seção 4.4 aborda o problema do uso de personalização no *SemRef* e explica a utilidade da priorização das correspondências semânticas no nosso trabalho. A Seção 4.5 especifica como utilizamos a localização e o perfil dos usuários para personalizar as consultas no *SemRef*. Finalmente, a Seção 4.6 conclui o capítulo com algumas considerações.

4.1. O Sistema SPEED

Esta seção detalha o sistema SPEED (*Semantic Peer-to-Peer Data Management System*) [Pires 2009], um PDMS baseado em semântica.

4.1.1. Arquitetura

No SPEED, os pontos são organizados de acordo com o conteúdo que compartilham. Existem três tipos de pontos e relacionamentos entre eles no sistema: os pontos semânticos, os pontos de integração e os pontos de dados. A Figura 16 ilustra a arquitetura do SPEED. Os esquemas de dados dos pontos no SPEED são representados através de ontologias [Noy et al. 2001], estruturadas na linguagem OWL (linguagem para definição e instanciação de ontologias na web) [OWL 2004].

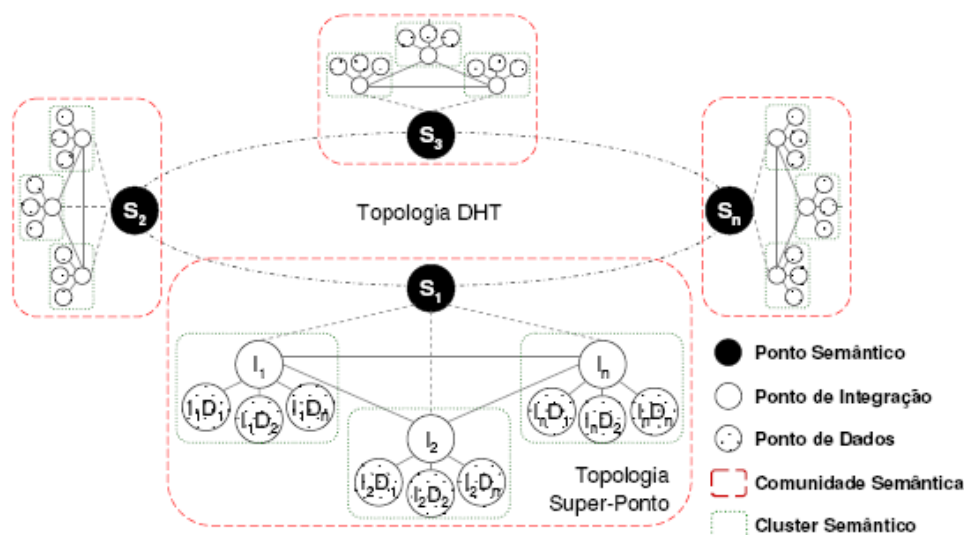


Figura 16 – Arquitetura do SPEED [Pires 2009]

O sistema foi projetado com duas topologias de redes distintas: a DHT (*Distributed Hash Table*), que organiza os *peers* semânticos, e a *super-peer* (ou super-ponto) para o gerenciamento de pontos de integração e pontos de dados. A rede DHT é utilizada para auxiliar um *peer* a encontrar outros *peers* com características comuns através de uma função *hash* e, posteriormente, formarem comunidades semânticas. Dentro das comunidades semânticas, os *peers* são organizados de acordo com a topologia *super-peer* [Pires 2009].

Na topologia *super-peer* [Yang et al. 2003] alguns pontos podem assumir papéis distintos em uma rede, com a eleição de pontos de maior capacidade computacional para coordenar um subconjunto de outros pontos da rede. Esses pontos eleitos são os *super-peers*. A busca nesse tipo de topologia é muito rápida em comparação a outras arquiteturas, já que o sistema possui seu espaço de busca particionado num conjunto menor de *peers* coordenados por *super-peers*, os quais possuem a informação de seus *peers* indexada.

Existem três tipos de pontos no SPEED: (i) ponto de integração; (ii) ponto semântico; e (iii) ponto de dados. Um ponto de integração ligado a vários pontos de

dados forma um *cluster* semântico. Cada *cluster* está ligado a um conjunto de conceitos ontológicos (ontologia do *cluster*) que representa os dados compartilhados pelos pontos daquele *cluster*. Este conjunto de conceitos é um subconjunto dos conceitos da ontologia do ponto semântico. A ontologia do *cluster* é armazenada no ponto de integração [Pires 2009].

Conjuntos de *clusters* semânticos que compartilhem informações a respeito de um mesmo domínio, junto com pontos semânticos, formam as comunidades semânticas. Cada *cluster* semântico possui um tipo diferenciado de ponto com diferentes responsabilidades e mais capacidade computacional. Esses pontos são chamados pontos de integração e possuem um conhecimento detalhado sobre os pontos de dados pertencentes ao seu *cluster*. Os pontos de integração se comunicam com outros pontos de integração e também com os pontos de dados de seu *cluster* semântico. Sua maior capacidade computacional é refletida em suas atribuições como, por exemplo, controlar e processar as consultas sobre os pontos de dados [Neves 2008]. Exemplos de *peers* de integração na Figura 16 são I_1 , I_2 e I_n .

Pontos de integração se comunicam também com o outro tipo de ponto do SPEED, o ponto semântico, responsável por armazenar e disponibilizar uma ontologia representativa de uma comunidade – uma ontologia de domínio, contendo elementos que caracterizam o conhecimento de um domínio. Na Figura 16, S_1 é um exemplo de ponto semântico. Uma responsabilidade desse tipo de ponto é gerenciar os metadados dos pontos de integração que se conectam a ele.

O outro tipo de ponto do SPEED é o ponto de dados. Um ponto de dados representa uma fonte compartilhando dados com outros pontos de dados no sistema, através de mapeamentos semânticos. Os pontos de dados são os locais onde são realizadas as consultas no sistema. Na Figura 16, I_1D_1 e I_1D_2 são exemplos de pontos de dados. Os pontos de dados são agrupados nos *clusters* semânticos de acordo com seus interesses semânticos, ou seja, tema de interesse abordado e estrutura de sua ontologia local. O tema de interesse é uma descrição do domínio semântico do ponto, enquanto a ontologia local descreve o esquema exportado [Pires 2009].

4.1.2. Geração de Correspondências entre Esquemas

As correspondências semânticas são os relacionamentos identificados entre os conceitos e propriedades dos esquemas dos pontos (esquemas locais).

No nosso trabalho, utilizamos ontologias para representação dos esquemas dos pontos participantes da rede do SPEED. Esses pontos são agrupados em um mesmo

domínio de conhecimento (como Educação, Saúde, etc.) e uma ontologia descrevendo o domínio está disponível para ser utilizada como *background knowledge*. As correspondências semânticas entre os pontos são estabelecidas para prover um entendimento comum de suas fontes de dados [Souza 2009]. Consideramos que as correspondências são determinadas entre pares de pontos que foram relacionados semanticamente de acordo com um processo de *clustering*.

Dizemos que $\{C\} = \{C_{ij}\}_{i < j}$ se refere ao conjunto de correspondências semânticas entre uma ontologia fonte (O_i) e uma ontologia destino (O_j). Como a normalização terminológica é um requisito necessário no qual as representações iniciais das duas ontologias são transformadas para um formato comum adequado para uma computação de similaridade, consideramos que O_i e O_j foram convertidas para um formato de representação uniforme, ou seja, os nomes dos elementos de O_i e O_j foram ajustados para se tornarem compatíveis com os nomes dos elementos encontrados na Ontologia de Domínio [Souza 2009].

Ontologias de domínio (OD) contêm conceitos e propriedades de um domínio de conhecimento em particular. Consideramos as ontologias de domínio como referências confiáveis que são disponibilizadas na web. Particularmente, nós as utilizamos para fazer a identificação de diferenças ou semelhanças conceituais entre duas ontologias sobrepostas.

Nesse sentido, duas ontologias passam por uma operação de comparação ontológica (*matching* semântico) [Pereira 2008] e têm seus conceitos e propriedades comparados com os conceitos ou propriedades equivalentes na OD e a correspondência semântica entre as duas ontologias iniciais é inferida baseada no relacionamento semântico existente entre os elementos da OD (chamaremos a partir de agora as ontologias que passam por essa operação de ontologias comparáveis). A ontologia de domínio possui um papel fundamental na definição das correspondências semânticas utilizadas nas consultas do SPEED [Souza 2009].

A Figura 17 mostra uma visão global da nossa abordagem para a especificação da semântica das correspondências entre duas ontologias O_1 e O_2 . Nessa figura, $O_1:x$ possui uma equivalência (denotada pelo símbolo \equiv) com $OD:k$ e $O_2:y$ possui uma equivalência com $OD:z$. Como “k” é subconceito de “z” na OD, inferimos que o mesmo relacionamento ocorre entre “x” e “y”. Então, nos concluímos que $O_1:x$ é subconceito de $O_2:y$.

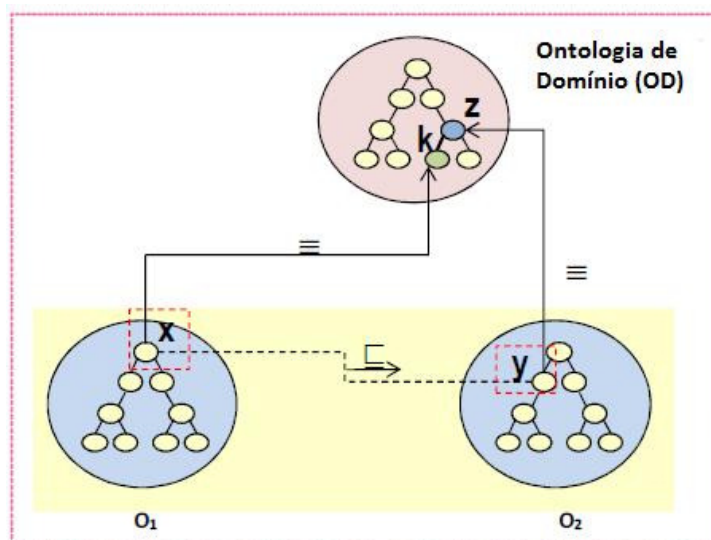


Figura 17 – Utilizando uma OD para especificar as Correspondências Semânticas [Souza 2009]

Nossa abordagem considera sete tipos de correspondências semânticas: [Souza 2009]

- Equivalência, denotado por $O_1x \equiv O_2y$;
- SubConceito e SuperConceito (Generalização e Especialização), denotados por $O_1x \subseteq O_2y$ (x subconceito de y) e $O_1x \supseteq O_2y$ (x superconceito de y);
- “Parte de” e “Todo” (Agregação e Composição), denotados por $O_1x \triangleright O_2y$ (x é parte de y) e $O_1x \triangleleft O_2y$ (x é composto de y);
- Aproximação (*isCloseTo*), denotado por $O_1x \approx O_2y$;
- Disjunção, denotado por $O_1x \perp O_2y$

onde x e y são elementos (conceitos ou propriedades) pertencentes às ontologias comparáveis O_1 e O_2 .

A fim de especificar essas correspondências, consideramos quatro aspectos: (i) o conhecimento semântico encontrado na OD; (ii) se os conceitos das ontologias dos *peers* compartilham superconceitos na OD; (iii) se esses superconceitos são diferentes do nó raiz; e (iv) a profundidade entre dois conceitos medida em nós.

A seguir, explicamos cada um dos tipos de correspondências semânticas existentes, provendo alguns exemplos utilizando um cenário ilustrativo que diz respeito a dispositivos eletrônicos, incluindo computadores e seus componentes. A Figura 18 ilustra a OD e a Figura 19 mostra as ontologias comparáveis do nosso cenário [Souza 2009].

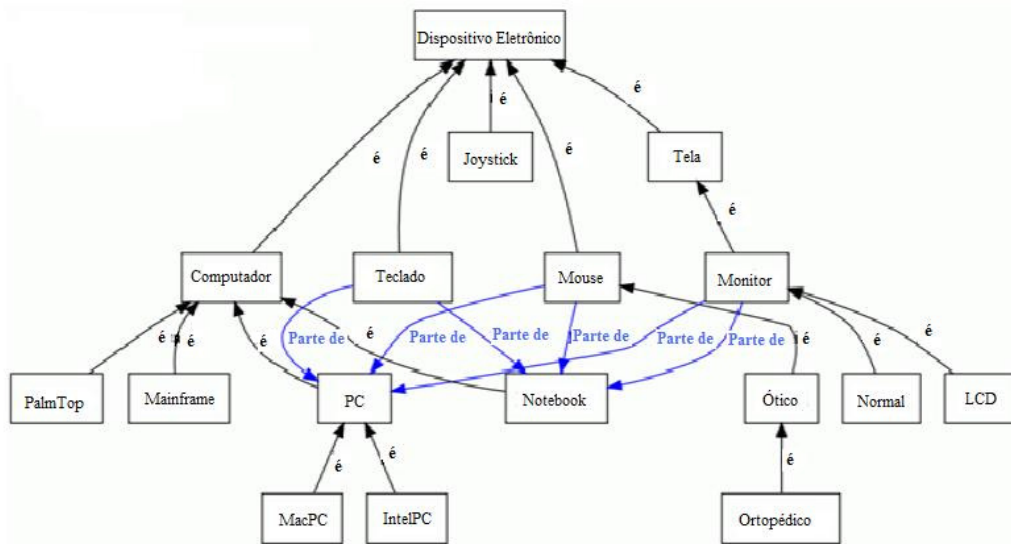


Figura 18 – Ontologia de Domínio [Souza 2009]

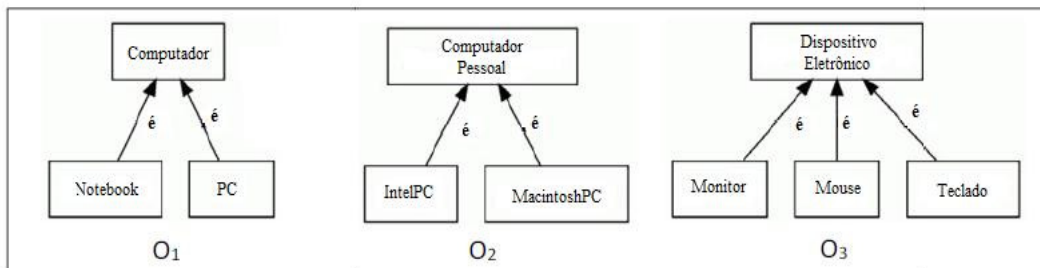


Figura 19 – Ontologias Comparáveis [Souza 2009]

Na nossa abordagem, se $O_1:x$ aponta para (se relaciona com) um conceito “k” da OD e $O_2:y$ se relaciona com o mesmo conceito “k”, podemos inferir que ambos os conceitos ou propriedades descrevem o mesmo conceito ou propriedade do mundo real (são equivalentes). Referindo-se ao nosso cenário, se temos $O_1:PC \equiv OD:PC$ e $O_2:ComputadorPessoal \equiv OD:PC$, então $O_1:PC \equiv O_2:ComputadorPessoal$.

A correspondência de subconceito diz que o conceito ou propriedade “x” de O_1 é menos genérico que o seu conceito (ou propriedade) relacionado “y” de O_2 . Por outro lado, a correspondência de superconceito expressa o fato de que “x” de O_1 é mais genérico que “y” de O_2 . Supondo que $O_2:ComputadorPessoal \equiv OD:PC$, $O_3:DispositivoEletrônico \equiv OD:DispositivoEletrônico$ e $OD:PC \sqsubseteq OD:DispositivoEletrônico$, então $O_2:ComputadorPessoal \sqsubseteq O_3:DispositivoEletrônico$.

A correspondência “parte de” diz que o conceito “x” de O_1 é parte/componente de do conceito relacionado “y” de O_2 , e a correspondência de “todo” expressa o fato de que “x” em O_1 é um agregado (uma composição) de “y” que está em O_2 . Um exemplo

da correspondência “parte de” pode ser considerarmos que $O_3:\text{Teclado} \equiv OD:\text{Teclado}$, $O_1:\text{PC} \equiv OD:\text{PC}$ e $OD:\text{Teclado} \triangleright OD:\text{PC}$, então $O_3:\text{Teclado} \triangleright O_1:\text{PC}$.

Dois conceitos ou propriedades de ontologias comparáveis são ditos aproximados (*close*) se eles são percebidos como pertencendo juntos a um contexto relevante comum, ou seja, dois conceitos estão sob o mesmo conceito do mundo real (o mesmo ancestral na OD). Considerando a OD, dois conceitos aproximados normalmente se sobrepõem (*overlap*) em algum grau. Se isso não acontece, eles devem ser declarados explicitamente como disjuntos. Nesse sentido, exemplos de conceitos aproximados semanticamente organizados sob o mesmo ancestral (Computador) são *notebook*, *palmtop*, e *mainframe*.

O último tipo de correspondência é a disjunção, a qual foi definida no sentido de identificar uma forte dissimilaridade entre os elementos das ontologias comparáveis. Essa correspondência diz que dois conceitos $O_1:x$ e $O_2:y$ são disjuntos se os seus elementos correspondentes (equivalentes) na OD “k” e “z”, respectivamente, são disjuntos, ou seja, “k” e “z” foram definidos na OD como sendo disjuntos. Isso significa que “x” de O_1 não sobrepõe y de O_2 .

4.1.3. Conexão de um Novo Ponto

Quando um ponto de dados se conecta ao SPEED, este primeiramente identifica a qual comunidade semântica deverá pertencer, através de palavras-chave dos domínios existentes, utilizando a estrutura DHT do sistema. Após essa etapa, a busca pelo *cluster* semântico a que ele vai pertencer é realizada através da operação de *matching* semântico [Pereira 2008] (a mesma operação que gera as correspondências semânticas) entre a ontologia que representa o esquema do ponto que deseja se conectar e a ontologia dos *clusters* (ver Figura 20).

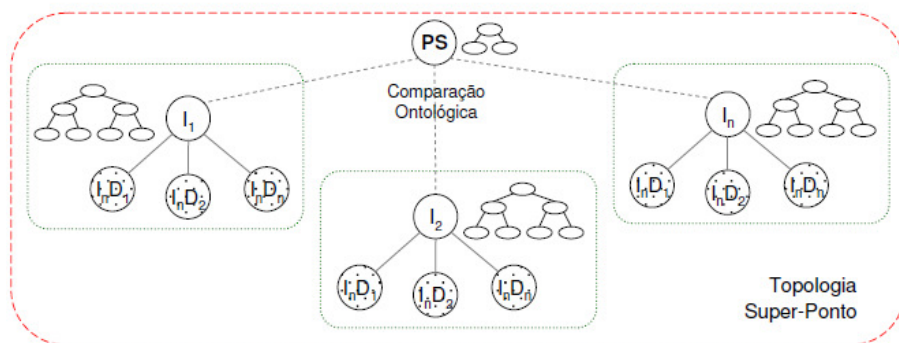


Figura 20 – Descoberta de *clusters* semânticos através de comparação ontológica [Pires 2009]

Esse *matching* também irá originar uma medida de similaridade semântica entre as ontologias, que ajudará a identificar o *cluster* da qual o ponto entrante do sistema fará

parte. Um limite de *cluster* pode ser definido para descartar *clusters* que apresentem baixo grau de similaridade semântica. Em seguida, o ponto solicitante ao SPEED se conecta ao *cluster* cujo valor de similaridade foi maior que o limite estabelecido. Os conceitos do esquema de dados (ontologia local) deste ponto são adicionados à ontologia do *cluster*. Neste caso, pode ser aplicado um processo de *merge* entre as ontologias.

Por outro lado, caso o valor de similaridade calculado seja inferior ao limite estabelecido, significa dizer que nenhum dos *clusters* daquela comunidade trata do interesse do ponto solicitante. Sendo assim, este ponto irá formar um novo *cluster*, cuja ontologia será formada inicialmente pelo seu esquema local e evoluirá com a conexão de outros novos pontos [Pires 2009].

4.1.4. Processamento de Consultas

As correspondências semânticas apresentadas na Seção 4.1.2 são particularmente importantes para o processamento de consultas no SPEED. Quando um usuário submete uma consulta num ponto de dados, a consulta, além de ser processada naquele ponto, é encaminhada até o ponto de integração daquele *cluster*. O ponto de integração identifica os pontos de dados capazes de responder àquela consulta, ao mesmo tempo em que a consulta é propagada para outros pontos de integração da mesma comunidade [Neves 2008].

Por meio das correspondências semânticas geradas entre os pontos da comunidade é que a consulta pode ser reformulada para os termos dos esquemas de outros pontos, sem modificação da intenção semântica original [Souza 2009; Neves 2008]. No final, os resultados recebidos dos pontos que executaram a consulta são integrados no ponto de integração que propagou a consulta, e o resultado final é enviado ao ponto que a submeteu.

O usuário possui a liberdade de escolher quais correspondências semânticas serão levadas em consideração no momento da reformulação da consulta. Essas escolhas, assim como a submissão da consulta em si, são feitas no módulo de consultas do sistema, que será explicado na próxima seção.

4.2. Reformulação de Consultas no SPEED

Esta seção descreve o módulo de reformulação de consultas do sistema SPEED, chamado *SemRef*. Serão descritas as características deste módulo e como ocorre a reformulação de consultas no SPEED. A seção também especifica os formatos de consulta suportados pelo *SemRef*, além da arquitetura e casos de uso do módulo.

4.2.1. Características

As consultas no SPEED podem ser submetidas nos *peers* de integração ou nos *peers* de dados do sistema. Nessa versão do nosso trabalho implementamos a reformulação e execução de consultas envolvendo apenas *peers* de integração. Esses *peers* de integração recebem a consulta, analisam, reformulam de acordo com o esquema de outros pontos e no final integram os resultados e os apresentam ao usuário.

Uma questão importante nesse processo é a reformulação da consulta entre pontos através de caminhos de correspondências semânticas disponíveis. Considerando a semântica implícita existente nas correspondências entre os pontos, é possível, no momento da reformulação, permitir o enriquecimento da consulta, provendo os usuários com resultados mais significativos [Souza 2009].

