

**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
SECRETARIA DA CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO**

GABRIEL ANDRÉ DUQUESNOIS DUBOIS BRITO

INTEGRAÇÃO DE OBJETOS DE APRENDIZAGEM NO SISTEMA ROSA - P2P

**Rio de Janeiro
2005**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

INSTITUTO MILITAR DE ENGENHARIA

GABRIEL ANDRÉ DUQUESNOIS DUBOIS BRITO

**INTEGRAÇÃO DE OBJETOS DE APRENDIZAGEM
NO SISTEMA ROSA - P2P**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientadora : Ana Maria de Carvalho Moura - Dr. Ing.

Rio de Janeiro
2005

c2005

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro - RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

005.758 Brito, Gabriel André Duquesnois Dubois

B862i Integração de Objetos de Aprendizagem no Sistema ROSA - P2P / Gabriel André Duquesnois Dubois Brito - Rio de Janeiro : Instituto Militar de Engenharia, 2005.

188p. : il., tab.

Dissertação (mestrado) - Instituto Militar de Engenharia – Rio de Janeiro, 2005.

1. Sistemas Distribuídos (P2P). 2. Integração de Dados. 3. WEB Semântica. 4. Banco de Dados Distribuídos. I. Título. II. Instituto Militar de Engenharia.

CDD 005.758

INSTITUTO MILITAR DE ENGENHARIA

GABRIEL ANDRÉ DUQUESNOIS DUBOIS BRITO

**INTEGRAÇÃO DE OBJETOS DE APRENDIZAGEM NO SISTEMA
ROSA - P2P.**

Dissertação de Mestrado apresentada no Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Mestre em Ciências em Sistemas e Computação.

Orientador: Ana Maria de Carvalho Moura - Dr. Ing.

Aprovada em 09 de junho de 2005 pela seguinte Banca Examinadora:

Prof^ª. Ana Maria de Carvalho Moura - Dr. Ing. – Presidente

Prof. Paulo de Figueiredo Pires, D. C. - UNI-RIO

Prof^ª. Maria Claudia Reis Cavalcanti, D. C. – IME

Prof. Jauvane Cavalcante de Oliveira, PhD – LNCC

Rio de Janeiro

2005

AGRADECIMENTOS

Agradeço ao Senhor Jesus Cristo, a Virgem Maria, a São Judas Tadeu, a Deus e a todos que me ajudaram a concluir com sucesso este trabalho.

À todos os meus familiares, amigos, professores e todas as pessoas que contribuíram de alguma forma para este trabalho

À minha mãe Valéria e ao meu pai Odilon. Eles foram fundamentais para mais esta conquista na minha vida. Não só pela ajuda financeira, mas também pelo incentivo, preocupação e interesse demonstrado.

À minha avó Liliane pela compreensão nos muitos momentos em que não pude visitá-la.

À minha esposa Adriana e ao meu filho Pedro Henrique, pela paciência e compreensão nos momentos em que não podemos ficar juntos para passear ou brincar.

Ao amigo Márcio Dutra pela prestividade, ajuda e caminhadas que serviram para exploração de várias idéias.

Ao amigo Fábio Coutinho pela dedicação nas explicações e manutenção da MEC ROSA.

Ao professor Paulo Pires, professora Maria Claudia e professor Jauvane de Oliveira, por fazerem parte da Banca Examinadora e pelas valiosas críticas e sugestões levantadas. Mais ainda ao professor Jauvane pela consultoria de rede prestada e pelo ambiente computacional cedido para realização dos testes com o protótipo do sistema proposto.

Ao professor Fábio Porto pela oportunidade oferecida e confiança depositada, sem a qual não teria sido possível minha entrada no IME e, conseqüentemente, a realização deste trabalho. Obrigado também pelas preciosas contribuições iniciais. Foram essências para determinar uma linha de estudo e um ponto de partida

À professora Ana Maria de Carvalho Moura por me orientar. Agradeço por toda a compreensão, tolerância, ajuda e confiança em mim depositada. Suas idéias, sugestões, correções e determinação foram essências para o sucesso deste trabalho.

SUMÁRIO

LISTA DE ILUSTRAÇÕES.....	09
LISTA DE TABELAS.....	11
1. INTRODUÇÃO.....	14
1.1 Motivação.....	15
1.2 Objetivo.....	16
1.3 Organização da dissertação.....	17
2. SISTEMAS PEER-TO-PEER (P2P).....	18
2.1 Visão Geral.....	18
2.2 Arquitetura.....	19
2.2.1 Tipos de Arquitetura.....	20
2.2.1.1 Parcialmente Centralizada ou Híbrida.....	20
2.2.1.2 Descentralizada ou Pura.....	22
2.2.2 Super-peers.....	23
2.2.2.1 Propriedades dos <i>Super-peers</i>	25
2.3 Problemas e Possíveis Soluções.....	27
2.4 Sistemas de Integração de Dados.....	30
2.4.1 Arquitetura.....	32
2.4.2 Problemas e Possíveis Soluções.....	34
2.4.2.1 Problemas de Integração de Dados Sob um Ponto de Vista Geral.....	35
2.4.2.2 Problemas Específicos de Integração de Dados em Ambiente P2P.....	37
2.4.3 Metadados e Ontologias no Suporte a Integração de Dados.....	39
2.4.3.1 Metadados.....	40
2.4.3.2 Ontologias.....	41
2.5 Trabalhos Relacionados.....	42
2.5.1 BestPeer - PeerD.....	42
2.5.2 Hyperion.....	47
2.5.3 Piazza.....	51
2.5.4 Edutella.....	53

2.5.5	Considerações Finais.....	56
3.	SISTEMA ROSA.....	59
3.1	Visão Geral	59
3.2	Arquitetura do Sistema ROSA.....	60
3.3	Modelagem de LOs.....	61
3.4	Modelo de Dados ROSA	65
3.5	Considerações Finais.....	67
4.	DEFINIÇÃO DO SISTEMA PROPOSTO – ROSA - P2P	69
4.1	Arquitetura Interna.....	69
4.2	Definição do ambiente P2P	71
4.2.1	Arquitetura.....	71
4.2.2	Estratégias Adotadas	73
4.2.2.1	Agregação de Peers e Agrupamento de <i>Super-peers</i>	73
4.2.2.2	Conexão de <i>Peers</i> ao Sistema.....	76
4.2.2.3	<i>Super-peer</i>	79
4.2.2.4	Quantidade de <i>Super-peers</i>	80
4.2.2.5	Eleição de Super-peers	81
4.2.2.6	Balanceamento do Sistema.....	84
4.2.2.7	Tabela de Propriedades dos <i>Peers</i> e <i>Super-peers</i>	88
4.2.2.8	Comunicação Entre Peers e Super-peers – Índices SP/P e SP/SP	90
4.2.2.9	Processamento de Consultas.....	96
4.2.3	Tolerância a Falhas	96
4.3	Definição do Sistema de Integração de Dados	99
4.3.1	Arquitetura de Integração	99
4.3.2	Estratégia Adotada	102
4.3.2.1	Vocabulários Controlados	102
4.3.2.1.1	Estratégia para Criação do Vocabulário Controlado Global e Local.....	103
4.3.2.1.2	Vocabulário Controlado Global.....	105
4.3.2.1.3	Vocabulário Controlado Local.....	107
4.3.2.1.4	Vocabulário Controlado de Palavras Chaves	110
4.3.2.2	SEVC – Serviço de Entrega de Vocabulários Controlados.....	112

4.3.2.3	Processamento de Consultas.....	115
4.3.2.3.1	Reenvio de Consultas.....	116
4.3.2.3.2	MEC ROSA – Máquina de Execução de Consultas do ROSA	118
4.3.2.3.3	Reescrita de Consultas	123
4.3.2.3.4	Estratégia de Processamento de Consultas.....	130
4.3.2.3.4.1	Tempo de Espera de Resultados	132
4.3.2.4	Integração de Dados.....	135
4.3.3	Considerações Finais.....	137
5.	PROTÓTIPO DO SISTEMA PROPOSTO	139
5.1	Detalhes da Implementação.....	139
5.2	Módulos Implementados	140
5.3	Interfaces do Usuário	141
5.3.1	Interfaces do ROSA - P2P	141
5.3.1.1	Tela de Abertura	142
5.3.1.2	Tela de Geração de Consulta	144
5.3.1.3	Tela de Configuração	147
5.3.2	Interfaces do Portal ROSA	148
5.3.2.1	Página Inicial	149
5.3.2.2	Página Principal	149
5.4.	Exemplo de Uso.....	150
5.5	Testes e Avaliação do Sistema Proposto.....	154
5.6	Considerações Finais.....	156
6.	CONCLUSÃO.....	157
6.1	Contribuições.....	159
6.2	Sugestões para Trabalhos Futuros	160
7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	161
8.	APÊNDICES	167
8.1	APÊNDICE 1 – DTD da Tabela de Propriedades	168
8.2	APÊNDICE 2 – DTD do Índice de Roteamento SP/P.....	169

8.3	APÊNDICE 3 – DTD do Índice de Roteamento SP/SP.....	170
8.4	APÊNDICE 4 – XMLSchema do Vocabulário Controlado Global	171
8.5	APÊNDICE 5 – Instância em XML do Vocabulário Controlado Global	172
8.6	APÊNDICE 6 – XMLSchema do Vocabulário Controlado Local	179
8.7	APÊNDICE 7 – Instância em XML do Vocab. Controlado Local de Informática .	180
8.8	APÊNDICE 8 – XMLSchema do Vocabulário Controlado de Palavras Chaves	185
8.9	APÊNDICE 9 – Instância em XML do Vocab. Controlado de Palavras Chaves....	186
8.10	APÊNDICE 10 – DTD da Estr. de Ident. e Localização dos Vocab. Controlados	187
8.11	APÊNDICE 11 – Instância XML da Estr. de Ident.e Local. dos Vocab. Contr.	188

LISTA DE ILUSTRAÇÕES

FIG. 2.1 Exemplo de um sistema P2P	19
FIG. 2.2 Arquitetura parcialmente centralizada ou híbrida.	21
FIG. 2.3 Arquitetura descentralizada ou pura.	22
FIG. 2.4 Arquitetura baseada em <i>super-peers</i>	24
FIG. 2.5 Funcionamento de um sistema de integração de dados em ambiente P2P.	31
FIG. 2.6 Arquitetura de integração de dados para sistemas P2P – <i>super-peers</i>	33
FIG. 2.7 Arquitetura do PeerDB.	43
FIG. 2.8 Arquitetura do Hyperion.	48
FIG. 2.9 Instâncias dos <i>peers</i> 1 e 2 - Projeto Hyperion.	49
FIG. 2.10 Mapeamento de tabelas - Projeto Hyperion	49
FIG. 2.11 Exemplo simplificado de mapeamentos de esquemas – Projeto Piazza.	52
FIG. 2.12 <i>Super-peers</i> dentro da topologia de HyperCuB.	54
FIG. 2.13 Exemplo de submissão de uma consulta – Rede Edutella.	56
FIG. 3.1 Arquitetura do sistema ROSA.	60
FIG. 3.2 Exemplo de LO lógico e físico.	62
FIG. 3.3 Mapa conceitual parcial do Curso de Mestrado em Sistemas e Computação.	63
FIG. 3.4 Modelo de dados ROSA.	65
FIG. 4.1 Arquitetura interna do sistema proposto – ROSA - P2P.	70
FIG. 4.2 Sistema proposto na arquitetura baseada em <i>super-peers</i>	72
FIG. 4.3 Camadas do sistema proposto.	73
FIG. 4.4 Agregações de <i>peers</i> e agrupamento de <i>super-peers</i>	75
FIG. 4.5 Agrupamento de <i>super-peers</i>	75
FIG. 4.6 Conexão simples de um <i>peer</i> ao sistema	77
FIG. 4.7 Eleição de um <i>super-peer</i> dentro de uma agregação	82
FIG. 4.8 Eleição de <i>super-peer</i> dentro de um agrupamento	83
FIG. 4.9 Balanceamento do agrupamento	86
FIG. 4.10 Uso da estratégia de balanceamento do agrupamento	87
FIG. 4.11 Exemplo de uma instância do índice da tabela de propriedades	90
FIG. 4.12 Índice de roteamento SP/P no sistema	91

FIG. 4.13 Exemplo de uma instância do índice de roteamento SP/P	92
FIG. 4.14 Índice de roteamento SP/SP no sistema.....	93
FIG. 4.15 Exemplo de uma instância do índice de roteamento SP/SP.....	95
FIG. 4.16 Arquitetura de integração de dados do sistema ROSA - P2P.	101
FIG. 4.17 Vocabulário controlado global parcial - predicado <i>é_pré-requisito_de</i>	106
FIG. 4.18 Vocabulário controlado local parcial - gerenciamento de banco de dados.....	109
FIG. 4.19 Vocabulário controlado de palavras-chaves - Domínio informática.	111
FIG. 4.20 Estrutura parcial de identificação e localização dos vocabulários controlados. ...	114
FIG. 4.21 Tela de solicitação de palavra chave.	117
FIG. 4.22 Exemplos de consultas.	118
FIG. 4.23 Reescrita da operação de seleção.	128
FIG. 4.24 Reescrita da operação de navegação.....	129
FIG. 4.25 Consulta final reescrita.	129
FIG. 4.26 Exemplo da integração de dados.	136
FIG. 5.1 Tela de abertura do ROSA - P2P.....	143
FIG. 5.2 Tela de exibição de resultados.	144
FIG. 5.3 Tela de geração da consulta do ROSA - P2P.....	146
FIG. 5.4 Tela de configuração do ROSA - P2P.....	147
FIG. 5.5 Página inicial do portal ROSA.	149
FIG. 5.6 Página principal do portal ROSA.....	150
FIG. 5.7 Mapa conceitual parcial do IME-RJ.....	151
FIG. 5.8 Mapa conceitual parcial da PUC-RJ.....	151
FIG. 5.9 formulação de consulta (exemplo).	152
FIG. 5.10 Resultado da consulta.	153

LISTA DE TABELAS

TAB. 2.1	Comparação entre arquiteturas P2P.	27
TAB. 2.2	Tabela de palavras chaves - Projeto <i>PeerDB</i>	45
TAB. 2.3	Comparação entre alguns dos principais sistemas P2P atuais.	57
TAB. 4.1	Propriedades dos <i>peers</i> e <i>super-peers</i>	89
TAB. 4.2	Índice de roteamento SP/P.	92
TAB. 4.3	Índice de roteamento SP/SP.	94
TAB. 6.1	Características do ROSA - P2P.	158

RESUMO

ROSA é um sistema de *e-learning* que permite a criação, armazenamento, re-uso e gerenciamento de objetos de aprendizagem – Learning Objects (LOs). Um LO, caracterizado por um conjunto de descritores de metadados, é uma coleção de material reutilizável usado para suportar aprendizagem, educação e/ou treinamento. Entretanto, o sistema ROSA ainda é um sistema centralizado, não permitindo a integração de LOs criados em outras instituições. Esta dissertação apresenta a evolução do ROSA em um sistema *peer-to-peer* (P2P), denominado ROSA - P2P, capaz de realizar a integração de LOs oriundos de diferentes *peers*. O ambiente P2P provê a interoperabilidade entre os *peers* ROSA - P2P, incluindo estratégias específicas para: conexão/desconexão de *peers* à rede P2P; definição e eleição de *super-peers*; balanceamento e redistribuição de *peers* no sistema; e alguns aspectos de tolerância à falhas. O sistema de integração de dados possui procedimentos de reenvio e reescrita de consultas baseados no seu significado semântico, assim como mecanismos para a correta integração dos respectivos resultados. Vocabulários controlados são utilizados para suportar estes processos, permitindo dentre outras coisas, a correta interpretação semântica dos dados e resolução de possíveis conflitos semânticos.

ABSTRACT

ROSA is an e-learning system, which enables the creation, storage, reuse and management of Learning Objects (LOs). A LO, characterized by a set of metadata descriptors, is a collection of reusable material used to support learning, education and/or training. However, since ROSA is still a centralized system, it does not provide yet a complete integration of LOs created in local ROSAs of other institutions. This thesis presents the evolution of ROSA into a *peer-to-peer* (P2P) system, named ROSA - P2P, which is able to carry out the integration of LOs from different peers. The P2P environment provides the interoperability between the ROSA - P2P peers, including particular strategies for: connection/disconnection of peers in the P2P network; definition and election of super-peers; balancing and redistribution of peers in the system; and some aspects concerning fault tolerance. The data integration system provides procedures for broadcasting and rewriting queries based on their semantic meanings, as well as mechanisms to perform the correct results integration. Controlled vocabularies are used to support these processes enabling, among other things, the correct data semantic interpretation and resolution of possible semantic conflicts.

1. INTRODUÇÃO

O sistema ROSA [Porto, Moura et al., 2003] é um sistema voltado para a área de Ensino a Distância (EAD), utilizado por profissionais da área educacional na preparação de aulas e/ou conteúdos instrucionais. No entanto ele é um sistema local, não permitindo nenhuma interoperabilidade entre as instituições que o utilizam. Para que o sistema ROSA possa assumir uma função real de ambiente cooperativo inter-institucional, torna-se necessário que conteúdos possam ser armazenados nas diversas instituições, e posteriormente intercambiados e integrados, de modo a fornecer respostas globais às consultas submetidas através dos diversos ROSAs locais.

Atualmente, devido ao progresso alcançado pela capacidade de interconexão, o crescimento de tecnologias associadas à computação distribuída e interoperabilidade passou a receber maior enfoque. Dentro deste contexto, a tecnologia *peer-to-peer* (P2P) [Brito e Moura, 2004] vem se destacando como uma das grandes topologias de sistemas distribuídos da atualidade. Esta tecnologia vem recebendo maior atenção tanto de pesquisadores quanto de empresas, uma vez que oferece benefícios com um baixo custo operacional. Dentre estes benefícios, destaca-se o compartilhamento de recursos, que pode ser mais objetivamente referenciado como o compartilhamento de serviços e conteúdos [Schollmeier, 2002].

Todavia, estes benefícios são utilizados segundo características particulares de cada sistema P2P. O compartilhamento de conteúdos, por exemplo, é utilizado por alguns sistemas somente como compartilhamento de arquivos, enquanto que a semântica dos dados é ignorada. Por outro lado, outros sistemas, a exemplo do ROSA, focam exatamente o uso desta semântica, favorecendo dentre outras coisas, a integração de dados, principalmente em se tratando de *peers* heterogêneos. Dentre o compartilhamento de serviços, pode-se destacar aplicações que compartilhem processamento (de consultas, por exemplo) e/ou armazenamento.

As arquiteturas de integração de dados para os sistemas *peer-to-peer* por sua vez, não possuem uma topologia bem definida [Brito e Moura, 2005]. Elas variam de acordo com o objetivo, arquitetura, funcionamento e características de cada sistema P2P. Desta forma, sistemas de integração de dados *peer* adotam uma semântica particular, possibilitando que, além de outros mecanismos, mapeamentos entre esquemas de exportação ou uso de chaves específicas sejam utilizados para o entendimento, resolução dos possíveis conflitos e integração dos dados residentes nos diversos *peers*.

Acoplado a cada novo sistema P2P, o sistema de integração de dados *peer-to-peer* se sobressai não só por ser um módulo fundamental para o funcionamento geral do sistema, mas por caracterizá-lo quanto à estratégia utilizada que provê a integração. Desta forma, sistemas ou arquiteturas de integração de dados *peer-to-peer* podem ser comumente referenciados como estratégia de integração de dados *peer-to-peer*.

Devido ao progresso alcançado na capacidade de interconexão, o crescimento da computação distribuída e o estabelecimento de padrões concebidos para suportar a heterogeneidade semântica passaram a receber maior enfoque. Neste contexto, a especificação e utilização de metadados [Moura, 2002] vêm ganhando papel de destaque. Estes podem ser utilizados para armazenar informações sobre os diversos esquemas, mapeamentos, associações, localização de fontes e outras informações, servindo desta forma como subsídio essencial para a integração de dados. Também podem ser descritos através de ontologias [Corcho, Lopez e Perez, 2002] ou estruturas similares, tal como vocabulários controlados, permitindo a correta interpretação e recuperação das informações, ao mesmo tempo em que viabilizam o intercâmbio entre os sistemas, permitindo pesquisas mais apuradas e restritas às informações realmente relevantes. Estes também colaboram no processo de reescrita de consultas e garantem um aumento da qualidade e precisão na resolução de conflitos e na geração de esquemas.

Em um sistema de integração de dados P2P, a definição de uma estratégia de processamento de consultas torna-se um elemento fundamental. Seu objetivo é garantir que consultas sejam reenviadas somente aos *peers* relevantes, além de reescrevê-las com base em alguma estrutura semântica [Brito e Moura, 2005]. Essa estrutura deve levar em consideração uma terminologia compatível com o domínio da consulta, de modo a recuperar todos os resultados relevantes existentes no ambiente P2P. Além disso, deve incluir uma linguagem de consulta expressiva, responsável pela execução das consultas, assim como por procedimentos de tolerância a falhas.

1.1 MOTIVAÇÃO

O sistema ROSA visa ser utilizado por instituições de ensino e profissionais da área educacional como repositório de seus conteúdos instrucionais. Seu funcionamento é local, e,

conseqüentemente, não proporciona integração entre os conteúdos dos diversos ROSAs locais.

Desta forma, uma grande contribuição a esse sistema seria dotá-lo desse recurso extra, permitindo ao ROSA a capacidade de interoperar seus conteúdos, mantendo porém suas características, principalmente no que se refere à semântica dos dados, e proporcionando aos usuários, não mais uma visão local às consultas, mas uma visão global e integrada, levando em consideração os conteúdos espalhados nos diversos ROSA locais.

Esta evolução do ROSA será de grande importância para as instituições que o utilizam, pois permitirá o compartilhamento de dados entre elas, até então não existente.

1.2 OBJETIVO

O objetivo desta dissertação é transformar o sistema ROSA em um sistema P2P, gerando o ROSA - P2P, viabilizando a integração de objetos de aprendizagem em ambiente distribuído. O que se pretende realizar na prática é um ambiente P2P onde usuários poderão submeter suas consultas e receber os respectivos resultados, resultante da integração dos dados residentes nos diversos *peers* ROSA - P2P. Para isso foi desenvolvida uma arquitetura de sistema baseada em *super-peers* [Brito e Moura, 2004], incluindo estratégias específicas para: conexão/desconexão de *peers* à rede P2P, tal como o agrupamento de *super-peers* por assunto; definição e eleição de *super-peers*; balanceamento e redistribuição de *peers* no sistema; e alguns aspectos de tolerância à falhas. Adicionalmente, foi definido um sistema de integração de dados P2P. Este é apoiado por estruturas semânticas, denominadas de vocabulários controlados, e possui mecanismos para o correto reenvio, reescrita e execução de consultas, assim como para integração dos respectivos resultados. Desta forma, cada consulta é reenviada aos *peers* relevantes, os quais a reescrevem com base na semântica do seu domínio e as executam com base na álgebra ROSA [Coutinho e Porto, 2004]. Os resultados parciais, oriundos de cada *peer* ROSA - P2P, são enviados ao *peer* solicitador, que os integra e os retorna ao usuário.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação encontra-se organizada em 6 capítulos, a saber:

O capítulo 1 apresenta a introdução, expondo a motivação, objetivo e organização dessa dissertação.

O capítulo 2 faz um levantamento das tecnologias relacionadas a sistemas P2P e integração de dados nesse ambiente. São abordados aspectos importantes quanto aos diferentes tipos de arquiteturas, problemas e respectivas soluções. Ao final, são apresentados alguns dos principais projetos P2P existentes, utilizados para um estudo comparativo dentre algumas de suas principais funcionalidades e características.

O capítulo 3 descreve o sistema ROSA, apresentando suas características mais importantes, a exemplo de mapas conceituais e o modelo de dados ROSA, oferecendo desta maneira uma visão geral do seu funcionamento.

O capítulo 4 apresenta o ROSA - P2P, destacando sua arquitetura interna e a estratégia definida para o ambiente P2P e integração de dados. São discutidos pontos relacionados a sua arquitetura, funcionamento e características particulares.

O capítulo 5 descreve alguns detalhes de implementação do sistema e testes realizados. Apresenta o protótipo ROSA - P2P e exhibe alguns resultados sobre a sua avaliação feita em um ambiente distribuído real, validando o seu funcionamento.

Finalmente, no capítulo 6, são apresentadas as conclusões dessa dissertação, suas contribuições e algumas sugestões para trabalhos futuros.

2. SISTEMAS *PEER-TO-PEER* (P2P)

Neste capítulo serão apresentados e discutidos os principais tópicos associados com a tecnologia de sistemas *peer-to-peer* (P2P). Também serão apresentados alguns dos principais trabalhos relacionados.

Ele está dividido em 5 seções. A seção 2.1 apresenta uma visão sobre a tecnologia P2P. Em seguida, a seção 2.2 descreve as principais arquiteturas, destacando suas características. A seção 2.3 aborda os principais problemas e respectivas soluções existentes nestes sistemas. Logo após, a seção 2.4 dá uma visão geral de sistemas de integração P2P, discutindo sobre sua arquitetura e abordando os principais problemas e respectivas soluções. Esta seção também apresenta conceitos relacionados a estruturas semânticas, a exemplo de metadados e ontologias. Finalmente na seção 2.5, são apresentados alguns dos principais projetos P2P existentes, focalizando a definição do ambiente P2P e a estratégia de integração utilizada, onde, ao final, é feito um estudo comparativo dentre algumas de suas principais funcionalidades e características.

2.1 VISÃO GERAL

Sistemas P2P nada mais são que o compartilhamento de recursos computacionais, serviços e conteúdos através da comunicação direta e descentralizada entre os sistemas envolvidos [Ooi, Shu e Tan, 2003]. Estes recursos e serviços incluem, dentre outras coisas, a troca de informações, ciclos de processamento e espaço de armazenamento em disco. Os sistemas P2P se aproveitam do poder computacional e da conectividade de computadores convencionais para, de forma barata, tornar estes recursos acessíveis entre os nós do sistema, denominados *peers*, conforme ilustrado na FIG. 2.1.

Em outras palavras, pode-se dizer que a idéia básica dos sistemas P2P é a simplicidade: compartilhar arquivos e programas, e permitir a comunicação direta com outras pessoas por meio da internet, sem a necessidade de suporte de um servidor centralizado. Em alguns casos, um servidor central pode ajudar as pessoas a se conectarem, mas, a partir daí, a comunicação é feita entre os *peers*. Futuramente, o modelo de tecnologia sem servidor poderá mudar os procedimentos de negócios de uma empresa. Redes P2P as ajudariam a criar suas próprias

áreas privadas para compartilhar arquivos, trocar informações, gerar bancos de dados e se comunicar de maneira instantânea e mais acessível, eliminando a necessidade de comprar servidores caros e até mesmo o suporte de consultores para montar bancos de dados. A tecnologia P2P também facilitará a participação da empresa em mercados virtuais, permitindo a colaboração direta com fornecedores, parceiros e clientes [Monteiro, 2003].

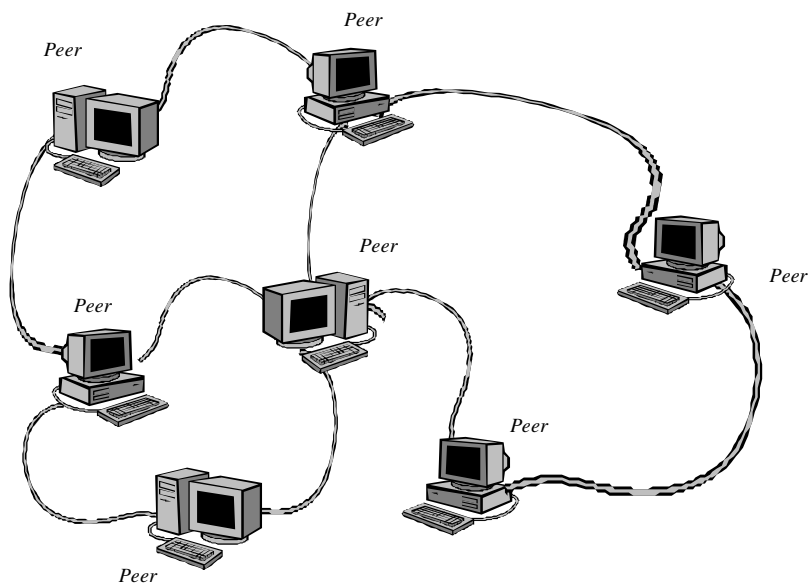


FIG. 2.1 – Exemplo de um sistema P2P

Porém, para um melhor entendimento da tecnologia P2P, faz-se necessário a definição dos tipos de arquitetura existentes, apresentados a seguir.

2.2 ARQUITETURA

Embora sistemas P2P não sejam uma novidade, existem ainda alguns conflitos quanto aos tipos de arquitetura, principalmente no que se refere à arquitetura centralizada (sistema cliente-servidor).

Um sistema cliente-servidor ou uma arquitetura centralizada, como alguns autores se referem, é um sistema distribuído que consiste de um nó central (servidor) com alto poder de processamento e vários nós ligados a ele (clientes) com menor poder de processamento. O

servidor é o único nó registrador, assim também como o único a prover serviços e conteúdos. Um cliente somente faz requisições de serviços e conteúdos a ele, nada mais. Conseqüentemente, não compartilha nenhum de seus recursos [Schollmeier, 2002].

Desta forma, conclui-se que esta arquitetura não é adequada para ser incluída como um tipo de arquitetura P2P, uma vez que o seu processamento é centralizado e o compartilhamento de recursos entre os nós não existe, fugindo completamente da idéia P2P (descentralizada). Existem inclusive, alguns autores que definem esta arquitetura como o oposto da arquitetura P2P [Singh, 2001] e [Thomas, Suchter e Rifkin, 1998].

Para ser realmente P2P, uma arquitetura deve permitir que os *peers* participantes compartilhem, ao menos, parte dos seus recursos (poder de processamento, capacidade de armazenamento, etc.). Eles são fundamentais para que *peers* possam prover seus conteúdos e serviços, como por exemplo, o compartilhamento de arquivos e serviços de impressão. Desta forma, *peers* são acessados por outros diretamente, não existindo assim nenhum *peer* intermediário, isto é, nenhum *peer* central necessário para prover o uso desses recursos. Cada *peer* atua como *servent*, podendo atuar ora como servidor e ora como cliente ao mesmo tempo [Schollmeier, 2002].

2.2.1 TIPOS DE ARQUITETURA

De forma a distinguir um sistema P2P que possui um *peer* central de um que não o possui, Schollmeier [Schollmeier, 2002] classifica a arquitetura dos sistemas *peer-to-peer* em 2 tipos básicos:

2.2.1.1 PARCIALMENTE CENTRALIZADA OU HÍBRIDA

Conforme apresentado na FIG. 2.2, a arquitetura parcialmente centralizada ou híbrida é aquela que contém um servidor central, responsável pelo mecanismo de busca e manutenção da infraestrutura, deixando a cargo dos *peers* participantes o compartilhamento de recursos, serviços e conteúdos de forma distribuída.

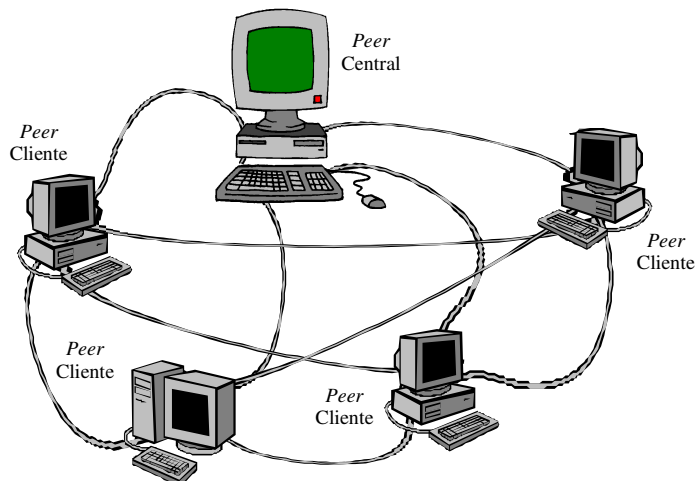


FIG. 2.2 – Arquitetura parcialmente centralizada ou híbrida

Um exemplo de um sistema deste tipo é o Napster [Napster], onde todos os clientes se conectam ao servidor central e informam quais dentre seus arquivos serão compartilhados. Assim, quando um *peer* deseja encontrar um arquivo, envia uma solicitação para o servidor. Este informa qual *peer* possui o referido arquivo e, a partir deste momento, o tráfego de dados é estabelecido entre os *peers*. Porém, se este *peer* central ficar inoperante ou falhar, a rede não mais funcionará, com exceção dos *peers* conectados no momento em que o *peer* central falhou, isto é, que estavam transferindo dados entre si, permanecendo até a completa transferência dos dados.

Escalabilidade, que diz respeito à capacidade que um sistema P2P poder crescer sem ficar sobrecarregado [Brito e Moura, 2004], é um outro problema desta arquitetura. Embora seja possível resolvê-lo temporariamente através de *upgrade* de *hardware* e balanceamento do sistema, o problema pode voltar a ocorrer. Isto vai depender da abrangência do sistema P2P em questão, isto é, o quanto ele é escalável.