O *SemRef* permite a elaboração de consultas em um ponto de submissão. O módulo ainda identifica a semântica da consulta e, a partir de correspondências semânticas entre os *peers*, reformula essa consulta (com ou sem enriquecimento) e a envia a outros pontos de integração que possam respondê-la satisfatoriamente [Neves 2008]. No escopo deste trabalho, a reformulação da consulta ocorre apenas entre dois *peers*, não prevendo ainda roteamento sobre diversos pontos da comunidade semântica.

4.2.2. Formato das Consultas

As consultas no *SemRef* podem ser formuladas utilizando Lógica Descritiva ALC (*Attribute Language with Complement*) [Baader et al. 2003] ou a linguagem de consultas SPARQL [SPARQL 2008].

A Lógica Descritiva (abreviada para DL, do inglês *Description Logics*) consegue explicitar e detalhar diversos domínios de aplicações graças ao seu poder de descrição de entidades através de operadores bem definidos e quantificadores, e também graças às regras de formação dessas descrições de algum objeto ou situação em particular. Os construtores da ALC são: (i) $\neg C$ (negação); (ii) $C \sqcap D$ (conjunção); (iii) $C \sqcup D$ (disjunção); (iv) $\forall R.C$ (restrição universal); e (v) $\exists R.C$ (restrição existencial limitada), onde C e D são conceitos e R é uma propriedade. Consideramos as ontologias dos *peers* no SPEED como uma tripla $O = \{C, R, I\}$, onde C é um conjunto de conceitos, R é um conjunto de propriedades e I é um conjunto de indivíduos (instâncias) [Souza 2009].

Uma consulta formulada utilizando a Lógica Descritiva é uma expressão $Q = C$, onde C é um conceito. Esse conceito pode ser atômico ou complexo (inclui propriedades, quantificadores, conjunções ou disjunções). Assim, uma consulta Q é uma

fórmula consistindo de uma disjunção de conjunções de conceitos C_1, \dots, C_N , onde $N \geq 1$, definida da seguinte forma [Souza 2009]:

- **Consulta em Lógica Descritiva:** Uma consulta Q expressa sob a ontologia de um ponto P_1 possui a seguinte forma: $Q = Q_1 \sqcup Q_2 \sqcup \dots \sqcup Q_M$, onde $Q_i = C_1 \sqcap C_2 \sqcap \dots \sqcap C_N$, e cada C_j é um conceito atômico, um conceito atômico negado ou um conceito atômico quantificado ($C_j, \neg C_j, \forall R.C$ ou $\exists R.C$);

Um exemplo de consulta formulada em Lógica Descritiva no *SemRef* é: $Q_1 = [\text{Teacher} \sqcap \text{Researcher}] \sqcup [\text{Student} \sqcap \text{Researcher}]$, que procura por indivíduos que são professores e pesquisadores ou estudantes que também são pesquisadores.

SPARQL é uma linguagem de consulta para documentos RDF, mas que também faz buscas em estruturas de arquivos OWL, ou seja, busca em ontologias em geral. É padronizada pela DAWG (RDF Data Access Working Group) da W3C. Vale ressaltar que as consultas submetidas em ALC no *SemRef* são traduzidas para o SPARQL para execução sobre as ontologias que representam os esquemas dos *peers*. Detalhes sobre essas traduções serão vistos quando tratarmos das questões de implementação do trabalho, discutidas no Capítulo 5.

As consultas em SPARQL consistem de um padrão de triplas (*triple patterns*): conjunções, disjunções e padrões opcionais de complementação. Essa idéia de tripla se baseia nas estruturas de grafo do RDF. SPARQL foi desenvolvida a partir de linguagens de consulta em RDF anteriores, como RDQL (RDF Data Query Language) [RDQL 2004] e SeRQL (*Sesame RDF Query Language*) [Broekstra et al. 2003], e sua implementação é compatível com uma série de plataformas.

A Figura 21 apresenta um exemplo de consulta em SPARQL, que deseja recuperar de uma ontologia todos os identificadores (IDs) de professores.

```
PREFIX rdf: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX prf: <http://www.owl-ontologies.com/Ontology1220896776.owl#>
SELECT
  ?ID
FROM
  <http://www.owl-ontologies.com/Ontology1220896776.owl>
WHERE
{
  ?x rdf:type prf:Teacher.
  ?x prf:ID ?ID
}
```

Figura 21 – Exemplo de Consulta SPARQL

Detalhando mais essa consulta a respeito da sintaxe da linguagem, temos os seguintes itens:

- **Prefixos:** No início há a declaração de prefixos com a palavra reservada *prefix*, que associa uma sigla com um URI (*Uniform Resource Identifier*) específico. Uma consulta pode incluir qualquer quantidade de prefixos desejada;
- **Select:** Palavra reservada que define os itens que serão retornados pela consulta (as variáveis em SPARQL são precedidas de “?” ou “\$”);
- **From:** Identifica a ontologia na qual a consulta irá executar;
- **Where:** Indica as condições para o retorno do resultado. Neste caso, como explicado, podemos observar o padrão de triplas do SPARQL. A tripla pode ser entendida como uma construção de sujeito, predicado e objeto;
- **?x:** A variável “x” é ligada a conceitos do tipo “*Teacher*” na ontologia;
- **Operador “ponto”:** O operador “.” indica uma concatenação de restrições. A segunda restrição indica que, dado que “x” contém um professor, seu ID deve ser atribuído à variável ID (?ID, que é a variável de retorno da consulta).

Sobre os resultados das consultas (tanto submetidas em Lógica Descritiva quanto em SPARQL), estes são definidos a seguir.

- **Resultado da Consulta:** Um resultado R de uma consulta Q_1 sob os termos da ontologia O_1 de um ponto P_1 que é reformulada sob os termos da ontologia O_2 de um ponto P_2 é um conjunto de instâncias $\{ID_1, ID_2, \dots, ID_N\}$ dos conceitos retornados das ontologias O_1 e O_2 , que são apresentados na forma de termos, que dependem do domínio de conhecimento e da reformulação produzida.

4.2.3. Reformulando Consultas

A utilização de semântica é uma das principais características do *SemRef*. Essa semântica auxilia para um processo de reformulação expandido e para a apresentação de resultados mais completos para as consultas. Na fase de reformulação da consulta, nem sempre um conceito presente no ponto de submissão terá conceitos equivalentes em pontos de destino onde a consulta será executada depois de reformulada. Sendo assim, se não é possível produzir um resultado exato para uma dada consulta ou se o usuário define que ele aceita que não sejam retornadas apenas respostas exatas, pode ser melhor produzir um resultado com respostas aproximadas do que não produzir resultado algum [Souza 2009].

Para este fim, consideramos que uma consulta formulada de acordo com os termos da ontologia de um *peer* fonte pode ser reformulada de forma exata ou enriquecida em outra consulta utilizando os termos de uma ontologia de um *peer* destino em função do conjunto de correspondências semânticas existente entre eles.

Utilizando esse conjunto de correspondências, as reformulações de uma consulta que podem ser produzidas são definidas da seguinte maneira:

- **Reformulação Exata:** Uma reformulação Q' de uma consulta Q é exata se cada conceito (ou propriedade) C' de Q' está relacionado com um conceito (ou propriedade) C de Q por meio de uma correspondência CO , onde $CO \in \{\equiv\}$ (equivalência);
- **Reformulação Enriquecida:** Uma reformulação Q' de uma consulta Q é enriquecida se cada conceito (ou propriedade) C' de Q' está relacionado com um conceito (ou propriedade) C de Q por meio de uma correspondência CO , onde $CO \in \{\supseteq, \supset, \cong, \triangleright, \triangleleft, \perp\}$.

O usuário possui ainda a liberdade de escolher o modo de reformulação para suas consultas: (i) modo restrito, onde apenas os resultados considerando as correspondências de equivalência são retornados (caso este resultado seja vazio, um resultado enriquecido será produzido), ou em outras palavras, apenas a reformulação exata será produzida; ou (ii) modo expandido, onde o resultado contemplará as correspondências semânticas escolhidas pelo usuário na opção de configuração de um conjunto de variáveis de enriquecimento, além da própria correspondência de equivalência, ou em outras palavras, ambas as reformulações (exata e enriquecida) serão produzidas.

Para considerar as outras correspondências no modo expandido (exceto equivalência), o usuário encontra no *SemRef* uma série de variáveis, chamadas de variáveis de enriquecimento, que podem ser selecionadas para as consultas.

Cada variável está ligada a uma ou mais correspondências semânticas para serem incorporadas na reformulação da consulta. São quatro variáveis de enriquecimento: (i) Aproximação, referente à correspondência de aproximação; (ii) Especialização, ligada à correspondência de superconceito; (iii) Generalização, referente à correspondência de subconceito; e (iv) Composição, ligada às correspondências de agregação. No caso da correspondência de disjunção, não existe uma variável, mas um operador fornecido na interface de consulta do módulo (detalhes no Capítulo 5).

O usuário submete uma consulta num *peer* P_1 e ela é reformulada apenas para um *peer* destino (P_2). Sendo assim, temos apenas duas ontologias (esquemas) de pontos envolvidas no processamento da consulta (O_1 e O_2) e um conjunto de correspondências semânticas entre elas. A Figura 22 ilustra esse cenário.

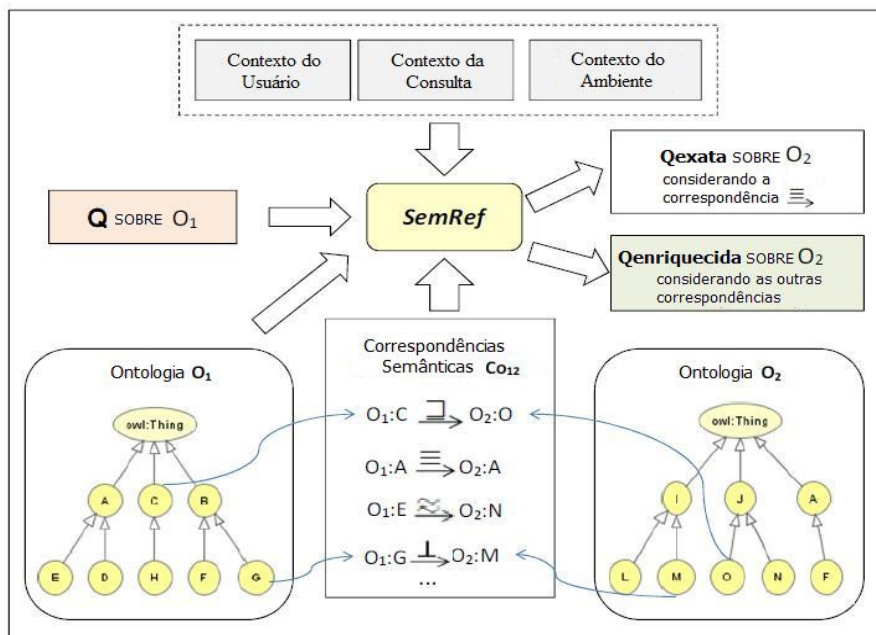


Figura 22 – Cenário das consultas no *SemRef* [Souza 2009]

O *SemRef* utiliza o conceito de contexto, como uma forma de melhorar todo o processo de reformulação da consulta. O contexto do usuário é identificado a partir do conjunto de variáveis de enriquecimento que permitem ao usuário expressar suas preferências a respeito do processo de reformulação (em outras palavras, o usuário decide que correspondências semânticas devem ser levadas em consideração na reformulação da consulta).

O contexto da consulta é adquirido a partir de sua semântica e do seu modo de reformulação (se no modo restrito ou no modo expandido). Finalmente, o contexto do ambiente é capturado utilizando alguns parâmetros definidos pelo usuário e leva em consideração a disponibilidade dos *peers* [Souza 2009].

4.2.4. Arquitetura do *SemRef*

Para prover o processamento da consulta envolvendo semântica (em especial a fase de reformulação), um componente chamado de *matcher* semântico foi desenvolvido [Pereira 2008]. O *matcher* é responsável por identificar e gerar as correspondências semânticas entre os *peers*. Não faz parte do escopo deste trabalho detalhar esse componente. Outro componente criado foi a ontologia de contexto (CODI, explicada a seguir), para armazenamento dos elementos contextuais envolvidos nas consultas (Figura 22) [Souza 2009].

Esses dois artefatos são importantes para o funcionamento do *SemRef*. Ilustramos a arquitetura do módulo de consultas do sistema SPEED na Figura 23. A

comunicação entre o usuário e o módulo (consultas submetidas e resultados) é feita através de uma interface gráfica.

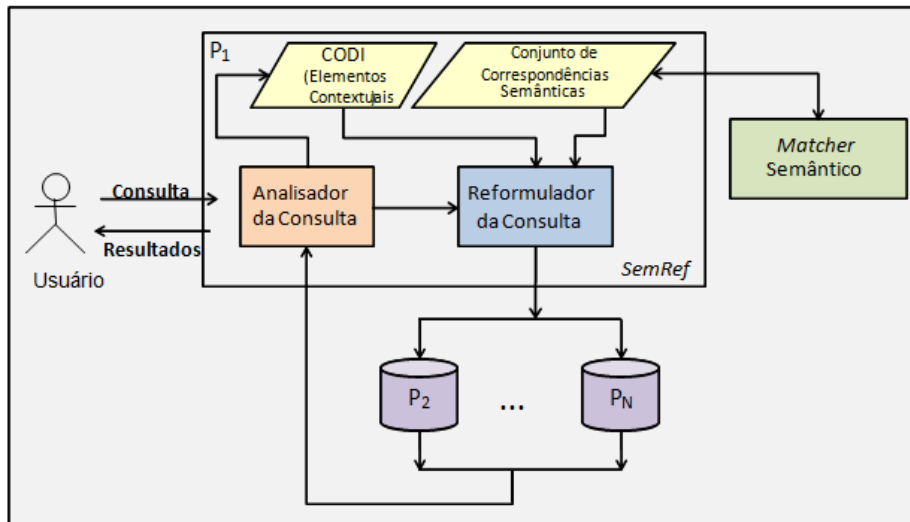


Figura 23 – Arquitetura do *SemRef*

Os principais componentes dessa arquitetura, seguidos de uma breve descrição, são [Souza 2009]:

- CODI: ontologia de contexto onde são armazenados os elementos contextuais envolvidos com as consultas (variáveis de enriquecimento e modo de reformulação da consulta), entidades da consulta e operadores, entre outros. A CODI foi codificada em OWL;
- *Matcher Semântico*: recebe como entrada duas ontologias (esquemas de dois *peers*) e gera como saída as correspondências semânticas existentes entre elas;
- Conjunto de Correspondências Semânticas (definido na Seção 4.1);
- Analisador da Consulta: responsável por analisar a semântica da consulta, identificando as entidades e os operadores. Também é responsável por receber os resultados, integrá-los e apresentá-los ao usuário;
- Reformulador da Consulta: componente chave do *SemRef*, responsável por verificar os elementos contextuais relacionados à consulta e as correspondências semânticas existentes entre os esquemas de dois *peers* para o processo de reformulação da consulta original, produzindo uma ou mais consultas reformuladas, de acordo com a Tabela 2.

Tabela 2 – Mapa de Consultas Reformuladas produzidas

Variáveis de Reformulação	Modo da Consulta		Consultas Reformuladas Produzidas
	Restrito	Expandido	
Aproximação Composição Generalização Especialização			
Pelo menos uma é Seleccionada	Verdadeiro	Falso	Exata Enriquecidas
Nenhuma é Seleccionada	Verdadeiro	Falso	Exata
Pelo menos uma é Seleccionada	Falso	Verdadeiro	Exata Enriquecidas (se a Exata retornar vazia)
Nenhuma é Seleccionada	Falso	Verdadeiro	Exata

4.2.5. Casos de Uso

Nesta seção, vamos apresentar as principais funcionalidades do *SemRef*, descritas a partir do diagrama de casos de uso ilustrado na Figura 24.

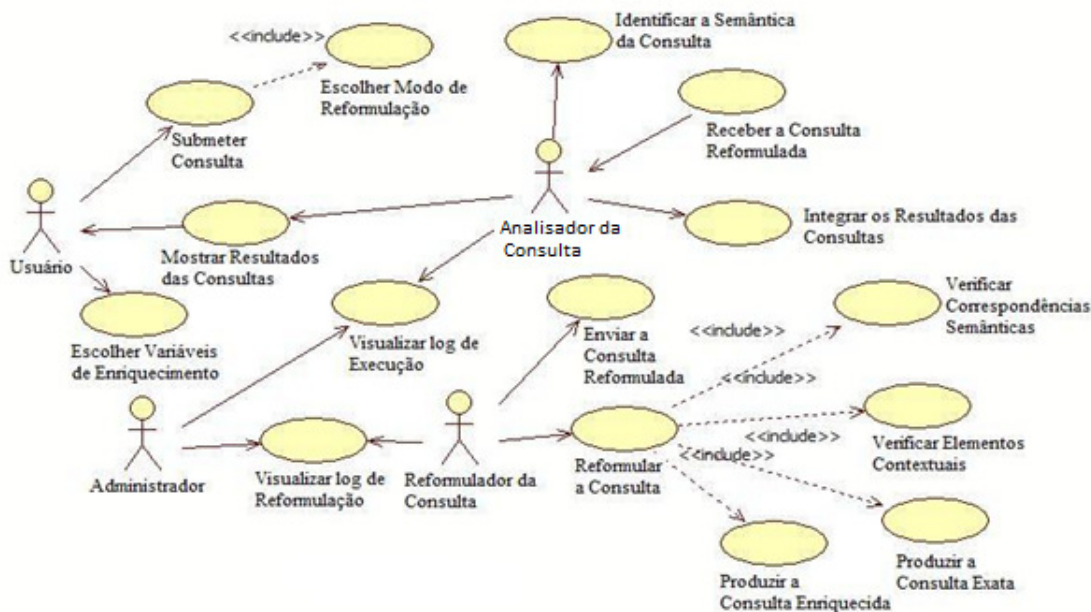


Figura 24 – Diagrama de Casos de Uso do *SemRef*

Como já explicado anteriormente, o usuário poderá submeter a consulta no *SemRef* a partir de uma interface gráfica, que permitirá também a seleção das variáveis de enriquecimento e do modo de reformulação da consulta.

Uma vez submetida a consulta, o Analisador da Consulta a recebe e identifica sua semântica, ou seja, os conceitos e operadores envolvidos (seja em ALC ou em

SPARQL). Depois, o Reformulador da Consulta verifica os elementos contextuais presentes, além das correspondências semânticas existentes e das variáveis escolhidas pelo usuário. Dependendo do modo de reformulação escolhido, uma ou duas reformulações podem ser produzidas (exata e/ou enriquecida).

Após a produção dos resultados, estes são enviados de volta ao Analisador da Consulta, que os recebe e os integra num resultado único para apresentação ao usuário. Cabe ressaltar duas funcionalidades não citadas anteriormente, mas presentes na Figura 24 – visualização de Log de Reformulação e visualização de Log de Execução. O *SemRef* gera esses dois logs para as sessões dos usuários no módulo. Seus formatos e utilidade serão discutidos no Capítulo 5, onde as questões de implementação do *SemRef* serão vistas em detalhes.

4.3. Um Exemplo Prático

No exemplo de uso do *SemRef*, vamos considerar dois pontos P_1 e P_2 , os quais pertencem a um domínio de Educação, com informações a respeito de estudantes, professores, e trabalhos de pesquisa de diferentes instituições. Cada ponto P_1 e P_2 possui uma ontologia descrevendo seu esquema – O_1 e O_2 , respectivamente.

Após a operação de *matching* entre essas ontologias, os conjuntos de correspondências semânticas entre os esquemas são gerados (um conjunto relacionando os conceitos de O_1 para O_2 , e o outro conjunto relacionando os conceitos de O_2 para O_1). Um trecho de um desses conjuntos de correspondências (de O_1 para O_2) está ilustrado na Figura 25.

```
<rdf:Description rdf:about="http://swrc.ontoware.org/ontology/portal#Student">
  <j.0:isSuperConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#UndergraduateStudent</j.0:isSuperConceptOf>
  <j.0:isSuperConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#GraduateStudent</j.0:isSuperConceptOf>
  <j.0:isSubConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#Person</j.0:isSubConceptOf>
  <j.0:isEquivalentTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#Student</j.0:isEquivalentTo>
</rdf:Description>
```

Figura 25 – Trecho de correspondências semânticas entre O_1 e O_2

Nessa figura, podemos verificar como o conceito *Student* de O_1 está relacionado a alguns conceitos presentes na ontologia O_2 . Por exemplo, *Student* de O_1 é subconceito (relação de generalização) de *Person* de O_2 . Essas informações são úteis para a fase de reformulação da consulta no *SemRef*. Vamos considerar a consulta deste exemplo prático como sendo formulada na Lógica Descritiva (definição e formatos na Seção 4.2.2), sendo composta neste caso por um conceito atômico.

Então, seguindo com o nosso exemplo, vamos considerar a submissão da consulta $Q_1 = Student$ a partir do esquema de P_1 . Vamos supor que o usuário escolhe o

modo de reformulação da consulta como expandido, indicando assim que o resultado da consulta contemplará as correspondências semânticas escolhidas pelo usuário na opção de configuração de variáveis de enriquecimento. Nesse exemplo, o usuário escolheu apenas a variável de especialização.

Ao executarmos a consulta Q_1 , o *SemRef* irá buscar as instâncias de *Student* em O_1 , além das instâncias de *Student* em O_2 (reformulação exata, decorrente do relacionamento de equivalência entre os conceitos). Além disso, uma reformulação enriquecida será gerada para recuperar as instâncias de *GraduateStudent* e *UndergraduateStudent* (devido aos relacionamentos de superclasse – vide Figura 25).

Temos então nesse exemplo quatro consultas para serem executadas: $Q_1 = Student$, $Q_2 = Student$, $Q_3 = GraduateStudent$ e $Q_4 = UndergraduateStudent$. A consulta Q_1 será executada em P_1 e as outras consultas em P_2 .

O *SemRef*, então, traduz essas consultas geradas para a linguagem SPARQL e as executa em P_1 e P_2 após as reformulações terem sido geradas. Por fim, integra os resultados vindos dos pontos e os apresenta ao usuário como um único. O propósito desse exemplo é ilustrar de uma maneira simples o funcionamento de uma consulta no *SemRef*, dadas algumas configurações escolhidas pelo usuário, e a utilização das correspondências semânticas.

Qualquer usuário que submetesse a consulta Q_1 e utilizasse as mesmas configurações do exemplo iria observar os mesmos resultados na mesma ordem. Nossas contribuições objetivam modificar esse cenário e serão especificadas em detalhes nas próximas seções.

4.4. Personalização das Consultas no *SemRef*

Os ambientes *peer-to-peer* são altamente dinâmicos e flexíveis, onde os pontos podem entrar e sair da rede a qualquer momento, podendo crescer a um número elevado de participantes, tornando difícil a busca por uma informação específica nessa rede de *peers*.

Com tanta informação, o universo de opções aumenta e o tempo e esforço requeridos para buscar um dado de interesse também crescem. Esses problemas são bem difíceis de serem tratados quando um usuário navega numa rede (P2P, web) à procura de uma informação específica, onde quantidades massivas de dados podem ser retornadas [Kostadinov et al. 2007]. A importância de se gerenciar esses dados para que os usuários possam especificar suas preferências e, assim, encontrarem o que procuram

mais rapidamente, é enorme e é um grande potencial para as tecnologias de bancos de dados [Agrawal et al. 2000].

No módulo de consultas do SPEED tínhamos uma ferramenta de consulta onde diferentes usuários, ao submeterem uma mesma consulta com os mesmos *peers* disponíveis, encontrariam os mesmos resultados. Nossas contribuições têm como objetivo a personalização das consultas no *SemRef*, ordenando os seus resultados de acordo com as preferências e com o perfil dos usuários.

O problema da personalização no *SemRef* parte principalmente do modo de trabalhar com duas “linhas de reformulação de consulta” a serem geradas e, posteriormente, executadas: uma consulta exata e as consultas enriquecidas. Para alcançar a personalização no módulo de consultas do SPEED, dois planos de trabalho foram propostos para fazer o elo das preferências e características do usuário com as consultas e seus resultados: (i) personalização dos resultados derivados da reformulação enriquecida, com a noção da escolha de uma priorização das variáveis de enriquecimento; e (ii) personalização dos resultados derivados do *peer* fonte e da reformulação exata, com a utilização de um perfil e da localização do usuário no momento de submissão de suas consultas.

A Seção 4.4.2 trata do plano de trabalho (i) e a Seção 4.4.3 especifica o plano de trabalho (ii). Essas contribuições estendem o módulo *SemRef* com novos componentes e funcionalidades. Damos o nome de *PSemRef* (*Personalized SemRef*) a esse módulo estendido, o qual será especificado na próxima seção.

4.4.1. Arquitetura do *PSemRef*

A Figura 26 mostra a arquitetura do *PSemRef*, com destaque para o módulo de personalização de consultas, que faz a ligação do perfil e da localização do usuário com as consultas submetidas, ordenando os resultados de acordo com essas características. O Analisador da Consulta utiliza as informações do módulo de personalização para ordenar os resultados das consultas submetidas no sistema.

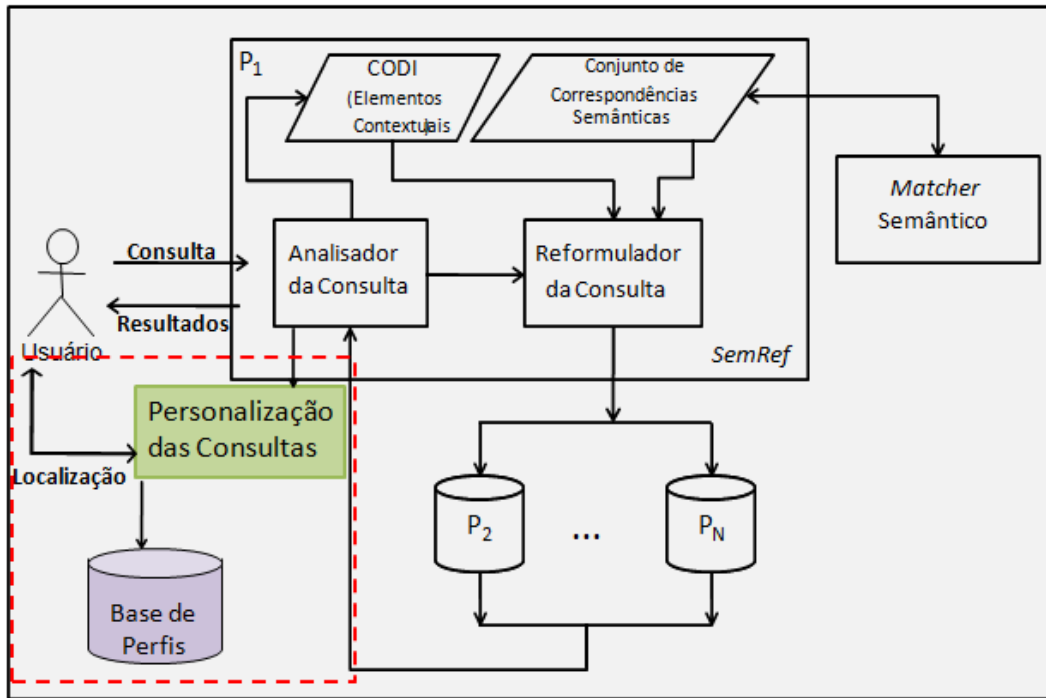


Figura 26 – Arquitetura do *PSemRef*

Para o módulo de personalização de consultas carregar o perfil da base de perfis e utilizar a localização nas consultas, o usuário deve efetuar um *login* no *PSemRef* para sua identificação no sistema. Esse *login* e também a escolha de prioridades para as variáveis de enriquecimento são destaque nos casos de uso do *PSemRef*, ilustrados na Figura 27.

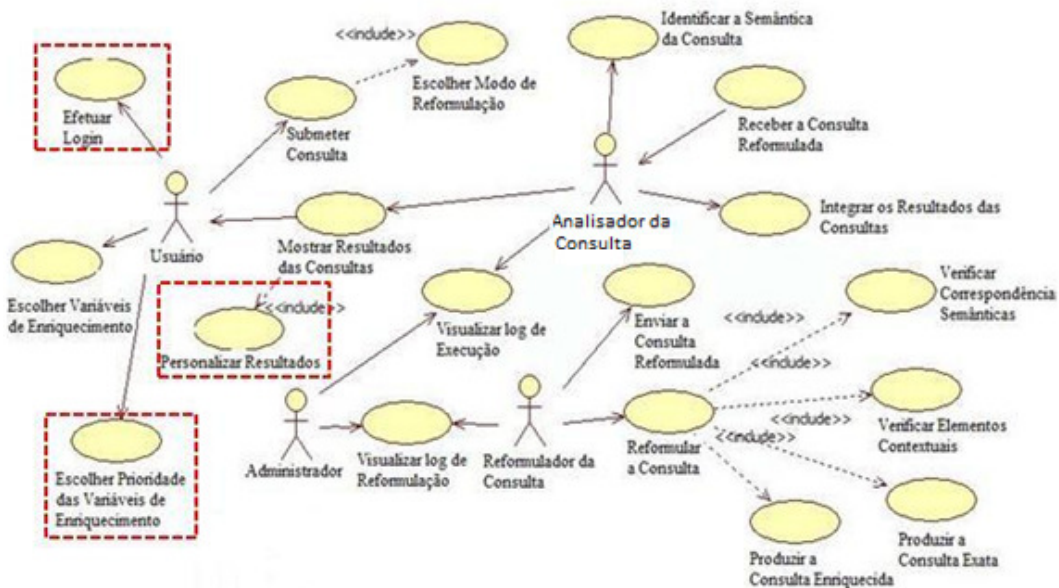


Figura 27 – Diagrama de Casos de Uso do *PSemRef*

Destacamos na Figura 27 as duas novas opções que estendem o *SemRef*: Efetuar *Login* e Escolher Prioridade das Variáveis de Enriquecimento. No *PSemRef* o

Analisador da Consulta (vide Figura 26), ao apresentar os resultados aos usuários, os exibe de uma forma personalizada e ordenada de acordo com as preferências e características destes usuários. Essas novas características e funcionalidades serão especificadas nas próximas seções.

Para ficar clara a relação da extensão do *PSemRef* ao módulo *SemRef*, considere a Figura 28. No módulo *SemRef*, o usuário poderia optar pela produção de duas reformulações para sua consulta: Exata (envolvendo o *peer* fonte e o *peer* destino) e Enriquecida (envolvendo o *peer* destino).

Após os resultados terem sido recuperados, a abordagem *PSemRef* foi desenvolvida de tal maneira que primeiramente os dois conjuntos de resultados (Exatos e Enriquecidos) são ordenados. Por fim, um *ranking* único de todo o conjunto de resultados da consulta do usuário é produzido, levando em consideração o Perfil do usuário, sua localização, e suas preferências na escolha da ordem de prioridade das variáveis de enriquecimento.

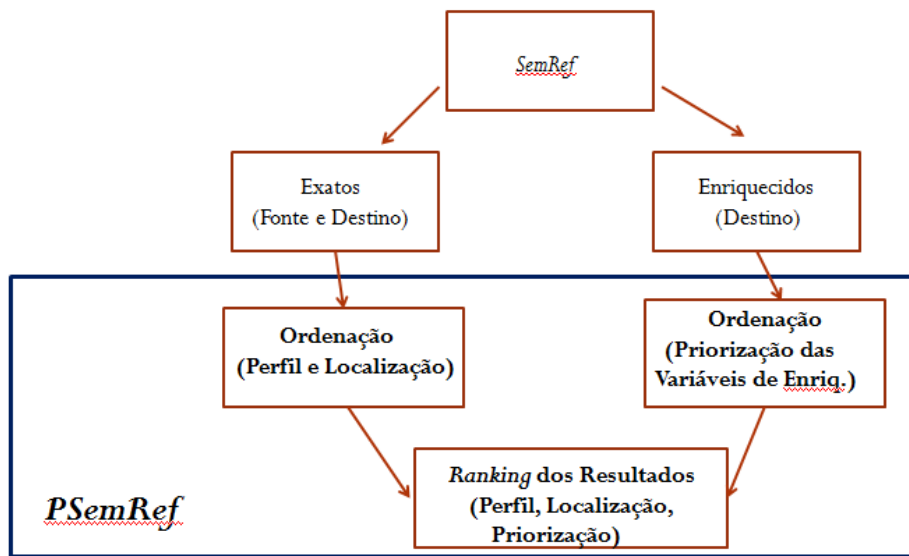


Figura 28 – Resumo da Especificação do *PSemRef*

4.4.2. Priorização das Variáveis de Enriquecimento

Como dito anteriormente, se o usuário submete uma consulta no *SemRef* e opta por uma reformulação exata, apenas a correspondência semântica de equivalência será levada em consideração no processo de reformulação. Caso o usuário opte por uma reformulação enriquecida, outras correspondências semânticas serão levadas em consideração para a produção dos resultados. A escolha das correspondências semânticas se dá por meio das variáveis de enriquecimento apresentadas na interface do *SemRef* (definidas na Seção 4.2.3).

Ao submeter suas consultas, um usuário pode selecionar quantas variáveis desejar, dependendo do seu interesse. Porém, se usuários diferentes selecionarem as mesmas variáveis na mesma consulta, irão observar resultados iguais. Com o objetivo de personalizar as consultas no SPEED, desenvolvemos o *PSemRef*. Uma de nossas extensões foi a de permitir aos usuários, além de selecionar as variáveis de enriquecimento, selecionar também a ordem de prioridade dessas variáveis nas consultas.

Sendo assim, cada usuário do *PSemRef* possui a liberdade de escolher quais variáveis são mais importantes para ele em uma determinada consulta, refletindo isso nos resultados retornados, que agora passarão a vir ordenados segundo essa priorização. Por exemplo, ao selecionar todas as variáveis de enriquecimento para sua consulta e o modo como expandido, o usuário pode escolher a variável de Composição como sendo a de maior prioridade. Então, ao serem exibidos os resultados da reformulação enriquecida, as instâncias derivadas das correspondências de agregação serão exibidas primeiramente, antes das instâncias oriundas dos outros tipos de correspondências.

A ordem padrão de resultados das consultas apresentada anteriormente no *SemRef*, considerando que o usuário optou pelo modo de consulta expandido, é a seguinte: (i) resultados do *peer* fonte; (ii) resultados derivados das correspondências de equivalência; e (iii) resultados derivados das outras correspondências semânticas. No *PSemRef*, ao invés dos resultados em (iii) serem exibidos de maneira igual para todos os usuários, eles estarão organizados num *ranking* de prioridade, ordenados de acordo com as preferências de cada usuário ao formular a consulta. Esse é um tipo de personalização, conhecido como personalização explícita, ou customização, visto que o usuário define abertamente (explicitamente) suas preferências para algo (nesse caso, as consultas).

O grande diferencial nessa proposta da priorização é o seu conceito. Embora os algoritmos para ordenação dos resultados enriquecidos (vistos no Capítulo 5) não sejam complexos, a idéia de priorização sobre as correspondências semânticas é algo muito importante para a organização dos resultados de consultas em PDMS, pois a reformulação é uma característica muito importante do processamento das consultas nestes sistemas.

Além disso, conforme a consulta vai sendo reformulada entre os pontos, muitas respostas podem ser encontradas, produzindo um resultado extenso. A divisão das respostas de acordo com a correspondência semântica que as originaram, aliada à

definição de um *ranking* de prioridades baseado em preferências traz benefícios aos usuários na visualização dos resultados de suas consultas.

Como os resultados das consultas são ordenados com base nas correspondências semânticas, cada reformulação gerada a partir da escolha de uma variável de enriquecimento gera uma nova consulta que é executada no *peer* destino (vide exemplo prático da Seção 4.3). O desempenho do *PSemRef* nos testes e experimentos com usuários (Capítulo 5) de reformulação de consultas envolvendo dois *peers* foi aceitável.

A execução de múltiplas consultas reformuladas pode ser prejudicial ao desempenho do sistema quando houver o roteamento da consulta original para vários *peers*. O processamento da consulta envolvendo roteamento para vários *peers* não faz parte do escopo deste trabalho.

É interessante que o *PSemRef* tenha todo o seu conjunto de resultados personalizado, ou seja, tanto os resultados recuperados do *peer* fonte quanto do *peer* destino. Nesta seção, tratamos da personalização dos resultados do *peer* destino que são derivados da reformulação enriquecida. A especificação da personalização dos resultados das consultas executadas no *peer* fonte e no *peer* destino (derivados da reformulação exata) será detalhada na próxima seção.

4.4.3. Utilização do Contexto do Usuário

Além de estender o *SemRef* com a funcionalidade de priorização das variáveis de enriquecimento, este trabalho também contribui com a ampliação do contexto do usuário, criando e utilizando perfis e informação de localização dos usuários no momento de submissão de suas consultas. Nesta seção, vamos abordar esta contribuição, a qual possui como objetivo personalizar os resultados oriundos do *peer* fonte e do *peer* destino (considerando apenas a reformulação exata).

Neste caso, resolvemos adotar uma técnica de personalização implícita, baseada em perfil do usuário e também em uma informação da localização deste usuário obtida no momento da submissão da sua consulta. Essas informações estendem e enriquecem o contexto do usuário que é levado em consideração nas consultas do *SemRef*, como visto na Figura 22 (Seção 4.2.3).

Conforme mostrado no Capítulo 2 (Seção 2.4), muitas abordagens descritas na literatura [Koutrika et al. 2004; Kostadinov et al. 2007; Campi et al. 2009] utilizam a técnica da personalização em seus sistemas a partir da criação de perfil para os usuários. No nosso trabalho também utilizamos um perfil [Ferraz 2010] (o qual possui como base

algumas características do usuário) como uma fonte confiável para a ordenação dos resultados das consultas submetidas no *PSemRef*.

No nosso ambiente, ao se fazer uma consulta por determinado conceito C_1 , vamos relacionar as propriedades desse conceito com informações do perfil do usuário para a ordenação das instâncias retornadas (o perfil dos usuários utilizado em nossa abordagem será especificado no Capítulo 5). Na reformulação exata entre os *peers*, um conceito C_2 no *peer* destino pode ser encontrado com as mesmas características (equivalente) de C_1 , fazendo então com que o mesmo critério para ordenação seja utilizado.

Porém, ao se aplicar outros tipos de correspondências em C_1 , outros conceitos com outras características podem ser encontrados, fazendo com que nosso critério do perfil não seja mantido para esses outros conceitos, principalmente quando a reformulação envolve a correspondência semântica de Aproximação (*closeness*), a qual não relaciona conceitos semanticamente muito semelhantes, o que faz com que o perfil do usuário não seja muito útil para ordenação destes conceitos aproximados. Por essa razão, optamos por dividir as frentes de trabalho entre a utilização de perfis para as consultas no *peer* fonte e reformulação exata, e a customização de prioridade das variáveis de enriquecimento para as reformulações enriquecidas.

A técnica de utilização de perfil relaciona algumas informações características do usuário (por exemplo, idade ou campo de estudo) com os esquemas das fontes de dados, ordenando assim as tuplas (ou instâncias, páginas web) de acordo com essas características.

Por exemplo, considerando um domínio semântico de Educação no SPEED, onde um usuário U_1 e um usuário U_2 vão submeter uma consulta M buscando por grupos de estudos. Considerando que o usuário U_1 possui afinidade com a área de Banco de Dados e o usuário U_2 com a área de Redes, é interessante que os grupos de estudos de Bancos de Dados cadastrados na rede do SPEED apareçam primeiro no resultado da consulta de U_1 , enquanto os grupos de estudos de Redes venham classificados primeiramente na consulta de U_2 .

Além disso, uma rede P2P como a do SPEED pode conectar pontos entre vários locais do mundo. Com isso, um bom critério para ordenação dos resultados das consultas, além dos perfis, é a localização do usuário que submete a consulta. Voltando ao nosso exemplo anterior, os resultados da consulta Q para o usuário U_1 continuarão mostrando primeiro os grupos de estudos da área de Banco de Dados, mas, além disso,

irão mostrar primeiro os grupos de estudo registrados do Brasil, considerando que U_1 submeteu sua consulta do Brasil. Para isso ser possível, além do perfil ser carregado no momento do *login* de um usuário, a sua localização também será recuperada no momento de utilização do sistema.

Essas novas informações (perfil e localização) podem ser consideradas como elementos contextuais (vide Seção 2.3). A localização do usuário é uma informação contextual dinâmica, pois não está armazenada, e sim é obtida no momento de utilização do sistema. O perfil do usuário é uma informação contextual estática, pois, diferentemente da localização, o perfil está armazenado e é sempre o mesmo para um usuário, independentemente do ambiente onde ele está submetendo suas consultas.

O diagrama de atividades da Figura 29 resume as principais tarefas (do usuário e do sistema) que envolvem desde a submissão de uma consulta por um usuário até a exibição dos resultados finais.

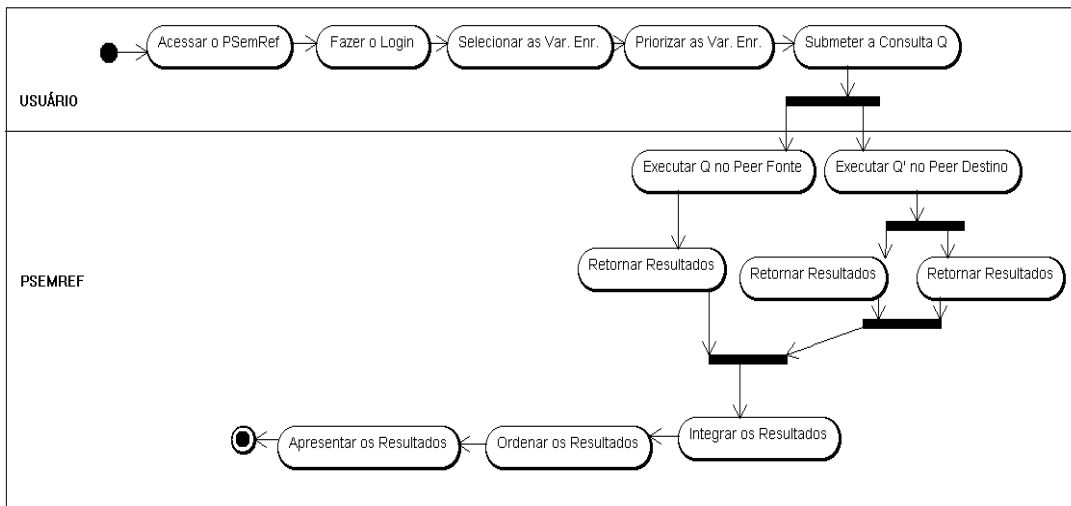


Figura 29 – Atividades do *PSEmRef*

O usuário primeiro acessa o *PSEmRef* e realiza o *login* no sistema para sua identificação (captura do perfil e sua localização). Após isso, ele pode selecionar um conjunto de variáveis de enriquecimento assim como sua ordem de prioridade para a submissão de uma consulta com modo expandido.

As atividades seguintes são todas de responsabilidade do *PSEmRef*. As consultas são reformuladas e executadas nos *peers* fonte e destino. Os conjuntos de resultados são integrados e, antes da exibição ao usuário, são ordenados de acordo com o perfil do usuário, localização, e ordem de prioridade das variáveis de enriquecimento escolhidas pelo usuário.