Contudo, pode-se referir a essa arquitetura como sendo uma derivação da arquitetura cliente/servidor, destacando-se como diferença o compartilhamento de recursos, serviços e conteúdos, os quais ocorrem de forma distribuída. Porém, pode-se observar que esta arquitetura também não é verdadeiramente *peer-to-peer*, pois embora haja o compartilhamento de serviços e conteúdos de forma distribuída, o processamento ocorre de forma centralizada, através do *peer* central.

2.2.1.2 DESCENTRALIZADA OU PURA

Conforme ilustrado na FIG. 2.3, a arquitetura descentralizada ou pura é aquela que possui todos os seus *peers* participantes de forma descentralizada, caracterizando-se assim, pela completa descentralização de seu funcionamento. Não existe um *peer* central e os mecanismos de busca e manutenção da infraestrutura, assim como o compartilhamento de recursos, serviços e conteúdos estão distribuídos pela rede, em cada *peer*. Um exemplo desta arquitetura é a rede Gnutella [Gnutella].

Nesta arquitetura, cada *peer* é responsável por manter informações sobre seus próprios dados e, conseqüentemente, ao receber uma solicitação de consulta, respondê-la e/ou reescreve-la aos *peers* a ele conectado. Para isto, esta arquitetura faz uso da técnica de "flooding" ou inundação, que consiste em "inundar" a rede com a consulta e retornar os resultados encontrados ao *peer* solicitante. Porém, para evitar que a rede fique saturada, as consultas são limitadas a um número máximo de *peers*. Pode-se observar então que, mesmo que um *peer* esteja "on-line" e possua a resposta à consulta, ele pode não ser consultado podendo a resposta à consulta ficar incompleta, diminuindo assim a confiabilidade do sistema.

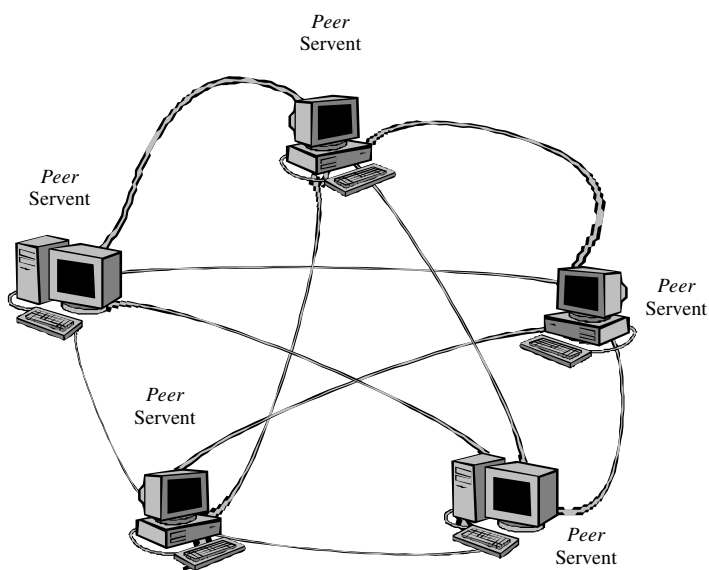


FIG. 2.3 – Arquitetura descentralizada ou pura

Para contornar esse problema, algumas adaptações foram feitas na arquitetura. Segundo [Aberer et al., 2003] e [Bawa et al., 2003], os sistemas P2P podem ser classificados quanto à

localização de recursos em sistema P2P estruturado e não-estruturado. Os sistemas P2P estruturados mantêm informações sobre quais recursos os *peers* oferecem. Desta forma, pesquisas podem ser diretas e conseqüentemente menos mensagens são necessárias, aumentando assim o desempenho, a confiabilidade e a eficiência do sistema. Em contraste, nos sistemas P2P não-estruturados, os *peers* não possuem informações sobre os recursos que outros *peers* oferecem e, desta forma, pesquisas são resolvidas com a técnica de *flooding* (inundação).

Além do seu funcionamento, a arquitetura descentralizada difere da parcialmente descentralizada quanto à tolerância a falhas, que é a capacidade que um sistema P2P possui de não se tornar inoperante ou deficiente na eventual presença de falhas [Brito e Moura, 2004], e quanto a escalabilidade. Caso ocorra uma falha de um *peer* ou ele esteja “off-line”, o sistema pode funcionar normalmente, isto é, a falha ou ausência de um *peer* não impacta no funcionamento do sistema. Quanto a escalabilidade, qualquer nó pode se unir ao sistema a qualquer momento e, assim compartilhar recursos, serviços e conteúdos com os demais *peers*. Não existe um número fixo de nós que podem se unir ao sistema, salvo pelo problema de volatilidade e capacidade de endereçamento, abordado na seção 2.3.

Conforme se pôde observar, a arquitetura descentralizada é o tipo de arquitetura realmente P2P, mas, com o intuito de se criar uma arquitetura mais eficiente, foi desenvolvida a arquitetura de *super-peers*, descrita a seguir.

2.2.2 SUPER-PEERS

A arquitetura baseada em *super-peers* pode ser vista como um melhoramento da arquitetura centralizada embutida nos sistemas descentralizados [Brito e Moura, 2004]. É formada por pequenos subconjuntos de *peers* ligados entre si, denominados de *super-peers*, que agregam outros *peers* conectados a eles. Esta arquitetura pode ser observada na FIG. 2.4.

A arquitetura centralizada é referenciada, pois *peers* submetem suas consultas ao seu respectivo *super-peer* que, na verdade, não deixa de ser um ponto centralizador. O sistema Kazaa [Kazaa] e Edutella [Nejdl et al., 2002] são alguns exemplos de sistemas que utilizam essa arquitetura.

Quando um *peer* deseja se conectar ao sistema, ele identifica com qual *super-peer*, dentre os existentes no sistema, deve se conectar. Durante este processo, *peer* e *super-peer* trocam

informações para se auto validarem, isto é, ambos verificam se podem fazer parte do mesmo sistema e conseqüentemente, em caso afirmativo, trocam informações sobre suas localizações, as quais são indexadas pelo *super-peer*.

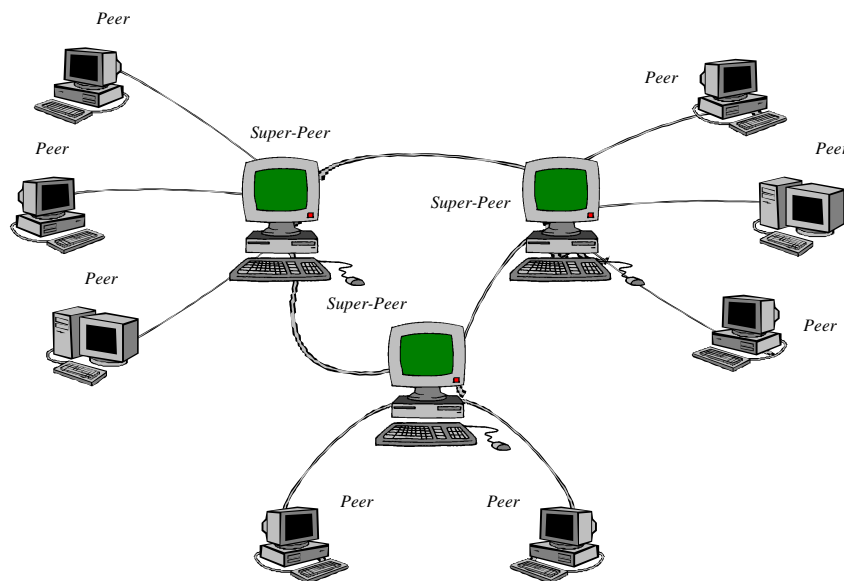


FIG. 2.4 – Arquitetura baseada em *super-peers*

Dependendo das características do sistema, o *peer* deve fornecer outras informações ao seu *super-peer*. Tais informações se referem aos recursos, serviços e/ou conteúdos que o *peer* deseja compartilhar. Dentro deste contexto, estratégias para manutenção dessas informações devem ser implementadas. No sistema P2P Edutella [Nejdl et al., 2002], por exemplo, essas informações são mantidas periodicamente enquanto os *peers* se registram ou atualizam suas informações regularmente. Quando esses procedimentos não ocorrem, indica que o *peer* deixou o sistema e todas as suas referências devem ser removidas. As atualizações são feitas através de *triggers*, acionados por uma notificação enviada pelo *peer*.

Uma vez conectado, o *peer* já se encontra apto a compartilhar recursos e submeter suas consultas. Sendo assim, o *super-peer* ao receber uma consulta, resubmete-a aos seus *peers* e aos demais *super-peers* do sistema. Para tal, o mesmo faz uso das suas informações indexadas. As respostas, por conseguinte, são retornadas diretamente ao *peer* solicitante. Vale observar que para responder a uma solicitação de consulta, o *super-peer* se comunica somente com os demais *super-peers*, que conseqüentemente se comunicam com seus *peers*.

2.2.2.1 PROPRIEDADES DOS *SUPER-PEERS*

Um *super-peer*, conforme o próprio nome já diz, não pode ser um *peer* comum. Ele possui algumas responsabilidades a mais, dentre as quais [Nejdl et al., 2003] destacam:

- *Agregação de peers*: refere-se ao conjunto de *peers* que são conectados ao *super-peer*, incluindo todo o processo de conexão, bem como sua manutenção dentro do sistema;
- *Roteamento de consultas*: diz respeito à busca da resposta à consulta do *peer* cliente dentro do sistema e do gerenciamento de uma consulta reenviada por um *super-peer*, incluindo, principalmente, questões relacionadas à otimização dessas consultas, tais como estratégias que utilizam o auxílio de ontologias e planos de consultas distribuídas.

Para ser um *super-peer*, um *peer* deve possuir algumas características físicas que permitam o bom funcionamento do sistema. Desta forma, a escolha ou a seleção de um *super-peer* se faz através da capacidade de cada *peer* em relação a essas características. Em [Zhu, Wang e Hu, 2003] elas são identificadas como:

- Estabilidade.

Um *super-peer* deve ser estável, caso contrário, sua instabilidade poderá prejudicar o desempenho do sistema como um todo.

- Largura de banda e acesso rápido

Um *super-peer* deve possuir uma largura de banda suficiente para permitir o tráfego de dados de maneira rápida e eficiente, uma vez que é um dos pontos centrais de processamento. Quando uma consulta for submetida a um *super-peer*, a largura de banda não pode ser fator limitante ao acesso rápido aos *peers* ou *super-peers* correspondentes.

- Poder de processamento

Um *super-peer* deve possuir um poder significativo de processamento para suportar o crescente tráfego na camada de rede, assim também como, a grande quantidade de consultas submetidas por seus clientes.

- Capacidade de armazenamento e memória

Um *super-peer* precisa ter espaço de armazenamento e memória extras para guardar as informações referentes aos seus *peers* e aos demais *super-peers*, bem como outras informações que sejam relevantes para o bom funcionamento do sistema.

Outro ponto importante se refere ao ciclo de vida de um *super-peer*, isto é, quanto tempo um *peer* está apto a ser um *super-peer*. Desta forma, por causa do dinamismo presente no sistema, devido principalmente às entradas e saídas de *peers* a qualquer momento, *super-peers* devem, em um período estipulado de tempo, verificar se suas capacidades em relação às suas características físicas, ainda são os melhores dentre os *peers* existentes no sistema. Esta condição determina se estes devem continuar a serem *super-peers* ou se nova eleição deve ocorrer para a indicação de outro *super-peer* [Brito e Moura, 2004]. Provavelmente, nesta eleição, uma estratégia para o balanceamento do agrupamento deverá ser implementada. Em [Triantafillou et al., 2003] pode-se encontrar com detalhes um algoritmo para eleição de *super-peers*, assim como uma técnica de balanceamento para sistemas P2P.

Quanto a quantidade de *super-peers* em um sistema P2P, ela pode ser definida e até mesmo parametrizada, de acordo com as características do sistema em questão. Para tal, um estudo prático deve ser realizado, focalizando obviamente, o bom desempenho do sistema.

A arquitetura baseada em *super-peers* é muito mais vantajosa comparada às demais arquiteturas apresentadas. A pesquisa, por exemplo, é muito mais rápida, uma vez que é feita somente entre os *super-peers* e não entre todos os *peers* do sistema. Por exemplo, uma pesquisa na arquitetura parcialmente centralizada ou descentralizada, é da ordem de $O(N)$, isto é, a consulta deve ser submetida a todos os *peers* participantes do sistema. Já a pesquisa realizada numa arquitetura baseada em *super-peers* é da ordem de $O(N/M)$, onde M é a média do número de *peers* conectados em cada *super-peer*. Assim, a consulta deve ser submetida somente aos demais *super-peers* do sistema. Fica claro perceber que o problema da técnica de inundação encontrado na arquitetura descentralizada é eliminado [Brito e Moura, 2004].

O gerenciamento acontece nos *super-peers* através do monitoramento dos seus *peers*. Isso faz com que a rede se torne mais robusta e portanto mais confiável. A escalabilidade se comporta similarmente ao da arquitetura descentralizada, possibilitando que qualquer nó se junte ao sistema e a qualquer momento compartilhe seus recursos. Quanto a tolerância à falhas, apresenta estratégias tanto para a falha do *super-peer* quanto do *peer*, aumentando desta forma, a confiabilidade do sistema.

A TAB. 2.1 apresenta uma breve comparação entre as arquiteturas P2P, a partir da qual se pode concluir que a arquitetura baseada em *super-peers* é a mais apropriada para o desenvolvimento e manutenção de sistemas P2P, uma vez que se apresenta com sendo a mais completa e vantajosa.

TAB. 2.1 – Comparação entre arquiteturas P2P

Arquitetura	Autonomia de gerenciamento	Escalabilidade	Tolerância a falha	Confiabilidade
Parc.centralizada	Sim	Sim	Não	Não
Descentralizada	Não	Sim	Sim	Sim
<i>Super-peers</i>	Sim	Sim	Sim	Sim

2.3 PROBLEMAS E POSSÍVEIS SOLUÇÕES

Os problemas existentes nos sistemas P2P estão, em muitos casos, relacionados aos encontrados em sistemas distribuídos. Porém alguns deles são bem específicos, e embora se trate de uma tecnologia já conhecida, algumas de suas soluções ainda estão em fase de estudo e aprimoramento. Nesta seção, serão apresentados alguns desses problemas e as respectivas soluções já encontradas:

- Volatilidade de capacidade de endereçamento

Algum tempo atrás a Internet era composta por computadores ligados e conectados continuamente, onde cada um deles estava associado a um endereço IP (protocolo Internet) único e invariante. Porém, atualmente, a realidade da Internet é bem diferente, onde a maior parte das conexões é temporária, principalmente no que se refere às máquinas-cliente, impossibilitando desta forma que o IP, que possui um endereçamento de 32 bits, seja capaz de mapear a infinidade de computadores que atualmente acessam e se unem à rede de modo dinâmico, instável e crescente.

Nos sistemas P2P, os *peers* não possuem endereço IP fixo, isto é, a cada conexão na rede, o *peer* ganha um novo endereço IP. Desta forma, observa-se a volatilidade de cada endereço, assim como um outro problema associado, referente ao dinamismo do ambiente em questão.

Em decorrência dessa limitação, um novo protocolo denominado ipv6 [ipv6], com endereçamento de 128 bits, está em fase de desenvolvimento. Isto permitirá a expansão exponencial da capacidade de endereçamento física da rede, resolvendo essa deficiência,

porém contribuindo ainda mais para o dinamismo das conexões, uma vez que não se tem controle sobre as mesmas.

- Dinamismo do ambiente

Como se pode observar no problema da volatilidade de endereço, o dinamismo das conexões faz com que os *peers* não referenciem todos os *peers* que já fizeram parte do sistema, dificultando desta forma, a definição de uma estratégia eficiente para prover o compartilhamento dos dados entre eles.

Assim, pode-se verificar que o dinamismo do ambiente se faz à medida que *peers* se unem e deixam a rede todo tempo, o que torna um esquema global uma estratégia difícil de ser construída e mantida, e portanto, uma opção aparentemente não prática, escalável e extensível [Ooi, Shu e Tan, 2003].

Uma possível solução para esse problema é o mapeamento de esquemas ou parte deles antes ou durante a execução da consulta. Portanto, quando uma consulta for submetida, o mapeamento será consultado e consultas serão refeitas e submetidas aos *peers* relevantes à consulta. Conseqüentemente, os dados serão integrados e retornados ao usuário.

- Confiabilidade

Em sistemas *peer-to-peer*, assim como em qualquer outro sistema, a confiabilidade é um ponto importante para a sua utilização e perpetuação. Contudo, a confiabilidade tratada por este item se refere a confiabilidade depositada nos dados compartilhados. A confiabilidade geral do sistema será tratada no item tolerância a falhas.

Como em sistemas P2P não se conhece a fonte de dados, isto é, os dados são fornecidos por pessoas remotamente localizadas, fica difícil obter um nível satisfatório de confiabilidade. Porém, com o intuito de obter algum, algumas soluções foram introduzidas, assim como também alguns aspectos de segurança. Dentre elas, pode-se destacar além da criptografia, assinaturas e certificados digitais [Oram, 2001].

- Tolerância à falhas

A tolerância a falhas está diretamente relacionada à confiabilidade geral do sistema, uma vez que possui procedimentos para combater as possíveis falhas que podem ocorrer em ambiente P2P.

Conforme visto na seção 2.2.1.1, a arquitetura parcialmente centralizada não apresenta tolerância a falhas e, conseqüentemente, não é uma arquitetura confiável. Isto se deve ao fato de existir um *peer* central (servidor) que, neste contexto, se caracteriza como sendo um ponto de falha. Uma vez que ele falhe, todo o sistema se torna inoperante. Em contraste com as demais arquiteturas, caso um *peer* falhe, o sistema continua funcionando, uma vez que estas arquiteturas não possuem um ponto único centralizador. Técnicas de tolerância a falhas são implementadas nestas arquiteturas de acordo com as características do sistema em questão.

Portando, uma solução capaz de oferecer uma saída emergencial para a arquitetura parcialmente centralizada é a replicação dos dados. Ela consiste na replicação do *peer* central (servidor) em um ou mais *peers* da rede, preferivelmente, os *peers* mais robustos. Desta forma, em caso de falha do *peer* servidor, os *peers* se conectariam a eles, em uma ordem pré-determinada. Este procedimento também pode ser utilizado pela arquitetura descentralizada e baseada em *super-peer*, onde os dados dos *peers* mais solicitados podem ser replicados em outros *peers*. Desta forma, uma vez que esses *peers* não possam ser localizados por qualquer motivo, seus dados certamente poderão, pois o mecanismo de busca redirecionará a consulta para um *peer* que possua a réplica dos dados procurados.

Porém neste ponto, ocorre um outro problema: replicação de dados. Como manter as réplicas sempre atualizadas e desta forma confiáveis? Este problema não é o objetivo desta seção, uma vez que esta solução não é aconselhada devido ao “workflow” necessário para provê-la. Porém, informações detalhadas podem ser encontradas em Valduriez e Özsu [Valduriez e Özsu, 2001].

- Redundância de informação

Uma vez que não se conhecem todos os *peers* que podem se unir ao sistema e muito menos seus conteúdos, fica claro perceber como o problema da redundância de informação afeta os sistemas P2P. *Peers* podem ter informações redundantes e desta forma exigir processamento desnecessário para sua localização e busca.

A solução encontrada neste caso é a implementação de controles sobre os dados dos *peers*, objetivando reduzir ao máximo a redundância.

- Escalabilidade

Na arquitetura parcialmente centralizada, embora esta solução só resolva o problema temporariamente, pode-se administrar o seu crescimento mediante ações de expansão ou balanceamento de recursos. Já na arquitetura descentralizada e baseada em *super-peers*, cada *peer* deve possuir sua própria solução de escalabilidade, uma vez que os mesmos possuem sua própria autonomia.

- Aspectos Legais

Uma vez que sistemas P2P permitem que outros *peer* se unam a ele e compartilhem grandes volumes de dados, muitas vezes desconhecidos, problemas relacionados a direitos de autoria e propriedade podem aparecer, podendo até mesmo comprometer a sua utilização. O Napster é um exemplo. Ele foi retirado de circulação por determinação da justiça, uma vez que compartilhava dados musicais sem o consentimento de seus autores e gravadoras e por possuir uma arquitetura parcialmente centralizada foi possível achar um responsável.

Se utilizasse uma arquitetura descentralizada ou baseada em *super-peers*, certamente ainda poderia estar em circulação, uma vez que não se saberia quem é o responsável, como ocorre atualmente no caso do Kazza [Kazza].

2.4 SISTEMAS DE INTEGRAÇÃO DE DADOS

Em um ambiente P2P, a estratégia para a integração de dados apresenta algumas diferenças quanto ao utilizado em ambiente WEB [Brito e Moura, 2005]. Essas diferenças dizem respeito principalmente à arquitetura, funcionamento e características particulares dos sistemas P2P. Como em um ambiente P2P cada *peer* atua simultaneamente como servidor e cliente, é conveniente que a integração ocorra em cada *peer*. Outra diferença fundamental é que, embora também não haja o uso de esquemas globais [Brito e Moura, 2005], o sistema de integração de dados nesse ambiente geralmente não apresenta nenhuma forma de visão integrada ao usuário. Isto ocorre, pois dependendo da arquitetura e estratégia utilizada para prover a comunicação entre os *peers*, muitos deles não sabem quem são os vizinhos dos seus vizinhos e assim por diante, fazendo com que consultas sejam reescritas e submetidas muitas vezes a uma grande quantidade de *peers* independente de serem aptos ou não a respondê-las.

A FIG. 2.5 mostra uma possível estratégia adotada para um sistema de integração de dados em ambiente *peer-to-peer*. Como será visto na seção 2.4.1, não existe a priori uma arquitetura padrão de integração bem definida nesse ambiente. Ela se encontra implícita dentro de cada sistema P2P e é desenvolvida de acordo com o objetivo, arquitetura, funcionamento e características do sistema P2P em questão.

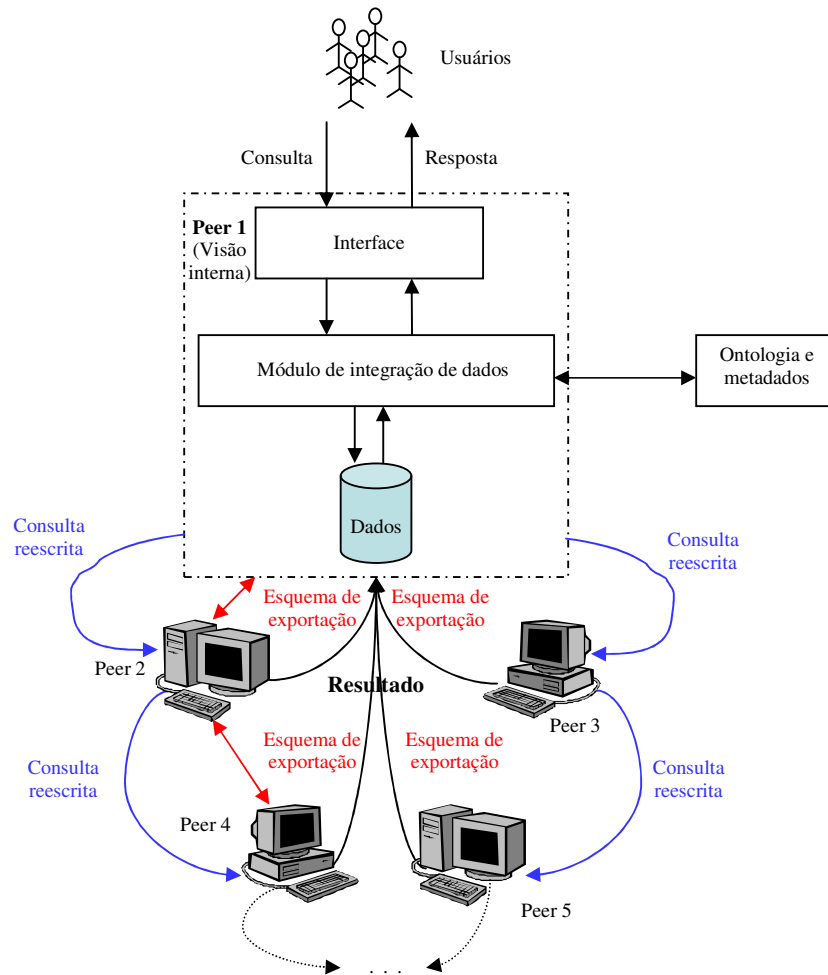


FIG. 2.5 – Funcionamento de um sistema de integração de dados em ambiente P2P

Neste exemplo simples, cada *peer* exporta os dados que os demais *peers* podem acessar. Observe que cada *peer* conhece somente os seus vizinhos, e assim sucessivamente. Desta forma, ao receber uma consulta de um *peer*, esta é reescrita com o auxílio de uma ontologia de domínio (ou outra estrutura que apresente a mesma funcionalidade) e reenviada aos *peers* relevantes aquele domínio. À medida que os dados são retornados ao *peer* solicitador, estes são integrados localmente e enviados ao usuário. Nesta etapa, ontologias também são

utilizadas como suporte à resolução de conflitos semânticos que usualmente ocorrem durante o processo de integração.

2.4.1 ARQUITETURA

O tipo de arquitetura, assim como o objetivo, funcionalidade e características particulares de cada sistema P2P afetam diretamente a definição e funcionamento do sistema de integração de dados. De fato, estes pontos estão diretamente envolvidos na definição, dentre outras coisas, das interfaces, recebimento, reenvio e reescrita de consultas, assim como do acesso às fontes, filtragem dos dados, resolução de conflitos e integração dos dados [Brito e Moura, 2005].

A FIG. 2.6 apresenta um exemplo de um sistema P2P que utiliza a arquitetura de *super-peers*. Seu sistema de integração faz uso de uma estrutura de metadados e de uma ontologia para auxiliar, dentre outras coisas, na localização dos dados, reescrita da consulta e resolução de conflitos. Com base nesta ontologia, as consultas dos usuários são reescritas e enviada aos *peers* que estão aptos a respondê-las. Cada *peer* retorna o resultado ao *peer* solicitador que os filtra, resolve os possíveis conflitos e integra-os, retornando a resposta da integração ao usuário.

Cada sistema de integração de dados num ambiente P2P adota uma semântica particular, possibilitando que mecanismos possam ser utilizados para prover a integração dos dados, como por exemplo mapeamentos entre esquemas de exportação, uso de chaves específicas e reescrita com ajuda de uma ontologia, que facilitam no processo de resolução de conflitos e compreensão semântica dos dados.

Apesar destes sistemas de integração serem referenciados como sendo uma nova arquitetura de integração [Ruzzi, 2004], ainda não existe um consenso quanto a uma topologia bem definida para uma arquitetura de integração de dados P2P. No entanto, ela pode ser definida como:

“Uma arquitetura para integração de dados *peer-to-peer* se constitui da estratégia de integração de dados adotada, que depende exclusivamente do objetivo, arquitetura, funcionamento e características particulares de cada sistema P2P”.

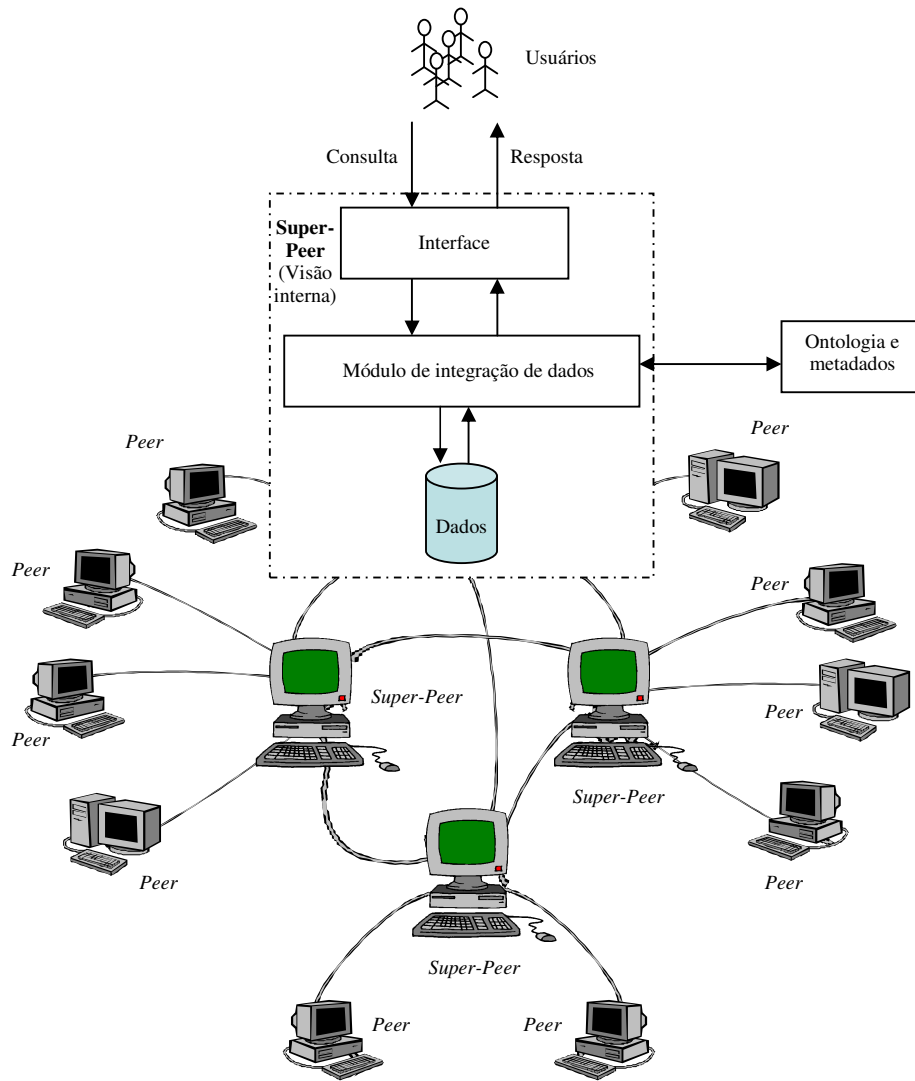


FIG. 2.6 – Arquitetura de integração de dados para sistemas P2P – *super-peers*

Contudo, sistemas para integração de dados *peer-to-peer* se baseiam, em parte, na abordagem virtual [Brito e Moura, 2005]. Embora não haja um sistema de mediação, os dados são mantidos nas fontes originais, isto é, em cada *peer*, e geralmente são integrados em tempo de execução da consulta.

O estudo do funcionamento do sistema de integração de dados de alguns dos principais sistemas P2P será apresentado na seção 2.5. Todavia, torna-se importante a apresentação sucinta de alguns sistemas P2P com o intuito de demonstrar que utilizam sistemas de integração bem diferentes, em função das particularidades específicas de cada um deles. O sistema Kazaa, por exemplo, tem como principal objetivo o compartilhamento de arquivos,

oferecendo portanto, um sistema de integração muito simples. Embora nenhum valor semântico seja agregado aos dados, existe um filtro que é realizado através das pesquisas “ad-hoc” submetidas pelos usuários, através da seleção de alguns poucos metadados. O resultado final, por sua vez, é simplesmente a união dos resultados retornados de cada *peer*. Por outro lado, um outro sistema chamado RACCOON [Li et al., 2004], tem como objetivo permitir que fontes de dados distintas possam integrar e compartilhar seus dados. De fato, este sistema considera o valor semântico dos dados residentes nas fontes existentes e, portanto, seu sistema de integração é mais sofisticado. Utiliza-se de uma linguagem rica para descrever os conteúdos das fontes de dados, descrevendo também informações sobre os esquemas e possíveis restrições. Mapeamentos semânticos são definidos entre fontes de dados, e durante a pesquisa, o sistema usa técnicas de mapeamento de esquemas para encontrar relações similares para a relação dada, sugerindo que o usuário escolha alguma.

Todavia, devido às características particulares desta tecnologia, os sistemas de integração de dados *peer-to-peer* trazem consigo novos problemas que se aglomeram aos já existentes. A abordagem destes problemas e possíveis soluções, principalmente dos novos desafios no ambiente P2P constituem o assunto da próxima seção.

2.4.2 PROBLEMAS E POSSÍVEIS SOLUÇÕES

Os principais problemas encontrados na integração de dados dizem respeito principalmente às diferenças estruturais e semânticas encontradas nos modelos para representação de dados de cada banco participante. Isto decorre do fato de existirem formas distintas de se modelar um problema do mundo real, podendo ocorrer diferentes representações de um mesmo problema: o modelo pode suportar diferentes representações ou os projetistas podem ter diferentes percepções da realidade [Tavares, 1999].

Além disso, existem diferenças quanto a heterogeneidade de hardware/software, dentre as quais é possível destacar diferenças quanto ao tipo de hardware, sistema operacional e protocolos de comunicação [Tatbul, 2001]. Porém estes não são a causa da inconsistência semântica na integração de dados e, portanto, não terão ênfase nesta dissertação.

Entretanto, com o desenvolvimento dos sistemas P2P e, conseqüentemente, dos sistemas de integração de dados P2P, novos problemas surgiram oriundos desta tecnologia, incrementando os já existentes. Desta forma, com o intuito de facilitar este estudo, os

problemas referentes à integração de dados serão divididos em: problemas de integração de dados sob um ponto de vista geral, oriundos da integração de banco de dados no contexto mais tradicional e WEB; e problemas específicos de integração de dados em ambiente P2P.