4.5. Considerações

Neste capítulo apresentamos o módulo de reformulação de consultas do sistema SPEED, chamado de *SemRef*, ambiente onde nosso trabalho foi realizado. Foram descritas suas características principais e fundamentos, e também apresentamos sua arquitetura e diagrama de casos de uso originais. Além disso, fornecemos um exemplo prático de consulta submetida no *SemRef*.

Uma consulta submetida por qualquer usuário em um mesmo ambiente da rede retornaria os mesmos resultados. Neste capítulo nós especificamos nossas contribuições que objetivam modificar esse cenário, personalizando as consultas submetidas no SPEED. Essas contribuições estenderam o módulo *SemRef* com novos componentes e funcionalidades. Esse módulo estendido recebeu o nome de *PSemRef* (*Personalized SemRef*).

Sugerimos duas frentes de trabalho para a especificação e criação do *PSemRef*. Uma das frentes de trabalho envolveu ordenar os resultados das consultas enriquecidas (Seção 4.4.2) e a outra envolveu a ordenação dos resultados das consultas exatas e também das submetidas no *peer* fonte (Seção 4.4.3). Com isso, desenvolvemos uma personalização explícita (com a escolha de prioridades para as variáveis de enriquecimento) e uma personalização implícita (a partir do uso de perfis e da informação de localização dos usuários).

Assim, cada usuário que submeter uma consulta Q irá observar uma ordenação de resultados diferente, levando em consideração as características e preferências de quem a submeteu. Isso auxilia o usuário a encontrar mais rapidamente uma informação desejada.

Em resumo, os resultados de qualquer consulta Q para um usuário U , envolvendo as ontologias O_1 e O_2 seguem a ordem: (i) resultados de O_1 e derivados da reformulação exata de O_2 ordenados segundo o perfil e localização de U ; e (ii) resultados derivados da reformulação enriquecida de O_2 , ordenados segundo a priorização das variáveis de enriquecimento, escolhida por U .

No Capítulo 5 mostraremos detalhes de implementação de nossas contribuições (p.ex. interfaces do sistema), assim como experimentos realizados no *PSemRef* envolvendo um estudo de caso.

Capítulo 5

Implementação e Estudo de Caso

Este capítulo tem como objetivo principal abordar as questões de implementação do nosso trabalho, detalhando os aspectos de desenvolvimento do módulo de reformulação de consultas com personalização. Fornecemos também exemplos mais detalhados de utilização do *PSemRef* envolvendo um estudo de caso, além de um *feedback* de usuários reais sobre nossa abordagem.

Este capítulo está organizado como segue. A Seção 5.1 mostra detalhes de implementação acerca do módulo *PSemRef* e do processo de reformulação e execução das consultas. A Seção 5.2 descreve as questões de implementação da personalização dos resultados das consultas. A Seção 5.3 mostra alguns experimentos mais detalhados de utilização do *PSemRef* enquanto a Seção 5.4 traz um *feedback* de usuários a respeito de nossa proposta. Finalmente, a Seção 5.5 conclui o capítulo com algumas considerações.

5.1. Implementação do Módulo *PSemRef*

Para a implementação do *PSemRef*, foi escolhida a linguagem de programação Java [Java 2010], com o auxílio da ferramenta Eclipse [Eclipse 2010]. Java foi escolhida por ser uma linguagem independente de plataforma, além de permitir (por meio do Eclipse) a criação de interfaces gráficas (do inglês GUI, *Graphical User Interface*) amigáveis necessárias ao sistema para sua fácil visualização e utilização por parte dos usuários.

Além disso, Java possui APIs (*Application Programming Interface*) [Java API 2010] que dão suporte à programação com ontologias, facilitando sua manipulação. Essas APIs [Jena 2010; Protégé API 2010] permitem o desenvolvimento de métodos de consultas sobre essas ontologias com o uso da linguagem SPARQL.

Nesta seção, apresentamos detalhes a respeito do processo de reformulação e execução das consultas, além de algumas funcionalidades da interface desenvolvidas para o módulo *PSemRef*, especificando suas características e funcionalidades.

5.1.1. Reformulação e Execução de Consultas

A reformulação de consultas é considerada a tarefa mais importante no processamento de consultas em ambientes distribuídos, pois permite que as consultas dos usuários sejam executadas em fontes distintas, com esquemas e dados diferentes.

No *PSemRef*, nós enriquecemos a reformulação das consultas utilizando semântica derivada de correspondências entre as ontologias dos *peers* no SPEED, e personalizamos os resultados a partir do perfil do usuário, de sua localização e de uma escolha de prioridades das variáveis de enriquecimento, que representam as correspondências semânticas.

Os usuários possuem duas maneiras para formulação de consultas: (i) Lógica Descritiva ou (ii) SPARQL. O primeiro passo do processamento das consultas, assim que o usuário as submete, é a verificação de configurações que envolvem a consulta, ou seja, o modo da consulta, assim como as variáveis de enriquecimento selecionadas.

No ambiente de reformulação de consultas, alguns aspectos são considerados:

- Existem dois *peers* – *peer* fonte P_1 e *peer* destino P_2 ;
- Os esquemas dos *peers* são representados por ontologias;
- Os *peers* (e suas ontologias) estão inseridos num mesmo domínio de conhecimento;
- Uma ontologia descrevendo o domínio (DO) está disponível; e
- Um conjunto de correspondências semânticas $Co[O_1, O_2]$ entre P_1 e P_2 foi gerado.

A Figura 30 ilustra um trecho de um arquivo de correspondências semânticas de um domínio de Educação para os conceitos *FullProfessor* e *University* de P_1 em relação a alguns conceitos de P_2 . Esse arquivo é gerado pelo *matcher* semântico (Seção 4.2.4).

```
<rdf:Description rdf:about="http://swrc.ontoware.org/ontology/portal#FullProfessor">
  <j.0:isDisjointWith>http://www.lehigh.edu/~zhp2/univ-bench.owl#AssociateProfessor</j.0:isDisjointWith>
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#VisitingProfessor</j.0:isCloseTo>
  <j.0:isPartOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#Course</j.0:isPartOf>
  <j.0:isPartOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#ResearchProject</j.0:isPartOf>
  <j.0:isSubConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#Professor</j.0:isSubConceptOf>
  <j.0:isEquivalentTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#FullProfessor</j.0:isEquivalentTo>
</rdf:Description>
<rdf:Description rdf:about="http://swrc.ontoware.org/ontology/portal#University">
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#College</j.0:isCloseTo>
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#ResearchGroup</j.0:isCloseTo>
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#Program</j.0:isCloseTo>
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#Department</j.0:isCloseTo>
  <j.0:isCloseTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#Institute</j.0:isCloseTo>
  <j.0:isSubConceptOf>http://www.lehigh.edu/~zhp2/univ-bench.owl#Organization</j.0:isSubConceptOf>
  <j.0:isEquivalentTo>http://www.lehigh.edu/~zhp2/univ-bench.owl#University</j.0:isEquivalentTo>
</rdf:Description>
```

Figura 30 – Correspondências Semânticas geradas entre P_1 e P_2

Caso o usuário submeta uma consulta $Q_1 = FullProfessor$ no modo restrito, o conceito *FullProfessor* (Q_{exata}) em P_2 também será consultado devido ao relacionamento de equivalência detectado nas ontologias dos *peers*. Se na mesma consulta Q_1 o usuário opta pelo modo expandido e tendo selecionado a variável de enriquecimento *Compose*, a reformulação enriquecida será produzida buscando pelos conceitos *ResearchProject* e *Course* ($Q_{enriquecida}$) em P_2 (relacionamento de agregação - *isPartOf*), além da reformulação exata. Na Seção 5.3 será visto o passo-a-passo de execução de consultas no *PSemRef* em detalhes, a partir de alguns testes realizados na ferramenta.

Nesse sentido, o algoritmo de reformulação do *PSemRef* recebe como entrada uma consulta Q submetida em P_1 , $Co[O_1, O_2]$, e as configurações escolhidas pelo usuário (modo da consulta e variáveis de enriquecimento). Como saída, o algoritmo produz uma consulta derivada da reformulação exata e consultas derivadas da reformulação enriquecida (uma consulta para cada variável de enriquecimento selecionada).

Algoritmo de Reformulação

O algoritmo original de reformulação [Souza 2009], em uma visão alto nível, é apresentado na Figura 31. Sua implementação foi feita na linguagem de programação Java. A versão completa do algoritmo está ilustrada no Apêndice A. As especificações foram feitas em inglês.

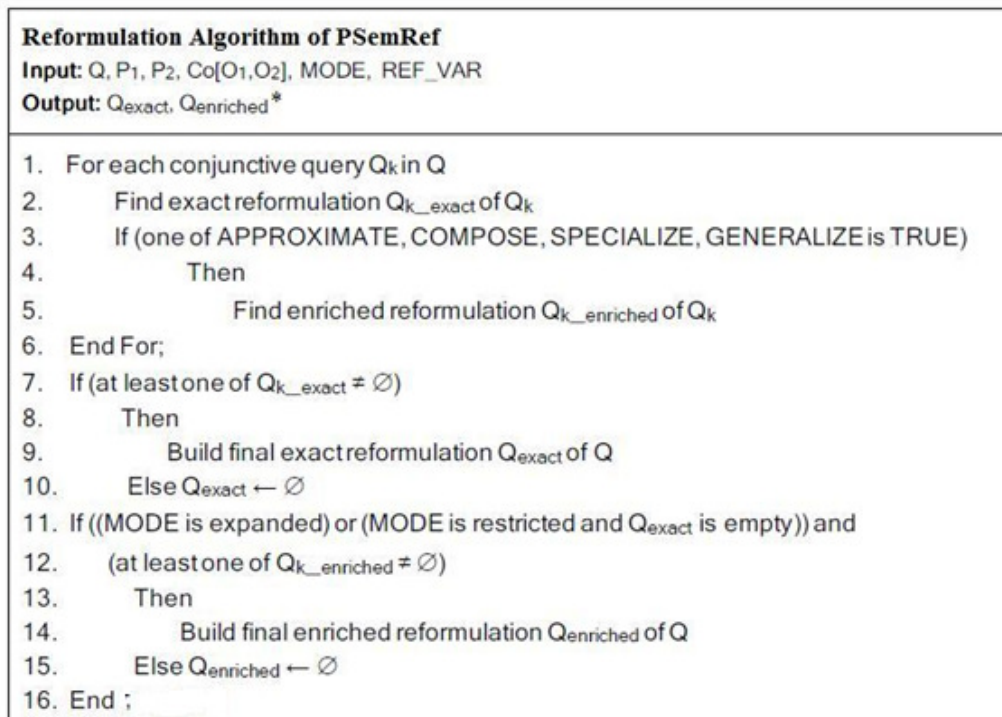


Figura 31 – Visão em alto nível do algoritmo de reformulação do *PSemRef* [Souza 2009]

Na Figura 31, chamamos atenção para o fato de a variável de saída $Q_{enriched}$ aparecer com “*” (significando a possibilidade de várias consultas), devido ao fato de que podemos, na versão corrente, gerar entre uma e quatro consultas reformuladas enriquecidas, dependendo da quantidade de variáveis de enriquecimento selecionadas. A variável de saída Q_{exact} não possui “*”, pois podemos gerar no máximo uma consulta reformulada exata.

Mais especificamente, com o objetivo de produzir as consultas reformuladas, o algoritmo realiza as seguintes tarefas [Souza 2009]:

- Recebe uma consulta Q na forma de uma disjunção de conjunções de conceitos da ontologia do *peer* P_1 (fórmula geral das consultas em Lógica Descritiva). Para cada conjunção (Q_k), cada conceito existente (C_j) será avaliado e os conceitos correspondentes em P_2 (relacionados com os conceitos de P_1 por algum tipo de correspondência semântica) serão adicionados a um dos três tipos de conjuntos:
 - S_1C_j – conjunto dos conceitos equivalentes a C_j ;
 - S_2C_j – conjunto dos conceitos relacionados a C_j por outro tipo de correspondência semântica (aproximação, subconceito, superconceito, parte de, todo). Cada conceito adicionado neste conjunto recebe um identificador com o nome da correspondência semântica que o relacionou com C_j (isso é útil para a ordenação dos resultados enriquecidos, que será vista na próxima seção). Esse conjunto S_2C_j é produzido se o modo de reformulação for expandido ou se for restrito e o conjunto S_1C_j retornar vazio; e
 - $Neg_S_2C_j$ – se existir um operador de negação em C_j , o *PSemRef* busca por conceitos disjuntos na ontologia do *peer* destino. Nesse caso, o conceito é adicionado a $Neg_S_2C_j$. Se não existir conceitos disjuntos, uma variável $BNeg$ é alterada para Verdadeiro e posteriormente no algoritmo o operador de negação é aplicado sobre os conceitos correspondentes ao conceito negado C_j (conceitos recuperador a partir das outras correspondências semânticas).
- Após o processamento de todos os conceitos de uma conjunção, o *PSemRef* verifica se existem correspondências exatas (equivalência) e se a conjunção não falha, ou seja, todos os conceitos existentes na conjunção possuem conceitos correspondentes no *peer* destino. Caso isso aconteça, a reformulação exata é construída para a conjunção Q_k ;

- Da mesma maneira, se existem correspondências enriquecidas e se a conjunção não falha, o *PSemRef* constrói a reformulação enriquecida para a conjunção Q_k (a checagem e reformulação são feitas para cada variável de enriquecimento);
- Finalmente, após processar todas as conjunções Q_k de Q , o *PSemRef* produz a consulta exata final Q_{exact} , na forma de uma disjunção das conjunções exatas e as consultas enriquecidas finais Q_{enriched} como disjunções das conjunções produzidas (divididas de acordo com a correspondência semântica que as originaram).

Métodos complementares que são chamados pelo algoritmo de reformulação foram definidos. O método *Build_Exact_Reformulation* é responsável por construir uma reformulação exata para uma conjunção. Similarmente, para a reformulação enriquecida, temos o método *Build_Enriched_Reformulation*. Com as reformulações produzidas para as conjunções, os métodos *Build_Final_Exact_Reformulation* e *Build_Final_Enriched_Reformulation* geram as disjunções finais a partir das reformulações das conjunções já executadas pelos outros métodos. Esses métodos complementares estão ilustrados no Apêndice B.

Execução das Consultas

Após as reformulações terem sido produzidas, elas são executadas no *peer* destino e a consulta original no *peer* fonte. O algoritmo de reformulação recebe como entrada uma consulta especificada em Lógica Descritiva (DL). Portanto, se o usuário submete uma consulta em SPARQL, haverá uma tradução desta para a DL para viabilização do processo de reformulação.

As consultas executadas sobre as ontologias dos *peers* são na forma da linguagem SPARQL. Como a saída do algoritmo de reformulação são as consultas especificadas em DL, haverá outra tradução - das consultas reformuladas para SPARQL - para execução no *peer* destino. Se o usuário tiver submetido sua consulta em DL, esta também será traduzida para SPARQL para execução no *peer* fonte.

As pesquisas sobre a linguagem SPARQL nos levaram à criação de *templates* para estas consultas, os quais serviram então de base para as traduções DL-SPARQL e SPARQL-DL. Os formatos das traduções são vistos nas Figuras 32 (um conceito), 33 (disjunção), 34 (conjunção) e 35 (negação) [Souza 2009].

A consulta por propriedades/atributos dos conceitos só pode ser realizada em SPARQL. O *template* para a consulta por propriedades pode ser visto na Figura 36. As consultas sobre as ontologias foram implementadas com o auxílio da API do Protégé

[Protégé API 2010]. Nas figuras a seguir, a expressão “ontologia” aparece nos *templates* da ferramenta no formato do URI da ontologia do *peer*.

DL		SPARQL
<u>Conceito</u>	<->	SELECT ?x FROM <ontologia> WHERE { ?x rdf:type <ontologia# <u>Conceito</u> > }

Figura 32 – Tradução consulta DL – SPARQL (um conceito)

DL		SPARQL
<u>Conceito1</u> \sqcup <u>Conceito2</u>	<->	SELECT ?x FROM <ontologia> WHERE { { ?x rdf:type <ontologia# <u>Conceito1</u> > } UNION { ?x rdf:type <ontologia# <u>Conceito2</u> > }}

Figura 33 – Tradução consulta DL – SPARQL (disjunção)

DL		SPARQL
<u>Conceito1</u> \sqcap <u>Conceito2</u>	<->	SELECT ?x FROM <ontologia> WHERE { { ?x rdf:type <ontologia# <u>Conceito1</u> > . ?x rdf:type <ontologia# <u>Conceito2</u> > }}

Figura 34 – Tradução consulta DL – SPARQL (conjunção)

DL		SPARQL
<u>-Conceito</u>	<->	<pre> SELECT ?x FROM <ontologia> WHERE { { ?x rdf:type ?y . <ontologia#Conceito> rdfs:subClassOf ?y } UNION { ?x rdf:type ?z . <ontologia#Conceito> rdfs:subClassOf ?y . ?z rdfs:subClassOf ?y . FILTER (?z != <ontologia#conceito>) }}</pre>

Figura 35 – Tradução consulta DL – SPARQL (negação)

```

SPARQL
SELECT ?x ?prop1 ?prop2
FROM <ontologia>
WHERE {
?x rdf:type <ontologia#Conceito> .
{
?x <ontologia#prop1> ?prop1
} UNION {
?x <ontologia#prop2> ?prop2
}
}
```

Figura 36 – Consulta em SPARQL (propriedades)

Após as traduções para o SPARQL e execução das consultas nos *peers*, os resultados são integrados e exibidos para o usuário de uma maneira personalizada. As questões de implementação envolvendo a personalização das consultas e a ordenação dos resultados serão vistas na Seção 5.2.

5.1.2. Interfaces Gráficas

Nesta seção vamos apresentar algumas telas que compõem o módulo *PSemRef*, detalhando suas características e funcionalidades. Todas as telas foram codificadas em inglês.

Tela Principal

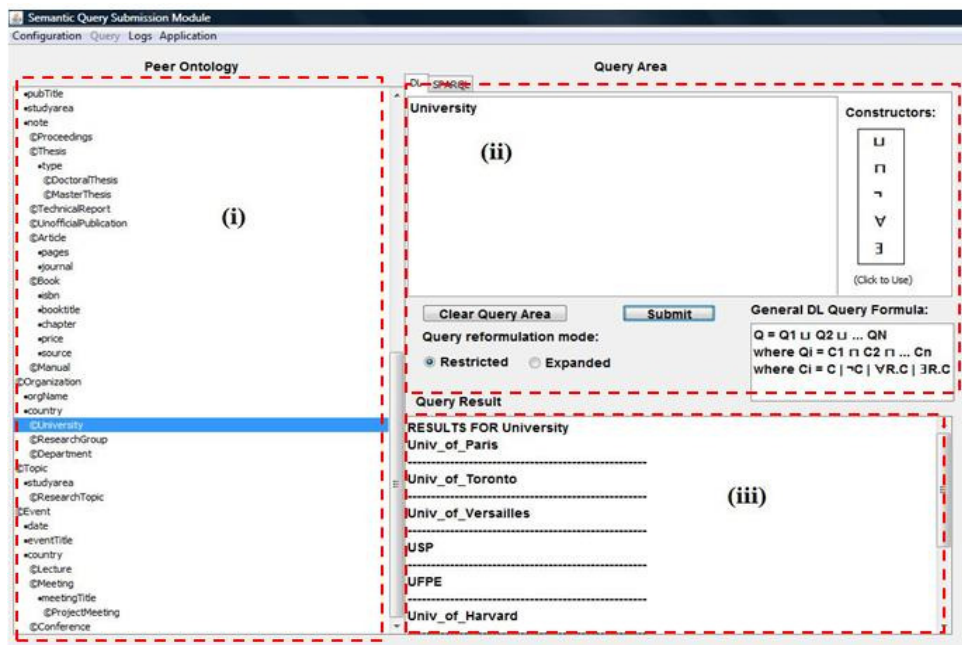


Figura 37 – Tela Principal do *PSemRef* – Consulta em DL

A Figura 37 ilustra a tela principal do *PSemRef*, composta de três grandes áreas. A área (i) mostra a estrutura de conceitos e propriedades da ontologia do *peer* fonte, montada com o auxílio da API Jena. Essa estrutura serve de base para a formulação das consultas dos usuários. Podemos observar alguns conceitos posicionados de forma diferente (uns mais à direita do que outros), referentes à idéia do relacionamento de hierarquia (subconceito e superconceito) existente entre eles. Por exemplo, nessa ontologia, existe um conceito chamado *Project*, que possui como atributos *studyArea* e *projectTitle* e que é superconceito de *SoftwareProject*.

A área (ii) é a área de formulação das consultas (neste caso, consultas formuladas em Lógica Descritiva, ou DL). No exemplo da Figura 37, foi submetida uma consulta buscando por Universidades. Mais à direita da consulta estão os construtores da DL para serem utilizados na formulação de consultas compostas. Os operadores presentes na interface são, na ordem de cima para baixo, (a) \sqcup (disjunção, ou operador de união), (b) \sqcap (conjunção, ou operador de interseção), (c) \neg (negação, ou operador de complemento), (d) \forall (restrição/quantificador universal) e (e) \exists (restrição/quantificador existencial). Nesta versão do nosso trabalho, não utilizamos os quantificadores nas consultas.

Ainda na área (ii), abaixo da área de formulação das consultas, temos dois botões, um que possui como função limpar a área de consultas e o outro que submete a consulta formulada. Também podemos enxergar as opções para o modo de

reformulação das consultas como restrito (gerar apenas a reformulação exata) ou expandido (gerar a reformulação exata e a enriquecida). Esses modos de reformulação foram definidos no Capítulo 4 (Seção 4.2.3). O último componente da seção (ii) é uma legenda para auxílio aos usuários, pois descreve a fórmula geral para formulação das consultas em DL. As consultas em nossa abordagem são formuladas por meio de uma disjunção de conjunções dos conceitos presentes na ontologia do *peer* fonte.

A área (iii) apresenta a área para os resultados das consultas. Os resultados são compostos por títulos informando que conceito(s) gerou os resultados e a seguir as instâncias recuperadas das ontologias dos *peers* envolvidos na consulta. Outra opção para a formulação das consultas no *PSemRef* é a linguagem SPARQL, como podemos observar na Figura 38. Nessa opção de consulta em SPARQL, as áreas (i) e (iii) destacadas na Figura 38 possuem as mesmas características das apresentadas na Figura 37.

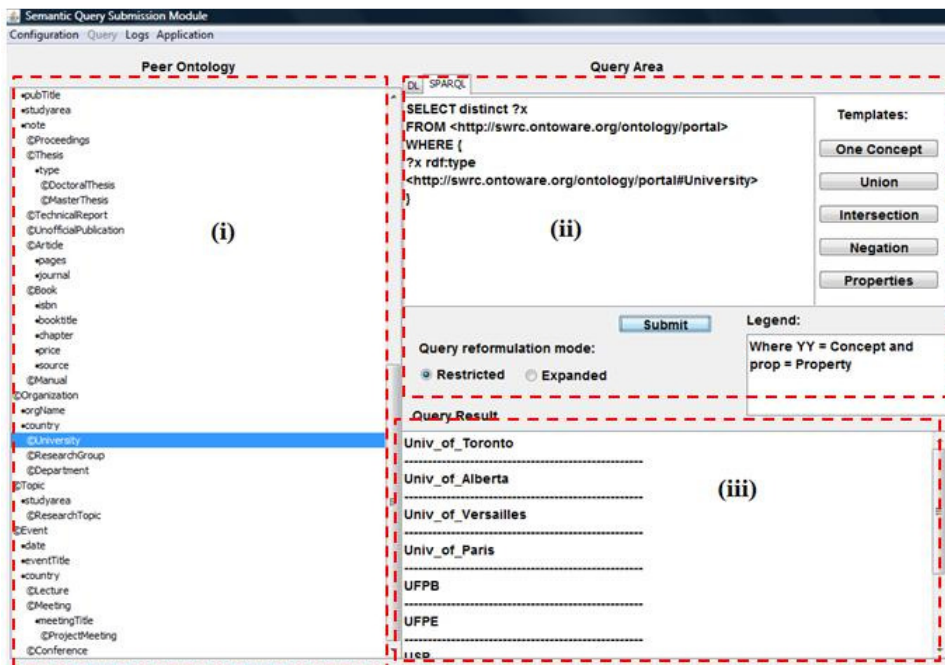


Figura 38 – Tela Principal do *PSemRef* – Consulta em SPARQL

A mudança ocorre na área (ii), pois na opção de SPARQL as consultas são formuladas com o auxílio de *templates*, localizados do lado direito da área de formulação das consultas, visto que a linguagem SPARQL não é intuitiva para a maioria dos usuários. Cada *template* possui a estrutura de uma consulta específica, nessa ordem, de cima para baixo, (a) busca por um conceito, (b) disjunção ou união de conceitos, (c) conjunção ou interseção de conceitos, (d) negação ou complemento de um conceito e (e) busca por propriedade(s) / atributo(s) de um conceito. Na Figura 38, a mesma consulta realizada na opção de DL foi submetida (busca por Universidades).

Outra mudança na opção de consultas em SPARQL é a legenda da tela principal, que agora auxilia o usuário no preenchimento do *template* escolhido para submissão de sua consulta. Por exemplo, ao selecionar o *template* de Propriedades, o usuário irá observar a estrutura mostrada na Figura 39.

```
SELECT ?x ?prop1 ?prop2
FROM <http://swrc.ontoware.org/ontology/portal>
WHERE {
  ?x rdf:type <http://swrc.ontoware.org/ontology/portal#YY1> .
  {
    ?x <http://swrc.ontoware.org/ontology/portal#prop1> ?prop1
  } UNION {
    ?x <http://swrc.ontoware.org/ontology/portal#prop2> ?prop2
  }
}
```

Figura 39 – *Template* SPARQL de uma consulta por Propriedades

Com o auxílio da legenda, ele preencherá no espaço “YY1” um conceito e nos espaços “prop1” e “prop2” os nomes das propriedades escolhidas. Os elementos de uma consulta SPARQL foram especificados no Capítulo 4 (Seção 4.2.2).

Outro componente da tela principal do *PSemRef* é a barra de *menus*, abaixo da barra de título da ferramenta, composta das seguintes opções: (i) *Configuration* – opções referentes à seleção das variáveis de enriquecimento e da ordem de prioridade dessas variáveis; (ii) *Logs* – acesso às telas de logs da ferramenta; e (iii) *Application* – composta das opções de *Login/Logout* de usuários e da opção *Exit*, a qual encerra a sessão do usuário no *PSemRef*. Nas seções seguintes destacamos essas opções.

Variáveis de Enriquecimento

A Figura 40 exibe a tela do *PSemRef* na qual o usuário seleciona quais variáveis de enriquecimento serão utilizadas para a reformulação de suas consultas. Essa tela é acessada a partir da opção *Configuration – Set Reformulation Variables* na barra de *menu* da tela principal.

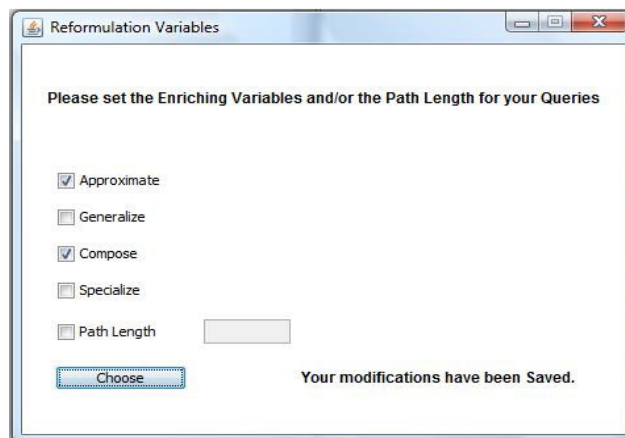


Figura 40 – Tela de escolha de Variáveis de Enriquecimento

Como dissemos anteriormente, cada variável possui relação com uma ou mais correspondências semânticas. Ao selecionar uma variável na tela mostrada na Figura 40, a correspondência semântica associada será incorporada na reformulação enriquecida da consulta. As variáveis se relacionam com as correspondências da seguinte maneira:

- *Approximate*: correspondência de aproximação (*isCloseTo*);
- *Generalize*: correspondência de subconceito (*isSubConceptOf*);
- *Specialize*: correspondência de superconceito (*isSuperConceptOf*);
- *Compose*: correspondências de agregação (*isPartOf* e *isWholeOf*).

A correspondência de disjunção não possui uma variável associada. Para considerar esse tipo de correspondência na reformulação enriquecida, um operador da DL é fornecido na tela principal (\neg), e um *template* específico pode ser utilizado na opção de consulta em SPARQL. A variável *path length* não é utilizada nessa versão da ferramenta, pois nosso ambiente consta de apenas dois *peers*. Essa variável controlará o roteamento da consulta (indicando a quantidade de *peers* pelos quais a consulta deverá ser reformulada) em versões futuras do nosso trabalho.

Os conceitos equivalentes (correspondência de equivalência) sempre serão retornados no momento de reformulação da consulta. Caso o usuário opte pelo modo de reformulação restrito, apenas a reformulação exata será produzida. Para se levar em consideração as variáveis de enriquecimento selecionadas a princípio, o usuário deverá escolher o modo de reformulação expandido. Existe um caso particular que é o de um usuário escolher o modo de reformulação restrito, porém a reformulação exata retornar vazia: nesse caso, a reformulação enriquecida é produzida de acordo com as variáveis de enriquecimento escolhidas. Essa opção objetiva evitar ao máximo que o usuário não encontre resultados vazios para suas consultas.

Telas de Logs

O *PSemRef* oferece aos seus usuários dois logs, um com detalhes sobre as reformulações e outro referente aos resultados das consultas submetidas durante suas sessões na ferramenta de consultas. As Figuras 41 e 42 ilustram as telas de logs de reformulação e de resultados, respectivamente.

Os trechos dos logs exibidos nas figuras a seguir dizem respeito a uma consulta por apenas um conceito (*Student*), no modo expandido, com a seleção de duas variáveis de enriquecimento (*Specialize* e *Generalize*), envolvendo o *peer* fonte P_1 e o *peer* destino P_2 .

No log de reformulação (acessado a partir da opção Logs – *View Reformulation LOG* na barra de *menu* da tela principal), o usuário pode observar, para cada consulta submetida, o respectivo modo de reformulação escolhido, as variáveis de enriquecimento selecionadas, além de sua ordem de prioridade para os resultados (nesse exemplo, não foi especificada uma ordem de prioridade). O log de reformulação pode ser visto na Figura 41, onde estão exibidas a consulta original (a qual é executada em P_1), a consulta exata (resultado da reformulação exata e executada em P_2) e as consultas reformuladas produzidas considerando cada variável de enriquecimento selecionada (também executadas em P_2).

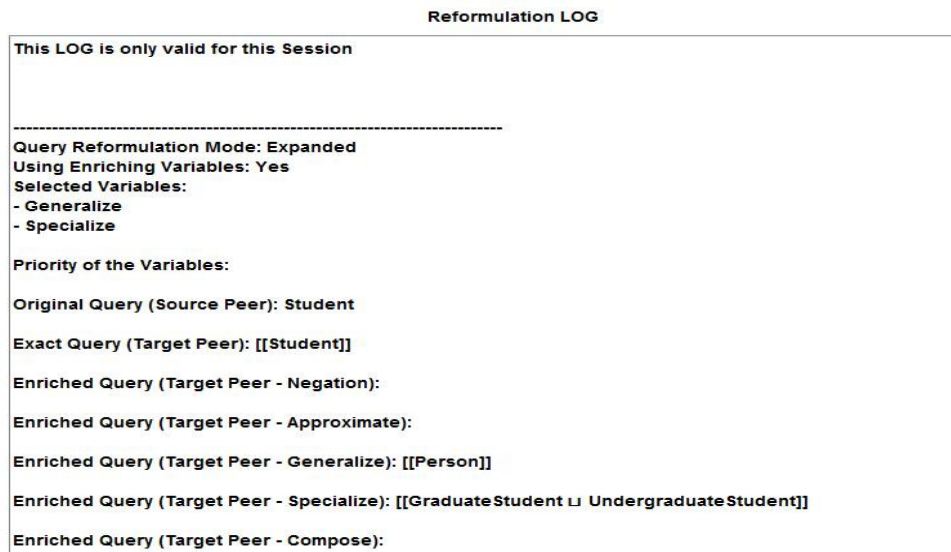


Figura 41 – Tela do Log de Reformulação

No nosso exemplo na Figura 41, a consulta *Student* produziu três consultas reformuladas: (i) *Student*, resultado da correspondência semântica de equivalência (notação: $P_1:Student \equiv P_2:Student$); (ii) *Person*, resultado da correspondência semântica de subconceito (notação: $P_1:Student \sqsubseteq P_2:Person$); e (iii) *GraduateStudent* \sqcup *UndergraduateStudent*, derivada da correspondência de superconceito (notação: $P_1:Student \sqsupseteq P_2: GraduateStudent$ e $P_1:Student \sqsupseteq P_2: UndergraduateStudent$). Caso a reformulação derivada de uma variável encontre mais de um conceito, como em (iii), é formada uma disjunção dos conceitos encontrados para execução da consulta.

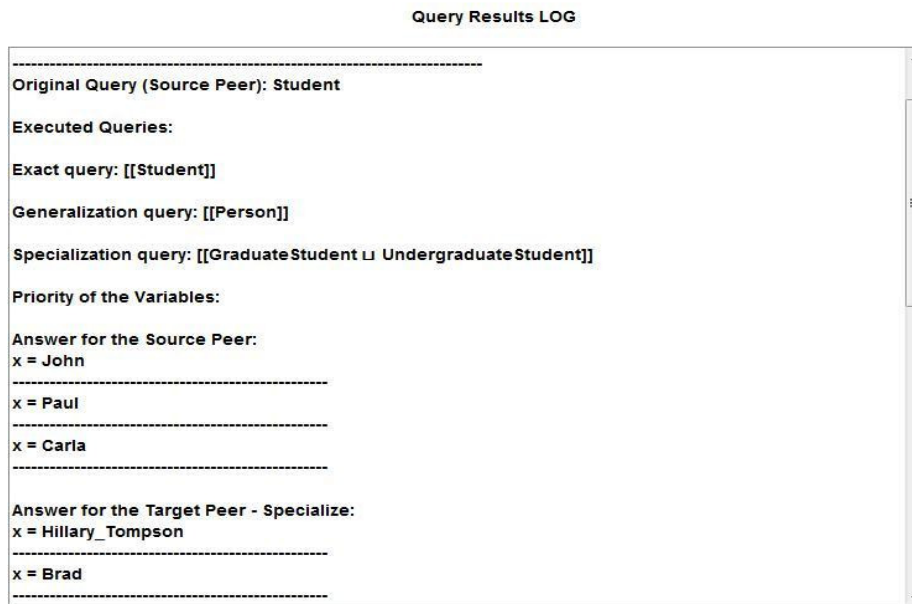


Figura 42 – Tela do Log de Resultados

No log de resultados (acessado a partir da opção Logs – *View Query Results LOG* na barra de *menu* da tela principal), o foco de informação para os usuários é outro. Para cada consulta submetida são informadas as consultas que foram de fato executadas (a consulta original e as consultas reformuladas produzidas). No nosso exemplo na Figura 42, podemos observar a consulta *Student* original e as consultas reformuladas (as quais também foram informadas no log de reformulação).

O log de resultados também foca nas respostas retornadas para os usuários, divididas em relação às reformulações. Na Figura 42 podemos enxergar as instâncias que foram recuperadas pela consulta original e um trecho das instâncias recuperadas pela consulta reformulada a partir da variável de enriquecimento de especialização. A ordem no log de resultados é sempre essa para as respostas: (i) resultados do *peer* fonte; e (ii) resultados do *peer* destino, para cada correspondência semântica utilizada a partir da seleção das variáveis de enriquecimento.

5.2. Implementação da Personalização

Na seção anterior, discutimos como foram implementadas as principais interfaces do *PSemRef*, e também os algoritmos e padrões que envolvem o processo de reformulação e execução das consultas. Nesta seção, iremos descrever as questões de implementação relacionadas ao principal objetivo deste trabalho: a personalização dos resultados das consultas.

5.2.1. Variáveis de Enriquecimento

Para a execução de uma reformulação enriquecida nas consultas, é necessária a escolha do modo de consultas expandido, além da seleção de uma ou mais variáveis de enriquecimento a serem consideradas (ou escolher o modo de consultas restrito e a reformulação exata retornar vazia). Além dessa escolha das variáveis, uma outra funcionalidade importante do *PSemRef* é a noção de uma ordem de prioridade para as variáveis selecionadas.

O objetivo da priorização das variáveis é determinar em que ordem os resultados enriquecidos serão exibidos para o usuário. O resultado derivado da reformulação enriquecida é dividido em grupos de acordo com a correspondência semântica que o originou, assim, um *ranking* desses grupos é formado na exibição final das respostas, baseado na ordem de importância escolhida das variáveis de enriquecimento.

A Figura 43 ilustra a tela da ferramenta onde o usuário define a ordem de prioridade das variáveis de enriquecimento.

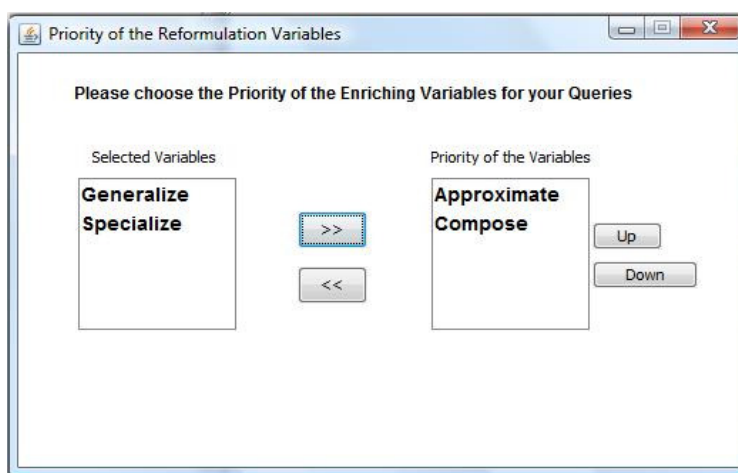


Figura 43 – Tela de Priorização das Variáveis de Enriquecimento

Nesse caso da Figura 43, as quatro variáveis foram selecionadas, então elas ficam à disposição na caixa do lado esquerdo (*Selected Variables*). O usuário decidiu então priorizar os resultados da reformulação enriquecida derivados das correspondências de aproximação e agregação movendo as variáveis *Approximate* e *Compose* para a caixa do lado direito (*Priority of the Variables*). Além disso, foi escolhida a maior prioridade para a variável *Approximate*, pois ela está situada acima da variável *Compose*.

Com esse cenário resumimos as características dessa interface: as variáveis dispostas na caixa *Selected Variables* não possuem prioridade, portanto suas respostas derivadas aparecem por último na área de resultados (sem uma ordem específica). As

variáveis com prioridade situam-se na caixa *Priority of the Variables*, com ordenação definida pela disposição das variáveis de cima para baixo.

Uma visão em alto nível do algoritmo de ordenação dos resultados enriquecidos é fornecida na Figura 44.

```
Enriched Results Ranking  
Input: EnrichedResults, VariablesPriority  
  
Boolean app = false;  
Boolean gen = false;  
Boolean spec = false;  
Boolean comp = false;  
For each element E in VariablesPriority  
  Test(E) //Approximate, Generalize, Specialize, Compose  
  Print correspondent result of EnrichedResults  
  Set correspondent variable to True  
End For;  
If any variable is False //app, gen, spec, comp  
  Print correspondent result of EnrichedResults  
  
End Method;
```

Figura 44 – Algoritmo de Ordenação dos Resultados Enriquecidos

Os parâmetros de entrada são o conjunto mapeado dos resultados enriquecidos (esse mapa é indexado pela variável de enriquecimento que originou o resultado) e a ordem de prioridade das variáveis de enriquecimento selecionada pelo usuário. Esse mapa é gerado pelo algoritmo de reformulação, apresentado na Seção 5.1.1.

Cada variável do *ranking* de prioridade é testada e, dependendo do seu valor, os resultados correspondentes são exibidos a partir do mapa dos resultados enriquecidos. No final, caso algum conjunto de resultados não tenha sido exibido, estes serão impressos por último (pois a variável de enriquecimento correspondente não foi selecionada como prioritária).

5.2.2. Perfil e Localização do Usuário

Para a personalização dos resultados do *peer* fonte e os derivados da reformulação exata, nossa abordagem visou à construção de um perfil para os usuários [Ferraz 2010] e também da captura de suas localizações no momento de submissão das consultas.

Nessa versão da ferramenta [Ferraz 2010], foi feita a especificação do perfil do usuário com algumas informações pessoais e uma informação útil para nosso cenário de estudo de caso, o qual envolve ontologias do domínio de Educação. Essa informação ligada ao nosso domínio é a “Área de Estudo” do usuário (área de estudo da

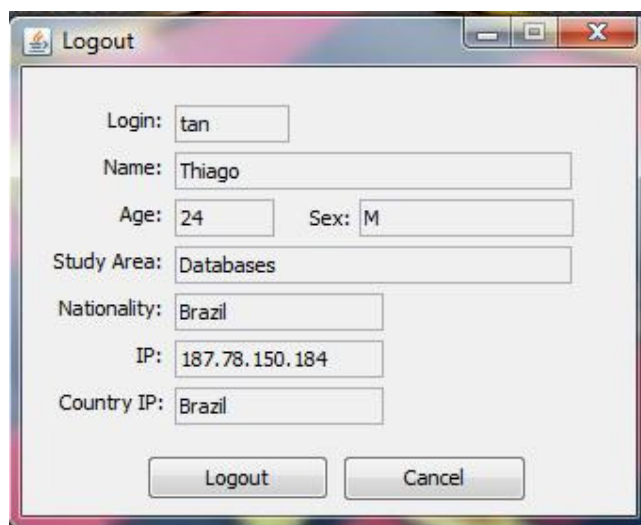
computação) e que iremos citar adiante em nosso trabalho como exemplo prático. Esse perfil não possui uma razão específica para sua definição, servindo como exemplo para os experimentos e testes, auxiliando na comprovação da nossa abordagem. Outros perfis genéricos envolvendo outros domínios semânticos poderiam ter sido definidos, com outros campos sendo utilizados como critérios para a personalização.

O perfil está armazenado em um banco de dados relacional – o HSQLDB (*HyperSQL Database*) [HSQLDB 2010] – escrito em Java e com suporte nativo de conectividade com essa linguagem de programação.

No banco de dados, as informações a respeito do usuário e que caracterizam seu perfil são: Nome, Login, Senha, Idade, Sexo, Nacionalidade, Área de Estudo. As áreas de estudo cadastradas foram: Banco de Dados, Engenharia de Software, Redes e Inteligência Artificial.

Para conseguirmos a informação de localização, implementamos uma captura do endereço IP do usuário no momento de suas consultas. Essa funcionalidade foi desenvolvida com o auxílio de um serviço de GeoIP [MaxMind 2010], o qual informa o país do usuário a partir do seu IP. O critério de país foi utilizado também para validação de nossa abordagem. A localização pode ser capturada sob diferentes granularidades (cidade, bairro, etc.).

No *PSemRef*, o usuário possui uma opção de login na ferramenta, para que seu perfil seja carregado do banco de dados e seu endereço IP seja analisado. A Figura 45 ilustra as informações da sessão de um usuário logado no *PSemRef*.