2.4.2.1 PROBLEMAS DE INTEGRAÇÃO DE DADOS SOB UM PONTO DE VISTA GERAL

Esta seção tem como objetivo apresentar os principais problemas que podem ocorrer em qualquer arquitetura de integração de dados. Esses problemas, geralmente semânticos e estruturais, são abordados por diversos autores [Tatbul, 2001], [Batista, 2003] e [Tavares, 1999]. Estes também foram bem estudados em [Brito e Moura, 2005]. Contudo, pode-se defini-los de maneira geral como sendo:

- Conflitos de modelo de dados

Referem-se ao uso de diferentes modelos de dados em diferentes esquemas. Por exemplo, um esquema pode ser representado em um modelo orientado a objetos, enquanto que o esquema integrado pode ser representado em um modelo de dados relacional.

Num processo de integração de dados, os esquemas locais das bases heterogêneas podem ser mapeados para um esquema global único utilizando um modelo de dados padrão, a exemplo do XML. A consulta é feita diretamente sobre o esquema global, porém, quando uma base de dados é acessada para fins de consulta, é feito um novo mapeamento para o respectivo modelo de dados.

- Conflitos de nomes

Diz respeito ao uso semântico dos dados. Na abordagem de banco de dados, o problema mais frequentemente encontrado no processo de integração recai sobre os problemas denominados de sinônimos e homônimos.

Sinônimos ocorrem quando o mesmo objeto do mundo real pode ser representado por diferentes nomes. Como exemplo, pode-se citar o nome de uma disciplina estar sendo referenciada em um esquema A por “banco de dados” enquanto que em outro esquema B é denominada simplesmente de “BD”. Para resolver este problema, o sistema de

integração deve reconhecer a equivalência semântica entre esses dados e mapear os diferentes nomes locais para um nome global equivalente.

Homônimos ocorrem quando diferentes objetos do mundo real são representados com o mesmo nome. Como exemplo, tem-se o termo “rede” que em um esquema pode significar “rede de computadores” enquanto que em outro pode estar referenciando uma “rede de pesca”. Para resolvê-los, o sistema de integração de dados deve reconhecer esta diferença semântica e mapear os nomes comuns para diferentes nomes globais.

Ontologias e metadados são geralmente utilizados pelo sistema de integração de dados para suportar e auxiliar na resolução destes conflitos.

- Conflitos estruturais

Referem-se aos conflitos existentes entre as representações estruturais dos esquemas, isto é, diferentes esquemas podem representar a mesma informação de diferentes maneiras. Por exemplo, um esquema pode utilizar uma única tabela para armazenar as informações referentes às disciplinas e seus respectivos instrutores, enquanto que outro esquema pode normalizar a mesma informação em uma tabela com as informações sobre as disciplinas e outra sobre os instrutores.

Para solucionar este problema, o sistema de integração de dados deve ou selecionar uma representação baseada em alguma delas, ou construir uma representação única comum de maneira que todas as incompatibilidades entre os esquemas possam ser suportadas em um esquema integrado.

- Conflitos semânticos

Conflitos semânticos são decorrentes dos problemas existentes entre as representações dos esquemas, isto é, que usam diferentes níveis de abstração para modelar a mesma entidade. Por exemplo, um esquema poderia distinguir entre carros e caminhões, enquanto que outro esquema poderia simplesmente modelar as mesmas informações com automóveis e, desta forma, falhar ao fazer a distinção entre carros e caminhões.

Para solucionar este problema, o sistema de integração de dados deve fazer uso de uma ontologia de domínio, a qual deve dar suporte e auxiliar no processo de integração dos dados, visando a identificação completa dos termos em comum.

- Conflitos de formato

Referem-se ao uso de diferentes valores para representar dados. Esses valores podem ser classificados quanto ao tipo de dados, domínio, escala, precisão e combinação. Por exemplo, um esquema pode armazenar o preço de um produto através da representação de um número inteiro e um outro esquema pode armazená-lo como sendo um ponto flutuante.

Esses conflitos podem ser solucionados pelo sistema de integração de dados através da definição de funções de transformação entre a representação local e a global. Algumas delas podem simplesmente ser constituídas de cálculos numéricos, tal como a conversão de centímetros para metros; outras podem ser mais complexas, requerendo tabelas para conversão ou algoritmos específicos de transformação.

- Ausência de dados

Diz respeito à representação de dados de um esquema que pode estar ausente em outro esquema. Por exemplo, um esquema pode armazenar a data da última modificação de uma tabela, enquanto que um outro esquema não faz qualquer referência a esta informação. Porém, podem ocorrer também casos onde a informação não foi gravada corretamente devido a atualizações incompletas ou erros de sistema.

2.4.2.2 PROBLEMAS ESPECÍFICOS DE INTEGRAÇÃO DE DADOS EM AMBIENTE P2P

Os problemas específicos de integração de dados em ambiente P2P ocorrem desde a submissão da consulta até a integração dos resultados. Porém, conforme visto na seção 2.4.1, as estratégias para integração de dados em ambiente P2P variam, dentre outras coisas, de acordo com a sua arquitetura e características específicas, razão pela qual um problema que afeta um sistema pode ser irrelevante para outro. Esta seção visa analisar os problemas comuns a todo sistema de integração de dados *peer-to-peer*, a saber:

- O que perguntar (WTA – what-to-ask)

Este problema, discutido em [Calvanese et al., 2004], consiste em descobrir o que o *peer* solicitador deve perguntar aos demais *peers* remotos para responder as consultas dos

usuários. O autor comprova que com o auxílio de uma linguagem de ontologia básica, utilizada para expressar o conhecimento entre os *peers*, e de um algoritmo específico para computá-la, é possível resolver este problema. Contudo ele também mostra que, mesmo enriquecendo ligeiramente esta linguagem, o problema nem sempre pode ser resolvido.

- Distribuição/colocação dos dados

Segundo Gribble [Gribble et al., 2001], este problema consiste de como distribuir os dados e trabalho entre os *peers* visando obter o menor custo possível para responder as consultas submetidas pelos usuários. O autor define o custo como sendo o custo de armazenamento, de processamento e de transferência de dados. Ainda são apresentados outros problemas relacionados que o afetam diretamente, como por exemplo heterogeneidade das fontes de dados e granularidade dos dados. Contudo, o autor mostra que este problema, mesmo na sua forma mais simples, é NP - Completo. Entretanto, algumas estratégias já desenvolvidas conseguem oferecer, em parte, um aumento considerável no desempenho do serviço de consulta. Este ganho se faz através da alocação dos *peers* e, conseqüentemente, dos dados em lugares estratégicos, fazendo com que o custo para responder as consultas submetidas pelos usuários seja otimizado.

- Processamento de consultas

Este problema está presente, implicitamente, no serviço de consulta oferecido pelos sistemas de integração de dados *peer-to-peer*. Consiste em obter, da melhor maneira possível, respostas consistentes em relação às restrições semânticas locais e ao mesmo tempo às restrições semânticas dos *peers* envolvidos, conforme verificado em [Bertossi e Bravo, 2004]. Diversas soluções são propostas dependendo da estratégia de integração adotada. Entretanto, um plano de consultas distribuído juntamente com um bom otimizador, assim como uma estratégia eficiente de distribuição de dados e técnicas para resolução de problemas semânticos são alguns pontos que devem ser considerados. Estes são fundamentais para que o processamento de consultas seja semanticamente correto e realizado com o menor custo possível.

- Dinamismo do ambiente P2P

Este problema também afeta os sistemas de integração de dados na WEB através do dinamismo das fontes WEB. Ele se faz à medida que *peers* se unem e se desconectam do sistema com frequência, fazendo com que as referências a estes *peers* fiquem inconsistentes, gerando um alto custo para mantê-las atualizadas. Isto dificulta, dentre

outras coisas, a definição de uma estratégia eficiente para prover a integração de dados entre eles.

Uma possível solução para esse problema é a manutenção de uma tabela de índices, a exemplo de um roteador. Portanto, para localizar um *peer* válido, basta consultar os dados contidos nesta tabela.

A partir dos problemas apresentados, é possível deduzir que algumas das soluções oferecidas necessitam do uso de uma estrutura de informação que possa oferecer algum valor semântico, visando melhorar a compreensão do que realmente o esquema deseja transparecer. O uso de metadados e ontologias despontam como tecnologias importantes na solução do processo de integração de dados nesse ambiente.

2.4.3 METADADOS E ONTOLOGIAS NO SUPORTE A INTEGRAÇÃO DE DADOS

Metadados e ontologias, no contexto da integração de dados, surgem como um conjunto de informações e regras que visam fornecer valor semântico aos dados oriundos das diversas fontes. Metadados, por oferecer de maneira fácil a descoberta, acesso, uso e re-uso dos dados que eles descrevem; e ontologias por possibilitar e facilitar o processo de integração, representando uma garantia do aumento da qualidade e precisão na resolução de conflitos, das consultas e dos esquemas gerados.

A linguagem XML [XML] já se tornou um consenso como linguagem padrão de interoperabilidade para a serialização de dados semi-estruturados ou estruturados na WEB. Porém, com o advento da WEB semântica [Berners-Lee et al., 2001], onde os dados devem ser lidos e interpretados por máquinas, o XML apresenta sérias limitações. A principal limitação é que o XML é puramente sintático, baseado em uma gramática bem definida, o que possibilita diferentes representações para uma mesma descrição. Como XML não impõe regras e permite diversas maneiras de representar um domínio, fica muito difícil reconstruir o documento com o seu significado semântico original [Vdovjak e Houben, 2001].

Dessa maneira, a pouca semântica oferecida pelo XML não é suficiente para resolver os problemas semânticos encontrados nos processos de integração de dados, principalmente depois que este problema se tornou mais complexo devido à necessidade de integrar um número cada vez mais crescente de fontes distribuídas e heterogêneas.

Um problema que antes alcançava poucos sistemas de bancos de dados, hoje atinge milhões de fontes de informação, tornando a interoperabilidade semântica um problema de difícil tratamento [Tavares, 1999]. Neste contexto metadados e ontologias ganham uma posição de destaque.

2.4.3.1 METADADOS

Metadados representam uma informação estruturada que descreve, explica, localiza, facilita a recuperação e uso, assim como o gerenciamento de uma fonte de informação. Metadados são frequentemente chamados de dados sobre dados ou informações sobre informações [NISO, 2004].

Existem 3 tipos principais de metadados [NISO, 2004]: *descritivos*, que descrevem um recurso com o propósito de descobri-lo e identificá-lo, utilizando por exemplo, no caso de um livro, elementos como título, autor e editora; *estruturais*, que indicam como objetos são colocados juntos, por exemplo, como páginas são ordenadas para formar um capítulo; *administrativos*, que provem informações para ajudar no gerenciamento de um recurso, como quando e como ele foi criado.

Metadados podem ser armazenados juntos ou separados do recurso que descreve. Se forem armazenados juntos, assegura que os metadados não correm o risco de se perder dos dados, por outro lado, se forem armazenados separados, podem facilitar o gerenciamento, ao mesmo tempo em que facilita a pesquisa e recuperação dos dados, uma vez que estarão melhor estruturados e mais simples de serem compreendidos.

Muitas iniciativas de padrões de metadados têm sido desenvolvidas para fins bens específicos. Algumas delas podem ser encontradas em [Yin et al., 2003] e [NISO, 2004].

No processo de integração de dados, metadados se destacam por oferecer, de uma maneira fácil, a descoberta, acesso e uso, assim também como o re-uso dos dados que eles descrevem.

2.4.3.2 ONTOLOGIAS

Inúmeras definições conceituando ontologias podem ser encontradas [Moura, Tanaka e Vieira, 2002] e [Corcho, Lopez e Perez, 2002]. Porém, a definição de Gruber [Gruber, 1993], é geralmente a mais citada, definindo esse conceito como sendo uma especificação explícita de uma conceitualização.

Segundo [Guarino, 1998], elas podem ser classificadas em: 1) ontologia de nível superior ou genérica, que descreve conceitos gerais como coisa, espaço, tempo, objeto, assunto, processo, ação e/ou metadados como entidades, relações e atributos; 2) ontologia de domínio, que descreve conceitualizações de domínios particulares, descrevendo o vocabulário relacionado a um domínio qualquer; 3) ontologia de tarefa, que descreve conceitualizações sobre a resolução de problemas, independentemente do domínio em que ocorram, isto é, descreve o vocabulário relacionado a uma atividade ou tarefa qualquer; 4) ontologia de aplicação, que depende tanto de um domínio quanto de uma tarefa particular, podendo ser uma especialização de ambas. Corresponde a papéis desempenhados por entidades do domínio quando da realização de certa atividade.

Para se desenvolver uma ontologia é necessária, além de outras coisas, uma linguagem de definição e um ambiente de desenvolvimento apropriado para a sua criação e gerenciamento. Atualmente, a linguagem OWL – Ontology Web Language [OWL], é a linguagem padrão recomendada pela W3C [W3C] para definição de ontologias, que já se integra a várias ferramentas de gerenciamento de ontologias, tais como o KAON [KAON] e Protégé [Protégé]. Na literatura podem ser encontradas várias outras ferramentas para criação de ontologias [Corcho, Lopez e Perez, 2002] e [Moura, 2002].

Ontologias são de grande importância, pois conseguem embutir significado sem ambigüidade através de vocabulários, permitindo interconexões semânticas entre os termos e regras de inferência e lógica sobre um determinado domínio de conhecimento. Além disso, facilitam a interpretação e recuperação das informações, ao mesmo tempo em que viabilizam o intercâmbio entre sistemas. Possibilitam também um mecanismo de pesquisa mais apurado e restrito às informações realmente relevantes, automação de tarefas que exijam raciocínio e sugestão de opções e caminhos, auxiliando o usuário no alcance dos seus objetivos.

No processo de integração de dados, elas possuem um papel fundamental dentre os quais se pode destacar:

- Explicitação do conteúdo das fontes de dados. Consiste em descrever a semântica referente aos dados das diversas fontes de dados, garantindo assim, um vocabulário compreensível a todos;
- Construção de um modelo de consulta global. Consiste em criar uma grande estrutura que pode ser utilizada para descobrir o que realmente os objetos significam. Auxilia na reescrita da consulta para as fontes ou *peers* relevantes;
- Auxílio lógico. Consistem em definir regras de mapeamentos e restrições de integridade que auxiliam o sistema de integração de dados na extração dos dados e no processamento de consultas, através da derivação lógica destas regras.

Como se pode ver, o uso de ontologias no processo de integração de dados é fundamental para a compreensão semântica dos dados oriundos das diversas fontes. Além de possibilitar e facilitar o processo de integração, este representa uma garantia do aumento da qualidade e precisão na resolução de conflitos e das consultas e esquemas gerados.

2.5 TRABALHOS RELACIONADOS

Existem muitos sistemas relacionados ao tema P2P, porém cada um deles é bem específico segundo características do seu sistema. Esta seção aborda alguns dos principais sistemas nesta área, focalizando principalmente, seu funcionamento e características segundo a arquitetura P2P e o sistema de integração de dados.

2.5.1 BESTPEER - PEERDB

O *PeerDB* [Ooi, Shu e Tan, 2003] é um sistema protótipo de gerenciamento de dados P2P construído no topo do *BestPeer* [Tan et al., 2002]. Este consiste de uma plataforma P2P genérica desenvolvida pela universidade de Singapura, com o objetivo de estudar como tecnologias P2P poderiam ser empregadas para o gerenciamento de dados distribuídos. Tem como objetivo desenvolver, de modo fácil e eficiente, aplicações P2P.

Possui várias características que o distinguem de outros sistemas P2P. Dentre elas pode-se destacar duas relacionadas ao processamento de consultas:

- Tecnologia de agentes móveis. Desde que agentes possam carregar tanto código quanto dado, eles podem efetivamente otimizar qualquer tipo de função. Desta forma, com agentes otimizando operações nos *peers*, a largura de banda da rede pode ser melhor utilizada.
- Reconfiguração dinâmica. Um *peer* pode se reconfigurar dinamicamente referenciando os “melhores” *peers* que o beneficiam como seu vizinho, baseando-se para isso, em uma simples regra: *peers* que se habilitam para responder uma consulta são comumente cotados para responder as consultas subsequentes. Desta forma, *peers* que beneficiam um ao outro serão agrupados e, portanto, consultas podem sempre ser respondidas por vizinhos mais próximos com maior probabilidade.

Sua arquitetura pode ser observada na FIG. 2.7. Esta é constituída por 3 camadas, a saber:

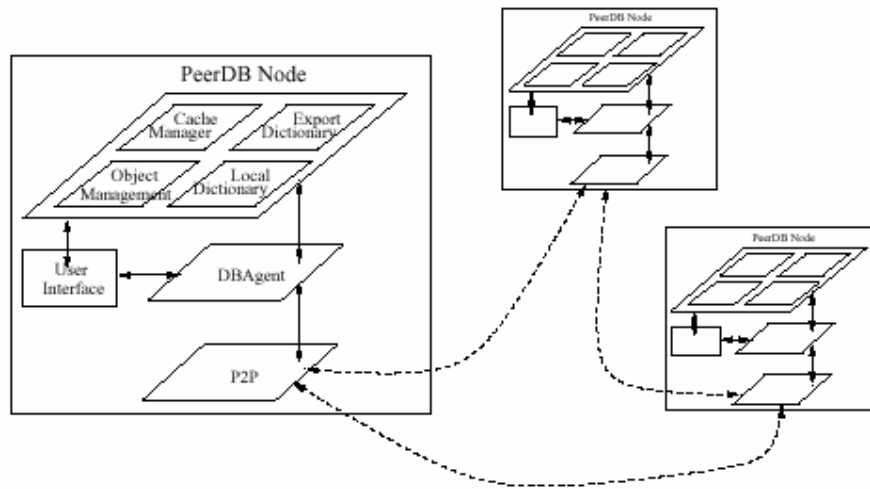


FIG. 2.7 – Arquitetura do *PeerDB*. Fonte: [Ooi, Shu e Tan, 2003].

- Camada P2P: provê funcionalidades da arquitetura P2P, como por exemplo, facilidades na troca de dados e descoberta de recursos;
- Camada de agente: utiliza agentes como “escravos” e provê um sistema de agente de banco de dados, chamado de “DBAgent”, oferecendo um ambiente para agentes móveis operarem. Cada *PeerDB* possui um agente mestre que gerencia a consulta do usuário, isto é, clona e despacha agentes trabalhadores para *peers* vizinhos, recebe respostas e apresenta-as aos usuários. Monitora também as estatísticas e gerencia as políticas de

reconfiguração de rede. Esta camada também é constituída por uma interface amigável, que permite aos usuários submeterem suas consultas e receberem suas respostas utilizando a linguagem SQL;

- Camada de gerenciamento de dados: conhecida também como camada de aplicação, tem como objetivo prover o armazenamento de dados e capacidade de processamento. É formada por quatro componentes que são fracamente integrados. O primeiro deles é o gerenciador de objetos, que tem como principais funcionalidades o armazenamento, manipulação e recuperação de dados do *peer*. É importante relatar que o gerenciador de objetos possui uma interface que apresenta facilidades de consulta SQL. Desta forma, o sistema também pode ser usado como um sistema gerenciador de banco de dados (SGBD) fora do *PeerDB*. O segundo é o dicionário local, que armazena para cada relação criada, os metadados associados. O terceiro componente é o dicionário de exportação que exporta metadados, os quais informam quais dados o sistema deseja que outros *peers* tenham acesso. Pode-se se notar que os metadados contidos no dicionário de exportação são um sub-conjunto dos existentes no dicionário local e desta forma, a distinção entre eles pode ser vista como uma visão lógica do conjunto total. O quarto e último componente é o gerenciador de *cache* que, além de servir como um armazenamento secundário de dados, determina o “caching” e políticas de reposição.

Uma das características do *PeerDB* é que ele visa integrar dados na sua forma relacional (relações/tabelas) e possui uma forte dependência e interatividade com os usuários na submissão de consultas. Assim, para tornar possível o compartilhamento dos dados heterogêneos dos *peers*, o *PeerDB* adota uma estratégia de mapeamento de esquemas. Para cada relação criada pelo usuário, metadados são mantidos para cada nome e atributos da relação. Eles são essencialmente palavras chaves/descrições providas pelo usuário na criação da tabela e servem como um tipo de sinônimo dos nomes e atributos das relações. Desta forma, DBAgentes são enviados para os *peers* com o objetivo de encontrar co-relações e trazer os respectivos metadados. Pode-se observar que, com a combinação de palavras chaves provenientes dos metadados das relações, o *PeerDB* é capaz de encontrar as relações que são potencialmente similares às relações da consulta.

Esta estratégia pode ser ilustrada através do seguinte exemplo. Suponha que existam 3 *peers* que compartilhem dados bibliotecários. *Peer* 1 (P1) define a relação *Livros_Românticos* (Código, Autor, Data_Publicação). *Peer* 2 (P2) define a relação *Livros_Gerais* (ID, Autoria) e por último, *Peer* 3 que define 2 relações denominadas *Livros_Didáticos* (Livro_No,

Autor_Livro) e Livros_Religiosos (Livro_No, Cristão_Responsavel). A TAB. 2.2 mostra as palavras chaves definidas para cada uma das relações de acordo com os respectivos *peers*.

TAB. 2.2 – Tabela de palavras chaves - Projeto *PeerDB*

Peer	Nomes	Palavras-chaves
P1	Livros_Românticos Código Autor Data_Publicação	Livros, Identificação, Código Autor, Autoria, Românticos
P2	Livros_Gerais ID Autoria	Livros, Identificação, Autoria, ID, Gerais
P3	Livros_Didáticos Livro_No Autor_Livro Livros_Religiosos Livro_No Cristão_responsável	Livros, Identificação, Numero, Autor, Didáticos, religiosos, Cristão

Agora, suponha que o usuário do *peer* 1, que conhece seu esquema, mas não o dos outros *peers*, submeta a seguinte consulta:

```
SELECT Código, Autor
FROM Livros_Românticos
WHERE Autor = 'Maria '
```

Como uma das palavras chaves para a relação Livros_Românticos é “Livros” e “Livros” também é uma palavra chave da relação do *peer* 2, assim como das relações do *peer* 3, essas relações podem ser mapeadas para a relação da consulta. Similarmente, pode-se verificar que os atributos da operação de seleção, código e autor, também possuem relação com as palavras chaves persistidas nos *peers* 2 e 3. No caso, código = identificação e autor = autoria. Contudo, note que para o *peer* 3, a consulta deve ser refeita em uma consulta de junção quando a ele submetida.

Semanticamente, pode-se observar que os dados residentes no *peer* 2 e 3 podem não ser os dados que o *peer* 1 está interessado, uma vez que eles podem ser livros que não sejam

românticos. Desta forma, é importante ter os metadados e, se possível, informações adicionais retornadas para o usuário antes de serem recuperadas das fontes originais.

Porém, com o intuito de adicionar maior valor semântico às relações, e desta forma minimizar os problemas semânticos existentes, o *PeerDB* faz uso de dois mecanismos. O primeiro consiste em explorar mais o uso dos metadados, como por exemplo, das afirmações de que dois atributos com significados semelhantes são mais prováveis para serem semanticamente similares. O segundo, por sua vez, diz respeito ao domínio dos atributos. Este pode prover alguma percepção, isto é, atributos não podem ser combinados se seus domínios não são compatíveis. Contudo, existe ainda a utilização de um tesouro [Gomes, 1990], porém este ainda se encontra em fase de desenvolvimento.

Como visto, o *PeerDB* faz uso de agentes para auxiliar no processamento de consultas. Esta estratégia é dividida em duas fases, a saber:

- Primeira fase: é utilizada a estratégia apresentada (mapeamento das relações), onde as relações pertinentes à consulta são retornadas para o *peer* que submeteu a consulta visando dois objetivos. O primeiro é permitir que usuários selecionem as relações mais relevantes, minimizando a sobrecarga de informações, uma vez que os dados podem ser sintaticamente o mesmo, mas semanticamente diferentes. Isto pode minimizar as transmissões de dados que não são úteis para o usuário, e conseqüentemente melhorar a utilização da largura de banda da rede. O segundo é permitir que o *peer* atualize sua estatística para facilitar pesquisas futuras.
- Segunda fase: tem início após o usuário ter selecionado as relações desejadas. Assim, as consultas podem ser reescritas e conseqüentemente direcionadas para os *peers* que contém as relações selecionadas, e as respostas são finalmente retornadas e armazenadas. Esta fase é completamente suportada por agentes. De fato, é o próprio agente que submete as consultas aos *peers* sendo ele mesmo quem interage com o sistema gerenciador de banco de dados.

O *PeerDB* adota, no seu sistema de integração de dados, uma estratégia de mapeamento de relações que, através do auxílio de uma tabela de palavras chaves (metadados) e de um tesouro (trabalho em andamento), consegue combinar relações baseadas nestas palavras que são comuns dentre as relações existentes, permitindo que consultas sejam enviadas aos *peers* relevantes, e ao mesmo tempo, resolvendo os problemas semânticos ocorridos, bem como possibilitando o compartilhamento dos dados residentes nos diversos *peers*. Para aprimorá-lo

ainda mais, existem planos para explorar questões relacionadas ao “caching” de dados assim como o processamento e otimização de consultas.

2.5.2 HYPERION

O projeto Hyperion [Arenas et al., 2003] está sendo desenvolvido pela Universidade de Toronto e Ottawa, no Canadá. Seu principal foco está centrado na especificação e no gerenciamento de metadados que permitam o compartilhamento e a coordenação de dados entre *peers* autônomos e independentes. Desta forma, assim como visões e mapeamentos GLAV (visão local e global) têm sido usados para integrar e trocar dados em um domínio comum, o projeto Hyperion considera que metadados são requeridos para compartilhar dados entre múltiplos contextos e, portanto, apresenta novas soluções para especificar e gerenciar tabelas de mapeamento. Uma tabela de mapeamento associa dados residentes em *peers* diferentes, provendo assim mecanismos superficiais para o compartilhamento de dados.

Além da tabela de mapeamento, o projeto Hyperion apresenta outros elementos chaves como a coordenação de dados e o mecanismo de regras. A coordenação de dados, entre outras coisas, envolve a reconciliação e integração de dados em tempo de execução da consulta e a manutenção da consistência dos dados contidos nos diferentes *peers*. Já o mecanismo de regras é utilizado para a criação e manutenção das tabelas e expressões de mapeamento.

Conforme se pode observar na FIG. 2.8, sua arquitetura é composta por três camadas, a saber:

- Camada de interface: permite que usuários submetam suas consultas e especifiquem se elas devem ser executadas somente localmente ou se os dados residentes em outros *peers* também devem ser considerados;
- Camada de dados (persistência): contém os dados locais (representados na forma relacional) e as tabelas e expressões de mapeamento, as quais são fundamentais para a troca de dados com outros *peers*;
- Camada P2P: é a mais importante do sistema e é formada por 3 componentes: o gerenciador de entendimento, responsável por estabilizar um entendimento entre *peers* semi-automaticamente, criando tabelas ou expressões de mapeamentos entre eles. Essas estruturas representam o conhecimento sobre os *peers* dentro de um determinado domínio e são tipicamente criadas manualmente por usuários administradores, os quais as

especificam através da assistência de um conjunto de regras naturais e metadados [Kementsietsidis et al., 2003]; gerenciador de consultas, responsável por executar as consultas locais ou globais. No caso de consultas globais, o gerenciador de consultas reescreve a consulta de acordo com os *peers* envolvidos, utilizando para isso, serviços do gerenciador de entendimento; e gerenciador de regras, responsável pelas políticas de consistência entre *peers*. Essas políticas são especificadas declarativamente através da camada de interface e possuem o formato ECA (evento-condição-ação).

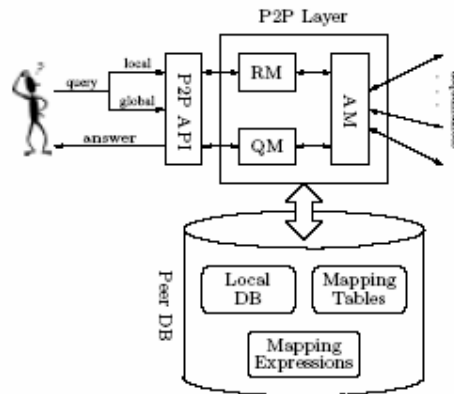


FIG. 2.8 – Arquitetura do Hyperion. Fonte: [Arenas et al., 2003].

Para possibilitar o compartilhamento dos dados heterogêneos dos *peers*, o Hyperion adota uma estratégia que consiste no mapeamento de expressões e de tabelas de mapeamento. O funcionamento desta estratégia pode ser demonstrado a partir das instâncias apresentadas na FIG. 2.9, baseada em [Arenas et al., 2003]. Considere a existência de 2 *peers* sobre o domínio de companhias aéreas, denominados de *peer 1* (companhia aérea X) e *peer 2* (companhia aérea Y).

O mapeamento de expressões é feito a partir de generalizações das expressões GLAV. Entretanto, essas expressões não são usadas para definir ou restringir um conjunto de instâncias válidas dos *peers*. Preferivelmente, são utilizadas como restrições para a troca de dados. Por exemplo, a expressão de mapeamento X_Passageiro (cód, nome) contém Y_Passageiro (cod, nome) que indica que todo passageiro do *peer 2* (companhia Y) deve ser considerado também um passageiro do *peer 1* (companhia X), isto é, uma consulta submetida ao *peer 1* que pergunte por todos os seus passageiros deve também ser submetida ao *peer 2*, de forma a recuperar todos os passageiros que nele persistem.

X_Passageiro

Cód.	Nome
1	João
2	José
3	Maria

X_Ticket

Cód.	N_vôo
1	X120
2	X130

X_Vôo

N_vôo	data	Origem	Destino	Vendidos
X120	01/12/04	Rio	São Paulo	130
X130	05/12/04	Rio	Recife	60
X140	17/02/05	Rio	Maranhão	100

*Peer 1 – Companhia X***Y_Passageiro**

Cód.	Nome
1	Caio
2	Joana
3	Carla

Y_Reserva

Cód.	N_vôo
1	Y125
2	Y439
3	Y512

Y_Frota

	Modelo	Capacidade
Y1	Boeing 747	360
Y2	Boeing 747	340
Y3	Boeing 777	400

Y_Vôo

N_vôo	data	Origem	Para	Vendidos	Aeronave
Y125	02/12/04	Galeão	CNG	100	Y2
Y439	09/12/04	S. Dumont	IRE	80	Y1
Y512	15/01/05	Galeão	IBA	110	Y3

CNG – Aeroporto de Congonhas – SP
 IRE – Aeroporto Internacional do Recife – PE
 IBA – Aeroporto Internacional da Bahia – BA

*Peer 2 – Companhia Y*FIG. 2.9 – Instâncias dos *peers* 1 e 2 - Projeto Hyperion

O mapeamento de tabelas, por sua vez, provê a definição de correspondências entre valores dos diferentes *peers*. Estas correspondências possuem uma importância particular, uma vez que no ambiente P2P, aonde não existe um padrão de nomes, os *peers* necessitam e dependem do uso de convenção de nomes internos. Como exemplo, pode-se definir os seguintes mapeamentos (FIG. 2.10):

Destino	Para
São Paulo	CNG
Recife	IRE

Tabela de mapeamento 1

N_Vôo X	N_Vôo Y
X120	Y125
X130	Y439

Tabela de mapeamento 2

FIG. 2.10 – Mapeamento de tabelas - Projeto Hyperion

A tabela de mapeamento 1 associa nomes de cidades da companhia X com os códigos dos aeroportos da companhia Y, enquanto que a tabela de mapeamento 2 associa vôos das duas companhias quando os respectivos vôos possuem a mesma cidade de destino. Pode-se observar que a tabela de mapeamento 1 não apresenta a cidade destino “Maranhão” da companhia X, uma vez que esta não está associada a nenhum código de aeroporto presente na companhia Y. O mesmo acontece para a tabela de mapeamento 2, onde o vôo “X140” da companhia X e o vôo “X512” da companhia Y não fazem referência a mesma cidade de destino.

Contudo, uma vez que os *peers* já se conhecem (foi feito o entendimento) e as tabelas de mapeamentos e os mapeamentos de expressões tenham sido feitos, os *peers* já podem submeter consultas ao sistema. Desta forma, suponha que o *peer* 1 submeta a seguinte consulta:

```
SELECT data
FROM X_Vôo
WHERE destino = 'São Paulo'
```

O gerenciador de consultas deve então reescrevê-la, de modo que esta possa ser submetida e entendida pelos demais *peers* relevantes, no caso o *peer* 2. As tabelas de mapeamentos e os mapeamentos de expressões são utilizados com essa finalidade e, portanto, a consulta é reescrita e submetida ao *peer* 2 da seguinte forma:

```
SELECT data
FROM Y_Vôo
WHERE para = 'CNG'
```

Entretanto, as tabelas de mapeamentos devem sempre estar atualizadas de modo a manterem-se consistentes. Esta manutenção é feita através de regras específicas. Detalhes sobre a manutenção das tabelas de mapeamento, assim como das funcionalidades de coordenação dos *peers* podem ser encontradas com mais detalhes em [Arenas et al., 2003].

Concluindo, o projeto Hyperion é um sistema que, através da especificação e do gerenciando de metadados, coordena e compartilha dados entre sistemas gerenciadores de banco de dados *peer*. A estratégia utilizada, por sua vez, caracteriza-se pelo uso de tabelas e expressões de mapeamento, assim como de mecanismos de coordenação de dados e de regras.

2.5.3 PIAZZA

O projeto Piazza [Tatarinov et al., 2003] está sendo desenvolvido na Universidade de Washington e da Pensilvânia, nos Estados Unidos. É um sistema de gerenciamento de dados P2P que provê o compartilhamento de dados heterogêneos em um ambiente distribuído e escalável.