The image shows a Windows-style dialog box titled "Logout". It contains several text input fields with the following values: Login: tan; Name: Thiago; Age: 24; Sex: M; Study Area: Databases; Nationality: Brazil; IP: 187.78.150.184; Country IP: Brazil. At the bottom, there are two buttons: "Logout" and "Cancel".

Figura 45 – Tela de Sessão de um usuário

Com essas informações do usuário é que o *PSemRef* é capaz de personalizar os resultados exatos das consultas, ou seja, resultados do *peer* fonte e os derivados da

reformulação exata. A justificativa para essa abordagem é fazer um elo das preferências do usuário com sua consulta original de fato, e também da reformulação de equivalência, que busca conceitos com as mesmas características dos conceitos envolvidos na consulta original.

Os resultados de uma consulta submetida no *PSemRef* são divididos em grupos: um grupo com os resultados exatos e do *peer* fonte e os demais grupos com os resultados enriquecidos, divididos de acordo com as variáveis de enriquecimento selecionadas. Vimos na seção anterior o critério para a ordenação dos resultados enriquecidos.

Antes dos resultados enriquecidos, os resultados da consulta original e os exatos são exibidos de forma personalizada. A Figura 46 ilustra uma visão em alto nível do algoritmo desenvolvido para a ordenação dos resultados do *peer* fonte e da reformulação exata.

```
Exact Results Ranking  
Input: ExactResults, UserProperties  
  
PerfectResults; ProfileResults; LocationResults; DifferentResults;  
Score;  
For each result R in ExactResults  
  If Property of R is equal to Property of UserProperties  
    Score += P;  
  If Location of R is equal to Location of UserProperties  
    Score += L;  
End For;  
Test (Score)  
If PL add R to PerfectResults;  
If P add R to ProfileResults;  
If L add R to LocationResults;  
Else add R to DifferentResults;  
Print PerfectResults, ProfileResults, LocationResults, DifferentResults  
  
End Method;
```

Figura 46 – Algoritmo de Ordenação dos Resultados Exatos

Os parâmetros de entrada são o conjunto de resultados exatos e as propriedades do usuário (propriedade do perfil e da localização). Cada resultado é testado para conferir se a propriedade do resultado é igual à propriedade do perfil do usuário e também se a informação de localização do resultado corresponde à localização do usuário. Um escore é atribuído para os resultados (P para perfil correspondente e L para localização correspondente).

Após isso, os resultados são divididos em grupos: (i) resultados com escore PL são chamados de resultados perfeitos (todas as informações iguais); (ii) resultados com escore P são os resultados do Perfil; (iii) resultados com escore L são os resultados de Localização; e (iv) resultados sem escore são os resultados Diferentes. Por fim os resultados são exibidos para o usuário. Podemos notar que os resultados de Perfil são exibidos primeiro do que os resultados de Localização (consideramos a informação do perfil mais importante do que a localização do usuário). Lembrando que nossos experimentos utilizaram como critério o domínio de Educação, com a informação de Perfil utilizada para personalização sendo a Área de Estudo do usuário e sua localização como o País.

Como exemplo prático, podemos supor que o usuário Thiago (vide Figura 45) realiza uma consulta $Q = Student$. No momento de exibição dos primeiros resultados (original e exato), os Estudantes que irão aparecer primeiro são os que satisfizerem os dois critérios para personalização (Banco de Dados e Brasil). Nesse caso, a ordenação dos resultados será: (i) Estudantes de Banco de Dados e do Brasil; (ii) Estudantes de Banco de Dados de outros países; (iii) Estudantes do Brasil de outras áreas; e (iv) Estudantes de outras áreas e de fora do Brasil.

A Figura 47 ilustra o cenário dos resultados personalizados a partir do perfil e localização de três usuários em uma consulta por Professores. Nem todos os conceitos presentes nas ontologias de Educação possuem as propriedades de área de estudo e de país (alguns conceitos possuem apenas uma dessas propriedades). Os conceitos que possuírem uma ou duas dessas propriedades serão ordenados no momento de exibição das respostas, caso contrário, não haverá personalização no resultado.



Figura 47 – Resultados personalizados

Caso a consulta envolva uma disjunção, por exemplo, Thiago submete a consulta $Q = Student \sqcup Event$, serão exibidos os resultados personalizados segundo a ordem dos conceitos da consulta (nesse caso, primeiro os resultados de Estudante e depois os resultados de Evento). O conceito Evento possui apenas a propriedade de país, então o único critério utilizado nesse caso para a personalização dos resultados será a localização do usuário.

Na Seção 5.3 serão descritos alguns experimentos realizados no *PSemRef* envolvendo um exemplo de uso, detalhando o passo-a-passo da execução de algumas consultas e exibindo todo o conjunto de resultados personalizado de acordo com as preferências do usuário.

5.3. Exemplo de Uso

A fim de demonstrarmos nossa abordagem, essa seção tem como objetivo ilustrar o passo-a-passo de utilização e funcionamento do *PSemRef* com alguns exemplos de consultas envolvendo um estudo de caso (exemplo de uso).

Nosso exemplo de uso do *PSemRef* é composto por dois *peers* P_1 e P_2 os quais pertencem ao domínio de Educação. Nesse cenário, os *peers* possuem dados complementares a respeito de pessoas do mundo acadêmico e suas atividades. Cada *peer* é descrito por uma ontologia – O_1 (*Semiprot.owl*) e O_2 (*Univbench.owl*). Consideramos como *background knowledge* a ontologia de domínio (OD) chamada

UnivCSCMO.owl. Fragmentos dessas ontologias estão ilustrados na Figura 48. A estrutura completa de conceitos das ontologias se encontra no Apêndice D.

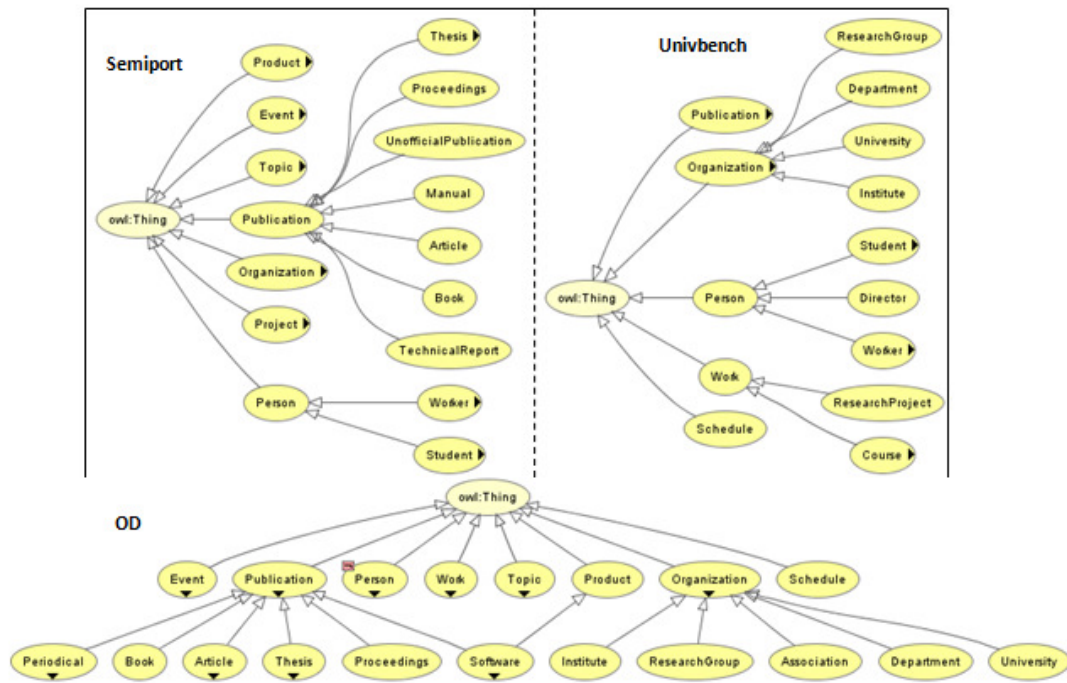


Figura 48 – Ontologias do Estudo de Caso

Para a realização dos testes, foram feitas algumas modificações nos arquivos das ontologias. Primeiramente, propriedades foram adicionadas em alguns conceitos para que estes se tornassem “personalizáveis” em nossas consultas, ou seja, com propriedades que pudessem ser comparáveis à informação do perfil do usuário e a sua localização. Essas propriedades adicionadas foram Área de Estudo e País. A Tabela 3 ilustra as áreas de estudo e países adicionados nas ontologias e os conceitos que possuem tais propriedades (chamados de conceitos personalizados) estão descritos na Tabela 4. Os conceitos sublinhados são aqueles que estão definidos apenas na ontologia O_1 . Os demais conceitos possuem definição em O_1 e em O_2 .

Tabela 3 – Áreas de Estudo e Países

Área de Estudo	País
Databases	Brazil
Software Engineering	USA
Networks	Canada
Artificial Intelligence	France

Tabela 4 – Conceitos Personalizados

Conceito	Propriedade(s)
Student	Área de Estudo e País
Faculty	Área de Estudo e País

<u>Project</u>	Área de Estudo
Publication	Área de Estudo
Organization	País
<u>Topic</u>	Área de Estudo
<u>Event</u>	País

Outra modificação para a viabilização dos experimentos foi a criação de instâncias em O_1 e O_2 para a realização das consultas. Essas instâncias estão ilustradas no Apêndice C. As correspondências semânticas entre O_1 e O_2 são identificadas por um processo de *matching* de ontologias, chamado *SemMatcher* [Pereira 2008].

Ilustramos na Tabela 5 o trecho de correspondências semânticas identificadas entre alguns conceitos personalizados.

Tabela 5 – Correspondências Semânticas entre O_1 e O_2

Conceito em O_1	Tipo de Correspondência	Conceito em O_2
GraduateStudent	isDisjointWith	UndergraduateStudent
GraduateStudent	isPartOf	Course
GraduateStudent	isPartOf	ResearchProject
GraduateStudent	isSuperConceptOf	MasterStudent
GraduateStudent	isSubConceptOf	Student
GraduateStudent	isEquivalentTo	GraduateStudent
University	isCloseTo	College
University	isCloseTo	ResearchGroup
University	isCloseTo	Program
University	isCloseTo	Department
University	isCloseTo	Institute
University	isSubConceptOf	Organization
University	isEquivalentTo	University
AssistantProfessor	isDisjointWith	AssociateProfessor
AssistantProfessor	isDisjointWith	FullProfessor
AssistantProfessor	isCloseTo	VisitingProfessor
AssistantProfessor	isPartOf	Course
AssistantProfessor	isSubConceptOf	Professor

Esse é o cenário de nosso estudo de caso para um experimento da personalização dos resultados do *PSemRef*. Executaremos uma mesma consulta a partir de alguns perfis e com configurações distintas para observarmos a diferença na ordenação das respostas. A Tabela 6 ilustra os perfis de usuário cadastrados em nossa base de dados. Para nosso experimento, faremos uso de três perfis: (i) usuário Thiago; (ii) usuário Mike; e (iii) usuário Jorge, destacados em negrito na Tabela 6.

Tabela 6 – Perfis de Usuário cadastrados

Login	Nome	Área de Estudo	País
tan	Thiago	Databases	Brazil
mk	Mike	Databases	USA
jh	John	Databases	Canada
lg	Ludovic	Databases	France
mb	Maria	Software Engineering	Brazil
kt	Kate	Software Engineering	USA
pm	Paul	Software Engineering	Canada
th	Thierry	Software Engineering	France
jtp	Jorge	Networks	Brazil
ld	Landon	Networks	USA
es	Eleni	Networks	Canada
dd	Didier	Networks	France
bpm	Bruno	Artificial Intelligence	Brazil
oq	Oliver	Artificial Intelligence	USA
kgj	Ken	Artificial Intelligence	Canada
cd	Catherine	Artificial Intelligence	France

A partir de cada perfil destacado da Tabela 6, executamos duas consultas em O_1 : (i) $Q_1 = University$; e (ii) $Q_2 = AssistantProfessor \sqcup GraduateStudent$. A Tabela 5 contém as correspondências semânticas identificadas que envolvem os conceitos utilizados nessas consultas. O primeiro passo para o experimento é cada usuário efetuar o *login* no *PSemRef*. Após o *login*, cada perfil é carregado e seu endereço IP obtido.

A fim de demonstrarmos nossa abordagem por completo, na captura do endereço IP, substituímos a localização dos usuários nesse experimento por seus países cadastrados no perfil.

Consulta Q_1 (University)

Para a realização dessa consulta, as configurações das variáveis de enriquecimento escolhidas por cada usuário estão ilustradas na Tabela 7. O usuário Jorge não selecionou nenhuma variável de enriquecimento.

Tabela 7 – Ordem de prioridade das variáveis de enriquecimento para Q₁

Usuário	Ordem das Variáveis	Usuário	Ordem das Variáveis
Thiago	Approximate	Mike	Generalize
	Generalize		Approximate

A consulta por Universidades é personalizada baseada apenas na localização dos usuários, devido à única propriedade de *University* (país). Sendo assim, o resultado dessa consulta submetida pelos usuários Thiago e Jorge é ordenado da mesma maneira. O conceito *University* possui apenas duas correspondências semânticas (*isCloseTo* e *isSubConceptOf*), por isso a escolha de apenas duas variáveis de enriquecimento para essa consulta, porém numa prioridade diferente.

A Figura 49 ilustra a tela principal do *PSemRef* com a consulta Q₁ submetida pelo usuário Thiago, destacando as configurações e primeiros resultados. Por questões de legibilidade, os resultados detalhados dos usuários estão descritos na Tabela 8.

Observamos a diferença na ordenação dos resultados da consulta Q₁ submetida por Thiago e por Mike, ilustrados na Tabela 8. Antes de cada conjunto de resultados, é mostrada no *PSemRef* a reformulação final (dividida por variável de enriquecimento no caso da reformulação enriquecida) que deu origem às respostas. Essas reformulações estão destacadas com sublinhado na Tabela 8, com seus resultados derivados mostrados em seguida. Nesses resultados, destacamos em negrito as informações que são correspondentes ao perfil e/ou localização do usuário que submeteu a consulta. As telas de logs do *PSemRef* contêm os detalhes a respeito do processo de reformulação das consultas, bem como dos resultados recuperados (vide Seção 5.1.2).

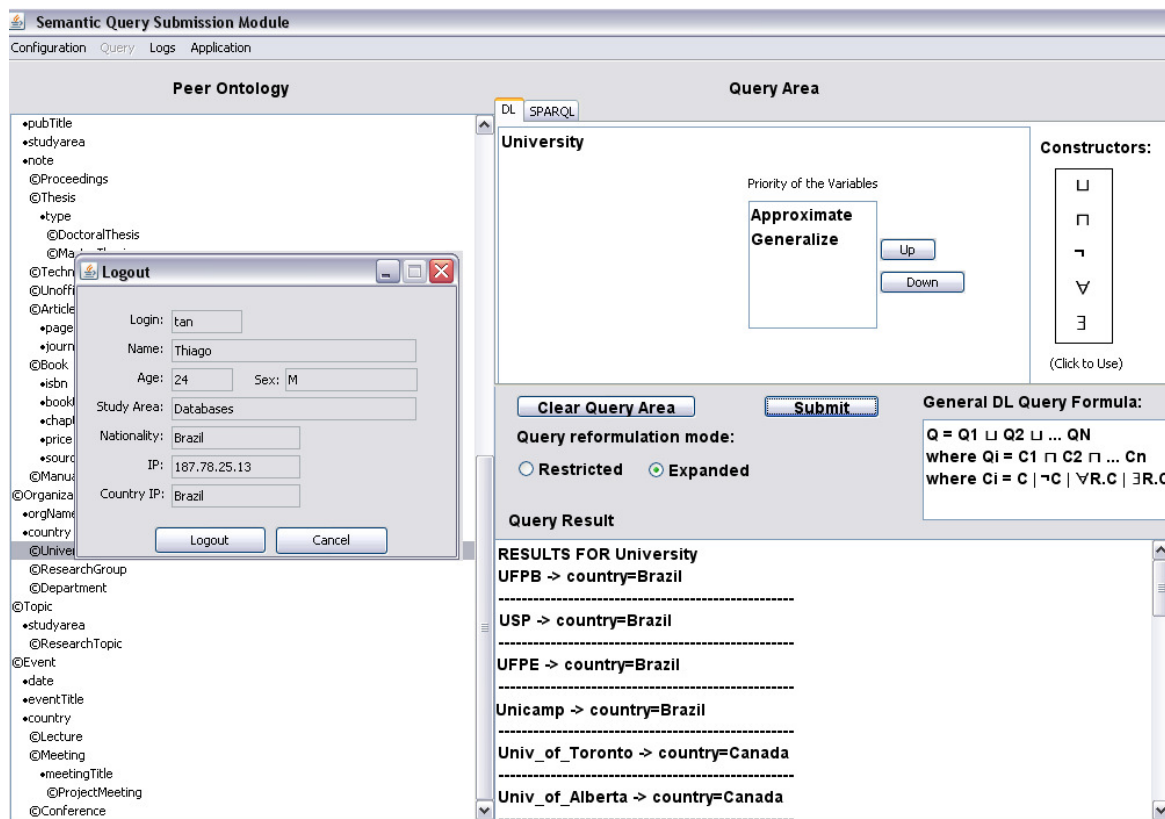


Figura 49 – Consulta Q_1 submetida por Thiago

Resumindo, a ordenação dos resultados das consultas no *PSemRef* segue a seguinte ordem: (i) resultados originados do *peer* fonte e da reformulação exata; e (ii) resultados originados da reformulação enriquecida, divididos por variável de enriquecimento.

Tabela 8 – Resultados para a consulta Q_1

Thiago (Databases, Brazil)	Mike (Databases, USA)
Q_1 original e $Q_{1\text{exact}}$ - University	
UFPB (Brazil)	University of Stanford (USA)
USP (Brazil)	University of Harvard (USA)
UFPE (Brazil)	University of Alberta (Canada)
UNICAMP (Brazil)	University of Toronto (Canada)
University of Toronto (Canada)	UFPE (Brazil)
University of Alberta (Canada)	USP (Brazil)
University of Stanford (USA)	UFPB (Brazil)
University of Harvard (USA)	UNICAMP (Brazil)
$Q_{1\text{enriched}}$ - isCloseTo (College \sqcup ResearchGroup \sqcup Program \sqcup Department \sqcup Institute)	$Q_{1\text{enriched}}$ - isSubConceptOf (Organization)

College_1	Green_ONG
Medicine	<u>Q_{1enriched} - isCloseTo (College ⊔ ResearchGroup ⊔ Program ⊔ Department ⊔ Institute)</u>
Butanta	College_1
<u>Q_{1enriched} - isSubConceptOf (Organization)</u>	Medicine
Green_ONG	Butanta

Podemos notar na tabela a personalização aplicada nos primeiros resultados por conta da localização de cada usuário (o país). Em seguida na Tabela 8, os resultados enriquecidos aparecem numa ordem diferente para cada usuário, devido à configuração de prioridade das variáveis de enriquecimento.

Consulta Q₂ (AssistantProfessor ⊔ GraduateStudent)

Nessa consulta, repetiremos os mesmos procedimentos da consulta Q₁. A diferença aqui é que veremos os resultados para as consultas dos três usuários, os quais também farão configurações diferentes nas variáveis de enriquecimento.

Tanto o conceito *AssistantProfessor* quanto *GraduateStudent* possuem duas propriedades (área de estudo e país), portanto os resultados serão personalizados de acordo com essas duas informações. Na abordagem *PSemRef*, o perfil tem prioridade maior do que localização do usuário, sendo assim, no primeiro conjunto de resultados (*peer* fonte e derivados da reformulação exata), a consulta será personalizada conforme a seguir: (i) perfil e localização; (ii) perfil; e (iii) localização.

Os resultados personalizados, no caso desta disjunção em Q₂, obedecerão a ordem dos conceitos na consulta: (i) resultados de *AssistantProfessor* originados do *peer* fonte e da reformulação exata; (ii) resultados de *GraduateStudent* originados do *peer* fonte e da reformulação exata e (iii) resultados de ambos os conceitos originados da reformulação enriquecida, divididos por variável de enriquecimento.

As configurações das variáveis de enriquecimento escolhidas por cada usuário para a consulta Q₂ estão ilustradas na Tabela 9, mudando a ordem de prioridade sobre as correspondências semânticas *isPartOf*, *isSubConceptOf* e *isCloseTo*.

Tabela 9 – Ordem de prioridade das variáveis de enriquecimento para Q₂

Usuário	Ordem das Variáveis	Usuário	Ordem das Variáveis	Usuário	Ordem das Variáveis
Thiago	Approximate	Mike	Compose	Jorge	Generalize
	Compose		Generalize		Approximate
	Generalize		Approximate		Compose

A Figura 50 ilustra a tela principal do *PSemRef* com a consulta Q₂ submetida pelo usuário Jorge, destacando as configurações e os primeiros resultados para cada conceito da consulta. Novamente por questões de legibilidade, os resultados detalhados dos três usuários para a consulta Q₂ estão descritos na Tabela 10, na qual sublinhamos as consultas executadas e destacamos em negrito nos resultados as informações que são correspondentes ao perfil e/ou localização do usuário que submeteu a consulta.

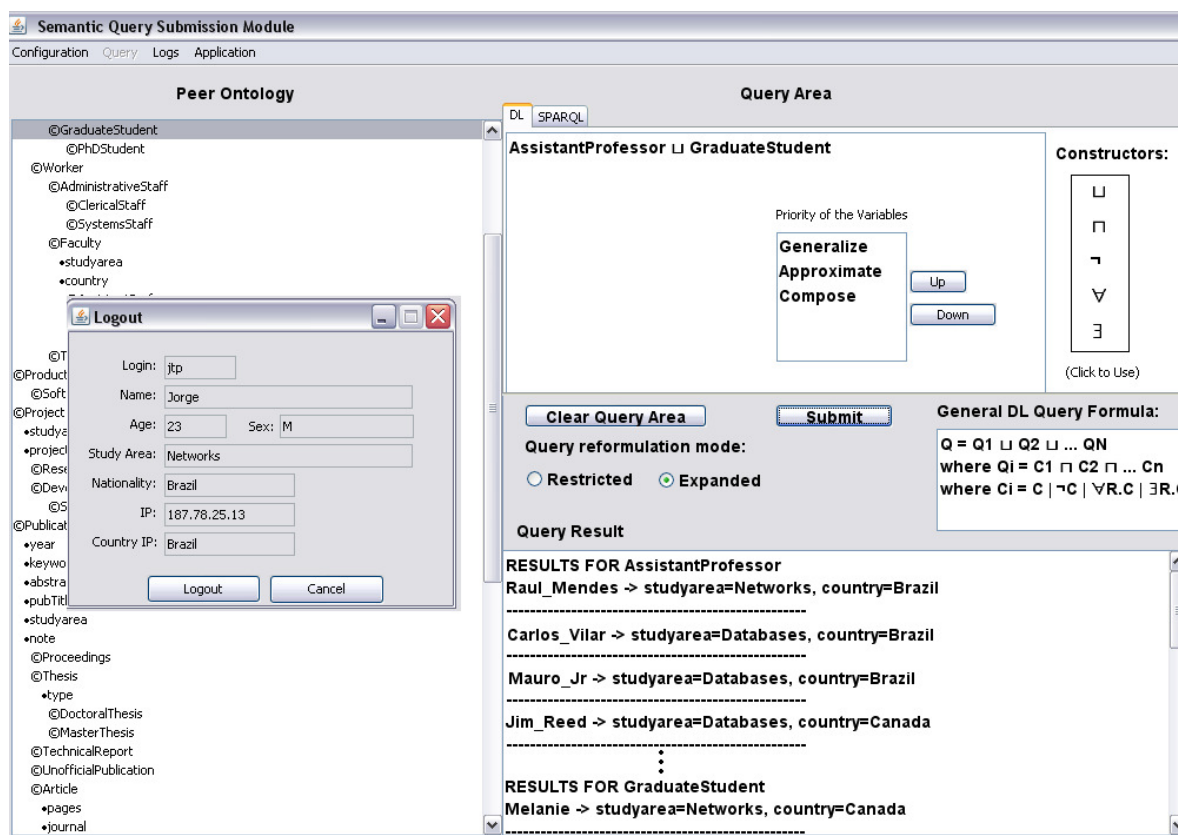


Figura 50 – Consulta Q₂ submetida por Jorge

Tabela 10 – Resultados para a consulta Q₂

Thiago (Databases, Brazil)	Mike (Databases, USA)	Jorge (Networks, Brazil)
<u>Q₂ original e Q_{2exact} - AssistantProfessor</u>		
CarlosV (Databases, Brazil)	JimNoan (Databases, USA)	RaulM (Networks, Brazil)
Mauro_Jr (Databases, Brazil)	JimReed (Databases, Canada)	CarlosV (Databases, Brazil)
JimReed (Databases, Canada)	CarlosV (Databases, Brazil)	Mauro_Jr (Databases, Brazil)
JimNoan (Databases, USA)	Mauro_Jr (Databases, Brazil)	JimReed (Databases, Canada)
RaulM (Networks, Brazil)	Craig (Artificial Intelligence, USA)	JimNoan (Databases, USA)
Craig (Artificial Intelligence, USA)	RaulM (Networks, Brazil)	Craig (Artificial Intelligence, USA)
<u>Q₂ original e Q_{2exact} - GraduateStudent</u>		
Marie (Databases, France)	Marie (Databases, France)	Melanie (Networks, Canada)
Melanie (Networks, Canada)	Marc (Artificial Intelligence, USA)	Brad (Networks, Canada)
Brad (Networks, Canada)	Brad (Networks, Canada)	Marc (Artificial Intelligence, USA)
Marc (Artificial Intelligence, USA)	Melanie (Networks, Canada)	Marie (Databases, France)
<u>Q_{2enriched} – isCloseTo (VisitingProfessor)</u>	<u>Q_{2enriched} – isPartOf (Course ⊔ ResearchProject)</u>	<u>Q_{2enriched} – isSubConceptOf (Professor ⊔ Student)</u>
Jim_Balenger	GIS_Web	Eloisa
<u>Q_{2enriched} – isPartOf (Course ⊔ ResearchProject)</u>	<u>Q_{2enriched} – isSubConceptOf (Professor ⊔ Student)</u>	Louis_P
GIS_Web	Eloisa	<u>Q_{2enriched} – isCloseTo (VisitingProfessor)</u>
<u>Q_{2enriched} – isSubConceptOf (Professor ⊔ Student)</u>	Louis_P	Jim_Balenger
Eloisa	<u>Q_{2enriched} – isCloseTo (VisitingProfessor)</u>	<u>Q_{2enriched} – isPartOf (Course ⊔ ResearchProject)</u>
Louis_P	Jim_Balenger	GIS_Web

Analisando a Tabela 10, podemos ver que os primeiros resultados personalizados são aqueles que estão relacionados por completo ao usuário, ou seja, com mesma área de estudo e país. Em seguida, a informação do perfil é relacionada, depois a localização, e por fim os demais resultados.

O primeiro conjunto de resultados contém a informação de propriedades dos conceitos. Em seguida, nos resultados enriquecidos, as propriedades podem ser diferentes, então o que é levado em consideração são as preferências do usuário no processo de reformulação a partir da escolha de prioridade das variáveis de enriquecimento.

5.4. *Feedback de Usuários*

Para termos alguma forma de validação por usuários, convidamos algumas pessoas que trabalham também no desenvolvimento do SPEED para executarem alguns testes no *PSemRef* por meio da submissão de consultas. Nessa versão de nossa ferramenta executamos esses testes apenas com os usuários desse grupo, por eles já serem familiarizados com os PDMS, e também com o processamento de consultas nesses sistemas.

Como o objetivo central do trabalho é a personalização ou, em outras palavras, adaptação às preferências e necessidades do usuário, é muito importante obter um *feedback* de outros usuários. Nesse sentido, fornecemos aos usuários um roteiro de utilização do *PSemRef* (passo a passo para submissão de consultas) e também exemplos de consultas que pudessem ser submetidas. Esse roteiro explicava como funcionava a seleção e priorização das variáveis de enriquecimento, a escolha do modo de reformulação das consultas e as opções de submissão das consultas.

Os usuários também tiveram total liberdade de executarem quaisquer consultas adicionais que desejassem antes de explanarem suas opiniões e sugestões sobre a ferramenta.

Os usuários deveriam submeter as consultas (do roteiro e as de livre escolha) em três cenários principais: (i) modo restrito; (ii) modo expandido selecionando quaisquer variáveis de enriquecimento desejadas, mas sem priorização; e (iii) modo expandido selecionando quaisquer variáveis de enriquecimento desejadas, com priorização.

Após os testes, pedimos aos usuários um levantamento a respeito do *PSemRef* através de um questionário. Esse questionário envolveu saber a opiniões dos usuários a respeito da utilidade das variáveis de enriquecimento e da sua priorização para as

consultas. Além disso, foi perguntado se os resultados das consultas submetidas refletiam os objetivos dos usuários e se o enriquecimento semântico é útil para a reformulação das consultas. Por fim, foi perguntado se os usuários consideravam a personalização de consultas útil em ambientes dinâmicos e distribuídos.

A grande maioria das respostas e opiniões a respeito dos experimentos e da ferramenta foi positiva. Para todos os usuários, os comentários negativos a respeito do *PSemRef* foram a respeito da usabilidade de sua interface gráfica (questões como posicionamento de componentes, textos de botões, etc.).

A maioria dos usuários considerou a utilização das variáveis de enriquecimento e a definição de uma priorização para elas como algo muito útil para as consultas. Todos informaram que os resultados encontrados em suas consultas foram úteis, principalmente devido ao enriquecimento semântico provido pelo *PSemRef* (resultados mais completos), além da personalização aplicada às consultas (resultados ordenados).

De uma maneira geral, as respostas e opiniões dos usuários demonstraram que a abordagem *PSemRef* mostrou-se útil para o sistema SPEED. Os testes foram bem executados e nenhum problema grave de utilização ou desempenho da ferramenta foi reportado.

5.5. Considerações

Neste capítulo abordamos os aspectos de implementação da nossa abordagem, falando a respeito dos requisitos técnicos e a seguir apresentando todas as interfaces gráficas que constituem o *PSemRef*, comentando sobre cada um de seus componentes em detalhes. As principais telas de nossa ferramenta envolvem a submissão da consulta, a manipulação das variáveis de enriquecimento e um conjunto de logs a respeito da sessão do usuário.

Apresentamos as principais características e algoritmos de nossa abordagem envolvendo o processamento da consulta, dando ênfase ao processo de reformulação. A consulta pode ser formulada em Lógica Descritiva ou em SPARQL, obedecendo a uma fórmula geral ou um conjunto de *templates* presentes nas interfaces. O algoritmo de reformulação gera vários conjuntos de consultas enriquecidas, dependendo das variáveis de enriquecimento selecionadas pelo usuário.

Especificamos aspectos de implementação da personalização das consultas, realizadas a partir do perfil dos usuários e suas localizações no momento de submissão das consultas. A informação do perfil possui prioridade maior na ordenação dos resultados do que a localização do usuário. A partir da escolha de prioridade das

variáveis de enriquecimento, os conjuntos de resultados enriquecidos são organizados num *ranking* para exibição final aos usuários.

Demonstramos a nossa abordagem a partir de um experimento envolvendo um estudo de caso do domínio de Educação com consultas submetidas por três usuários com perfis e localizações distintos. Apresentamos também um *feedback* de usuários reais sobre nossa proposta, o qual se mostrou (junto com os experimentos) muito positivo para a validação da abordagem *PSemRef*.

Os perfis de usuários podem ser definidos para outros domínios semânticos, com a ressalva da dificuldade existente na especificação de um perfil “universal”, ou seja, um perfil que envolva informações para personalização diante de vários domínios semânticos. As abordagens de personalização de consultas apresentadas (Capítulo 2) direcionaram a criação dos perfis a tarefas (domínios) específicas.

O que pode ser feito no sistema SPEED é a criação de perfis categorizados de acordo com o domínio semântico (utilizando como referencial a ontologia de domínio) do *peer* acessado para a submissão de consultas, já que uma consulta não trafega para outras comunidades semânticas. Sendo assim, os critérios de personalização do perfil (semântico) seriam mantidos e utilizados para as consultas submetidas em qualquer *peer*.

O Capítulo 6 apresenta as conclusões deste trabalho, destacando nossas contribuições e indicando trabalhos futuros que podem ser realizados para melhoria de nossa abordagem.

Capítulo 6

Conclusões

Este capítulo tem como objetivo apresentar algumas considerações finais sobre os principais tópicos abordados nesta dissertação, incluindo as contribuições alcançadas e indicações para trabalhos futuros.

6.1. Considerações Finais

Um PDMS é um ambiente no qual há uma grande quantidade e heterogeneidade de fontes de dados, além da alta dinamicidade com que estas fontes se tornam disponíveis. Essas peculiaridades tornam quase impraticável que uma consulta por algum tipo de informação seja precisa. Como resultado, os usuários que fazem consultas em ambientes P2P normalmente se deparam com quantidades massivas de resultados, muitos destes irrelevantes. Considerando isso, os usuários, ao acessarem um PDMS, deveriam ser capazes de especificar suas preferências ao realizarem consultas no sistema.

Este trabalho foi desenvolvido no SPEED, um PDMS baseado em semântica o qual adota uma arquitetura de *super-peers*. Neste ambiente, os *peers* de integração são responsáveis pelo encaminhamento de uma consulta aos *peers* de dados em seu *cluster*, assim como para outros *peers* de integração.

Uma consulta, ao ser submetida e executada em um ponto origem, também é reformulada sob os termos do esquema de um ponto destino – representado por ontologias – a partir de um conjunto de correspondências semânticas existentes entre os elementos dos esquemas. A consulta submetida pode gerar dois tipos de reformulação: (i) exata e (ii) enriquecida.

O nosso trabalho envolveu o processamento de consultas no SPEED entre dois pontos de integração – embora possa ser estendido para um cenário mais amplo. Mais especificamente, nossa abordagem recebeu o nome de *PSemRef* e visou à personalização de consultas submetidas no SPEED, tanto no âmbito da reformulação exata quanto da enriquecida.

Os resultados derivados do *peer* origem e da reformulação exata foram ordenados a partir de um perfil do usuário, além de sua localização no momento de

realização da consulta. O usuário, ao definir uma ordem de prioridade para as variáveis de enriquecimento, especifica a ordem de exibição dos resultados derivados da reformulação enriquecida. Nossa abordagem foi especificada em detalhes tanto no aspecto conceitual quanto de implementação.

Apresentamos alguns trabalhos que estão sendo realizados envolvendo personalização de consultas e também de reformulação de consultas em alguns PDMS. Comparando estes trabalhos com o nosso, em nossa abordagem nós aplicamos a personalização de consultas em um ambiente P2P. Mais ainda, levamos em consideração as preferências dos usuários no momento da reformulação da consulta, além de no momento da exibição das respostas em si, provendo estes usuários com um conjunto ordenado de resultados de acordo com seu perfil e preferências.

Abordamos o problema da personalização de consultas no SPEED e especificamos nossa proposta para solucioná-lo. Discutimos também os aspectos de implementação de nossa abordagem, além de exibirmos um estudo de caso para utilização do *PSemRef* e um *feedback* de usuários reais sobre nosso trabalho. Nesta versão, criamos um perfil de usuário direcionado ao nosso estudo de caso, o qual envolveu o domínio de Educação.

6.2. Contribuições

As contribuições deste trabalho são descritas a seguir.

- Utilização do perfil do usuário para personalização das consultas;
- Captura de uma informação contextual estática (perfil do usuário) e uma informação contextual dinâmica (localização do usuário) para personalização das consultas;
- *Ranking* dos resultados das consultas, baseado nas informações contextuais capturadas e nas preferências do usuário;
- Escolha das preferências dos usuários na seleção e ordem de prioridade das variáveis de enriquecimento para a reformulação das consultas;
- Especificação e implementação do módulo *PSemRef*, além de testes e experimentos realizados com a ferramenta.

6.3. Trabalhos Futuros

Consideramos que nosso trabalho atingiu o objetivo proposto. Porém, a abordagem *PSemRef* merece algumas extensões e melhorias na forma de trabalhos futuros, os quais são especificados a seguir.

- **Otimização da Reformulação das Consultas:** o algoritmo de reformulação das consultas pode gerar algumas imperfeições, pois quando ocorre a reformulação de um conceito, este pode ser substituído por um conjunto de conceitos utilizando diferentes correspondências semânticas. Como resultado, esses conceitos que surgem podem estar em conflito ou serem redundantes na consulta reformulada;
- **Personalização de Propriedades:** no *PSemRef*, uma consulta com propriedades apenas é possível de uma única maneira, bastante limitada, sem personalização. Pretendemos ampliar as possibilidades de consultas com propriedades, permitindo dessa maneira a aplicação da personalização nesse tipo de consultas;
- **Ampliação de Perfis:** pretendemos em trabalhos futuros utilizar perfis mais amplos com informações variadas que possam ser utilizados de uma maneira satisfatória em consultas no SPEED envolvendo outros tipos de domínio. Pretendemos dividir o conjunto de perfis de acordo com os domínios semânticos existentes no SPEED (utilizando como referencial a ontologia de domínio);
- **Melhorias da Interface Gráfica:** muitas observações feitas pelos usuários durante os testes de consultas envolveram questões de interface do *PSemRef*. Pretendemos realizar uma série de melhorias para deixar a interface mais amigável, deixando dessa maneira a ferramenta de consultas do SPEED com uma usabilidade mais agradável em todos os aspectos.

Referências Bibliográficas

[Abbar et al. 2009] Abbar S., Bouzeghoub M., Lopes S. (2009): Context-Aware Recommender Systems: A Service-Oriented Approach. In: VLDB'09, Lyon, France.

[Adjiman et al. 2006] Adjiman P., Chatalic P., Goasdoué F., Rousset M.C., Simon L. (2006): Distributed reasoning in a P2P setting: Application to the semantic web. Journal of Artificial Intelligence Research (JAIR).

[Adjiman et al. 2007] Adjiman P., Goasdoué F., Rousset M.C. (2007): SomeRDFS in the Semantic Web. In: Journal on Data Semantics, LNCS, vol. 8, p. 158-181.

[Agrawal et al. 2000] Agrawal R., Wimmers E. (2000): A Framework for Expressing and Combining Preferences. In: Proceedings of the ACM SIGMOD International Conference Management of Data (SIGMOD'00), New York, USA, p. 297-306.

[Amazon 2009] Amazon, <http://www.amazon.com/>, Novembro de 2009.

[Baader et al. 2003] Baader F., Calvanese D., McGuinness D., Nardi D., Patel-Schneider P. (2003): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.

[Broekstra et al. 2003] Broekstra J., Kampman A. (2003): SeRQL: A Second Generation RDF Query Language. In: Proceedings of the SWAD-Europe Workshop on Semantic Web Storage and Retrieval. Vrije Universiteit, Amsterdam, Netherlands.

[Campi et al. 2009] Campi A., Mazuran M., Ronchi S. (2009): Domain Level Personalization Technique. In: Proceedings of the 3rd International Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB 2009), Lyon, France.

[Castano et al. 2003] Castano S., Ferrara A., Montanelli S., Zucchelli D. (2003): HELIOS: a General Framework for Ontology-based Knowledge Sharing and Evolution in P2P Systems. In: Proceedings of DEXA'03 2nd Web Semantics Workshop, Prague, Czech Republic.

[Castano et al. 2005] Castano S., Montanelli S. (2005): Semantic Self-Formation of Communities of Peers. In: Proceedings of the ESWC Workshop on Ontologies in Peer-to-Peer Communities, Heraklion, Greece.

[Dey et al. 2000] Dey A. K., Abowd G. D. (2000): Towards a Better Understanding of Context and Context-Awareness. In: Proceedings of the CHI Workshop on the What, Who, Where, When and How of Context-Awareness, The Hague, Netherlands.

[Eclipse 2010] Eclipse IDE, <http://www.eclipse.com/>, Janeiro de 2010.

[Ferraz 2010] Ferraz G. (2010): Preferências do Usuário para Personalização de Consultas. Monografia de Conclusão de Curso. Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

[Ghidini et al. 2006] Ghidini C., Serafini L. (2006): Reconciling Concepts and Relations in Heterogeneous Ontologies. ESWC 2006, p. 50-64.

[Giannopoulos et al. 2009] Giannopoulos G., Dalamagas T., Sellis T. (2009): Collaborative Ranking Function for Web Search Personalization. In: Proceedings of the 3rd International Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB 2009), Lyon, France.

[Herschel, Heese 2005] Herschel S., Heese R. (2005): Humboldt Discoverer: A semantic P2P index for PDMS. In: Proceedings of the International Workshop Data Integration and the Semantic Web, Porto, Portugal.

[HSQLDB 2010] HyperSQL, <http://hsqldb.org/>, Abril de 2010.

[Java 2010] Java, <http://www.java.com/>, Janeiro de 2010.

[Java API 2010] Java Platform API Specification, <http://www.java.sun.com/javase/6/docs/api>, Janeiro de 2010.

[Jena 2010] Jena Framework, <http://jena.sourceforge.net/javadoc/index.html>, Janeiro de 2010.

[Kazaa 2007] Kazaa (2007), <http://www.kazaa.com/>, Outubro de 2009.

[Kostadinov et al. 2007] Kostadinov, D., Bouzeghoub M., Lopes S. (2007): Query Rewriting based on User's Profile Knowledge. In: Actes des 23èmes Journées Bases de Données Avancées, Marseille, France.

[Koutrika et al. 2004a] Koutrika G., Ioannidis Y. (2004): Personalization of Queries in Database Systems. In: Proceedings of the 20th International Conference on Data Engineering (ICDE'04), p. 597-608, Boston, Massachusetts, USA.

[Koutrika et al. 2004b] Koutrika G., Ioannidis Y. (2004): Personalization of Queries based on User Preferences. In: Proceedings of Preferences: Specification, Inference, Applications (Preferences 2004), Schloss Dagstuhl, Germany.

[Koutrika 2006] Koutrika, G. (2006): Personalization of Structured Queries with Personal and Collaborative Preferences. In: ECAI Workshop about Advances on Preference Handling, Riva del Garda, Italy.

[Mandreoli et al. 2008] Mandreoli F., Martoglia R., Sassatelli S., Villani G. (2008): Building a XML Data Sharing with SUNRISE. In: 3rd International Workshop on Database Technologies for Handling XML Information on the Web (DataX'08), Nantes, France.

[MaxMind 2010] MaxMind – GeoIP, <http://www.maxmind.com/app/ip-location>, Abril de 2010.

[Napster 2007] Napster (2007), <http://free.napster.com/>, Outubro de 2009.

[Neves 2008] Neves T. (2008): Desenvolvimento do Módulo de Reformulação de Consultas no Sistema SPEED. Monografia de Conclusão de Curso. Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

[Noy et al. 2001] Noy N. F., McGuinness D. L. (2001): Ontology development 101: A guide to creating your first ontology. Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory.

[OWL 2004] OWL Web Ontology Language (2004), <http://www.w3.org/TR/owl-features/>, Fevereiro de 2010.

[Pereira 2008] Pereira T. (2008): Mapeamento Semântico de Ontologias no SPEED. Monografia de Conclusão de Curso. Universidade Federal de Pernambuco (UFPE), Recife, PE, Brasil.

[Pires 2009] Pires C. (2009): Ontology-based Clustering in a Peer Data Management System. PhD Thesis, Federal University of Pernambuco (UFPE), Recife, PE, Brazil.

[Priberam 2009] Dicionário Priberam da Língua Portuguesa, <http://www.priberam.pt/DLPO/>, Novembro de 2009.