Nele, cada *peer* possui um esquema representando sua “visão do mundo” que é, provavelmente, diferente das demais representadas pelos *peers* participantes. Desta forma, para possibilitar o compartilhamento dos dados, o sistema de integração de dados utiliza uma estratégia de mapeamentos de esquemas. Ele assume que os *peers* estão interessados em compartilhar seus dados e, desta forma, dispostos a definir os respectivos mapeamentos entre seus esquemas. O conjunto desses mapeamentos é que definirá a semântica do sistema. Assim, quando um usuário de um *peer* submete uma consulta sobre um esquema, o sistema de respostas de consultas expande recursivamente qualquer mapeamento relevante para o consulta, recuperando dados dos *peers* relevantes, que são então enviados como respostas aos usuários.

É importante ressaltar que, embora não tenha sido encontrada uma descrição sucinta dos componentes de sua arquitetura, esta pode ser definida através do próprio sistema em funcionamento, destacando-se principalmente o mapeamento e processamento de consultas.

Os mapeamentos de esquema, por sua vez, podem ser de dois tipos: descrição do *peers* e descrição dos dados armazenados. A descrição dos *peers* é feita através do mapeamento de dois ou mais esquemas, onde são definidas as correspondências entre “visões do mundo” de diferentes *peers*. A descrição de dados armazenados é o mapeamento de um esquema armazenado para um esquema de *peer*. Estes mapeiam os dados armazenados em um *peer* para “visões do mundo” de cada *peer*.

A FIG. 2.11 apresenta um exemplo simplificado onde ocorrem mapeamentos de esquemas. Os dados persistidos nas bases de dados são mapeados para o esquema de cada *peer*, referenciados como descrição dos dados armazenados. Acima deles, um novo esquema de *peer* é criado (descrição de *peers*), o qual se caracteriza por ser um *peer* virtual e sem dados para estocar, composto somente pelo mapeamento dos esquemas dos *peers* de origem, no caso, *peer* PUC e *peer* UERJ.

A construção dos mapeamentos é realizada através de um processo constituído de duas fases. A primeira, chamada de comparação de esquemas, consiste em descobrir similaridades

ou um conjunto de correspondências dentre os esquemas, as quais identificam elementos similares que podem ser mapeados. Por exemplo, uma comparação entre as disciplinas das instituições PUC e UFRJ (ver FIG. 2.11) poderia incluir a correspondência: PUC.disciplina ~ UFRJ.matéria. A segunda fase objetiva aperfeiçoar os mapeamentos realizados pela primeira fase, de modo que estes fiquem mais precisos. Para tal, as correspondências são selecionadas e, sobre estas, são aplicadas técnicas automáticas de combinação e até mesmo intervenções humanas.

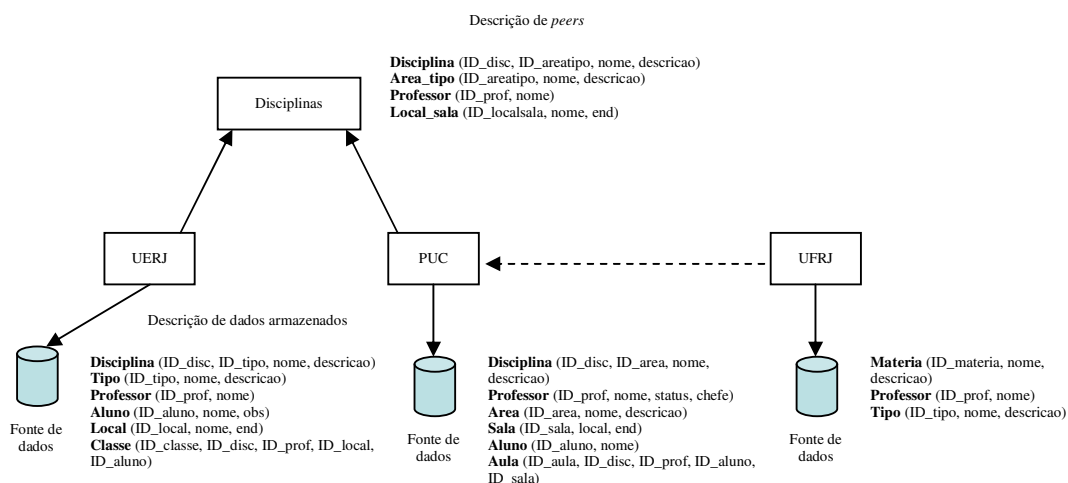


FIG. 2.11 – Exemplo simplificado de mapeamentos de esquemas – Projeto Piazza

Contudo, pode-se verificar que as correspondências concretizadas pela primeira fase são simplesmente declarações de similaridade e não tem, praticamente, nenhuma semântica envolvida. Desta forma, com o intuito de minimizar os problemas semânticos que possam vir a ocorrer, o Piazza adota uma abordagem que se caracteriza por duas propriedades chaves: o uso conjunto de heurísticas e algoritmos e a exploração de experiências anteriores. A primeira se baseia em técnicas de aprendizagem de máquina. A segunda utiliza técnicas de exploração de experiências passadas, onde informações sobre mapeamentos válidos já existentes são reutilizadas para o mapeamento de novos esquemas.

O serviço de consulta do projeto Piazza é bem específico. Utiliza um algoritmo próprio de reformulação de consultas [Halevy et al., 2003] que, além de outras coisas, apóia-se nos mapeamentos de esquemas (que embutem significado semântico aos elementos) para reescrever as consultas, garantindo que as mesmas sejam submetidas somente aos *peers* realmente capazes de respondê-las. Uma estrutura de índices está sendo desenvolvida para auxiliar na identificação e otimização de busca. Desta forma, ao receber uma consulta, o

serviço de consulta expande recursivamente qualquer mapeamento relevante para o consulta, recuperando os dados dos *peers* relevantes, que são então enviados como respostas aos usuários.

A exemplo de vários sistemas gerenciadores de banco de dados P2P, o Piazza também está interessado em explorar a definição destes mapeamentos e de informações com maior riqueza semântica, de modo a aprimorar sua representação do conhecimento e conseqüentemente a reformulação de consultas.

2.5.4 EDUTELLA

O projeto Edutella [Edutella] caracteriza-se como uma plataforma múltipla para estender, especificar e implementar uma infra-estrutura de metadados em RDF [RDF] para redes P2P, visando o compartilhamento de recursos educacionais. Ele foi desenvolvido em cima do JXTA [JXTA], um *framework* para implementação de aplicações P2P e provê alguns serviços específicos, dentre os quais resumidamente, pode-se destacar:

- Serviço de mapeamento: provê a tradução entre diferentes vocabulários de metadados dos *peers*, permitindo assim a interoperabilidade entre eles. Gerencia os mapeamentos entre os diferentes esquemas e utiliza esses mapeamentos para traduzir consultas em cima de um esquema X para consultas em cima de um outro esquema Y. São realizados através do uso de regras e mecanismos de transformações locais;
- Serviço de Mediação: define visões que unem dados provenientes de diferentes fontes de metadados e reconcilia informações conflitantes;
- Serviço de replicação: provê persistência dos dados, disponibilidade e balanceamento do “workload”, enquanto mantém a integridade e consistência dos mesmos;
- Serviço de roteamento: roteia as consultas que recebe para os *peers* e *super-peers* apropriados;
- Serviço de anotação: anota informações sobre dados armazenados na rede Edutella;
- Serviço de agrupamento: utiliza informações semânticas para conectar e agrupar *peers*;
- Serviço de consulta: é o serviço mais básico dentro da rede Edutella. Provê uma interface para a submissão de consultas e utiliza outros serviços e estratégias para o envio e tradução das mesmas.

O projeto Edutella é um sistema mais robusto do que os demais vistos. Porém, antes de o detalharmos, faz-se necessário apresentar algumas características quanto a sua arquitetura e funcionamento. No Edutella, cada *peer* descreve seus recursos (dados e esquema) através da linguagem RDF e RDFS [RDFS] e uma linguagem de consulta denominada RDF-QEL [Nejdl et al., 2002] é utilizada como linguagem de troca de consultas. Esta serve como formato de intercâmbio de consultas comum, utilizadas para tradução pelas linguagens de consulta locais. Uma estrutura baseada em *wrapper* é responsável por traduzir linguagens de consulta local para o modelo de consultas padrão Edutella [Nejdl et al., 2002].

Sua arquitetura é baseada na arquitetura de *super-peers*. Cada um deles emprega índices de roteamento que, explicitamente, reconhecem a heterogeneidade semântica das redes P2P baseada em esquemas. Por essa razão incluem, além das informações usuais dos índices, informações sobre estes esquemas. Também são responsáveis pelo roteamento de mensagens e integração / mediação dos metadados. Estendendo um pouco mais, eles estão organizados dentro da topologia HyperCuP [Schlosser et al., 2002], conforme ilustra a FIG. 2.12. Esta topologia capacibilta a organização dos *peers* em uma estrutura de grafo recursiva, permitindo consultas eficientes e garantindo que as mesmas não sejam redundantes.

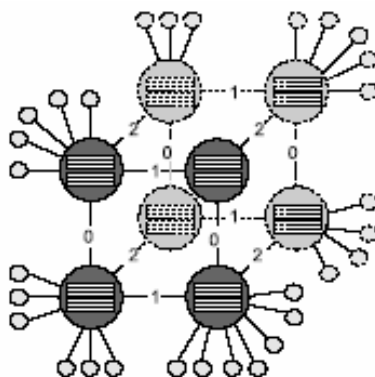


FIG. 2.12 – *Super-peers* dentro da topologia de HyperCuB

Pode-se definir os índices de roteamento como sendo de dois tipos:

- *Super-peer/Peer* (SP/P): Contém as informações que um *super-peer* deve saber a respeito dos *peers* que se encontram a ele conectado. Dentre tais informações pode-se destacar as que dizem respeito à localização, esquemas, conteúdos e metadados. Elas são mantidas enquanto os *peers* se registram e/ou atualizam suas informações periodicamente. O fato desses procedimentos não ocorrerem indica que o *peer* deixou o

sistema e que todas as suas referências devem ser removidas. As atualizações são feitas através de *triggers*, acionados por uma notificação enviada pelo *peer*;

- *Super-peer/Super-peer (SP/SP)*: refere-se às informações que um *super-peer* deve saber a respeito dos *super-peers* ao qual está conectado. Estas informações são extraídas e resumidas dos índices SP/P, e por essa razão contém basicamente o mesmo tipo de informação. Atualizações são baseadas em mensagens dos *peers* conectados, isto é, sempre que um índice SP/P é atualizado, esta mudança deve ser propagada para o índice SP/SP em questão. As consultas são enviadas para os *super-peers* vizinhos baseadas nas informações contidas neste índice (SP/SP) e são reenviadas aos *peers* conectados com base nos índices SP/P.

Peers são organizados no sistema através de uma estratégia de agrupamento de *peers* baseada em algumas características particulares, possibilitando assim, que os *peers* com características semelhantes estejam fisicamente agrupados, permitindo, dentre outras coisas, que o processamento de consulta possa ser otimizado.

A rede Edutella, assim como outros sistemas gerenciadores de dados *peer*, não utiliza um esquema global para prover a integração dos dados. Isto se deve ao fato de que os seus *peers* possuem esquemas heterogêneos e se unem e saem da rede a todo o momento. Desta forma, para prover a integração dos seus dados, o sistema Edutella utiliza uma estratégia baseada nos índices de roteamentos, metadados e mapeamento de esquemas.

Assim, quando um *super-peer* recebe uma consulta, o serviço de consulta, utiliza-se das funcionalidades P2P e das informações contidas no seu índice de roteamento para encontrar onde os dados estão localizados, identificando os *peers* relevantes à consulta. A seguir, essa informação é utilizada, assim como as informações de metadados e mapeamentos de esquemas e as contidas no índice de roteamento para geração de um plano de consulta distribuída. A ideia principal é fazer com que cada *peer*, ao receber a consulta, submeta-a localmente e repasse-a ao seu *super-peer* e *peers*, para que os mesmos possam respondê-las.

A FIG. 2.13 mostra um exemplo da utilização desta estratégia. Após serem feitos os mapeamentos e tabelas de índices, o *peer* 1 (P_1) pode submeter consultas. Desta forma, a consulta C_1 “Quais são os livros (título e autor) que tratam do assunto de banco de dados?” submetida por P_1 é traduzida para que possa ser submetida ao seu *super-peer* e, a partir deste, aos demais *peers* do agrupamento e aos demais *super-peers* relevantes do sistema, no caso SP_3 e seu *peer*, e assim sucessivamente. A tradução da consulta é suportada, principalmente,

pelos mapeamentos dos esquemas e metadados, seleção dos *super-peers/peers* relevantes para respondê-las e pelo índice de roteamento de cada um deles.

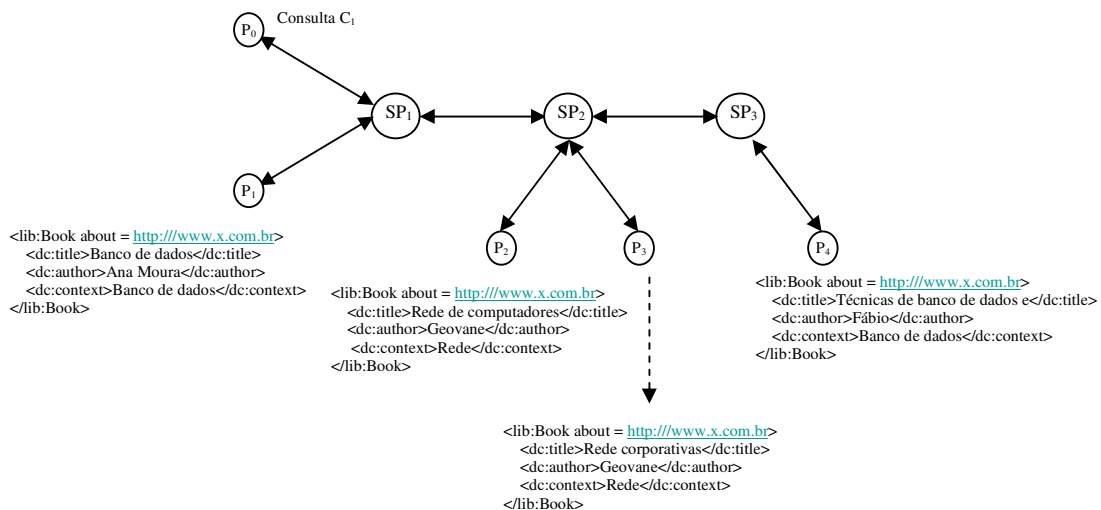


FIG. 2.13 – Exemplo de submissão de uma consulta – Rede Edutella

Além destas, muitas outras questões são abordadas pelo sistema Edutella. Referem-se a estratégias de mediação, linguagem de esquemas/consultas e tradução das consultas, as quais podem ser encontradas com mais detalhes em [Nejdl et al., 2003]. Contudo, devido principalmente à sua robustez, o sistema Edutella destaca-se dos demais, uma vez que utiliza tecnologias mais recentes como RDF e arquitetura baseada em *super-peers*, assim como estratégias bem mais complexas e elaboradas.

2.5.5 CONSIDERAÇÕES FINAIS

Os projetos apresentados possuem as principais características de um sistema gerenciador de dados P2P, dentre as quais pode-se destacar escalabilidade, tolerância a falhas, confiabilidade e mesmo grau de autonomia dos *peers*, além de serem dotados de mecanismos para tratamento de dados heterogêneos e de rico valor semântico.

Quanto ao sistema de integração, apesar de utilizarem estratégias distintas, possuem características bem similares principalmente com relação à resolução de conflitos semânticos. Para tal, metadados e tesauros se destacam como ferramentas utilizadas para embutir valor

semântico aos dados, possibilitando desta forma, que a exploração do conhecimento e mapeamentos de esquemas possam ser feitos para a realização da integração dos dados.

Contudo, com a finalidade de melhor destacar suas especificações, a TAB. 2.3 apresenta um resumo das principais funcionalidades e características de cada um deles, a saber:

TAB. 2.3 – Comparação entre alguns dos principais sistemas P2P atuais

Sistema	Arquitetura P2P	Representação de dados	Ferramenta semântica	Mapeamento de esquemas	Processamento de consultas	Linguagem de consulta	Particularidades
PeerDB	Descentralizada	Relacional	Metadados e Tesouro	Através do uso de tabelas com palavras-chaves	Uso de DBAgents, os quais gerenciam o processamento. Forte interação com o usuário	SQL	Estratégia baseada na junção de palavras-chaves
Hyperion	Descentralizada	Relacional	Metadados	Através do uso de tabelas de mapeamento	Reescrita da consulta segundo a tabela de mapeamento	SQL	Coordenação de dados em tempo de execução
Piazza	Descentralizada	XML e XMLS	Metadados	Através do uso de heurísticas e algoritmos	Reescrita com base nos mapeamentos. Utiliza uma estrutura de índices particular	XQuery	Utiliza técnicas de aprendizagem de máquina e exploração de experiências anteriores
Edutella	<i>Super-peers</i>	RDF e RDFS	Metadados	Através do uso de mecanismos de transformações locais e regras	Geração de um plano de consulta distribuído, baseado no índice de roteamento e nos mapeamentos	RQL-QEL	Utilização de índices entre SP/P e SP/SP, serviço de agrupamento de <i>peers</i> e uso de um modelo de consultas padrão próprio

Como é possível observar, as estratégias de integração de dados abordadas variam principalmente quanto as técnicas utilizadas para prover os mapeamentos de esquemas e as estratégias adotadas para prover o processamento de consultas. Isto ocorre porque cada

sistema P2P possui objetivo, arquitetura e características particulares, as quais refletem diretamente no funcionamento dos respectivos sistemas de integração de dados (ver seção 2.4.1), tornando difícil a escolha de um “melhor” ou “mais apropriado” sistema de integração de dados.

Contudo, dentre os sistemas apresentados, o Edutella é o que mais se sobressai, uma vez que utiliza a arquitetura baseada em *super-peers*. Conforme descrito na seção 2.2.2.1, esta arquitetura se destaca pela maior agilidade no processamento de consultas e na autonomia de gerenciamento. Seu sistema de integração de dados também se sobressai, visto que o uso de RDF na integração de dados já embute alguma semântica nesta representação, além do que considera a integração de dados numa arquitetura de *super-peers*, refletindo a tendência das pesquisas atuais na área.

Algumas características e funcionalidades apresentadas por esses sistemas serviram de base para a definição de alguns pontos importantes no ROSA - P2P, dos quais podem ser citados: uso dos índices de roteamento SP/P e SP/SP (específico do projeto Edutella), reescrita das consultas e localização de *peers* relevantes às consultas.

Sendo assim, o estudo dos principais sistemas atuais sobre gerenciamento de dados P2P foi de suma importância no desenvolvimento do ROSA - P2P, sugerindo e incentivando idéias até então não observadas.

3. SISTEMA ROSA

Este capítulo tem como objetivo apresentar o sistema ROSA, destacando suas características mais importantes e oferecendo uma visão geral do seu funcionamento.

Para um melhor entendimento, este capítulo será dividido em 4 seções. A seção 3.1 é a primeira delas e permite o entendimento do sistema ROSA de forma simples, porém abrangente. Logo após, a seção 3.2 exibe a arquitetura do sistema, especificando-a em alguns níveis. Em seguida, a seção 3.3 discute a modelagem de LOs, apresentando o conceito de mapa conceitual. Finalmente na seção 3.4, é apresentado o modelo de dados ROSA.

3.1 VISÃO GERAL

O sistema ROSA (Repository of Objects with Semantic Access) [Porto, Moura et al., 2003] é um sistema voltado para a área de Ensino a Distância (EAD), que se propõe a auxiliar profissionais da área educacional, ajudando-os na busca de conteúdos didáticos armazenados no sistema que forneçam base para preparação de suas aulas e/ou conteúdos instrucionais. Tem como objetivo armazenar objetos de aprendizagem – Learning Objects (LOs), que representam de fato os conteúdos instrucionais, e explorar seu acesso de acordo com o contexto semântico em que foram criados. Este contexto é determinado por relacionamentos semânticos entre LOs e expressos através de um mapa conceitual, a partir de um modelo de dados bem definido.

O ROSA utiliza o banco de dados XML nativo Tamino¹ e oferece uma álgebra e uma linguagem de consulta própria, a ROSAQL [Porto, Moura e Silva, 2004], através das quais consultas de conotação mais semântica podem ser feitas a exemplo de: “que tópicos *compreendem* a disciplina de banco de dados?”; “que disciplinas são *pré-requisito* para o ensino de Redes?” Nesses exemplos, *compreendem* e *pré-requisito* fazem parte de um conjunto de predicados pré-definidos que associam os diversos LOs. Questões relacionadas a sinônimos, termos específicos/genéricos e associados são resolvidos por tesauros de domínio, que auxiliam no processamento de consultas.

¹ <http://www.softwareag.com/tamino/>

3.2 ARQUITETURA DO SISTEMA ROSA

A arquitetura do sistema ROSA está baseada na arquitetura tradicional de banco de dados [Elmasri e Navathe, 2000]. Ela tem por objetivo criar e armazenar LOs, além de prover suporte à recuperação e pesquisa de LOs segundo determinadas características [Porto, Moura, Fernandez et al., 2003].

A FIG. 3.1 exibe a arquitetura do ROSA. Conforme pode-se observar, ela é especificada através de 4 níveis, definidos como:

- **Nível externo:** corresponde ao sistema na forma tal como é exibido ao usuário;
- **Nível conceitual:** corresponde à definição do modelo conceitual. Define os conceitos e os relacionamentos em certo domínio de aplicação, utilizando para isso uma representação através de mapas conceituais;
- **Nível lógico:** corresponde à definição do modelo de dados. Define os objetos de aprendizagem e seus metadados;
- **Nível físico:** corresponde à camada de persistência dos dados. Define como e onde os dados serão armazenados em meio físico.

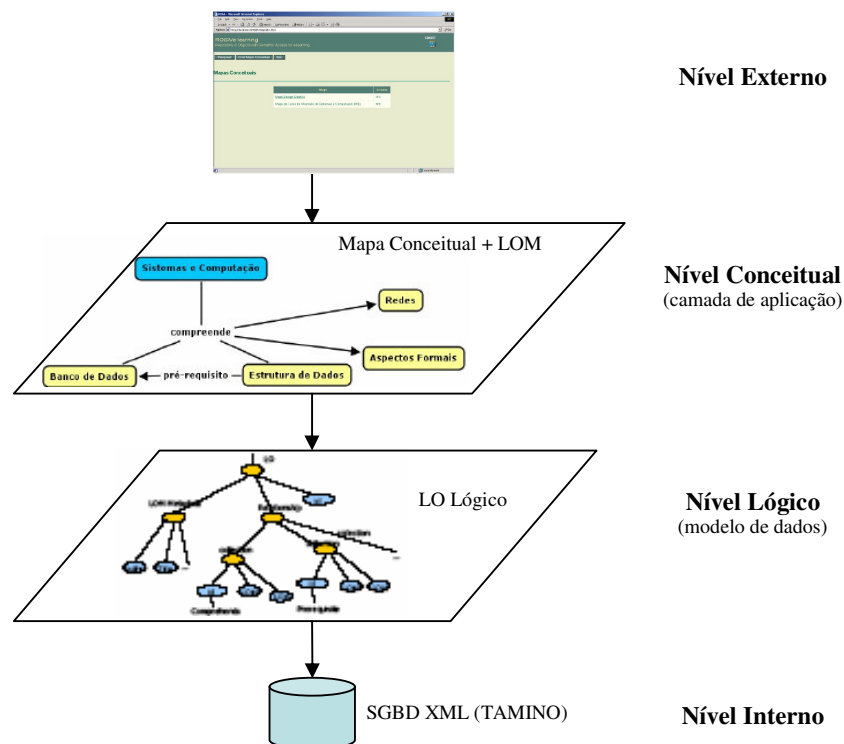


FIG. 3.1 – Arquitetura do sistema ROSA

3.3 MODELAGEM DE LOs

No contexto do ROSA, um LO é uma estrutura de dados bem definida para representar conteúdos instrucionais. Eles podem ser de dois tipos: **LO Lógico** e **LO Físico**. Um LO Lógico representa uma coleção de LOs, podendo conter muitos LOs físicos. Estes se associam a outros LOs através de predicados semânticos; e o LO Físico, que é o próprio LO armazenado em meio físico, a exemplo de imagens, vídeos ou documentos do tipo .doc, .ppt, etc. Porém, estes não são representados nos mapas conceituais.

Os LOs são constituídos, na sua estrutura, por um conjunto de características e propriedades, denominadas de **metadados**, e por um conjunto de **associações**. Os metadados são baseados numa extensão do padrão IEEE-LOM², e contém informações tais como *identificador, título, palavras chaves, idioma, versão, nível de agregação*, etc. As associações são predicados que expressam os relacionamentos que um LO tem com outro(s) LO(s). Estes são caracterizados segundo um domínio específico e são classificados em dois tipos: **predicados de agregação**, a exemplo de *compreende, abrange, é formado por, é constituído por, possui, caracteriza, etc*; e **predicados de associação** (ou predicados semânticos), a exemplo de *pré-requisito, fundamenta, é base para, necessita, requer, determina, gera, influencia, usa, provê, etc*. Predicados podem também contemplar propriedades de equivalência específicas, tais como: *reflexiva, simétrica e transitiva*.

Desta forma, associações (expressas através de predicados) são definidas por um nome, que as caracterizam e pelo tipo de relacionamento aos quais pertencem. Assim, é possível utilizar vários predicados num mesmo mapa conceitual significando o mesmo tipo de relacionamento. Esta característica possibilita um uso mais flexível sobre a forma pela qual um conteúdo pode ser expresso, sem perder a semântica sobre a relação [Fernandez, 2004].

A FIG. 3.2 apresenta um exemplo parcial de uma base de dados para o sistema ROSA. Pode-se observar a presença de um LO lógico e de um LO físico, assim como dos respectivos metadados e relacionamentos (somente no caso do LO lógico).

```
.  
. .  
. .  
. .  
<LOLógico>  
  <identificador>001</identificador>  
  <geral>  
    <título>banco de dados</título>
```

² <http://ltsc.ieee.org/>

```

<idioma>pt-BR</idioma>
<descrição>disciplina do curso de banco de dados</descrição>
<nívelDeAgregação>disciplina</nívelDeAgregação>
</geral>
<cicloDeVida>
  <status>final</status>
  <contribuinte>
    <papel>autor</papel>
    <data>01/05/2005</data>
  <entidade>
    <idEntidade>A-001</idEntidade>
    <nome>Ana Maria Moura</nome>
  </entidade>
</contribuinte>
</cicloDeVida>
<associações>
  <predicado>
    <nome>compreende</nome>
    <objetos>
      <id>002</id>
      <id>003</id>
      <id>004</id>
    </objetos>
  </predicado>
</associações>
</LOLógico>
<LOFísico>
  <geral>
    <identifíer>004</identifíer>
    <título>QUEL</título>
  </geral>
  <aspectosTécnicos>
    <formato>doc</formato>
    <localização>c:/ROSA/arquivos</localização>
  </aspectosTécnicos>
</LOFísico>
.
.
.

```

FIG. 3.2 – Exemplo de LO lógico e físico

Neste exemplo, o LO lógico corresponde a uma *disciplina* denominada de *banco de dados* cuja autoria é de *Ana Maria Moura*. Estas informações estão representadas através dos metadados título e nível de agregação em geral, e pelo nome (do contribuinte) em ciclo de

vida. Este LO também possui 3 associações definidas com os LOs 002, 003 e 004 através do predicado *compreende*. Esta informação se encontra representada através do metadado id (do o outro LO) em associações -> predicado -> objetos. Dentre esses LOs, pode-se verificar que o de id 004 é um LO físico, cujo arquivo possui a extensão *doc* e está localizado em *c:/ROSA/arquivos*. Estas informações são representadas através dos metadados formato e localização em aspectos técnicos.

Contudo, a representação dos metadados dos LOs e das associações semânticas existentes entre eles formam um estrutura rica em conhecimento, correspondente ao mapa conceitual, que permite consultas semânticas e visões abstratas dos recursos de informações. Este por sua vez é representado por um grafo direcionado, onde os nós representam os LOs (que podem ser um curso, disciplina ou tópico), identificados pelos seus nomes, e as arestas representam relacionamentos entre eles, a exemplo dos predicados em RDF (Resource Description Framework).

Com o intuito de esclarecer esta representação, considere a FIG. 3.3, que representa um mapa conceitual parcial do Curso de Mestrado em sistema e Computação.

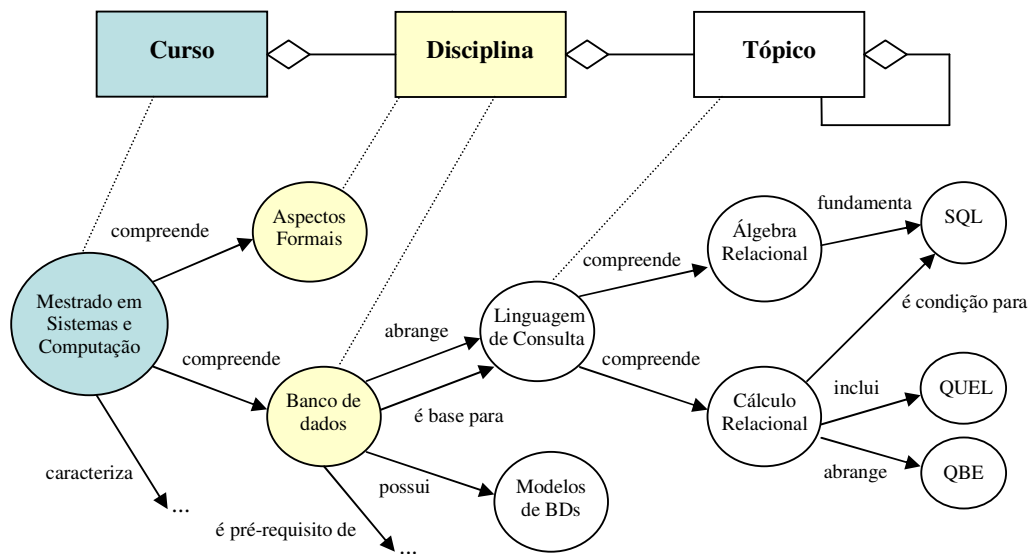


Fig. 3.3 – Mapa conceitual parcial do Curso de Mestrado em Sistemas e Computação

Analisando este mapa, pode-se verificar que o curso de *Mestrado em Sistemas e Computação* compreende as disciplinas: *Aspectos Formais*, *Banco de Dados*, etc. A disciplina *Banco de dados* abrange e é base para o tópico *Linguagem de Consulta*, possui o tópico

Modelo de dados, etc. A partir deste ponto, tópicos podem associar-se somente a outros tópicos. Como exemplificado, o tópico *Álgebra Relacional* fundamenta o tópico *SQL*.

Com relação aos predicados, pode-se observar tanto a existência de predicados de agregação, a exemplo de *compreende* e *abrange* e de predicados de associação, a exemplo de *fundamenta*.

Fazendo uso da sua expressividade, é possível explorar este mapa conceitual através de seus relacionamentos e resolver questões através da simples navegação do mesmo. A seguir, pode-se verificar algumas dessas questões:

I. Quais disciplinas são *compreendidas* pelo curso de *Mestrado em Sistemas e Computação*?

Resposta: *Aspectos Formais e Banco de Dados*

II. Quais são os tópicos fundamentados por outro(s) LO(s)?

Resposta: *SQL*

III. Quais tópicos são *abrangidos* por tópicos que são *compreendidos* pela disciplina *Linguagem de Consulta*?

Resposta: *QBE*

Essas consultas refletem a variedade de indagações que podem ser respondidas pela simples navegação do mapa conceitual. Este auxilia o usuário na visualização das instâncias e permite que consultas mais sofisticada possam ser feitas ao sistema com mais facilidade e precisão, como é o caso das consultas *ad hoc* feitas através do ROSAQL.

Um mapa conceitual pode ou não estar associado a um tesauro. Contudo, é interessante que esteja, pois este não serve apenas para contextualizar o mapa, mas também para auxiliar nas pesquisas submetidas pelos usuários, uma vez que define termos em um determinado domínio, relacionado-os a outros termos equivalentes, sinônimos e associados

Contudo, embora o Sistema ROSA não possua um esquema conceitual associado, a representação de LOs e de suas associações feitas através de um mapa conceitual é fundamentado no modelo de dados ROSA. Este representa as instâncias geradas e será visto na próxima seção.

3.4 MODELO DE DADOS ROSA

O modelo de dados ROSA foi gerado com a intenção de encontrar um modelo que melhor represente as instâncias geradas a partir do ROSA. Este está baseado na semântica do modelo RDF, e permite representar dado e metadado indistintamente, fornecendo suporte a esquemas e utilizando como linguagem padrão o XML. O RDF define como estrutura principal do seu modelo o conceito de *declaração*. Esta é a forma utilizada para definir associações entre recursos na Web, onde um recurso (qualquer objeto identificado por uma *URI- unified resource identification*) é ligado a um outro recurso ou objeto através de um predicado [Porto, Moura, Fernandez et al., 2003].