[Protégé API 2010] Protégé-OWL API, <http://protege.stanford.edu/plugins/owl/api/>, Janeiro de 2010.

[RDQL 2004] RDQL – A Query Language for RDF (2004), <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109>, Novembro de 2009.

[Sendhikumar et al. 2008] Sendhikumar S., Geetha T. V. (2008): Personalized Ontology for Web Search Personalization. In: Proceedings of the First Bangalore Annual Compute Conference (Compute'08), p. 1-7, Bangalore, India.

[Souza 2009] Souza D. (2009): Using Semantics to Enhance Query Reformulation in Dynamic Distributed Environments. PhD Thesis, Federal University of Pernambuco (UFPE), Recife, PE, Brazil.

[Souza et al. 2009a] Souza D., Neves T., Salgado A. C., Tedesco P., Kedad Z. (2009): Using Semantics to Enhance Query Reformulation in Dynamic Environments. In: 13th East European Conference on Advances in Databases and Information Systems (ADBIS'09), p. 78-92, Riga, Latvia.

[Souza et al. 2009b] Souza D., Neves T., Salgado A. C., Tedesco P. (2009): SemRef: A Semantic-based Query Reformulation Tool for Dynamic Environments. In: 24th Brazilian Symposium on Databases (SBBD'09), p.7-12, Fortaleza, Brazil.

[Souza et al. 2009c] Souza D., Neves T., Salgado A. C., Tedesco P., Kedad Z. (2009): Semantic-based Query Reformulation in Dynamic Environments. In: 25e Journées Bases de Données Avancées (BDA'09), Namur, Belgique.

[SPARQL 2008] SPARQL Query Language for RDF (2008), <http://www.w3.org/TR/rdf-sparql-query/>, Fevereiro de 2010.

[Stefanidis et al. 2005] Stefanidis K., Pitoura E., Vassiliadis P. (2005): On Supporting Context-Aware Preferences in Relational Database Systems. In: Proceedings of the 1st International Workshop on Managing Context Information in Mobile and Pervasive Environments, in conjunction with MDM 2005, Cyprus.

[Stefanidis et al. 2007a] Stefanidis K., Pitoura E., Vassiliadis P. (2007): Adding Context to Preferences. In: Proceedings of the 23rd International Conference on Data Engineering (ICDE'07), p. 846-855, Istanbul, Turkey.

[Stefanidis et al. 2007b] Stefanidis K., Pitoura E., Vassiliadis P. (2007): A Context-Aware Preference Database System. International Journal of Pervasive Computing and Communications, vol. 3, n. 4, p. 439-460.

[Stefanidis et al. 2009] Stefanidis K., Drosou M., Pitoura E. (2009): "You May Also Like" Results in Relational Databases. In: Proceedings of the 3rd International Workshop on Personalized Access, Profile Management and Context Awareness in Databases (PersDB 2009), Lyon, France.

[Stuckenschmidt et al. 2005] Stuckenschmidt H., Giunchiglia F., van Harmelen F. (2005): Query processing in ontology-based peer-to-peer systems. In: V. Tamma, S. Crane, T. Finin, and S. Willmott, editors, Ontologies for Agents: Theory and Experiences. Birkhuser.

[Sung et al. 2005] Sung L. G. A., Ahmed N., Blanco R., Li H., Soliman M. A., Hadaller D. (2005): A Survey of Data Management in Peer-to-Peer Systems. School of Computer Science, University of Waterloo.

[Tatarinov, Halevy 2004] Tatarinov I., Halevy A. (2004): Efficient Query Reformulation in Peer Data Management Systems. In: Proceedings of the SIGMOD International Conference Management of Data (SIGMOD'04), p. 539-550, Maison de la Chimie, Paris, France.

[Yang et al. 2003] Yang B., H. Garcia-Molina. (2003): Designing a Super-Peer Network. In: Proceedings of the 19th International Conference on Data Engineering (ICDE'03), Bangalore, India.

[Ziegler et al. 2008] Ziegler P., Dittrich K., Hunt E. (2008): A Call for Semantic Data Integration. In: Proceedings of the IEEE 24th International Conference on Data Engineering Workshop (ICDEW'08).

Apêndice A

Algoritmo de Reformulação

Neste Apêndice ilustramos o algoritmo de reformulação completo do nosso trabalho [Souza 2009].

Reformulation Algorithm of PsemRef

```
For each  $Q_k$  in  $Q$  /* for each conjunctive query in  $Q$  */
   $B \leftarrow \text{TRUE}$  /*  $B$  will be used to stop the search if some concept of  $Q_k$  has no
    correspondent concept in  $P_2$  */
  While (there is still a concept  $C_j$  in  $Q_k$  to process) and ( $B = \text{TRUE}$ )
     $S_1C_j \leftarrow \emptyset$  /* set of concepts that are equivalent to  $C_j$  */
     $S_2C_j \leftarrow \emptyset$  /* set of concepts related to  $C_j$  by other kind of correspondence,
      except disjointness */
     $\text{Neg\_}S_2C_j \leftarrow \emptyset$  /* set of concepts related to  $C_j$  by disjointness correspondence */
    For each equivalence assertion between  $C_j$  and a concept  $C'$  /*  $C'$  is in  $O_2$  */
      Add  $C'$  to  $S_1C_j$ 
    End For;
    For each other kind of assertion involving  $C_j$  /* different from equivalence */
      If  $\text{SPECIALIZE} = \text{TRUE}$ 
        Then
          If there is a concept  $C'$  in  $P'$  such that  $C' \sqsubseteq C_j$  /* subConceptOf */
            Then
              Add  $C'$  to  $S_2C_j$ 
            End If;
          End If;
        End If;
      If  $\text{APPROXIMATE} = \text{TRUE}$ 
        Then
          If there is a concept  $C'$  in  $P'$  such that  $C' \approx C_j$  /* closeTo */
            Then
              Add  $C'$  to  $S_2C_j$ 
            End If;
          End If;
      If  $\text{GENERALIZE} = \text{TRUE}$ 
        Then
          If there is a concept  $C'$  in  $P'$  such that  $C' \sqsupseteq C_j$  /* superConceptOf */
            Then
              Add  $C'$  to  $S_2C_j$ 
            End If;
          End If;
      If  $\text{COMPOSE} = \text{TRUE}$ 
        Then
          If there is a concept  $C'$  in  $P'$  such that  $C' \triangleright C_j$  or  $C' \triangleleft C_j$  /* related through
            a part of or a whole of correspondence */
            Then
              Add  $C'$  to  $S_2C_j$ 
            End If;
          End If;
        End If;
      End If;
    End If;
  End For;
```

```

        If Cj is negated
        Then
            If there is a concept C' in P' such that C' ⊥ Cj /* they are disjoint */
            Then
                Add C' to Neg_S2Cj
                BNeg ← TRUE
            End If;
        End If;

    End If;
End For; /* End of the loop related to the assertions different from ≡ */

If (S1Cj = ∅ and S2Cj = ∅ and Neg_S2Cj = ∅)
Then
    B ← FALSE /* there is no correspondence between Cj and concepts of P2 */
End If;
End While; /* End of the loop processing concepts */

B1 ← TRUE;
If any S1Cj = ∅
Then B1 ← FALSE
End If; /* Checking if there was an empty set concerning exact correspondences */

B2 ← TRUE;
If any S2Cj = ∅
Then B2 ← FALSE

If B2 = TRUE or BNeg = TRUE
/* if there were enriching correspondences and no resulting empty set, then we can build the
enriched reformulation for the current Q */
Then
    Qk_enriched ← Build_Enriched_Reformulation (Qk, S2C1, ... S2Cp, Neg_S2C1, ... Neg_S2Cp)
Else
    Qk_enriched ← ∅
End If;

End For; /* End of the loop processing the conjunctive queries Qk */

If (at least one of Qk_exact ≠ ∅) /* at least one of Qk's exact reformulations is not empty */
Then
    Q_exact ← Build_Final_Exact_Reformulation (Q, Q1_exact, ..., Qm_exact)
Else
    Q_exact ← ∅
End If;

If ((MODE is expanded) or (MODE is restricted and Q_exact is empty)) and
(at least one of Qk_enriched ≠ ∅)

/* If MODE is expanded or MODE is restricted and Q_exact is empty; and
at least one of Qk's enriched reformulations is not empty */

Then
    Q_enriched ← Build_Final_Enriched_Reformulation (Q, Q1_enriched, ..., Qm_enriched)
Else
    Q_enriched ← ∅
End If;

End;

```

Figura A.1 – Algoritmo de Reformulação [Souza 2009]

Apêndice B

Métodos Complementares da Reformulação

Neste Apêndice ilustramos os métodos complementares em pseudocódigo que auxiliam no processo de reformulação do *PSemRef* [Souza 2009].

```
Build_Exact_Reformulation ( $Q_k, S_1C_1, \dots, S_1C_p$ )  
 $Q'_k \leftarrow \emptyset$   
For each  $S_1C_i$   
   $e_i \leftarrow \emptyset$   
  For each  $C$  in  $S_1C_i$   
    If  $C_i$  is negated in  $Q_k$   
      Then  $e_i \leftarrow e_i \sqcup \neg C$   
    Else  $e_i \leftarrow e_i \sqcup C$   
  End If;  
End For;  
 $Q'_k \leftarrow Q'_k \cap e_i$   
End For;  
return ( $Q'_k$ )  
End_Build_Exact_Reformulation
```

Figura B.1 – Método *Build_Exact_Reformulation* [Souza 2009]

```
Build_Enriched_Reformulation ( $Q_k, S_2C_1, \dots, S_2C_p, \text{Neg\_}S_2C_1, \dots, \text{Neg\_}S_2C_p$ )  
 $Q'_k \leftarrow \emptyset$   
For each  $i$   
   $e_i \leftarrow \emptyset$   
  For each  $C$  in  $S_2C_i$   
    If  $C_i$  is negated in  $Q_k$   
      Then  $e_i \leftarrow e_i \sqcup \neg C$   
    Else  $e_i \leftarrow e_i \sqcup C$   
  End If;  
End For;  
For each  $C$  in  $\text{Neg\_}S_2C_i$   
   $e_i \leftarrow e_i \sqcup C$   
End For;  
  
   $Q'_k \leftarrow Q'_k \cap e_i$   
End For;  
returns ( $Q'_k$ )  
End_Build_Enriched_Reformulation
```

Figura B.2 – Método *Build_Enriched_Reformulation* [Souza 2009]

Build_Final_Exact_Reformulation (Q, Q_{1_exact}, ..., Q_{m_exact})

Q'_exact ← ∅

For each Q_{i_exact}

 Q'_exact ← Q'_exact ∪ Q_{i_exact}

End For

Return(Q'_exact)

End Build_Final_Exact_Reformulation;

Figura B.3 – Método *Build_Final_Exact_Reformulation* [Souza 2009]

Build_Final_Enriched_Reformulation (Q, Q_{1_enriched}, ..., Q_{m_enriched})

Q'_enriched ← ∅

For each Q_{i_enriched}

 Q'_enriched ← Q'_enriched ∪ Q_{i_enriched}

End For

Return(Q'_enriched)

End Build_Final_Enriched_Reformulation;

Figura B.4 – Método *Build_Final_Enriched_Reformulation* [Souza 2009]

Apêndice C

Instâncias Cadastradas para o Estudo de Caso

Neste Apêndice, ilustramos as instâncias cadastradas nas ontologias do nosso estudo de caso, envolvendo o domínio de Educação. As ontologias envolvidas foram a *Semiport.owl* e a *Univbench.owl*. A ilustração está organizada em tabelas com as colunas ordenadas da seguinte maneira: (i) Conceito; (ii) Identificador; (iii) País relacionado; e (iv) Área de Estudo relacionada. Nem todas as instâncias possuem ambos os dados (iii) e (iv) preenchidos.

Tabela C.1 – Instâncias cadastradas em *Semiport*

Conceito	Identificador	País	Área de Estudo
Conference	conferencefr2	France	
Conference	conferencefr1	France	
Conference	conferencecan2	Canada	
Conference	conferencecan1	Canada	
Conference	conferenceusa2	USA	
Conference	conferenceusa1	USA	
Conference	SBES	Brazil	
Conference	SBBD	Brazil	
Event	Evento2	Brazil	
Event	Evento1	Brazil	
Topic	topicai2		Artificial Intelligence
Topic	topicai1		Artificial Intelligence
Topic	topicnt2		Networks
Topic	topicnt1		Networks
Topic	topicdb2		Databases
Topic	topicdb1		Databases
Topic	topicsw2		Software Engineering
Topic	topicsw1		Software Engineering
University	Univ of Versailles	France	
University	Univ of Paris	France	
University	Univ of Toronto	Canada	
University	Univ of Alberta	Canada	
University	Univ of Stanford	USA	
University	Univ of Harvard	USA	
University	UFPB	Brazil	
University	USP	Brazil	

University	UFPE	Brazil	
Publication	publicationsw2		Software Engineering
Publication	publicationsw1		Software Engineering
Publication	publicationia1		Artificial Intelligence
Publication	publicationia		Artificial Intelligence
Publication	publicationnt2		Networks
Publication	publicationnt1		Networks
Publication	publicationdb2		Databases
Publication	publicationdb1		Databases
TechnicalReport	TechnicalReport_39		Databases
TechnicalReport	TechnicalReport_37		Artificial Intelligence
Publication	publicationdb		Databases
Article	Article_26		Databases
Publication	publicationnt		Networks
MasterThesis	MasterThesis_42		Networks
Article	Article_27		Software Engineering
Article	Article_29		Software Engineering
Project	Neural Networks		Artificial Intelligence
Project	Robots		Artificial Intelligence
Project	MySQL1		Databases
Project	Oracle1		Databases
Project	Requisitos1		Software Engineering
Project	ToolCase1		Software Engineering
Project	PDMS1		Networks
Project	Wireless1		Networks
AssistantProfessor	Jim_Reed	Canada	Databases
AssistantProfessor	Tim_Bell	Canada	Databases
FullProfessor	John_Queue	Canada	Software Engineering
Lecturer	Cristina_Holmes	Canada	Artificial Intelligence
FullProfessor	Mark_Holt	Canada	Software Engineering
FullProfessor	Tsonga	France	Networks
FullProfessor	Lilly	France	Databases
FullProfessor	Cristina_Hilton	France	Software Engineering
FullProfessor	Ulisses	Brazil	Software Engineering
AssistantProfessor	Mauro_Jr	Brazil	Databases
AssistantProfessor	Carlos_Vilar	Brazil	Databases
AssistantProfessor	Raul_Mendes	Brazil	Networks
FullProfessor	Davies	USA	Networks
AssistantProfessor	Jeremy	USA	Artificial Intelligence
AssistantProfessor	Craig	USA	Artificial Intelligence
AssistantProfessor	JimNoan	USA	Databases
Student	Paul	USA	Networks
Student	John	Canada	Artificial Intelligence
UndergratuateStudent	Joanna	France	Software Engineering
GraduateStudent	Marie	France	Databases
GraduateStudent	Melanie	Canada	Networks

GraduateStudent	Marc	USA	Artificial Intelligence
UndergraduateStudent	Jairo	Brazil	Networks
UndergraduateStudent	Ana	Brazil	Software Engineering
PhDStudent	Joey	USA	Databases
PhDStudent	Ross	Canada	Networks
Student	Carla	Brazil	Databases

Tabela C.2 – Instâncias cadastradas em *Univbench*

Conceito	Identificador	País	Área de Estudo
University	Univ of Nice	France	
University	Univ of Lyonnais	France	
University	Univ of Quebec	Canada	
University	Univ of Montreal	Canada	
University	Univ of Princeton	USA	
University	Univ of NY	USA	
University	UFRJ	Brazil	
University	Unicamp	Brazil	
University	UFPI	Brazil	
Publication	publicationsw3		Software Engineering
Publication	publicationsw4		Software Engineering
Publication	publicationia3		Artificial Intelligence
Publication	publicationia4		Artificial Intelligence
Publication	publicationnt3		Networks
Publication	publicationnt4		Networks
Publication	publicationdb3		Databases
Publication	publicationdb4		Databases
TechnicalReport	TechnicalReport_90		Databases
TechnicalReport	TechnicalReport_80		Artificial Intelligence
Publication	publicationdb12		Databases
Article	Article_265		Databases
Publication	publicationnt122		Networks
TechnicalReport	TechnicalReport_390		Networks
Article	Article_2704		Software Engineering
Article	Article_2977		Software Engineering
FullProfessor	John_Kent	Canada	Databases
FullProfessor	Lex_L	Canada	Databases
FullProfessor	Adam_S	Canada	Software Engineering
FullProfessor	July_TR	Canada	Artificial Intelligence
FullProfessor	Polly_H	Canada	Software Engineering
FullProfessor	Lloriz	France	Networks
FullProfessor	Toulalan	France	Databases
FullProfessor	Mauresmo	France	Software Engineering
FullProfessor	Ubiratan	Brazil	Software Engineering
FullProfessor	Teo_J	Brazil	Databases
FullProfessor	Robson_Abrantes	Brazil	Databases

FullProfessor	Rodrigo_Ilo	Brazil	Networks
FullProfessor	Danny_S	USA	Networks
FullProfessor	Jordie	USA	Artificial Intelligence
FullProfessor	Nathan	USA	Artificial Intelligence
Student	Flen_Green	USA	Networks
Student	Louis_P	Canada	Artificial Intelligence
UndergraduateStudent	Yasmine	France	Software Engineering
GraduateStudent	Fabienne	France	Databases
GraduateStudent	Brad	Canada	Networks
GraduateStudent	Patrick	USA	Artificial Intelligence
UndergraduateStudent	Nilo	Brazil	Networks
UndergraduateStudent	Tatiana	Brazil	Software Engineering
Student	Eloisa	Brazil	Databases

Apêndice D

Ontologias de Educação

Neste Apêndice, ilustramos a estrutura completa das ontologias do domínio de educação utilizadas em nosso estudo de caso do Capítulo 5.



Figura D.1 – Ontologia Semiport.owl



Figura D.2 – Ontologia Univbench.owl

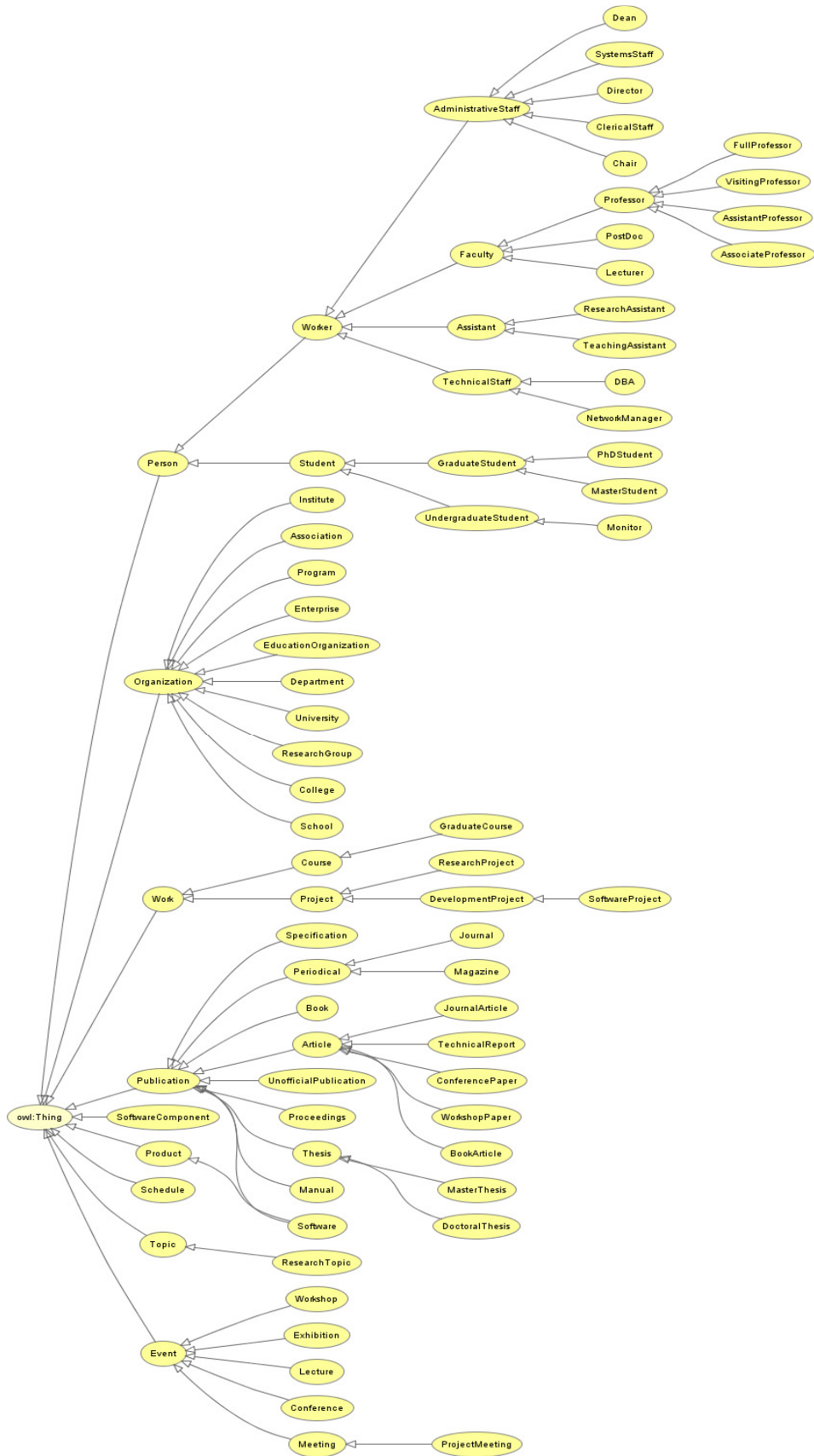


Figura D.3 – Ontologia de Domínio UnivCSCMO.owl