Desta forma, no modelo ROSA, *associações* são expressas através de predicados nos moldes do RDF. Estes relacionam LOs segundo uma semântica particular. A participação de LOs em predicados é capturada no Modelo de Dados através de *declarações* formando triplas $T(\text{sujeito}, \text{predicado}, \langle \text{objeto} \rangle)$. Nas declarações, *sujeito e objetos* são objetos do tipo *Recurso Complexo*, sendo este uma unidade básica do Modelo de onde derivam as demais estruturas: *Objetos de Aprendizagem (LO)* e *Relacionamentos*. *LOs* representam os objetos de interesse do Domínio de EAD, enquanto *Relacionamentos* provêm estruturas de composição para LOs. A FIG. 3.4 apresenta o modelo de dados ROSA.

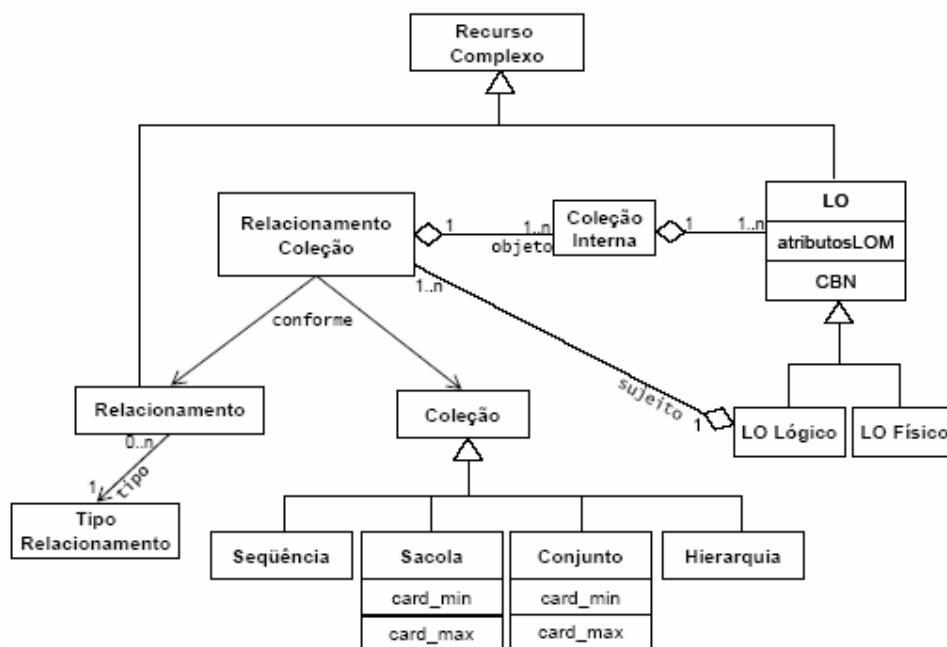


FIG. 3.4 – Modelo de dados ROSA

Este modelo corresponde à última versão e sua descrição detalhada encontra-se em [Coutinho, 2004] e [Coutinho e Porto, 2004]. Como pode-se observar, ele está expresso num diagrama de classes UML. A classe *Recurso Complexo* provê uma representação comum para LOs e relacionamentos. A classe *LO* especializa a classe *Recurso Complexo* e possui os atributos: *atributosLOM* e *CBN*. O *atributosLOM* armazena o subconjunto de atributos do padrão IEEE-LOM adotado pelo ROSA, juntamente com um atributo que identifica unicamente o LO. O *CBN* (Contexto de Busca Navegacional) é um atributo que armazena as informações geradas pela busca navegacional. Essas informações correspondem a uma espécie de histórico da busca, contendo todos os caminhos pelos quais o LO foi encontrado. A classe *LO* é especializada em duas subclasses: *LO Lógico* e *LO Físico*. *LO Lógico* representa a parte conceitual do objeto de aprendizagem e a classe *LO Físico* representa a parte física do objeto de aprendizagem. Como visto na seção 3.3, ambos LOs lógicos e físicos possuem um conjunto comum de atributos advindos do padrão IEEE-LOM, contudo, os atributos de cunho pedagógico são pertinentes apenas aos LOs lógicos, enquanto que os *atributos técnicos* (formato, tamanho, etc.) dizem a respeito apenas aos LOs físicos. Somente os LOs lógicos participam das associações semânticas.

A classe *Relacionamento* representa os predicados existentes no sistema. Suas instâncias são classificadas de acordo com um tipo de relacionamento (representado no modelo como *Tipo Relacionamento*). Um *Tipo Relacionamento* define as propriedades de equivalência comuns aos *Relacionamentos* daquele tipo. As propriedades de equivalência podem ser: *reflexiva, simétrica e transitiva*.

A classe *Coleção* é especializada em classes que representam os possíveis tipos de coleções. Assim, *Coleção* é uma classe que representa a semântica dos tipos de coleções de forma genérica. As classes *Sacola* e *Conjunto* incluem atributos que identificam a cardinalidade mínima e máxima. Esses atributos introduzem restrições sobre as coleções, determinando o número máximo (*cardinalidade máxima*) e mínimo (*cardinalidade mínima*) de elementos que podem ser extraídos da coleção.

Uma associação parte sempre de um único LO (o sujeito) chegando (por meio de um predicado) a uma coleção de LOs (o objeto). Sendo assim, o *sujeito* da declaração ROSA será sempre uma instância da classe *LO Lógico*, enquanto o restante da declaração pode ser enxergado numa instância de *Relacionamento Coleção*. Essa instância está associada a uma instância da classe *Relacionamento* a qual representa o *predicado* da declaração ROSA, e de forma similar também está associada a uma instância de *Coleção* que representa o tipo de coleção. Finalizando, a instância *Relacionamento Coleção* agrega uma ou mais instâncias de

Coleção Interna representando o *objeto* da declaração. A classe *Coleção Interna*, por sua vez, representa para o modelo ROSA uma estrutura básica, de maneira análoga à tupla para o modelo relacional, com o objetivo de auxiliar na implementação da operação de junção entre LOs na álgebra ROSA, implementada pela máquina de execução de consultas ROSA [Coutinho, 2004], discutida na seção 4.3.2.3.2 .

É importante ressaltar que esta versão do modelo de dados ROSA é uma evolução do modelo inicialmente desenvolvido [Porto, Moura, Fernandez et al., 2003], que tinha como principal foco o processamento de suas triplas (sujeito, predicado, objeto). Sendo assim, alguns pontos abordados não estão presentes no ROSA implementado, a exemplo das diferenças semânticas impostas pelos tipos de coleções.

3.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os pontos mais importantes a respeito do sistema ROSA, permitindo seu entendimento pelos leitores desta dissertação. Contudo, o sistema ROSA é muito mais extenso e complexo. Possui funcionalidades mais avançadas que abrangem toda a parte gerencial de dados, e vão desde a criação de mapas conceituais até a exportação dos mesmos.

Quanto a sua implementação, utiliza a tecnologia Servlet [Servlet] e faz uso da linguagem de programação JAVA [JAVA], com a qual implementa as regras de negócio, e das linguagens JSP [JSP], JavaScript [JavaScript] e HTML [HTML].

Embora não possua um esquema conceitual, o ROSA faz uso do seu modelo de dados para garantir que as instâncias estejam sintaticamente corretas. Entretanto, este pode ser um ponto negativo na construção do ROSA - P2P, principalmente durante a integração de dados, onde esquemas conceituais devem ser utilizados para o entendimento sintático dos dados existentes nas diversas bases participantes. Esta questão será discutida em detalhes quando da definição da estratégia de integração de dados do ROSA - P2P.

Diversos trabalhos foram desenvolvidos com base no sistema ROSA. Estes permitem que o ROSA torne-se um sistema cada vez mais robusto, evoluindo em consonância com a tendência das tecnologias atuais, como é o caso do ROSA - P2P. Dentre estes, pode-se citar a representação e acesso aos LOs através de topic maps [Fernandez, 2004] e uma abordagem baseada em lógica para representação e busca de LOs [Costa, 2005].

O sistema ROSA encontra-se atualmente disponível para testes no seguinte endereço eletrônico: www.des.ime.eb/~Rosa.

4. DEFINIÇÃO DO SISTEMA PROPOSTO – ROSA - P2P

O sistema proposto no contexto desta dissertação se constitui da integração de objetos de aprendizagem no sistema ROSA em ambiente P2P, denominado ROSA - P2P.

O que se pretende realizar na prática é um ambiente P2P onde usuários poderão submeter consultas, seja através de um portal [Toledo, 2002], denominado de portal ROSA, seja através de cada *peer* ROSA - P2P. Desta forma, quando uma consulta for submetida por um usuário, por exemplo, através de um *super-peer*, este a reescreverá e a submeterá para si próprio, armazenando o resultado em *cache*. Em seguida, reenviará a consulta original aos seus próprios *peers* e aos demais *super-peers* relevantes, ativando um relógio que controlará o tempo em que o mesmo aguardará pelas respostas. Todavia, quando todos os resultados tiverem sido retornados ou o tempo de espera de resultados tiver sido esgotado, os dados residentes em *cache* serão integrados e a resposta da integração será retornada ao usuário que submeteu a consulta (*peer* solicitador).

Para um melhor entendimento, o sistema proposto será dividido em 3 seções. A seção 4.1 é a primeira delas e apresenta a arquitetura interna do ROSA - P2P. Logo após, a seção 4.2 discute a definição do seu ambiente P2P, seguida pela seção 4.3, que trata da definição do seu sistema de integração de dados.

4.1 ARQUITETURA INTERNA

Esta seção visa apresentar e especificar de modo abrangente os componentes compostos por cada *peer* e *super-peer* do sistema ROSA - P2P. Contudo, a especificação de cada um destes, assim como da forma como os *peers* e *super-peers* estão localizados e representados, será apresentada em detalhes durante o restante deste trabalho.

A FIG. 4.1 ilustra a arquitetura interna do sistema, exibindo seus módulos e respectivos componentes, a saber:

- **Módulo de interoperabilidade:** formada pelo componente interoperador (P2P), possui as características e funcionalidades necessárias para a formação e manutenção de uma

rede P2P, tais como estabelecimento de conexões, atualizações de índices de roteamento, eleição de *super-peers* e balanceamento da rede;

- **Módulo de processamento de consulta:** formada pelo componente interface do usuário, é responsável por oferecer um ambiente mais amigável de comunicação entre os usuários e o componente de processamento de consultas. Este adota uma estratégia dividida em 2 fases, apresentada com detalhes na seção 4.3.2.3;
- **Módulo de gerenciamento de dados:** é formada por dois componentes: vocabulários controlados, que são ferramentas que facilitam e possibilitam a interpretação e recuperação das informações, ao mesmo tempo em que viabilizam o intercâmbio entre os sistemas, permitindo pesquisas mais apuradas e restritas às informações realmente relevantes; e *cache/integrador de dados*, que armazena temporariamente os resultados parciais das consultas. Uma vez obtido os resultados dos *peers* e/ou *super-peers* para os quais a consulta foi reenviada, é realizada sua integração e o resultado final da consulta é exibido ao usuário.

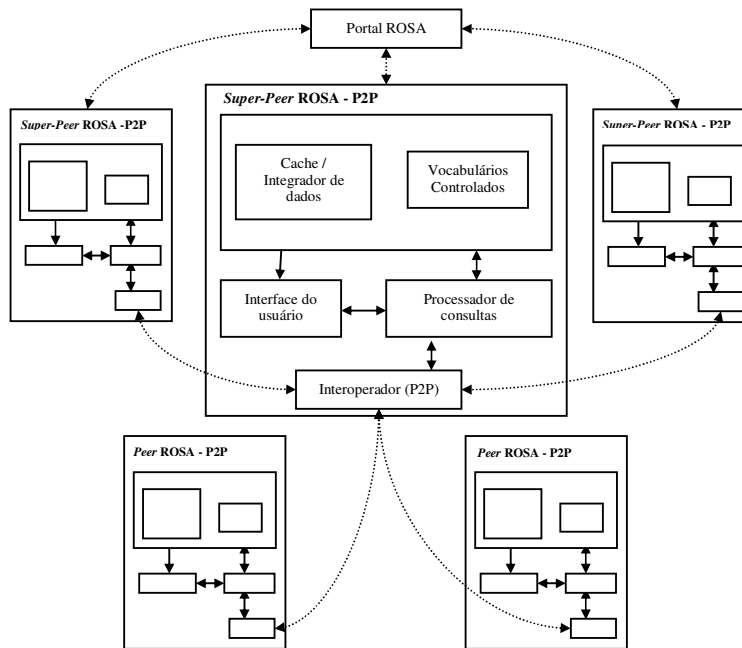


FIG. 4.1 – Arquitetura interna do sistema proposto – ROSA - P2P

4.2 DEFINIÇÃO DO AMBIENTE P2P

A definição do ambiente P2P é uma das tarefas mais importantes e complexas a serem desenvolvidas no contexto desta dissertação. Sua especificação é fator decisivo para o bom funcionamento do sistema, visto que este serve de base para a sua construção. Sendo assim, para um melhor entendimento, ele será dividido quanto a sua arquitetura, estratégias adotadas e tolerância a falhas, discutidos ao longo desta seção.

4.2.1 ARQUITETURA

O ambiente de execução proposto para o sistema é fazer com que os vários *peers* ROSA - P2P estejam fisicamente distribuídos em *sites* diferentes, de forma a serem acessados via Internet. Desta forma, tem-se o portal ROSA disponível em um servidor Web e os demais ROSA - P2P em *sites* fisicamente distribuídos na WEB.

Analisando o ambiente de execução pretendido e as diferenças básicas entre as arquiteturas P2P existentes (vide seção 2.2), fica claro perceber que a arquitetura P2P que melhor se enquadra neste ambiente é a arquitetura baseada em *super-peers*, uma vez que possui redução de tempo e largura de banda para pesquisa, estratégias contra falhas do sistema, gerenciamento através dos *super-peers* e um índice aceitável de confiabilidade e escalabilidade, muito embora a arquitetura descentralizada estruturada também pudesse ser utilizada.

A arquitetura baseada em *super-peers* se caracteriza pela formação de pequenos subconjuntos de *peers* de maior poder computacional interligados entre si, denominados *super-peers*. Estes são responsáveis pelo gerenciamento de recursos, tendo cada um deles outros *peers* conectados a si. Nesse ambiente todos os *peers* conterão o sistema ROSA - P2P, porém, somente aqueles dotados de algumas características mais acentuadas poderão ser eleitos *super-peers* (seção 4.2.2.5). Desta forma, para responder a uma solicitação de consulta, o *super-peer* se comunicará com os demais *super-peers* relevantes à consulta, que conseqüentemente se comunicarão com seus *peers*. A FIG. 4.2 ilustra o sistema proposto na arquitetura baseada em *super-peers*.

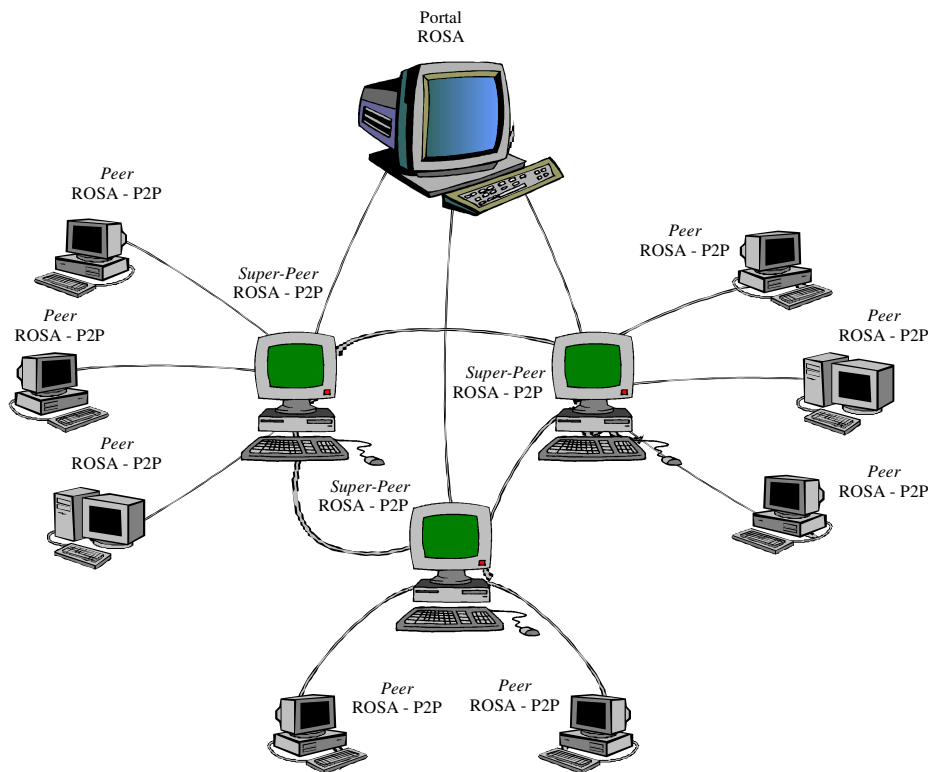


FIG. 4.2 – Sistema proposto na arquitetura baseada em *super-peers*

É importante ressaltar que, embora o portal ROSA referencie alguns *super-peers*, ele não faz parte da arquitetura P2P em questão. Encontra-se em uma camada acima e serve apenas como um ponto de entrada para usuários que não possuem o ROSA - P2P possam submeter suas consultas via WEB e receberem os respectivos resultados. Este portal possui as funcionalidades necessárias para estas operações, tais como os módulos referentes à integração de dados (seção 4.3.2.4). Porém, a máquina onde ele se encontra serve para a persistência de alguns serviços oferecidos pelo sistema proposto, tais como o SD – Serviço de Diretório, que permite que novos *peers* que desejem se conectar ao sistema possam solicitar a relação de *super-peers* existentes e o SEVC – Serviço de Entrega de Vocabulários Controlados, que consiste em persistir todos os vocabulários controlados (global, local e de palavras chaves) nesta máquina, externa ao ambiente P2P, fazendo com que um *peer*, ao se conectar ao sistema pela primeira vez, receba-os via rede. Estes serviços serão apresentados na seção 4.2.2.2 e 4.3.2.2, respectivamente. A FIG. 4.3 mostra, em um nível mais alto de abstração, as camadas do sistema proposto.

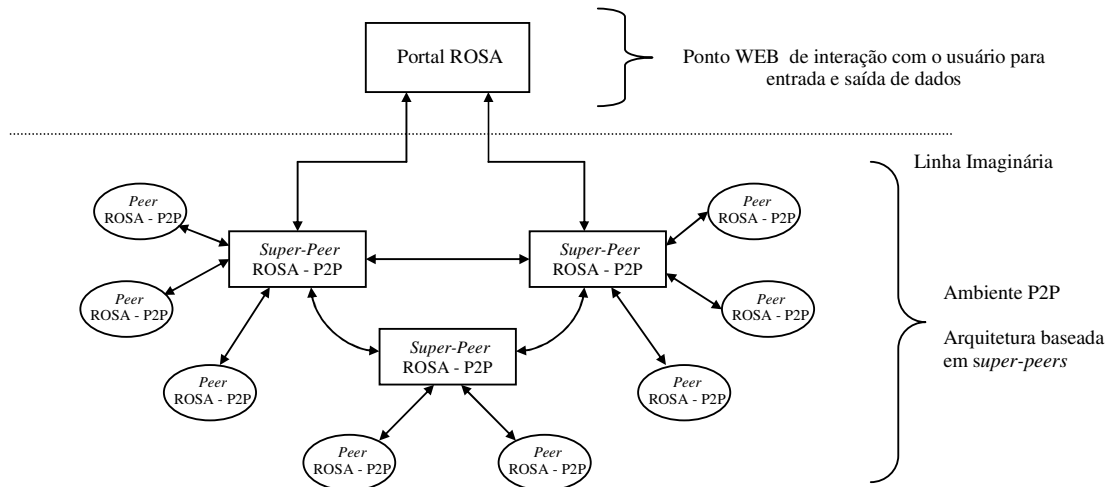


FIG. 4.3 – Camadas do sistema proposto

Uma vez definida a arquitetura P2P, é preciso especificar as estratégias adotadas, ressaltando suas particularidades com maiores detalhes.

4.2.2 ESTRATÉGIAS ADOTADAS

Esta seção apresenta as estratégias definidas e adotadas para o ambiente P2P. Ela especifica em detalhes, questões relacionadas com a formação da rede P2P, conexão e reconexão de *peers*, definição, quantidade e eleição de *super-peers*, assim como a estratégia utilizada para o balanceamento do sistema e de comunicação entre os *peers* e *super-peers*.

4.2.2.1 AGREGAÇÃO DE PEERS E AGRUPAMENTO DE SUPER-PEERS

Segundo [Nejdl et al., 2003], o agrupamento de *peers* está baseado na idéia de unir *peers* à *super-peers* que possuam características similares. Desta forma, a quantidade de mensagens enviadas pela rede será menor, uma vez que cada agrupamento de *peers* é conhecido, sendo possível direcioná-las somente para os agrupamentos alvos de maior interesse.

Porém, no ROSA - P2P, o agrupamento de *peers* será referenciado como uma agregação de *peers*. Esta mudança se faz necessária, pois além de facilitar o entendimento da estratégia definida para o sistema, os *peers* se agregam a um *super-peer* existente formando uma agregação. Porém, os *super-peers* de mesmo domínio, estes sim se agrupam entre si.

Portanto, a estratégia adotada será baseada na agregação de *peers*, porém será mais abrangente. Esta parte da idéia de agregações e a estende ao conceito de agrupamento dessas agregações, denominado de agrupamento de *super-peers*. *Peers* serão agregados inicialmente através de duas características importantes: assunto e localização, enquanto que *super-peers* serão agrupados somente através do assunto tratado pelas agregações. Contudo, o sistema apresenta algumas exceções, permitindo também a criação de agregações baseadas somente no assunto tratado pelos *peers*. Neste caso, a localização geográfica dos mesmos é ignorada, para formar agregações somente por assunto. De fato, as agregações baseadas nas características de assunto e localização são utilizadas para a formação da rede, podendo ser evoluídas também para agregações somente por assunto, adaptando-se ao crescimento dinâmico da rede, permitindo conexões avançadas e o balanceamento otimizado do sistema. Casos referentes à criação de agregações somente por assunto serão discutidos com mais detalhes ainda neste capítulo.

Como o sistema ROSA armazena conteúdos instrucionais, considera-se que a granularidade do assunto será feita em função do curso que cada instituição oferece. Quanto à localização, será representada segundo sua localização geográfica, cuja granularidade será feita inicialmente pelo país de origem. Estas informações são fornecidas pelo usuário no momento da instalação do sistema, podendo ser alteradas caso haja alguma modificação posterior.

Desta forma, para se formar uma agregação, basta que um *peer* se conecte a um *super-peer* semelhante ao seu assunto e localização. Por exemplo, uma instituição que ofereça o curso de Medicina localizada no Brasil será conectada a um *super-peer* que possui estas mesmas características. O sistema suporta também que uma instituição trate de mais de um assunto. Neste caso, uma instituição que ofereça o curso de Medicina, Informática e Direito localizada no Brasil será conectada a um *super-peer* que atenda a estas mesmas características. De fato, esta particularidade da estratégia evita que um grande número de assuntos e, conseqüentemente de agregações, sejam criados desnecessariamente.

Um agrupamento é formado quando o número de agregações semelhantes em assunto for superior ou igual a duas. Por exemplo, dois ou mais *super-peers* que tratem de um mesmo assunto formam um agrupamento, mesmo se forem de países distintos.

Para facilitar o entendimento dessas definições, a FIG. 4.4 exibe alguns exemplos de agregações de *peers* e de um agrupamento de *super-peers*, segundo características similares.

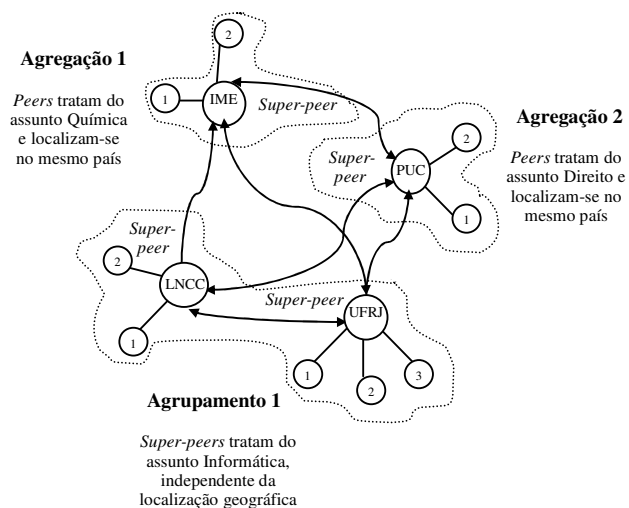


FIG. 4.4 – Agregações de *peers* e agrupamento de *super-peers*

Contudo, com o intuito de se destacar a estratégia de agrupamento de *super-peers*, a FIG. 4.5, baseada em [Nejdl et al., 2003], exibe um exemplo mais detalhado sobre o seu funcionamento. Neste exemplo, os *super-peers* de 3 a 5 tratam de um assunto, que não é tratado nos *super-peers* 0 a 2. Desta forma, uma consulta que trate deste assunto, submetida pelo *super-peer* 0, será direcionada para um *super-peer* que a atenda, tais como aqueles numerados de 3 a 5. Para que isso seja possível, cada *super-peer* deve possuir uma referência para cada tipo de agrupamento existente no sistema. Isso garante maior confiabilidade ao sistema, já que evita que *peers* relevantes à consulta deixem de ser consultados. No caso, o *super-peer* 0 possui uma referência ao agrupamento com a característica de assunto da consulta, através da conexão com o *super-peer* 3. Este, ao receber a consulta, submete-a aos demais *super-peers* do agrupamento, no caso, os de números 4 e 5.

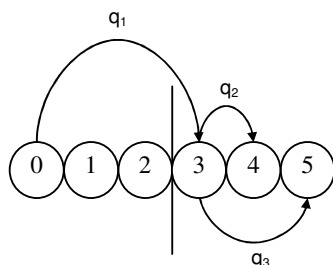


FIG. 4.5 – Agrupamento de *super-peers*

Cada *super-peer* pertencente a um mesmo agrupamento deve possuir uma referência para os demais *super-peers* do agrupamento. Isto é necessário para que todos os *super-peers* do agrupamento relevantes à consulta sejam consultados e, desta forma, a confiabilidade do sistema seja mantida.

Esta estratégia muito ajudará na organização física dos *peers* no sistema, sobretudo no que se refere à otimização do processamento de consultas. *Peers* com características comuns ficarão geograficamente próximos uns aos outros, facilitando sua localização e proporcionando maior desempenho e confiabilidade, uma vez que as consultas serão submetidas somente aos *super-peers* relevantes, isto é, os que tratarem de um mesmo assunto.

4.2.2.2 CONEXÃO DE PEERS AO SISTEMA

Um dos principais problemas encontrados na especificação do sistema foi decidir como se daria a criação de um novo *peer* pela primeira vez no ambiente, já que nesta etapa inicial, o sistema ainda não tem conhecimento sobre a localização física de cada *peer* ROSA - P2P na rede. Vale observar que a localização é dinâmica, podendo mudar a qualquer momento.

Uma solução viável encontrada, que não altera as características da arquitetura de *super-peers*, foi permitir que a máquina onde se encontra o portal ROSA, que não faz parte do ambiente P2P e possui IP estático, atue também como um servidor de serviços. Dentre esses serviços, pode-se destacar o SD – serviço de diretório. Este consiste em manter uma estrutura de dados com a relação dos *super-peers* existentes no sistema, permitindo que os novos *peers* que desejarem se conectar ao sistema recebam-na via rede. Esta estrutura de dados visa auxiliar no processo de conexão do *peer* e é mantida com informações atualizadas sobre os metadados desses *super-peers*. Porém, com a finalidade de prevenir possíveis falhas nesta máquina, esta deve ser replicada em um outro servidor, possivelmente em uma outra rede. Desta forma, em caso de falha, os *peers* deverão se dirigir à máquina de contingência, conforme será discutido na seção 4.2.3.

Para possibilitar que o sistema possa sempre se comunicar sem problemas, todo *peer* deve possuir uma porta física do seu hardware com o status “listening”. Esta porta deverá ser utilizada pelo sistema para se comunicar com os demais *peers* participantes. Sua escolha deverá ser feita de acordo com as portas ainda disponíveis pela IANA (Internet Assigned Numbers Authority) [IANA]. IANA é uma organização que registra o uso das portas físicas

dos computadores para aplicações. Para solicitar o registro de uma porta, basta acessar o seu site¹ e fornecer alguns dados, dentre os quais pode-se destacar o número da porta e a descrição da aplicação.

Desta forma, quando um *peer* deseja se conectar ao sistema pela primeira vez, ele envia uma consulta ao serviço de diretório, através do seu IP e da porta escolhida, solicitando a relação de *super-peers* disponíveis. Assim, o *peer* verifica quais, dentre os *super-peers* existentes, contém suas características de assunto e localidade (vale observar que estas foram informadas no momento da instalação de cada sistema ROSA - P2P). Uma vez identificado, o *peer* envia uma solicitação de conexão ao respectivo *super-peer* que a valida, verificando realmente se o mesmo pode fazer parte da rede P2P ou não. Esta validação é realizada através da autenticação do novo *peer* atestando se realmente o sistema ROSA - P2P está presente no *peer*.

Uma vez estabelecida a conexão, o *peer* deve fornecer, através do envio de suas propriedades (metadados) ao seu já definido *super-peer*, informações sobre sua localização, identificação, se deseja ser um *super-peer* ou não, características físicas, dentre outras. A partir deste momento, o *peer* já se encontra apto a compartilhar recursos e, desta forma, já poderá submeter suas consultas. Este caso simples de conexão de um *peer* ao sistema pode ser observado na FIG. 4.6.

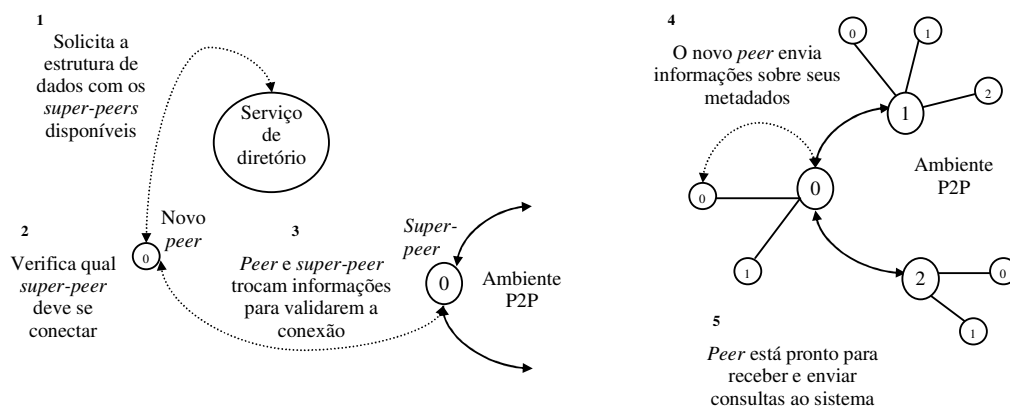


FIG. 4.6 – Conexão simples de um *peer* ao sistema

Porém, quando o *peer* não encontrar nenhum *super-peer* com as suas características de assunto e localização geográfica, duas situações podem ocorrer:

¹ <http://www.iana.org/cgi-bin/usr-port-number.pl>

- Este deve se tornar um *super-peer*, e portando se unir ao sistema informando seus metadados a todos os demais *super-peers* e ao serviço de diretório, recebendo ao mesmo tempo, as mesmas informações sobre cada tipo de agregação e/ou agrupamento existente. Definições sobre a quantidade de *super-peers*, assim como técnicas para eleição e manutenção do *super-peer*, serão discutidas mais adiante. Porém, caso tenha optado por não se tornar um *super-peer*, o novo *peer* será questionado e poderá optar por ser um *super-peer* temporário até que outro *peer*, de mesmas características e com desejo de ser um *super-peer*, entre no sistema e assuma este papel. Contudo, ainda que opte por não sê-lo, uma estratégia de balanceamento deverá ser executada de modo a fazer com que um novo *super-peer* seja eleito dentre os *peers* existentes em uma agregação ou agrupamento cujo assunto seja semelhante ao assunto do novo *peer*. Caso esta agregação ou agrupamento correspondente ao assunto do novo *peer* não exista, este não poderá se conectar ao sistema. Caso exista, o sistema permitirá a criação de uma agregação somente por assunto, adaptando-se às situações impostas pela dinamicidade do sistema. De fato, esta particularidade da estratégia permite a conexão de qualquer *peer* ao sistema, com a condição de que exista pelo menos uma agregação correspondente ao assunto do novo *peer*. Neste ponto, pode-se verificar que o sistema permite a criação de agregações semelhantes em assunto, independente da localização. Assim, os *peers* existentes serão então redistribuídos e balanceados, possibilitando que o novo *peer* possa agora se conectar a um *super-peer* correspondente ao seu assunto.
- A rede P2P do sistema ainda não foi montada. Esta situação indica que este *peer* é o primeiro da rede, devendo se tornar o primeiro *super-peer* do sistema. Seus metadados serão então informados ao serviço de diretório, iniciando assim a construção da rede P2P.

Devido ao alto grau de dinamismo do ambiente P2P, causado principalmente pela entrada e saída de *peers* a todo o momento no sistema, *peers* devem possuir mecanismos que permitam uma conexão rápida ao sistema. Desta forma, quando um *peer* que se desconectou da rede desejar se reconectar, basta que este se conecte ao seu *super-peer* (informações sobre a localização do *super-peer* ficam armazenadas no *peer*). Caso este não encontre o seu *super-peer* (no caso de ter deixado o sistema ou ter se tornado um *peer* comum), deverá se juntar ao sistema como se fosse a primeira vez e obter um novo *super-peer*. Ao mesmo tempo, para agilizar este processo, os *super-peers* deverão manter informações sobre os seus *peers* que deixaram o sistema por certo período de tempo (parametrizável pelo administrador do sistema), como por exemplo, uma semana. Passado este tempo, as referências

correspondentes deverão ser removidas. Desta forma, quando um *peer* se reconectar ao sistema durante este período, só precisará atualizar seus metadados e não enviá-los novamente. No caso do serviço de diretório, assim que um *super-peer* deixar o sistema ou se tornar um *peer* comum, suas referências devem ser removidas. Isto se faz necessário para evitar que um novo *peer* fique tentando se conectar a um *super-peer* que não existe mais no sistema. No sistema P2P Edutella [Edutella], por exemplo, essas informações são mantidas enquanto os *peers* se re-registram periodicamente ou atualizarem suas informações regularmente. Quando esses procedimentos não ocorrem, indica que o *peer* deixou o sistema e todas as suas referências devem ser removidas.

O estabelecimento da conexão com o sistema através de um ponto inicial (no caso com a utilização do serviço de diretório) foi a solução encontrada para esta etapa do sistema. Futuramente, poderá se verificar a possibilidade de um novo *peer* se unir ao sistema sem a existência de um ponto inicial, como por exemplo, através de um “scan” na Internet a procura de nós que possuam a porta referente ao sistema com o status “listening”, e desta forma estabelecer a conexão. Atualmente este procedimento é inviável, uma vez que redes possuem mecanismos que bloqueiam mensagens *broadcasting*.

Alguns casos particulares de conexão de um *peer* ao sistema serão discutidos nas seções 4.2.2.4 e 4.2.2.6.

4.2.2.3 SUPER-PEER

No sistema proposto, optou-se por considerar que toda instituição de ensino será automaticamente um *super-peer*. Do contrário, os dados referentes às características físicas de um *peer* devem ser fornecidos pelo usuário no momento da instalação do sistema, assim como a informação se o novo *peer* deseja ou não ser um *super-peer*.

É exatamente neste momento que o sistema se depara com um ponto comprometedor: qual é a quantidade de *super-peers* que o sistema poderá suportar para que a pesquisa não seja prejudicada? É fácil observar que, quanto maior for a quantidade de *super-peers*, maior será a quantidade de mensagens entre eles.

Outro ponto importante se refere ao ciclo de vida de um *super-peer*, isto é, quanto tempo um *peer* está apto a ser um *super-peer*. Desta forma, *super-peers* devem, em um período estipulado de tempo, verificar se suas capacidades em relação às suas características físicas

(seção 2.2.2.1) ainda são os melhores dentre os *peers* existentes na sua agregação ou no seu agrupamento. Esta condição determina se estes devem continuar a serem *super-peers* ou, se nova eleição deve ocorrer para a indicação de outro *super-peer*. Nesta eleição, uma estratégia para o balanceamento do agrupamento deve ser implementada. No sistema proposto, essas questões serão discutidas nas seções a seguir.

4.2.2.4 QUANTIDADE DE *SUPER-PEERS*

A quantidade de *super-peers* existente será dinâmica, determinada conforme a quantidade máxima de *peers* que um *super-peer* pode suportar para o bom funcionamento do sistema. Esta quantidade será parametrizada segundo uma avaliação do tempo de resposta a determinadas consultas submetidas em máquinas dotadas de hardware semelhantes, porém de diferentes localizações. Desta forma, a quantidade ideal de *super-peers* no sistema estará sendo balanceada indiretamente.

A título de exemplo, considere essa quantidade como sendo de 25 *peers*. Neste caso, quando um novo *peer* desejar se conectar a um *super-peer* que já possua 25 *peers*, este avisará ao novo *peer* que o limite máximo de *peers* com os quais pode se conectar estourou. Desta forma, o novo *peer* verificará se existe outro *super-peer* que atenda as suas características, com o qual possa se conectar. Neste ponto, três decisões podem ser tomadas:

- Caso o *super-peer* exista, e este possua menos que 25 *peers*, o novo *peer* se conectará a ele;
- Caso o *super-peer* exista, porém este possua também 25 *peers* ou mais, o novo *peer* se deparará com o mesmo problema;
- Caso não exista, este *peer* será analisado quanto a sua opção de ser um *super-peer*. Em caso afirmativo, este se tornará um *super-peer*, formando uma nova agregação e, conseqüentemente, um novo agrupamento, ou fazendo parte do agrupamento já existente. Neste ponto, pode-se verificar que o sistema permite também a criação de agregações semelhantes tanto em assunto quanto em localização. De fato, a extensão desta estratégia aumenta a flexibilidade do sistema quanto a novas conexões. Caso tenha optado por não ser um *super-peer*, o novo *peer* será questionado e poderá optar por ser um *super-peer* temporário até que outro *peer* assuma este papel. Porém, caso ainda não opte por sê-lo, uma estratégia de balanceamento deverá ser executada de modo a fazer com que um novo

super-peer, seja eleito dentre os *peers* existentes no *super-peer* inicialmente solicitado para conexão, ou seja, dentre os *peers* da respectiva agregação, ou dentre os existentes no respectivo agrupamento (em último caso). Neste ponto, caso o novo *super-peer* tenha sido eleito dentre os *peers* de uma agregação correspondente às características de assunto e localização do novo *peer*, a estratégia de agrupamento inicial é mantida. Porém, caso o novo *super-peer* tenha origem de uma agregação que não corresponda à localização do *peer* (caso onde este foi eleito dentre os *peers* do agrupamento), o sistema permitirá a criação de uma agregação somente por assunto, adaptando-se às situações impostas pela dinamicidade do sistema. Assim, os *peers* existentes serão redistribuídos e balanceados, possibilitando que o novo *peer* possa agora se conectar a um *super-peer* com menos de 25 *peers* correspondente condicionalmente ao seu assunto. Esta última situação será melhor discutida na seção 4.2.2.6, referente ao balanceamento do sistema.

É possível observar também que à medida em que um *super-peer* é mais procurado, poderá ficar com mais *peers* do que outro do mesmo agrupamento, desbalanceando o agrupamento. A estratégia utilizada pelo sistema para resolver este problema de desbalanceamento do agrupamento será descrita na seção 4.2.2.6 com maiores detalhes.

4.2.2.5 ELEIÇÃO DE SUPER-PEERS

No sistema proposto, a estratégia inicial de agregação de *peers* está baseada nas características de assunto e localização e a de agrupamento de *super-peers* na característica de assunto. Desta forma, a eleição de *super-peers* se fará entre os *peers* de cada agregação e/ou agrupamento, e não entre todos os *peers* do sistema, otimizando assim o tempo para a eleição de novos *super-peers*, ao mesmo tempo em que facilita o entendimento do seu algoritmo. Isto é possível porque os *peers* deverão permanecer dentro da mesma agregação ou agrupamento, isto é, mesmo que um *peer* de uma agregação ou agrupamento possua melhores condições para ser o *super-peer* de outra agregação (diferente em assunto) ou agrupamento, ele não poderá fazê-lo.

A eleição de *super-peers* dentro de agrupamentos se faz necessária, visto que as agregações que os compõem podem possuir *super-peers* que, embora sejam os melhores de cada agregação, não são os melhores para o agrupamento como um todo. Este fato acarretaria

na diminuição do desempenho do sistema, já que um *super-peer* com melhores características (vide seção 2.2.2.1) poderia estar ocupando este papel.

Assim, a eleição de um *super-peer* pode ser feita de duas maneiras: dentro de uma agregação ou de um agrupamento. No caso em que é realizada dentro de uma agregação, a cada período de tempo, que pode ser parametrizado, o *super-peer* verifica dentre os seus *peers*, quais deles desejam ser *super-peers* e quais dentre esses possuem as características físicas mais acentuadas para assumir este papel (observe que o *super-peer* possui informações sobre os seus *peers*, dentre as quais as informações que o caracterizam como um *super-peer*). Caso esta situação se mantenha, nada muda. Caso contrário, como ilustrado na FIG. 4.7, o *peer* ganhador é eleito pelo *super-peer* como o novo *super-peer*, recebendo do antigo *super-peer* as informações referentes aos demais *peers*, assim como sobre os outros *super-peers*.

Uma vez eleito, o novo *super-peer* deverá atualizar as informações referentes aos *super-peers* no serviço de diretório, bem como enviar uma mensagem a todos os seus *peers* e a todas as agregações conectadas a ele, avisando que um novo *super-peer* assumiu este papel, a partir do qual, suas referências são atualizadas.

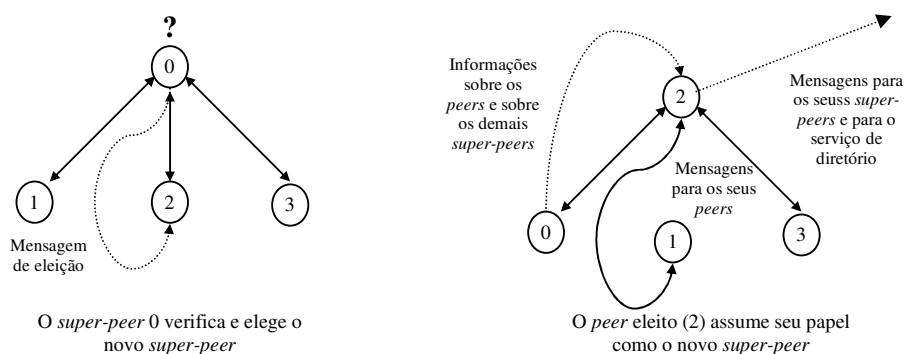


FIG. 4.7 – Eleição de um *super-peer* dentro de uma agregação

No caso em que é realizada dentro de um agrupamento, a eleição se dará da mesma forma, porém, antes de iniciá-la, os *super-peers* deverão verificar quantos *super-peers* já existem dentro do agrupamento, selecionando os melhores *peers* de cada um. Por exemplo, caso um agrupamento contenha 3 *super-peers*, cada um deles deverá selecionar os seus 3 melhores *peers* dentre os seus *peers*. Em seguida, as informações sobre estes *peers* devem ser trocadas entre os *super-peers*, possibilitando desta forma, que cada *super-peer* as analise e identifique os 3 melhores *peers* do agrupamento. Uma vez identificados, o *super-peer* possuidor do melhor *peer* o elego como o novo *super-peer* da sua agregação, e assim se segue, até que todos os demais *peers* se tornem os novos *super-peers*. Esta situação se aplica ao caso onde os melhores *peers* estão distribuídos dentre os *super-peers* do agrupamento.

Porém, pode ocorrer a situação de um *super-peer* possuir mais do que um *peer*, que esteja dentre os 3 melhores para ser um dos *super-peers* do agrupamento, conforme mostra a FIG. 4.8. Neste caso, o *super-peer* que não possui um dos 3 melhores *peers*, solicita ao *super-peer* em questão que eleja um desses seus *peers* como o novo *super-peer* da sua agregação. Conseqüentemente, o *peer* eleito assume seu papel, permitindo que o antigo *super-peer* se torne um dos seus *peers*.

Um detalhe importante se refere ao fato de que o *super-peer* possuidor de mais de um *peer* candidato (*super-peer* “C” na FIG. 4.8.2) fica esperando o pedido do *super-peer* que não possui *peer* candidato (no caso, o *super-peer* “B” na FIG. 4.8.3) e só depois de liberar o *peer* para esse papel é que elege o seu melhor *peer* como o novo *super-peer* da sua agregação, sendo o seguinte melhor para o *super-peer* solicitante (na FIG. 4.8.4, a agregação solicitante recebe o *peer* “C_3” do *super-peer* “C” para ser seu novo *super-peer*).

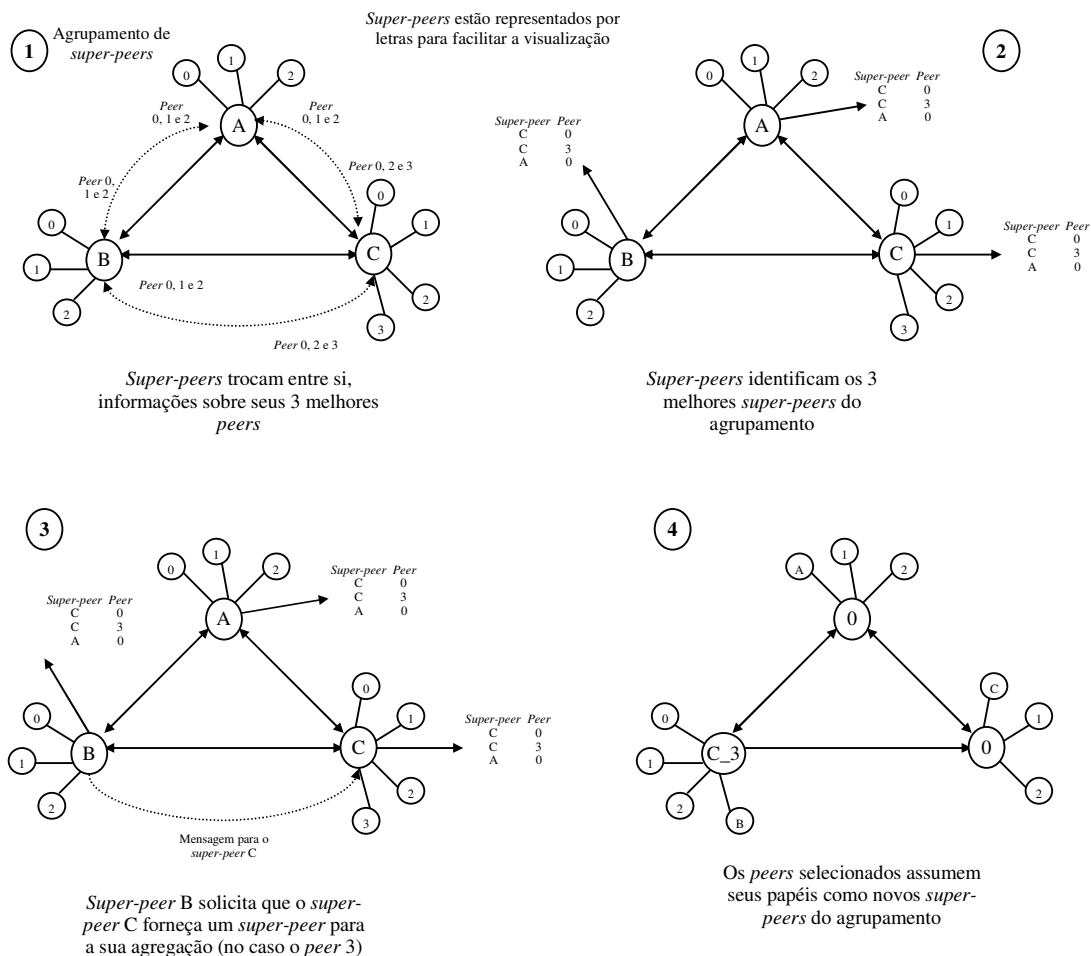


FIG. 4.8 – Eleição de *super-peer* dentro de um agrupamento

Após a rotina de eleição de *super-peers* dentro do agrupamento, o sistema deve fazer, quando necessário, uma redistribuição dos *peers* entre os *super-peers*. Este procedimento visa manter o agrupamento balanceado, aumentando a eficiência do sistema e conseqüentemente das futuras buscas. Esta estratégia é descrita na seção a seguir.

4.2.2.6 BALANCEAMENTO DO SISTEMA

A estratégia utilizada pelo sistema para mantê-lo balanceado, consistiu basicamente da redistribuição dos *peers* entre os *super-peers* desbalanceados do agrupamento em questão, mantendo-o apto a responder com mais rapidez a uma grande quantidade de consultas.

Assim, o balanceamento se faz necessário quando um *super-peer* possui mais *peers* do que outro do mesmo agrupamento, considerando uma margem de erro aceitável (conforme será visto adiante). São várias as situações que conduzem ao desbalanceamento, dentre as principais, pode-se considerar:

- O tempo que um *peer* passa como *super-peer*. Como já visto anteriormente, quando um novo *peer* deseja entrar no sistema, este solicita ao serviço de diretório a relação com as referências de todos os *super-peers* do sistema. Como o *super-peer* em questão já assume este papel por um longo período, ele provavelmente ocupará a primeira posição nesta tabela. Conseqüentemente, qualquer novo *peer* tentará se conectar primeiro a este, podendo ocorrer um desbalanceamento;
- Eleição de um novo *super-peer* dentro de um agrupamento. Uma vez que um *super-peer* possua melhores *peers* candidatos do que os demais *super-peers* do agrupamento, ele os elegerá como os novos *super-peers* do agrupamento, conforme verificado na seção 4.2.2.5. Desta forma, este ficará com menos *peers* e, dependendo da situação, o agrupamento poderá se tornar desbalanceado;
- O número máximo de *peers* conectado a um *super-peer* foi alcançado. Isto significa que o número máximo de *peers* conectado ao *super-peer* foi atingido. Essa última situação, por se tratar de uma particularidade de conexão do sistema, será discutida mais adiante, com mais detalhes.

Para manter o agrupamento balanceado, o *super-peer* com maior número de *peers* (esta informação já se encontra no *super-peer*) verifica, em determinados períodos de tempo, quais

super-peers estão desbalanceados, considerando uma margem de erro aceitável quanto a diferença entre a quantidade de *peers* destes *super-peers*. Ambos os parâmetros período de tempo e margem de erro aceitável devem ser parametrizados pelo administrador do sistema. O período de tempo define quando o *super-peer* com a maior quantidade de *peers* do agrupamento será testado quanto ao seu desbalanceamento dentro do respectivo agrupamento. Futuramente, visando uma maior automação e eficiência, esta funcionalidade poderá ser definida e implementada através de um *trigger*. A margem de erro indica o número máximo de *peers* que um *super-peer* pode ter em relação a outro *super-peer* do agrupamento, de modo a não interferir no bom funcionamento do sistema. Uma vez identificados, o *super-peer* deve, através da função IQP (Identificadora da Quantidade de *Peers*), verificar quantos *peers* cada *super-peer* desbalanceado deve possuir para que o agrupamento se torne balanceado e, conseqüentemente, enviar referências de seus *peers* para cada um desses *super-peers* até que estes se tornem balanceados. Após o balanceamento, as referências sobre os *peers* devem ser atualizadas em cada *super-peer*.

$$\text{Função IQP} = (\lfloor \sum P / N \rfloor)$$

Onde,

- P é o número de *peers* conectado a cada *super-peer* desbalanceado;
- N é o número total de *super-peers* desbalanceados.

Para um melhor entendimento do processo, suponha uma margem de erro aceitável de 5 *peers* em um agrupamento de 4 *super-peers*, conforme mostra a FIG. 4.9a. Neste exemplo, o *super-peer* 0 possui 18 *peers*, seguido pelo *super-peer* 1 com 5 *peers*, o *super-peer* 2 com 7 *peers* e por último o *super-peer* 3 com 13 *peers*. O *super-peer* 0, através de suas informações sobre os demais *super-peers* do agrupamento, identifica que somente ele e os *super-peers* 1 e 2 estão desbalanceados (o *super-peer* 3 está dentro da margem de erro aceitável). Ao executar a função IQP, identifica que ele e os demais *super-peers* desbalanceados devem ter 10 *peers* cada, para que o agrupamento se torne balanceado. Assim, envia 5 dos seus *peers* para o *super-peer* 1 e 3 dos seus *peers* para o *super-peer* 2, de modo que cada um passe agora a ter 10 *peers* cada, conforme ilustra a FIG. 4.9b. Ao final da transferência dos *peers*, ambos os *super-peers* atualizam suas informações sobre seus atuais *peers* (FIG. 4.9c) e desta forma, o balanceamento se torna completo.

Neste ponto, pode-se verificar outra extensão da estratégia, que permite que *peers* se conectem a *super-peers* diferentes de sua localização geográfica. Este fato, embora pareça, não prejudica o desempenho do sistema, visto que as consultas são reenviadas aos *super-peers* com base no(s) assunto(s) tratado(s) por eles, conforme será visto na seção 4.3.2.3.1 Sendo assim, embora se perca tempo tendo que direcioná-las a *peers* que possam estar geograficamente mais distantes dos *super-peers*, esta perda não se equipará ao ganho obtido pelo balanceamento do agrupamento, que possibilita que *super-peers* e *peers* processem a consulta em paralelo. Como estão balanceados (cada *super-peer* possui a mesma quantidade de *peers*), nenhum deles ficará ocioso enquanto outros trabalham.

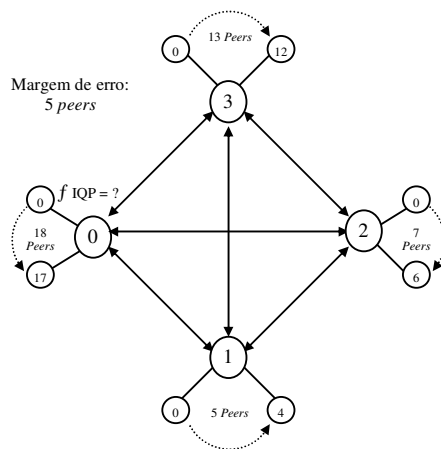


FIG. 4.9a – Agrupamento com margem de erro de 5 peers

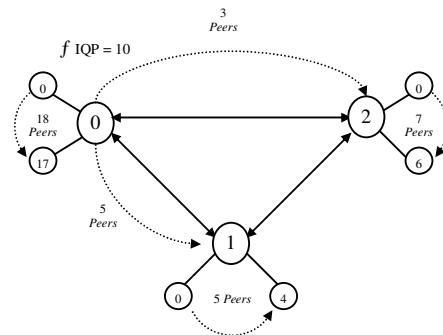


FIG. 4.9b – Transferência de peers do super-peer 0 para os super-peers 1 e 2

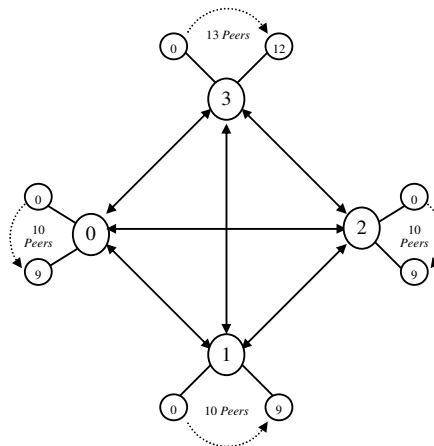
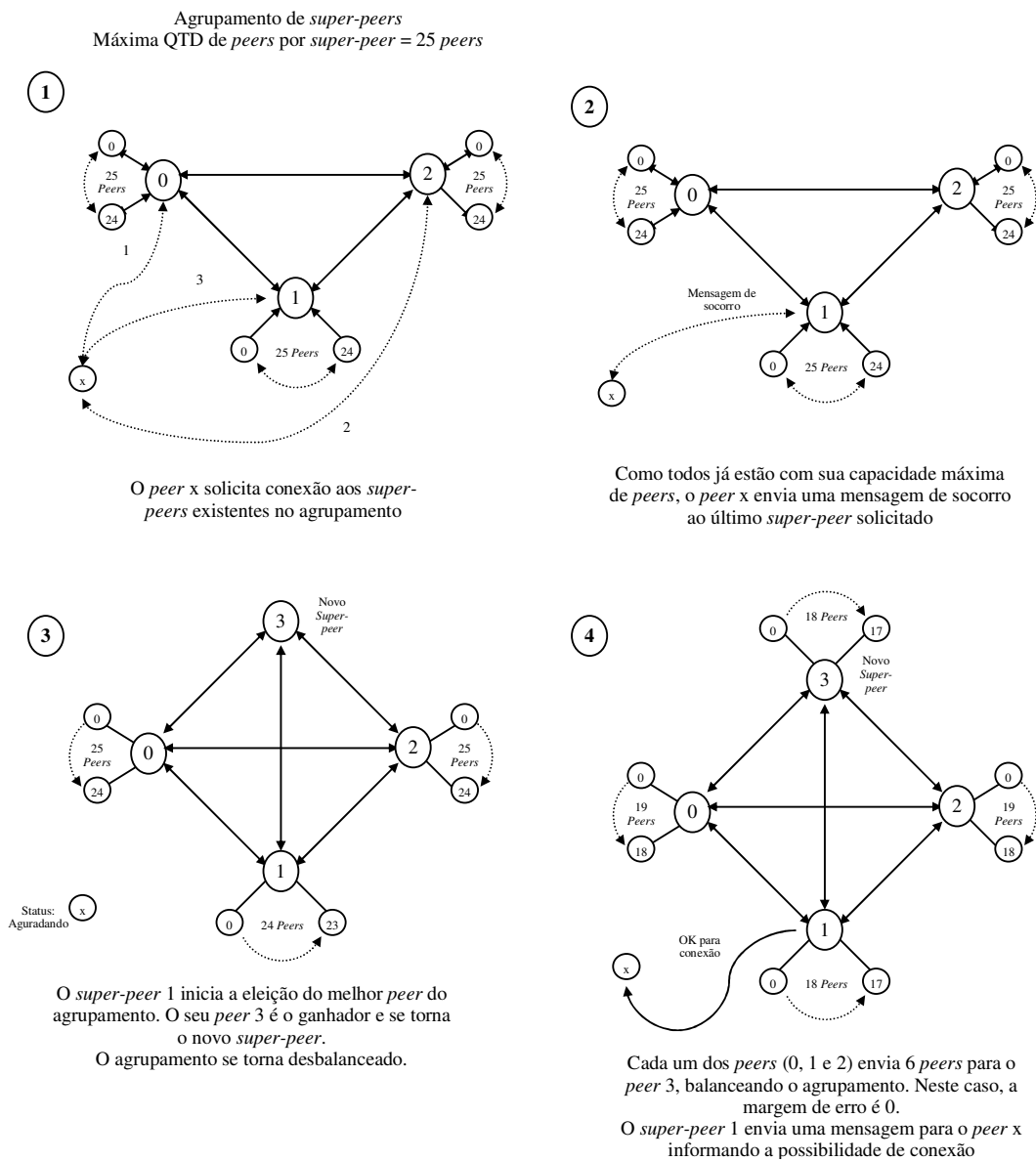


FIG. 4.9c – Agrupamento balanceado

FIG. 4.9 – Balanceamento do agrupamento

Conforme mencionado nesta seção e ilustrado na FIG. 4.10, uma particularidade de conexão do sistema pode ocorrer quando um *peer*, que optou por não ser um *super-peer*, deseja entrar no sistema, mas os *super-peers* do agrupamento com o qual deve se conectar não suportam mais a entrada de novos *peers*, visto que o limite máximo de *peers* foi alcançado. Neste caso, para possibilitar a entrada do novo *peer* ao agrupamento, e conseqüentemente ao sistema, o último *super-peer* com o qual o *peer* tentou se conectar recebe um pedido de “socorro”. A partir daí é iniciada a eleição de um novo *super-peer* dentro do agrupamento, com a ressalva de que este novo *super-peer* não irá substituir nenhum outro. Pelo contrário, ele irá ser mais um dentre os *super-peers* existentes no agrupamento. Neste ponto, pode-se perceber que o sistema se encontra desbalanceado, devendo ser re-balanceado.



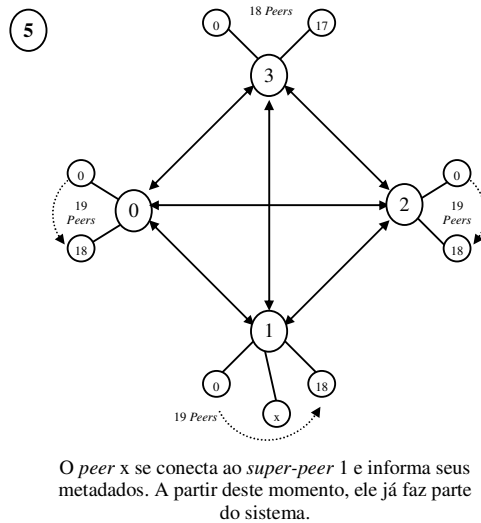


FIG. 4.10 – Caso particular de conexão de um novo *peer* ao sistema. Uso da estratégia de balanceamento do agrupamento.

4.2.2.7 TABELA DE PROPRIEDADES DOS PEERS E SUPER-PEERS

Cada *peer* e/ou *super-peer* do sistema possui uma tabela de propriedades. Esta consiste em armazenar informações particulares sobre eles, sem as quais não seria possível realizar certos procedimentos, tais como rotinas de conexão e re-conexão de *peers* e eleição de *super-peer*, assim como sua própria identificação no sistema.

Essas informações são fornecidas principalmente pelo usuário quando da utilização do sistema pela primeira vez ou quando altera suas configurações, conforme será demonstrado na seção 5.2.1.3. Todavia, algumas delas são fornecidas pelo próprio sistema. A TAB. 4.1 apresenta a definição destas informações.

TAB. 4.1 – Propriedades dos *peers* e *super-peers*

Informação	Descrição
Localização do <i>peer</i>	Informação sobre o IP (Internet Protocol) do <i>peer</i> . Identifica o <i>peer</i> na rede através de um endereço físico exclusivo.
Localização antiga do <i>peer</i>	Informação sobre o IP antigo do <i>peer</i> . Identifica o endereço físico antigo do <i>peer</i> . No caso de uma re-conexão, pode existir a possibilidade de se obter um novo endereço IP fazendo com que as referências a este <i>peer</i> se tornem inconsistentes. Este metadado permite identificar esta mudança, possibilitando que todas as respectivas referências possam ser atualizadas pelo novo endereço IP do <i>peer</i> .
Localização do <i>super-peer</i>	Informação sobre o IP do <i>super-peer</i> a qual pertence o <i>peer</i> . Permite identificar o endereço IP do <i>super-peer</i> , permitindo que o <i>peer</i> possa se conectar e se comunicar com o mesmo.
Características para agregação	Informações sobre o(s) assunto(s) e a localização geográfica do <i>peer</i> , as quais são fundamentais para o funcionamento da estratégia de agrupamento e reenvio de consultas (seção 4.3.2.3.1).
Identificação do <i>peer</i>	Informação sobre a identificação do <i>peer</i> . Esta identificação se refere ao nome extenso da instituição ou do usuário.
Tipo de <i>peer</i>	Informação se o <i>peer</i> é instituição de ensino ou não. Caso seja, automaticamente estará apto a ser um <i>super-peer</i> .
<i>Super-peer</i>	Informação se o <i>peer</i> é um <i>super-peer</i> ou não.
Opção por ser <i>super-peer</i>	Informação se o <i>peer</i> deseja ser um <i>super-peer</i> ou não.
Características físicas	Informações sobre a estabilidade, largura de banda, poder de processamento e capacidade de armazenamento e memória do <i>peer</i> , as quais são necessárias para a estratégia de eleição de novos <i>super-peers</i> .

A tabela de propriedades foi implementada através de uma instância em XML e de seu esquema de validação em DTD (apêndice 1). A FIG. 4.11 apresenta um exemplo desta tabela.

```
<?xml version="1.0" encoding="UTF-8"?>
<Peer>
  <propriedades>
    <IPAtual>127.0.0.1</IPAtual>
    <IPAntigo>127.0.0.1</IPAntigo>
    <IPSuperpeer>192.168.0.1</IPSuperpeer>
    <assunto>Informática</assunto>
    <Localização>Brasil</Localização>
    <nome>IME-RJ</nome>
    <tipo>Instituição de ensino</tipo>
    <superpeer>Sim</superpeer>
    <desejaSerSuperpeer>Sim</desejaSerSuperpeer>
    <caracteristicasFisicas>
      <estabilidade>Alta</estabilidade>
      <larguraBanda>Alta</larguraBanda>
      <poderProcessamento>Médio</poderProcessamento>
      <capacidadeArmazenamento>Alta</capacidadeArmazenamento>
    </caracteristicasFisicas>
  </propriedades>
</Peer>
```

FIG. 4.11 – Exemplo de uma instância do índice da tabela de propriedades

4.2.2.8 COMUNICAÇÃO ENTRE PEERS E SUPER-PEERS – ÍNDICES SP/P E SP/SP

Segundo Nejdil et al. [Nejdil et al., 2003], o uso de índices de roteamento nos *super-peers* reduz significativamente o tempo de distribuição da consulta entre os *peers* que são relevantes. Desta forma, embora muitas estruturas de índices sejam apresentadas na bibliografia, às vezes faz-se necessário que seja definida uma estrutura própria, de acordo com as características do sistema. O uso de uma tabela *hash*, por exemplo, não seria adequada para o ambiente P2P proposto, uma vez que consultas submetidas ao sistema são semanticamente ricas, não se adaptando a esse tipo de índice. Portanto, de acordo com o estudo de sistemas similares, conforme apresentado na seção 2.5, o sistema se baseará em índices de roteamento, tais quais semelhantes aos existentes no sistema Edutella.

Com o intuito de permitir a comunicação e pesquisa eficiente entre os *peers*, o sistema adotará uma estratégia que utilizará duas estruturas de dados, denominadas de índices de roteamento. Uma será referente à comunicação entre o *super-peer* e seus respectivos *peers* (SP/P) e a outra entre um *super-peer* e seus *super-peers* (SP/SP).

Embora não faça parte da arquitetura P2P, o portal ROSA também faz parte do sistema. Desta forma, torna-se necessário definir uma estrutura de dados para que ele possa localizar os *super-peers* relevantes às consultas submetidas pelos usuários. O mesmo é necessário para o serviço de diretório que armazena as referências de todos os *super-peers* do sistema. Essa estrutura de dados será a mesma utilizada para a comunicação entre um *super-peer* e seus *super-peers* (SP/SP) e será detalhada mais adiante.

Índice de roteamento SP/P:

Cada *super-peer* possui um índice de roteamento *super-peer/peer*, conforme mostra a FIG. 4.12. Ela contém informações sobre os metadados de todos os seus *peers* e a informação referente à quantidade destes *peers* que se encontram “on-line”, a qual são de suma importância para o funcionamento do sistema.

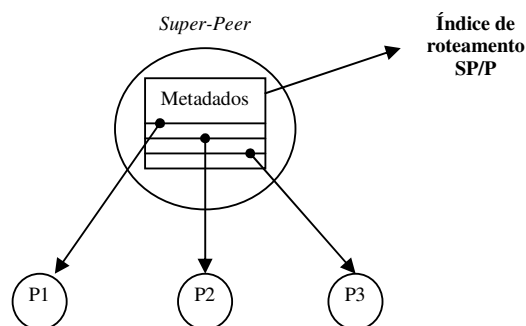


FIG. 4.12 – Índice de roteamento SP/P no sistema

Essas informações são fornecidas por todos os *peers* conectados ao *super-peer*, através do envio de um sub-conjunto das informações contidas na tabela de propriedades e também por ele próprio e podem ser definidas conforme a TAB. 4.2.

TAB. 4.2 – Índice de roteamento SP/P

Informação	Descrição
Localização do <i>peer</i>	Informação sobre o IP do <i>peer</i> .
Características para agregação	Informações sobre o(s) assunto(s) e a localização geográfica do <i>peer</i> .
Tipo de <i>peer</i>	Informação se o <i>peer</i> é instituição de ensino ou não.
Opção por ser <i>super-peer</i>	Informação se o <i>peer</i> deseja ser um <i>super-peer</i> ou não.
Status	Informação se o <i>peer</i> está “on-line” ou “off-line”. O <i>peer</i> com o status “on-line” significa que está ativo no sistema.
Características físicas	Informações sobre a estabilidade, largura de banda, poder de processamento e capacidade de memória do <i>peer</i> .
Quantidade de <i>peers</i>	Informação sobre o total de <i>peers</i> “on-line”. Indica o total de <i>peers</i> “on-line” contidos na agregação.

Sua implementação foi realizada através de uma instância em XML e de seu esquema de validação em DTD, os quais podem ser visto na FIG. 4.13 e no apêndice 2, respectivamente.

```

<?xml version="1.0" encoding="UTF-8"?>
<Peers>
  <peer>
    <IP>127.0.0.1</IP>
    <assunto>Engenharia elétrica</assunto>
    <localizacao>Brasil</localizacao>
    <tipo>Instituicao de ensino</tipo>
    <desejaSerSuperpeer>Sim</desejaSerSuperpeer>
    <status>on-line</status>
    <caracteristicasFisicas>
      <estabilidade>Medio</estabilidade>
      <larguraBanda>Medio</larguraBanda>
      <poderProcessamento>Medio</poderProcessamento>
      <capacidadeArmazenamento>Medio</capacidadeArmazenamento>
    </caracteristicasFisicas>
  </peer>
</Peers>

```

FIG. 4.13 – Exemplo de uma instância do índice de roteamento SP/P

Como em qualquer ambiente dinâmico, essas informações devem ser mantidas atualizadas, para não comprometer nem o desempenho e nem a confiabilidade do sistema. Portanto, cada *peer* é responsável pela manutenção de suas informações, para os quais *triggers* serão implementados de modo a fazer com que quaisquer alterações destas informações sejam detectadas e enviadas ao(s) respectivo(s) *super-peer(s)* para as devidas atualizações.

Por outro lado, conforme já discutido na seção 4.2.2.2, caso um *peer* deixe o sistema, o *super-peer*, além de mudar o seu status para “off-line”, deve manter as suas respectivas informações por um período de tempo (parametrizável), como por exemplo, uma semana. Passado este tempo, as referências devem ser removidas. Isto faz com que o índice de roteamento SP/P mantenha-se atualizado, ao mesmo tempo em que permite que a conexão ou a re-conexão de um *peer* ao sistema seja mais rápida.

O índice de roteamento SP/P é uma estrutura de dados fundamental para que consultas possam ser direcionadas eficientemente aos *peers* relevantes e, conseqüentemente, serem respondidas mais rapidamente. Desta forma, ao receber uma consulta relevante a sua agregação, o *super-peer* analisará as informações de metadados de cada um dos seus *peers*, reenviando-a automaticamente para aqueles aptos a respondê-la, otimizando assim todo o seu processamento.

Índice de roteamento SP/SP

O índice de roteamento *super-peer/super-peer* está presente no portal ROSA, assim como em cada *super-peer* do sistema, conforme ilustra a FIG. 4.14.

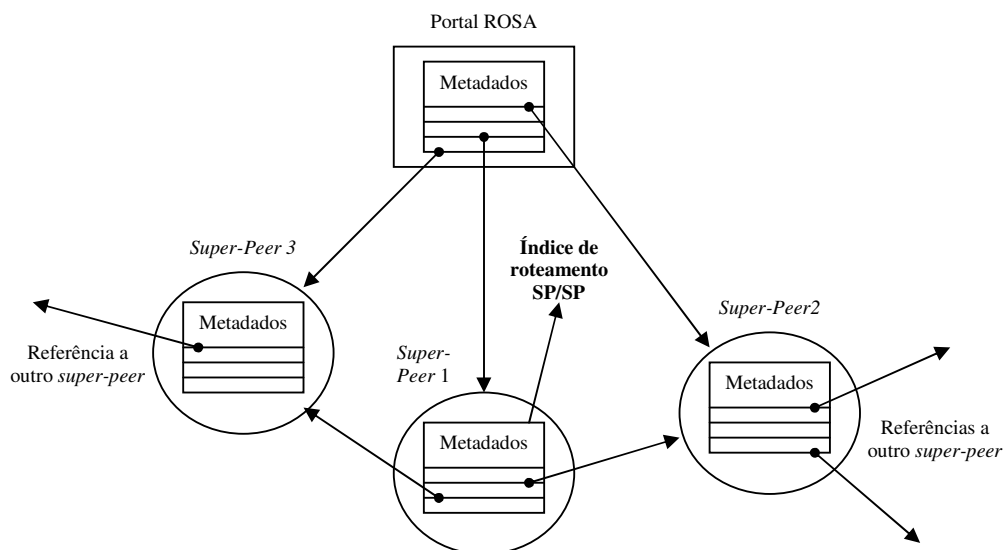


FIG. 4.14 – Índice de roteamento SP/SP no sistema

Ele contém as informações sobre os metadados de cada *super-peer* conectado ao portal ROSA e/ou ao *super-peer*, sendo também de suma importância para o funcionamento do sistema. Este índice também está presente no serviço de diretório, onde é povoado com referências a todos os *super-peers* do sistema.

Essas informações são fornecidas pelos *super-peers*, através do envio de um sub-conjunto das informações contidas na tabela de propriedades, ao portal ROSA, ao serviço de diretório e aos *super-peers* a eles conectados, assim como por eles próprios. São basicamente as mesmas contidas no índice de roteamento SP/P, porém são referentes ao *super-peer*, conforme mostra a TAB. 4.3.

TAB. 4.3 – Índice de roteamento SP/SP

Informação	Descrição
Localização do <i>super-peer</i>	Informação sobre o IP do <i>super-peer</i> .
Características para agrupamento	Informações sobre o(s) assunto(s) e a localização do <i>super-peer</i> .
Status	Informação se o <i>super-peer</i> está “on-line” ou “off-line”.
Quantidade de <i>super-peers</i>	Informação sobre o total de <i>super-peers</i> “on-line”. Indica o total de <i>super-peers</i> “on-line” a qual se está conectado.
Quantidade de <i>super-peers</i> que tratam do mesmo assunto	Informação sobre o total de <i>super-peers</i> “on-line” e que tratam do mesmo assunto. Indica o total de <i>super-peers</i> “on-line” do agrupamento. É utilizada quando se deseja saber para quanto <i>super-peers</i> uma determinada consulta está sendo reenviada (seção 4.3.2.3.4.1).

Sua implementação foi realizada através de uma instância em XML e de seu esquema de validação em DTD, os quais podem ser vistos na FIG. 4.15 (exemplo) e no apêndice 3 respectivamente.


```

<?xml version="1.0" encoding="UTF-8"?>
<Peer>
  <superpeer>
    <IP>127.0.0.1</IP>
    <assunto>Engenharia elétrica</assunto>
    <localizacao>Brasil</localizacao>
    <status>on-line</status>
  </superpeer>
  <qtdSuperpeers>1</qtdSuperpeers>
  <qtdSuperpeersMesmoAssunto>1</qtdSuperpeersMesmoAssunto>
</Peer>

```

FIG. 4.15 – Exemplo de uma instância do índice de roteamento SP/SP

A manutenção deste índice ocorre da mesma maneira que a do índice de roteamento SP/P, isto é, cada *super-peer* é responsável pela manutenção de suas informações exportadas. *Triggers* serão implementados de modo a fazer com que quaisquer alterações destas informações sejam detectadas e enviadas ao portal ROSA, ao serviço de diretório e ao(s) respectivo(s) *super-peer(s)* para as devidas atualizações.

Quando o portal ROSA, o serviço de diretório ou um *super-peer* detecta ou recebe uma mensagem de que um *super-peer* está ausente (o *super-peer* pode ter deixado o sistema ou se tornado um *peer* comum), estes além de mudarem o seu status para “off-line”, devem manter as suas respectivas informações por algum período de tempo (parametrizável), como por exemplo, uma semana. Passado este tempo, as referências devem ser removidas. Isto faz com que o índice de roteamento SP/SP mantenha-se atualizado, ao mesmo tempo em que permite que o portal ROSA ou um *super-peer* não perca tempo tentando localizar o *ex-super-peer*, permitindo desta forma que a distribuição da consulta seja mais rápida. Este procedimento, como se pode observar, é bem semelhante ao realizado no índice de roteamento SP/P. Com relação ao serviço de diretório, permite que este informe aos novos *peers* somente referências a *super-peer* válidos, aumentando o desempenho e qualidade das conexões iniciais no sistema.

O índice de roteamento SP/SP é uma estrutura de dados fundamental para que consultas possam ser direcionadas eficientemente aos *super-peers* relevantes e, conseqüentemente, serem respondidas mais rapidamente. Desta forma, ao receber uma consulta, o portal ROSA ou o *super-peer* analisará as informações de metadados de cada um dos *super-peers* a ele conectado, reenviando a consulta aos *super-peers* relevantes e estes aos seus *peers*, otimizando assim todo o processamento. No caso do serviço de diretório, atua oferecendo aos

novos *peers* todas as possibilidades de conexão possíveis segundo suas características particulares, possibilitando que este possa se unir ao sistema.

4.2.2.9 PROCESSAMENTO DE CONSULTAS

O objetivo desta seção é definir as características do processamento de consultas que são pertinentes ao funcionamento do ambiente P2P e não do sistema como um todo (as quais serão discutidas na seção 4.3.2.3). Assim, elas podem ser caracterizadas como:

- Identificação da consulta. Diferentemente de alguns sistemas existentes, as consultas não possuirão um identificador único na rede, gerado em tempo de execução. Porém, possuirão duas identificações: uma referente ao IP do *peer* que a submeteu (ou do portal ROSA) e a outra do *super-peer* responsável por ela (*super-peer* remetente).
- Reenvio de consultas. Devida a estratégia de agrupamento utilizada, consultas serão enviadas às agregações e/ou aos agrupamentos pertinentes. Desta forma, um *super-peer* ao recebê-la, verificará no seu índice de roteamento se o *super-peer* remetente pertence ao seu agrupamento. Caso pertença, não será necessário retransmiti-la, pois o *super-peer* remetente a fará. Caso não pertença, deverá fazê-la, retransmitindo a consulta aos demais *super-peers* do agrupamento.

Esta seção finaliza a definição sobre o funcionamento do ambiente P2P. Porém, como em qualquer sistema, falhas podem acontecer a qualquer momento, principalmente em se tratando de um ambiente tão instável, como o P2P. Desta forma, mecanismos de tolerância a falhas devem ser implementados de modo a fazer com que o sistema não se torne inoperante, conforme tratado na próxima seção.

4.2.3 TOLERÂNCIA A FALHAS

Tolerância a falhas é a capacidade que um sistema possui de não se tornar inoperante ou deficiente na eventual presença de falhas [Brito e Moura, 2004].

Como mencionado na seção 2.2.2.1, tolerância a falhas é uma das características da arquitetura baseada em *super-peers*. Ela é de vital importância para manter a confiabilidade do sistema, uma vez que possui procedimentos para combater as possíveis falhas que podem ocorrer, tão comum em ambiente P2P.

Estas possíveis falhas vão desde a ausência momentânea de um *peer* até um *timeout* de uma consulta. Porém, esta seção tem como objetivo oferecer uma visão básica sobre o assunto e, portanto, aborda somente as que se referem à falha física do *peer*, classificadas em:

- Falha de um *super-peer*. A falha de um *super-peer* pode ser detectada tanto pelos seus *peers* quanto pelos *super-peers* a ele conectados. Portanto, no caso de ser feita pelos seus *peers*, os mesmos devem imediatamente se re-conectar ao sistema utilizando o procedimento referente à primeira conexão de um *peer* ao sistema (seção 4.2.2.2) e, desta forma, se conectar a um outro *super-peer*, ou ainda se tornar um *super-peer*. É importante ressaltar que o interesse aqui é exclusivamente contornar a falha do *super-peer*. Quando a falha é detectada pelos *super-peers*, a referência ao *super-peer* que falhou deve ser removida do índice de roteamento SP/SP. Caso um dos *peers* que se re-conectou torne-se um *super-peer*, este informará aos demais *super-peers* sobre seu novo papel, a partir do qual as devidas atualizações nos índices de roteamento pertinentes devem ser feitas;
- Falha de um *peer*. A falha de um *peer* é um processo bem mais simples em relação a falha de um *super-peer*. Ela é detectada pelo *super-peer* da agregação em questão. Desta forma, caso ocorra a falha de um *peer*, o seu *super-peer* não poderá lhe direcionar nenhuma consulta, porém poderá exibir ou informar ao *peer* solicitador, as informações relevantes à consulta que se encontram no seu índice de roteamento (o índice de roteamento possui informações sobre os metadados dos *peers*). Obviamente, elas serão exibidas com o status de “off-line”, porém servirão como base para a tomada de decisão do usuário, uma vez que os mesmos saberão que elas existem no sistema e poderão re-submeter a mesma consulta em um outro momento, a fim de obtê-las completamente;
- Falha do portal ROSA. A falha do portal ROSA é detectada pelos usuários que não possuem o ROSA - P2P, porém desejam submeter consultas ao sistema. Caso esta máquina fique inoperante, os usuários serão direcionados a uma máquina de contingência, semelhante à máquina original em hardware e software, assim como em dados e configurações, possibilitando que os usuários possam acessar o sistema normalmente;

- Falha do serviço de diretório. A falha do serviço de diretório é detectada por um novo *peer* que deseje se conectar ao sistema e pelos *super-peers* já existentes. Em ambos os casos, o novo *peer* ou o *super-peer* devem se dirigir à máquina de contingência.

Concluindo, fica claro perceber que os procedimentos apresentados em relação à tolerância a falhas, têm como principal objetivo manter o sistema funcionando, independentemente da situação em que se encontra. Estes procedimentos permitem aumentar a confiabilidade do sistema, tornando-o mais robusto e permitindo que o mesmo jamais se torne inoperante.

4.3 DEFINIÇÃO DO SISTEMA DE INTEGRAÇÃO DE DADOS

A definição do sistema de integração de dados é de vital importância para que os dados possam ser devidamente e otimadamente localizados, filtrados, recuperados, armazenados e integrados. No contexto do sistema ROSA - P2P, pode-se considerá-lo como sendo o seu coração, uma vez que o sistema visa principalmente a integração dos objetos de aprendizagem oriundos dos diversos *peers*.

Sendo assim, para um melhor entendimento, sua definição será dividida quanto a sua arquitetura e estratégia adotada, sendo estes os assuntos tratados neste capítulo.

4.3.1 ARQUITETURA DE INTEGRAÇÃO

Como se pode verificar em [Brito e Moura, 2005], não existe ainda uma topologia bem definida quanto à(s) arquitetura(s) de integração de dados *peer-to-peer*. Ela(s) se define(m) de acordo com o objetivo, arquitetura, funcionamento e características de cada sistema P2P, impossibilitando desta maneira, que uma referência a uma arquitetura já definida possa ser utilizada.

Desta forma, baseando-se no objetivo, arquitetura, funcionamento e características do sistema ROSA - P2P, uma arquitetura válida para integração dos dados pode ser definida. No entanto, de forma a melhor definir as características desta arquitetura de integração, vale observar algumas premissas e/ou decisões já consideradas no desenvolvimento do sistema, tais como:

- Objetivo: integrar os objetos de aprendizagem oriundos dos diversos *peers* ROSA, de acordo com o ambiente P2P criado e descrito na seção 4.2;
- Arquitetura: baseada na arquitetura de *super-peers*, segundo seção 4.2.1;
- Funcionamento: utiliza uma estratégia de agrupamento de *super-peers* de acordo com o assunto e localização. Possui algoritmo para eleição de *super-peers* e balanceamento do sistema. A comunicação entre os *peers* é feita através de estruturas específicas, com suporte a tolerância a falhas;

- Características: os dados dos *peers* agregam valores semânticos. Não possui esquema conceitual de seus dados.

Com base nestas informações, uma definição da arquitetura de integração de dados para o sistema ROSA - P2P já pode ser estabelecida. Como se pretende integrar os objetos de aprendizagem oriundos dos diversos *peers* ROSA e sabendo-se que tais objetos possuem valor semântico, pode-se dizer que a arquitetura de integração possuirá recursos extras, além do que simplesmente compartilhar arquivos. Esses recursos dizem respeito à utilização de ontologias, metadados ou qualquer estrutura que vise embutir algum valor semântico aos dados dos *peers* participantes e, conseqüentemente, auxiliar no entendimento do seu real significado.

As características condizentes com o funcionamento do sistema estão diretamente relacionadas com a sua arquitetura, que é baseada em *super-peers*. Assim, a arquitetura de integração de dados deverá seguir este mesmo padrão, permitindo meios para que a integração aconteça.

A FIG. 4.16 exhibe a arquitetura ou sistema de integração de dados definido para o sistema ROSA - P2P. Pode-se observar que as características do ambiente P2P foram mantidas, não comprometendo o funcionamento do sistema atual. Cada *peer* possui seu próprio sistema de integração de dados. Isto se deve ao fato de cada *peer* possuir um ponto de entrada de consultas, servindo também como ponto de integração dos resultados. Uma interface amigável atua na interação com o usuário, possibilitando que as consultas possam ser submetidas de maneira fácil, com a exibição dos respectivos resultados de forma clara. O módulo de integração de dados é o responsável por todo o processo que se estende desde a submissão da consulta até a exibição do(s) resultado(s) ao(s) usuário(s). Neste processo estão incluídas, dentre outras etapas: a localização das fontes de dados; o reenvio e a reescrita de consultas; o processamento de consultas; a resolução de conflitos; e a integração dos dados. Existem também vocabulários controlados que agregam valor semântico aos dados, atuando e auxiliando em todas essas etapas.

Desta forma, quando uma consulta for submetida por um usuário, o respectivo *super-peer* a reescreverá e a submeterá para si próprio, armazenando o resultado em *cache*. Em seguida, reenviará a consulta original aos seus *peers* e aos demais *super-peers* relevantes, ativando um relógio que controlará o tempo que o mesmo aguardará pelas respostas. Cada um destes *peers* ou *super-peers* que receberem a consulta deverão reescrevê-la e submetê-la para si próprio,

porém o resultado deverá ser enviado ao *peer* que submeteu a consulta pela primeira vez (*peer* solicitador), que o armazenará em *cache*. No caso, estes *super-peers* também devem reenviar a consulta para cada um de seus *peers*, de modo que todos os *peers* relevantes possam respondê-la. Todavia, quando todos os resultados tiverem sido retornados ou o tempo de espera de resultados estiver sido esgotado, os dados residentes em *cache* serão integrados e a resposta da integração retornada para o usuário.

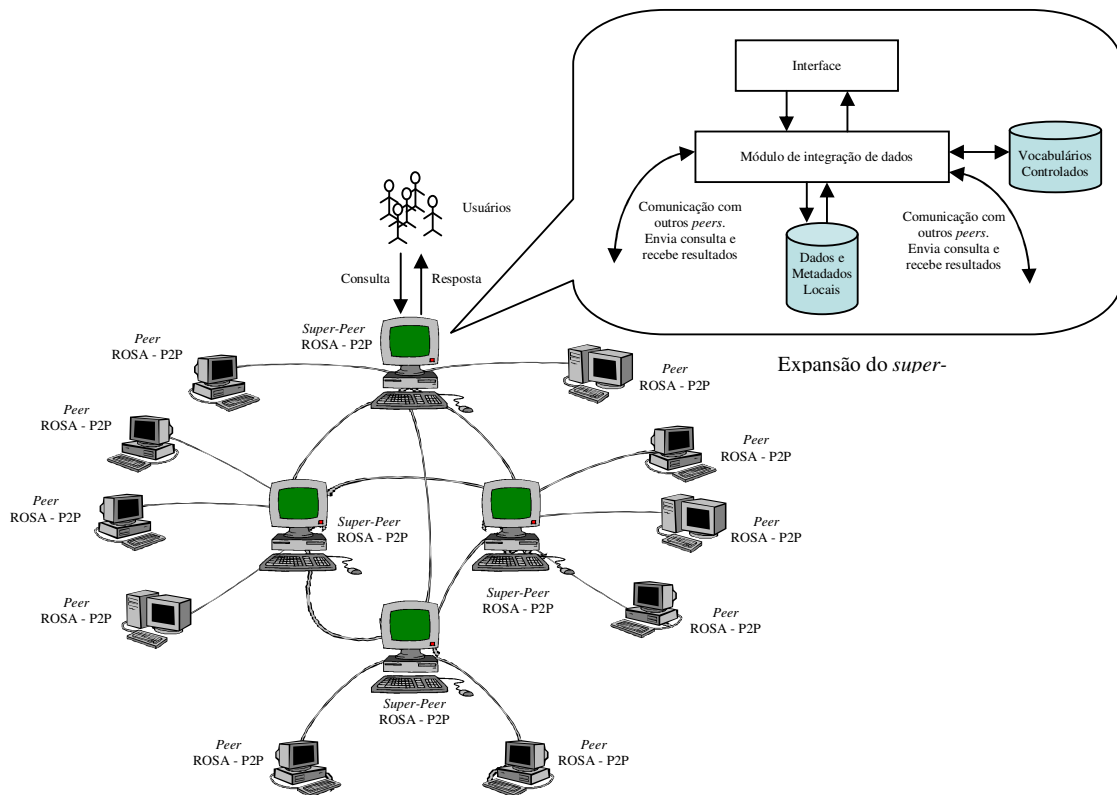


FIG. 4.16 – Arquitetura de integração de dados do sistema ROSA - P2P

Uma vez definida a arquitetura/sistema de integração de dados, é preciso especificar como se dará o seu funcionamento, ressaltando principalmente, suas características e particularidades quanto à estratégia adotada.

4.3.2 ESTRATÉGIA ADOTADA

A definição da estratégia de um sistema de integração de dados é um grande desafio em debate na construção de informações globais e uma tarefa complexa a ser gerenciada. Portanto, todos os detalhes e aspectos de otimização devem ser cuidadosamente observados, de modo a obter o mínimo de complexidade com o máximo de simplicidade, otimização, produtividade e confiabilidade.

Sendo assim, de modo a facilitar o seu entendimento, a definição da estratégia do sistema de integração de dados do ROSA - P2P será dividida em vocabulários controlados, SEVC – serviço de entrega de vocabulários controlados, processamento de consultas e integração dos dados, detalhados a seguir.

4.3.2.1 VOCABULÁRIOS CONTROLADOS

Como visto na seção 2.4.3.2, ontologia é uma ferramenta essencial no processo de integração, principalmente quando os dados são oriundos de diferentes fontes de informação ou de esquemas distintos, possibilitando que o real significado de um dado ou de uma informação seja corretamente compreendido.

Muito embora ainda se encontre na literatura referência a tesouros e taxonomias [McGuinness, 2003] como ontologias, preferiu-se adotar as definições de Gruber [Gruber, 1993] e Corcho [Corcho, Lopes e Pérez, 2002], que consideram uma ontologia como um conjunto de regras de conceitos, relacionamentos e restrições e/ou regras sobre um determinado domínio de conhecimento.

Desta forma, dentro do contexto do sistema ROSA - P2P, se utilizará a terminologia de vocabulários controlados ao invés de ontologias. Estes então, surgem como ferramentas que facilitam e possibilitam a interpretação e recuperação das informações, ao mesmo tempo em que viabilizam o intercâmbio entre os sistemas, permitindo pesquisas mais apuradas e restritas às informações realmente relevantes. Tornam-se indispensáveis para: a correta localização dos *peers* relevantes à consulta; na reescrita da consulta pelos *peers* receptores; na resolução de possíveis conflitos; e na sugestão de opções e caminhos associados com a pesquisa,

auxiliando o usuário no alcance dos seus objetivos e na automação de tarefas que exijam raciocínio. Isto pode ser observado durante a busca de resultados, onde a operação de transitividade pode ser aplicada, aumentando a qualidade e precisão no processo de integração de dados.

De modo a contemplar esses benefícios, o sistema proposto necessita de 3 vocabulários controlados distintos. Os dois primeiros, denominados de vocabulário controlado global e local, por serem mais complexos, passaram por um processo preliminar para sua criação. O terceiro, denominado de palavras chaves, surgiu diante da necessidade de se detectar, em tempo de execução da consulta, de que assunto a mesma se tratava, permitindo seu encaminhamento aos *peers* relevantes.

Ambos os vocabulários controlados foram definidos através da linguagem XML esquema [XMLSchema]. Optou-se por esta linguagem, ao invés de uma linguagem específica de definição de ontologias a exemplo da OWL [OWL], pelos seguintes motivos: 1) os vocabulários controlados não necessitam de uma riqueza maior de conceitos, relacionamentos entre esses conceitos, axiomas e restrições, conforme será discutido nas próximas seções; 2) existência de uma API específica acoplada à linguagem de programação JAVA [JAVA], que permite a correta interpretação de documentos descritos em XML. Se a linguagem OWL fosse utilizada, seria necessário um interpretador OWL compatível com JAVA para possibilitar esta interpretação. Embora isso já exista, iria requerer um tempo maior de estudo; 3) familiaridade com a linguagem XML, que permitiu que os vocabulários controlados pudessem ser construídos rapidamente e com qualidade, atingindo o objetivo a qual eles se propõem.

As definições dos vocabulários controlados do sistema serão apresentadas a seguir:

4.3.2.1.1 ESTRATÉGIA PARA CRIAÇÃO DO VOCABULÁRIO CONTROLADO GLOBAL E LOCAL

A estratégia utilizada para a definição dos vocabulários controlados global e local está baseada nas técnicas apresentadas em [Buccella e Cechich, 2003] e [Noy e McGuinness, 2001]. Consiste, primeiramente, em conhecer e caracterizar as fontes de dados, assim como definir os termos relacionados. Para tal, dois estágios principais são apresentados:

- Análise das fontes de informações: implica na análise completa das fontes de informações, isto é, quais e como são armazenadas, seu significado (semântica), etc. No sistema ROSA, esta análise tem o seguinte resultado:

- Não existe a noção de esquema conceitual no ROSA, o qual possibilitaria a aplicação de restrições e tipagem de dados. O ROSA considera apenas mapas conceituais de dados, que são baseados em um modelo conceitual de dados (ver FIG. 3.4);

- Um mapa conceitual está baseado em 3 classes de LOs (curso, disciplina e tópico) e um conjunto de predicados;

- O mapa conceitual é ele próprio a instância de dados;

- As instâncias são armazenadas em XML

- Definição dos termos: equivale à escolha de uma lista de termos ou conceitos de acordo com o mapa conceitual do sistema ROSA: curso, disciplina, tópico e lista de predicados.

Na seção 3.4 foi apresentado o modelo de dados do sistema ROSA, que mostra como se dão os relacionamentos entre objetos de aprendizagem no sistema. No entanto, todos os mapas são criados de forma a obedecer as restrições da realidade institucional brasileira, tal como apresentado na FIG. 3.4, onde cada uma dessas classes (curso, disciplina e tópico) representam LOs, cujas instâncias e predicados, através dos seus relacionamentos, constituem um mapa conceitual. Assim sendo, percebe-se que a integração se dará no nível das próprias instâncias, fazendo com que ambos os vocabulários controlados global e local sejam definidos de acordo com os termos usuais que as representam.

Seria interessante apontar aqui a necessidade de um controle que verificasse discrepâncias de ordem sintática e semântica durante a criação de um mapa conceitual. Por exemplo, um curso, que na nossa realidade identifica um programa dentro de uma instituição, não pode abranger tópico, apenas disciplinas. Disciplinas e tópicos podem compreender outros tópicos e um curso não pode jamais ser pré-requisito de uma disciplina. Assim, no desenvolvimento do sistema proposto, assume-se que um mapa conceitual esteja bem definido, do ponto de vista sintático e semântico.

A identificação estrutural de cada LO é dada através do próprio descritor LOM, sob o metadado *nível de agregação*, que indica se o mesmo se refere a um curso, disciplina ou

tópico, solucionando a priori, os possíveis conflitos estruturais. Desta forma, o papel dos vocabulários controlados global e local se reserva apenas aos problemas de caráter semântico. Suas definições podem ser vistas nas seções a seguir:

4.3.2.1.2 VOCABULÁRIO CONTROLADO GLOBAL

Devido às suas características genéricas, o vocabulário controlado global será utilizado por todos os *peers* do sistema. Este é formado por um vocabulário de sinônimos, segundo os predicados existentes no sistema ROSA, e por algumas propriedades específicas inspiradas no uso de tesouros, acrescidas de propriedades atribuídas aos predicados dos LOs, tais como “transitividade” e “inversoDe”. De fato, este vocabulário controlado será de suma importância na reescrita da operação de navegação ou “browsing” (seção 4.3.2.3.2), permitindo que a consulta seja reescrita de forma a obter como resultado todos os dados pertinentes, independentemente da semântica utilizada por cada *peer* para descrever um dado predicado.

O vocabulário controlado global, definido através de uma instância em XML e de seu esquema de validação em XML esquema, pode ser visto nos apêndices 5 e 4 respectivamente. A FIG. 4.17 mostra parte deste vocabulário, tomando como exemplo o predicado *é_pré-requisito_de*.

```
<predicado>
  <nome>é_pré-requisito_de</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>true</transitivo>
  <inversoDe>requer</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>fundamenta</nomeEquivalente>
    <nomeEquivalente>é_base_para</nomeEquivalente>
    <nomeEquivalente>é_condição_para</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>fundamenta</nome>
  <use>é_pré-requisito_de</use>
</predicado>
<predicado>
```

```

    <nome>é_base_para</nome>
    <use>é_pré-requisito_de</use>
</predicado>
<predicado>
    <nome>é_condição_para</nome>
    <use>é_pré-requisito_de</use>
</predicado>

```

FIG. 4.17 – Vocabulário controlado global parcial - predicado *é_pré-requisito_de*

Pode-se verificar que cada elemento “predicado” possui:

- Nome do predicado: permite sua identificação;
- Tipo de predicado: especifica se é um predicado associativo ou de agregação (definição do sistema ROSA);
- Transitivo: indica se o predicado é transitivo ou não, permitindo desta forma a extensão da consulta;
- Inverso de: revela se o predicado é inverso e possibilita que uma consulta possa retornar também o sujeito e não só o objeto;
- Recursivo: especifica se o predicado é recursivo ou não;
- Equivalente: indica quais são os predicados sinônimos ao predicado inicial, permitindo que estes valores possam fazer parte da consulta reescrita.

Contudo, pode-se verificar ainda que alguns elementos “predicado” possuem somente o elemento “nome do predicado” e “use”. É o caso dos predicados *fundamenta*, *é_base_para* e *é_condição-para*. Estes predicados são sinônimos ao predicado *é_pré-requisito_de* e o elemento “use” indica o predicado sinônimo que deve ser utilizado para a reescrita da consulta. Isto faz com que todas os demais elementos (tipoDePredicado, transitivo, inversoDe, recursivo e equivalente) não sejam reproduzidos para todo elemento “predicado” desnecessariamente, contribuindo para a clareza do entendimento do vocabulário controlado.

Um pequeno exemplo de uso do vocabulário controlado global, baseado na FIG. 4.17 e na sintaxe do MEC ROSA [Coutinho, 2004], pode ser verificado a seguir. Suponha a consulta1, definida como:

- 1) Selecione os LOs que possuam título igual a *banco de dados distribuídos* e que sejam pré-requisito de algum outro LO.

```
Consulta: select|LOs@lom/geral/titulo = banco de dados distribuídos  
browsing|LOs@(é_pré_requisito_de )
```

Com o auxílio do vocabulário controlado global, esta consulta será reescrita pelo módulo de processamento de consultas (seção 4.3.2.3), gerando uma nova consulta, definida como:

```
Consulta: select|LOs@lom/geral/titulo = banco de dados  
browsing|LOs@((é_pré-requisito_de or fundamenta) or  
(é_base_para or é_condição_para))
```

O resultado obtido pela consulta 1 não será completo, pois não compreende todos os valores que semanticamente atendem a consulta, isto é, podem existir predicados que sintaticamente são diferentes, mas que semanticamente representam a mesma informação. Já o resultado obtido pela consulta reescrita será completo, pois compreenderá todos os valores que semanticamente atendem a mesma.

Portanto, pode-se concluir que o vocabulário controlado global é de grande importância semântica para o sistema, uma vez que permite que consultas sejam reescritas corretamente abrangendo todas as possibilidades semânticas dos termos em questão, alcançando assim todos os resultados existentes.

Contudo, a utilização do vocabulário controlado global será exemplificado e discutido em maiores detalhes na seção 4.3.2.3.3, referente à reescrita de consultas.

4.3.2.1.3 VOCABULÁRIO CONTROLADO LOCAL

O vocabulário controlado local se caracteriza por especificar vocabulários de acordo com cada assunto e idioma tratado no sistema, referente respectivamente a um domínio de conhecimento e ao país de origem. Desta forma, o sistema possuirá tantos vocabulários controlados locais quantos forem esses assuntos e idiomas. Por exemplo, *peers* que tratem do assunto informática localizados no Brasil se basearão no vocabulário controlado local de informática no idioma português; *peers* que tratem do assunto informática localizados no China se basearão no mesmo vocabulário controlado local de informática no idioma chinês;

peers que tratem do assunto medicina localizados no Brasil se basearão no vocabulário controlado local de medicina no idioma português; *peers* que tratem do assunto medicina localizados na Espanha se basearão neste mesmo vocabulário controlado local de medicina no idioma espanhol; e assim por diante para os demais *peers*. Percebe-se então que, ao contrário do vocabulário controlado global, somente os *peers* relativos a um mesmo domínio é que possuirão um vocabulário controlado local daquele assunto, de acordo com o idioma utilizado.

O vocabulário controlado local está baseado na estrutura de um tesouro e é formado por um vocabulário de sinônimos segundo os LOs (curso, disciplina e tópico) e por termos equivalentes, genéricos, específicos e associados. De certo será de vital importância na reescrita da operação de seleção (seção 4.3.2.3.2), permitindo que a consulta seja reescrita de forma a selecionar todos os dados pertinentes, independentemente da semântica utilizada por cada *peer* para descrever um dado LO.

O vocabulário controlado local, definido através de uma instância em XML e de seu esquema de validação em XML esquema, pode ser visto nos apêndices 7 e 6 respectivamente. Este vocabulário teve como base o tesouro já existente no ROSA, relativo ao domínio de Banco de Dados. A FIG. 4.18 exhibe parte deste vocabulário controlado, tomando como exemplo o LO *gerenciamento de banco de dados*.

```
<LO>
  <nivelAgregacao>curso</nivelAgregacao>
  <nivelAgregacao>disciplina</nivelAgregacao>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>gerenciamento de banco de dados</nome>
  <equivalente>
    <nomeEquivalente>BD</nomeEquivalente>
    <nomeEquivalente>banco de dados</nomeEquivalente>
  </equivalente>
  <associado>
    <nomeAssociado>aplicações de banco de dados</nomeAssociado>
    <nomeAssociado>sistemas de banco de dados</nomeAssociado>
  </associado>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>database management</termoldioma>
    </idioma>
    <idioma>
```

```
<nomeldioma>Francês</nomeldioma>
<termoldioma>base de données</termoldioma>
</idioma>
</idiomas>
</LO>
```

FIG. 4.18 – Vocabulário controlado local parcial - gerenciamento de banco de dados

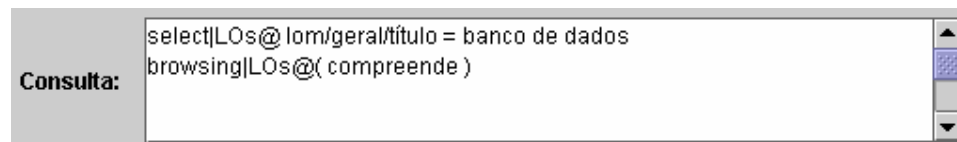
Pode-se constatar que cada elemento “LO” compreende:

- Nível de agregação: indica se o LO é um curso, disciplina ou tópico;
- Nome do LO: permite sua identificação;
- Equivalente: indica quais são os LOs sinônimos ao LO inicial, permitindo que estes valores possam fazer parte da consulta reescrita;
- Associado: identifica os termos associados ao LO, permitindo que estes possam ser também exibidos ao usuário, auxiliando-os no alcance do seu objetivo.
- Idioma: identifica o LO em outros idiomas, possibilitando o uso do sistema por *peers* que trabalhem com qualquer idioma. Estes podem ser adicionados ao vocabulário controlado local, bastando apenas sua especificação quanto a cada LO. No caso, somente a língua Inglesa e Francesa foram especificadas e implementadas, servindo como exemplo e validando o uso do sistema em outro idioma.

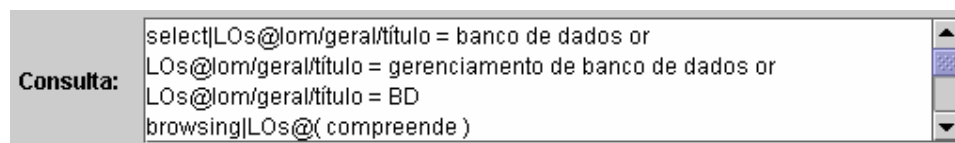
Semelhantemente ao vocabulário controlado global, o vocabulário controlado local também possui o elemento “use” em alguns de seus LOs. Sua finalidade é idêntica, fazendo com que o vocabulário controlado local possa ser melhor estruturado, contribuindo assim para a clareza do documento XML.

Um pequeno exemplo de uso do vocabulário controlado local, baseado na FIG. 4.18 e na sintaxe do MEC ROSA, pode ser visto a seguir. Suponha a consulta 1, definida como:

- 1) Selecione os LOs que possuam título igual a *banco de dados* e que *compreendam* algum outro LO.



Esta consulta, com o suporte do vocabulário controlado local, será reescrita pelo módulo de processamento de consultas, gerando uma nova consulta, definida como:



The image shows a window titled "Consulta:" containing a SQL query. The query is: `select|LOs@lom/geral/titulo = banco de dados or
LOs@lom/geral/titulo = gerenciamento de banco de dados or
LOs@lom/geral/titulo = BD
browsing|LOs@(compreende)`

A análise e conclusão deste exemplo são similares aos exibidos na seção anterior. De fato, o resultado obtido pela consulta reescrita será bem mais completo, pois compreenderá todos os valores que semanticamente atendem a consulta, independente inclusive, do idioma da consulta.

Portanto, pode-se concluir que o vocabulário controlado local, assim como o global, é de grande importância semântica para o sistema, uma vez que permite que consultas sejam reescritas de forma a abranger todas as possibilidades semânticas dos termos em questão, alcançado assim todos os resultados existentes na base de dados.

Exemplos de utilização do vocabulário controlado local serão exemplificados e discutidos em maiores detalhes na seção 4.3.2.3.3.

4.3.2.1.4 VOCABULÁRIO CONTROLADO DE PALAVRAS CHAVES

O vocabulário controlado de palavras chaves é um vocabulário diferente dos demais vistos até aqui. Ele é formado por um vocabulário relacionado segundo cada assunto existente no sistema proposto e permite detectar, em tempo de execução da consulta, de que assunto a mesma trata, permitindo desta forma o encaminhamento da consulta aos *peers* aptos a respondê-la.

Este vocabulário controlado se apresenta como um conjunto de termos semanticamente relacionados, cobrindo de modo abrangente um domínio de conhecimento. Contudo, da mesma forma que os demais vocabulários controlados, este é de vital importância ao sistema, pois permite que o processamento de consultas seja otimizado, uma vez que as consultas serão identificadas por assunto, sendo submetidas somente aos *peers* que tratem deste assunto.

O vocabulário controlado de palavras chaves também é definido através de uma instância em XML e de um esquema de validação em XML esquema, apresentado nos apêndices 9 e 8

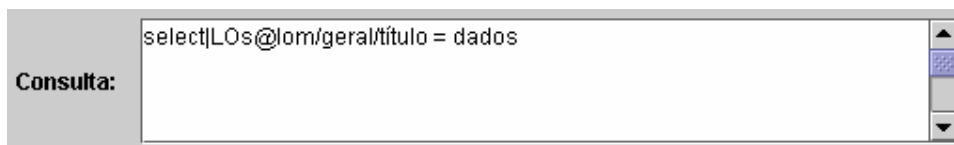
respectivamente. Todavia, a FIG. 4.19 apresenta parte deste vocabulário, tomando como exemplo o assunto *informática*.

```
<palavra_chave>
  <nome>informatica</nome>
  <nome>BD</nome>
  <nome>BDs</nome>
  <nome>banco de dados</nome>
  <nome>banco</nome>
  <nome>dados</nome>
  <nome>rede</nome>
  <nome>rede de computadores</nome>
  <nome>software</nome>
  <nome>programa</nome>
  <nome>linguagem</nome>
  <nome>bio</nome>
  <nome>bioinformatica</nome>
  <nome>data</nome>
  <nome>database</nome>
  <nome>network</nome>
  <nome>gerenciamento de banco de dados</nome>
  <assunto>
    <nomeAssunto>Informatica</nomeAssunto>
  </assunto>
</palavra_chave>
```

FIG. 4.19 – Vocabulário controlado de palavras-chaves - Domínio informática

Para cada elemento “palavra_chave” existem vários termos relacionados a um mesmo assunto, como por exemplo, o assunto informática que está relacionado a banco de dados, rede, programa, software, etc. O elemento “assunto” identifica o assunto que se relaciona com cada termo expresso pelo elemento “nome”.

O uso do vocabulário controlado de palavras chaves pode ser visto através do seguinte exemplo baseado na FIG. 4.19 e na sintaxe do MEC ROSA. Suponha a seguinte consulta: “Selecione os LOs que possuam título igual a *dados*”.



Antes de enviá-la aos demais *peers* do sistema, o processamento de consultas deve verificar a qual assunto ela está relacionada. Para isso, compara o valor do título, no caso

“dados”, a todos os valores de palavras chaves existentes no vocabulário de palavras-chaves. Ao encontrar o valor, o processamento de consultas verifica a qual assunto ele se refere. Este procedimento se repete até o final do documento XML. Pode-se verificar neste ponto, que a palavra “dados” pode se relacionar a dois ou mais assuntos distintos, permitindo assim que a consulta seja submetida a todos os *peers* que realmente possam respondê-la.

Pode-se afirmar que o vocabulário controlado de palavras chaves é imprescindível para a otimização do processamento de consultas do sistema, uma vez que permite que estas sejam submetidas somente aos *peers* que possam realmente respondê-las, fazendo com que o desempenho do sistema não seja afetado com processamento e tráfego desnecessários, o que poderia comprometer o tempo de resposta ao usuário.

Detalhes sobre o uso do vocabulário controlado de palavras chaves, assim como exemplos mais expressivos, serão discutidos na seção 4.3.2.3.1.

4.3.2.2 SEVC – SERVIÇO DE ENTREGA DE VOCABULÁRIOS CONTROLADOS

Um dos problemas encontrados em relação aos vocabulários controlados, principalmente ao vocabulário controlado local, foi identificar o melhor local para sua instalação inicial. Como visto na seção 4.3.2.1.3, existem tanto vocabulários controlados locais quanto forem a quantidade de assuntos tratados no sistema. Imagine, por exemplo, um *peer* que trate do assunto informática. Deveria este *peer* possuir somente o vocabulário controlado local de informática ou todos os demais vocabulários? Como este *peer* se comportaria ao receber uma consulta que trata de um outro assunto que não o seu?

A resposta à primeira questão é óbvia quando fundamentada em duas premissas vistas na seção 4.3.2.1.3, que afirmam que *peers* devem se basear no vocabulário controlado local correspondente ao seu assunto e, somente os *peers* que tratam deste mesmo assunto é que devem possuí-lo. Os demais vocabulários controlados locais, além de utilizar espaço físico de armazenamento no *peer*, seriam inúteis, uma vez que o *peer* não os utilizaria.

A resposta à segunda questão será discutida em detalhes na seção 4.3.2.3.4. No entanto, pode-se esclarecer a priori que um *peer*, ao identificar que uma consulta trata de um outro assunto, não a reescreverá, até mesmo porque não possui o respectivo vocabulário para suportar este processo. Ele somente irá submetê-la a um *super-peer* do agrupamento do

respectivo assunto, que se encarregará de reescrevê-la e submetê-la para si e, obviamente, retransmitir aos seus *peers* e demais *super-peers* daquele agrupamento.

O serviço de diretório, como especificado na seção 4.2.1, é um serviço residente na máquina do portal ROSA, externo ao ambiente P2P, que permite que novos *peers* que desejem se conectar ao sistema possam solicitar a relação de *super-peers* existentes. Desta forma, estes receberão uma estrutura de dados com estas informações, possibilitando sua conexão.

Assim, de forma a viabilizar uma solução, foi necessário criar mais um serviço, denominado de serviço de entrega de vocabulários controlados (SEVC), atuando na máquina do portal ROSA. Esse serviço consiste em persistir todos os vocabulários controlados (global, local e de palavras chaves) nesta máquina, externa ao ambiente P2P, fazendo com que um *peer*, ao se conectar ao sistema pela primeira vez, receba-os via rede. Vale ressaltar que, em caso de indisponibilidade da máquina onde se localiza o SEVC, o que provocará sua inoperância, os novos *peer* devem se dirigir a máquina de contingência (seção 4.2.3) para utilizarem o serviço.

Esta estratégia foi criada principalmente para resolver o problema do vocabulário controlado local. Como cada *peer* só deveria possuir um vocabulário controlado local referente ao seu domínio de conhecimento, como decidir qual dos vocabulários controlados locais deveria persistir com o *peer* no momento da instalação do sistema, uma vez que o sistema deve ser genérico e cada usuário só informa o seu domínio de conhecimento quando da utilização do sistema pela primeira vez? Para melhor compreensão, imagine um usuário que acabou de instalar o sistema. Neste momento, o sistema ainda não sabe de que assunto este *peer* irá tratar e manter todos os vocabulários controlados locais no sistema seria, no mínimo, desvantajoso, uma vez que existem tantos vocabulários controlados locais quantos forem o número de assuntos tratados pelo sistema. Assim, quando o usuário informar o assunto do respectivo *peer*, o sistema irá contactar o SEVC e solicitar os vocabulários controlados global e local referente ao seu domínio de conhecimento, bem como o vocabulário controlado de palavras chaves. Desta forma, somente os vocabulários controlados relevantes aquele *peer* serão transferidos e armazenados, otimizando objetivamente e estrategicamente seu espaço em disco.

Neste ponto, pode-se verificar que os vocabulários controlados global e de palavras chaves já poderiam estar persistidos no *peer*, uma vez que são únicos para todo o sistema. Porém, optou-se por resgatá-los junto com o vocabulário controlado local através do sistema

de entrega de vocabulários controlados, já que estes também podem sofrer atualizações ao longo do tempo, como por exemplo, a inclusão de outros idiomas ao seu vocabulário, acarretando num problema ainda maior: o de inconsistência de vocabulários. Seria necessário propagar tais atualizações aos demais *peers* do sistema, que nem sempre estariam disponíveis, prejudicando o sistema como um todo. Assim, optou-se por instalar em cada *peer* recém criado os vocabulários da sua fonte mais atual, no caso, os residentes junto ao portal ROSA, ao invés de esperar uma advertência do seu *super-peer* ou do sistema de entrega de vocabulários controlados para a atualização do(s) mesmo(s).

A FIG. 4.20, exibe a estrutura parcial de identificação e localização dos vocabulários controlados tratados no sistema, implementada em XML. Seu objetivo consiste em estruturá-los de acordo com seu tipo, assunto, nome e localização de modo a facilitar, sobretudo, na manutenção do sistema assim como auxiliar o SEVC na identificação e localização destes vocabulários de um modo fácil e eficaz. Sua utilização é mais expressiva em se tratando dos vocabulários controlados locais, uma vez que estes são numerosos e poderiam, por exemplo, possuir nomes totalmente diferentes e estarem armazenados em lugares completamente distintos. Porém, seu uso se justifica no momento em que se desejar alterar alguns dos dados como nome do arquivo e sua localização, bastando apenas a sua alteração no documento XML e não no código do programa, processo inclusive custoso, sendo considerado como um erro de análise e programação na construção de programas [Pressman, 2002].

```

<Vocabularios>
  <vocabulario>
    <tipo>global</tipo>
    <nomeArquivo>VocGlobal</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/global</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>local</tipo>
    <assunto>Medicina</assunto>
    <nomeArquivo>VocMedicina</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/local</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>palavra_chaves</tipo>
    <nomeArquivo>VocPC</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/PC</localizacao>
  </vocabulario>
</Vocabularios>

```

FIG. 4.20 – Estrutura parcial de identificação e localização dos vocabulários controlados

A estrutura em questão, assim como seu DTD, podem ser verificados nos apêndices 11 e 10 respectivamente. Contudo, faz-se necessário exemplificar sua forma de uso e, conseqüentemente, como o sistema realiza a entrega dos vocabulários. Suponha que um novo *peer*, tratando do assunto medicina, acabe de se integrar ao sistema. Ele se conectará ao servidor do portal ROSA e solicitará ao SEVC os vocabulários controlados necessários (global, local e de palavras chaves), informando para tal, seu domínio de conhecimento. Para o caso do vocabulário controlado local, o SEVC primeiramente comparará para cada vocabulário controlado do tipo = “local”, se seu assunto corresponde ao assunto do novo *peer*, isto é, assunto = “medicina”. Uma vez localizado, este fato é consumado, pois os assuntos tratados já estão pré-definidos no sistema, o SEVC identificará então o nome e a localização do vocabulário controlado local, recuperando o arquivo e enviando-o ao *peer* solicitante, o qual, a partir deste momento, já estará apto a enviar suas consultas e receber seus respectivos resultados. O mesmo procedimento ocorrerá para resgate e envio dos demais vocabulários controlados (global e de palavras chaves).

4.3.2.3 PROCESSAMENTO DE CONSULTAS

Segundo [Arenas et al., 2003], o processamento de consultas é o serviço mais importante em uma rede P2P. Este consiste basicamente da distribuição das consultas entre os *peers*. Porém, de acordo com as características de cada sistema P2P, diversas estratégias podem ser implementadas para otimizar este processamento e, desta forma, proporcionar um maior desempenho do sistema num menor tempo de resposta aos usuários.

Por usufruir de um rico conteúdo semântico, o sistema proposto utilizará uma estratégia de processamento de consultas que possa suportá-lo. De acordo com os trabalhos existentes analisados nesta área (ver seção 2.5), percebe-se que dois pontos são essenciais para o seu funcionamento: reenvio e reescrita de consultas. Estes, acrescidos da MEC ROSA e de uma estratégia bem estabelecida, especificam em detalhes o funcionamento do processamento de consultas, conforme descritos nas seções a seguir.

4.3.2.3.1 REENVIO DE CONSULTAS

O reenvio de consultas é uma das principais tarefas do processamento de consultas. Esta etapa consiste em retransmitir uma consulta submetida por um *peer* aos demais do sistema, possibilitando desta maneira, que a mesma seja processada, assim como respondida por todos.

Porém, enviar cada consulta a todos os *peers* do sistema não seria eficiente, uma vez que o processamento de consultas gastaria poder de processamento desnecessário ao enviá-la aos *peers* que não são aptos a respondê-la. Desta forma, a maneira encontrada para embutir inteligência ao processamento de consultas, a fim de otimizar seu funcionamento e beneficiar o sistema como um todo, foi primeiramente localizar os *super-peers* relevantes aquele domínio. Assim, cada consulta somente será enviada aos *super-peers* que realmente puderem respondê-la, que se encarregarão de reenviá-la aos seus respectivos *peers*, já que todos tratam do mesmo domínio.

Contudo, saber de que assunto a consulta trata é uma tarefa difícil de ser realizada, pois os atributos da consulta, além da grande diversidade de nomes, podem ser expressos através de diversas formas, além de possuírem significados diferentes capazes de provocar inconsistências, a exemplo das sinonímias e homonímias, impossibilitando desta forma, seu entendimento semântico (ver seção 2.4.2.1). Para auxiliar neste processo, o processamento de consultas faz uso do vocabulário controlado de palavras chaves que visa agregar valor semântico a metadados específicos, permitindo sua correta interpretação e possibilitando a identificação da consulta quanto a um domínio de conhecimento específico.

Observando-se o padrão de metadados utilizado pelo sistema ROSA e os tipos de consultas que podem ser realizadas, verificou-se que “título” e “palavra-chave” seriam os metadados mais apropriados para auxiliar no processo da descoberta do domínio de assunto das consultas. “Título” especifica o nome de um curso, disciplina ou tópico relevantes na elaboração da maioria das consultas e “palavras-chaves” sugerem um valor de seleção de dados, atuando como um filtro. Assim, quando da submissão de uma consulta ao sistema, o processamento de consultas deve verificar se “título” e/ou “palavras-chaves” fazem parte da consulta. Caso façam, esses valores devem ser comparados aos termos existentes no vocabulário controlado de palavras chaves. Uma vez encontrados, os respectivos assuntos são retornados e a consulta reenviada aos *peers* que tratam daquele domínio de assunto. Caso esses termos não façam parte da consulta ou os valores de pesquisa não sejam encontrados,

impossibilitando a definição do assunto da consulta, uma solicitação é enviada ao usuário solicitando-lhe uma palavra que forneça algum valor semântico à sua consulta. Exemplos destas palavras são exibidos de modo a persuadi-lo a informar a palavra correta, possibilitando assim, que o domínio seja identificado e a consulta possa ser reenviada aos *super-peers* corretos, conforme exhibe a FIG. 4.21.

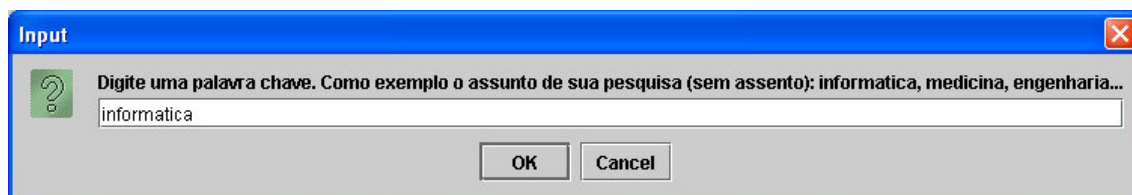
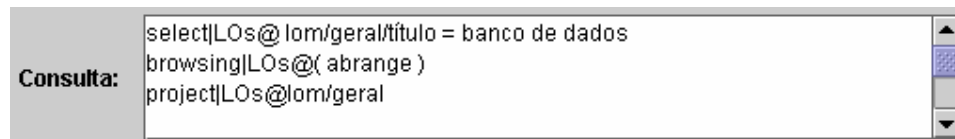


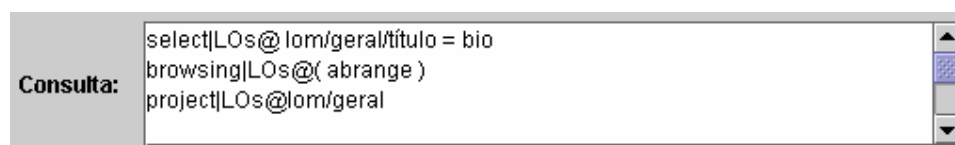
FIG. 4.21 – Tela de solicitação de palavra chave

Para um melhor entendimento, a FIG. 4.22 apresenta exemplos distintos de consultas utilizando a interface de consultas do sistema ROSA - P2P.

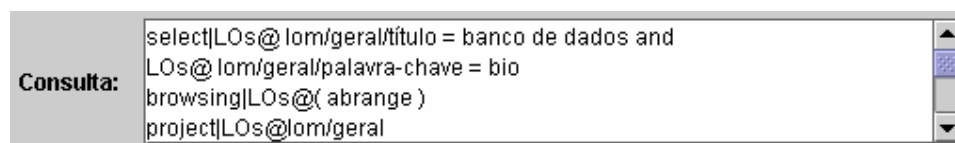
Consulta 1) Projete os LOs que possuem título igual a “banco de dados” e que abrangem outros LOs



Consulta 2) Projete os LOs que possuem palavra-chave igual a “bio” e que abrangem outros LOs



Consulta 3) Projete os LOs que possuem título igual a “banco de dados” e palavra-chave igual a “bio” e que abrangem outros LOs



Consulta 4) Projete os LOs que possuem nível de agregação = “disciplina” e que abrangem outros LOs

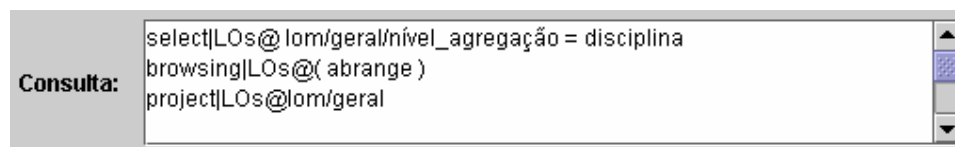


FIG. 4.22 – Exemplos de consultas

Analisando a submissão da consulta 1, o processamento de consultas identifica o metadado título e compara seu valor aos existentes no vocabulário controlado de palavras chaves (ver apêndice 9), onde o assunto “informática” é identificado. Este mesmo procedimento ocorre para a consulta 2, considerando metadado palavra-chave ao invés de título. Os assuntos “informática” e “medicina” são identificados, pois “bio” se refere tanto a informática quanto a medicina (ver apêndice 9). Na consulta 3, ambos os metadados são considerados e, conseqüentemente, informática e medicina são retornados como assuntos. No caso da consulta 4, como o metadado título nem palavra-chave foram encontrados, o usuário é solicitado a informar uma palavra que embute valor semântico a consulta, como exemplo bioinformática, identificando informática e medicina como assuntos. Em ambos os casos a consulta é reenviada aos *peers* relevantes aos respectivos assuntos.

A estratégia definida e utilizada pelo processamento de consultas é muito simples diante da complexidade do problema em questão. Este pode ser considerado um ponto positivo, visto que facilita seu entendimento e implementação. Seu uso é fundamental, pois otimiza o processamento de consultas fazendo com que tempo e processamento não sejam utilizados desnecessariamente, aumentando o desempenho do sistema e diminuindo o tempo de resposta ao usuário.

4.3.2.3.2 MEC ROSA - Máquina de Execução de Consultas do ROSA

MEC ROSA é a máquina de execução de consultas criada para o sistema ROSA, e que também servirá de subsídio para a execução de consultas no ambiente ROSA - P2P. Esta máquina é fruto do trabalho desenvolvido no mestrado de Fábio Coutinho, cujo conteúdo

mais detalhado pode ser encontrado em sua dissertação [Coutinho, 2004]. Contudo, serão apresentadas as suas principais características e particularidades, assim como os principais operadores e operações definidas na álgebra ROSA, fornecendo desta forma, uma visão geral da máquina e do seu funcionamento, possibilitando seu entendimento e caracterizando o seu uso no sistema proposto.

Ela foi instanciada a partir do framework QEEF [Ayres, 2003], que visa a construção de máquinas de execução de consultas com características gerais que permitam sua fácil adaptação e extensão a novas funcionalidades. Possui uma álgebra específica, denominada de álgebra ROSA, que é composta por um conjunto de operadores criados para manipular os dados do modelo ROSA. Estes operadores são definidos como: projeção, seleção, navegação, junção, operações de conjunto, fecho transitivo e seleção de predicados.

Seu objetivo consiste em processar consultas resolvendo as operações desta álgebra, possibilitando que o sistema ROSA possa ser consultado declarativamente, ao mesmo tempo em que o torna independente quanto a outras linguagens, a exemplo do XQuery [XQuery]. A MEC ROSA provê também um nível avançado de especificação de consultas, o que sustenta um rico processamento de consultas.

Entretanto, como o sistema ROSA foi inicialmente construído para ser um sistema local, o uso da MEC ROSA ficou restrito somente a uma base de dados, impossibilitando assim, a geração de um plano de consulta distribuído, o que seria ideal para o ambiente distribuído do ROSA - P2P. Porém, foi definida uma estratégia para utilização da MEC ROSA em ambiente distribuído, permitindo sua utilização no sistema proposto. Sua apresentação será exibida na seção 4.3.2.3.4.

Os operadores da MEC ROSA, bem como suas especificações, podem ser encontrados em detalhes em [Coutinho, 2004]. Todavia, faz-se necessário a descrição de alguns desses operadores, tais como o de seleção, projeção e navegação, assim como das operações relacionadas, tais como scan, projeção, seleção e navegação, uma vez que foram os únicos operadores e operações a serem implementadas e fundamentais para a maioria das propostas de consultas do ROSA - P2P, conforme descritas a seguir:

- **Scan**

A operação de scan é fundamental para a mais simples das consultas. Ela provê o acesso às fontes de dados, o que faz com que se torne um pré-requisito na escrita de qualquer consulta.

Sua execução é simples e consiste unicamente em permitir o acesso aos dados pelas demais operações. Nos exemplos citados neste trabalho, com o intuito de melhor expressar seu conteúdo, optou-se não apresentá-la. Todavia, ela se encontra implícita nos mesmos.

A seguir é apresentado um exemplo do operador scan, onde “LOs” é o nome da base acessada:



A screenshot of a query input field. On the left, there is a grey vertical bar with the word "Consulta:" in white text. To the right of this bar is a white rectangular input box with a thin grey border. Inside the input box, the text "scan|LOs" is entered in a monospaced font.

- **Projeção**

A forma geral do operador de projeção, simbolizado por π , é:

$$CR = \pi(l) (CE)$$

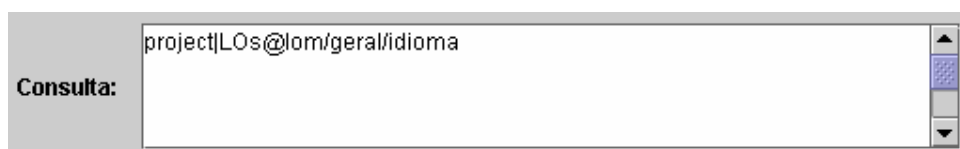
onde, CE e CR são a coleção de entrada e a coleção resultante, respectivamente. l contém uma lista dos únicos metadados que estarão presentes, isto é, terão seus valores recuperados nos LOs da coleção CR.

O operador π recupera todas as coleções internas de CE com seus respectivos LOs, sendo que seus metadados são limitados aos metadados discriminados em l , ou seja, apenas os valores dos metadados listados em l constarão dos LOs recuperados a partir das coleções internas de CE. Esses LOs, juntamente com as coleções internas a que pertencem, irão compor a coleção CR. A ausência de l determina que os LOs sejam retornados com seu conteúdo original, logo, $CR \equiv CE$.

A operação de projeção, que utiliza o operador de projeção, funciona como uma espécie de reconstrução do LO a ser projetado. O processo consiste em percorrer a árvore de nós do LO, construindo uma nova árvore cujos nós representem apenas os metadados presentes na lista de projeção. Assim, esta nova árvore passa a substituir aquela que está presente no LO.

Alguns exemplos de consultas que fazem uso do operador de projeção podem ser vistas a seguir:

- 1) Liste todos os idiomas dos LOs



A screenshot of a query input field. On the left, there is a grey vertical bar with the word "Consulta:" in white text. To the right of this bar is a white rectangular input box with a thin grey border. Inside the input box, the text "project|LOs@lom/geral/idioma" is entered in a monospaced font. To the right of the input box, there are three small vertical buttons: a grey one at the top, a blue one in the middle, and a grey one at the bottom.

2) Projete o título e a descrição de cada LO

Consulta: project|LOs@lom/geral/título, LOs@lom/geral/descrição

• **Seleção**

A forma geral do operador de seleção, simbolizado por σ , é:

$$CR = \sigma(v) (CE)$$

onde, CE e CR representam a coleção de entrada e a coleção resultante, respectivamente. v simboliza a condição de seleção sobre os valores dos metadados do LO e é formada pelos operadores da lógica booleana (and, or, not) e operadores de comparação (<, >, ≤, ≥, =, ≠, in).

O operador σ seleciona as coleções internas da coleção CE que satisfazem à condição v . Uma coleção interna é considerada válida quando todos os seus LOs são válidos (satisfazem a v). Toda coleção interna de CE considerada válida fará parte da coleção resultante (CR), caso contrário, a coleção interna não é selecionada e nenhum de seus LOs estarão presentes em CR.

A operação de seleção, que utiliza o operador de seleção, prevê que seus termos componentes encontrem-se na *forma normal conjuntiva* [Valduriez e Özsu, 2001], a qual corresponde a uma conjunção de disjunções, tal como:

$$(T_{11} \vee T_{12} \vee \dots \vee T_{1n}) \wedge \dots \wedge (T_{m1} \vee T_{m2} \vee \dots \vee T_{mn})$$

onde T_{ij} é um termo simples.

Alguns exemplos de consultas que utilizam o operador de seleção podem ser vistas a seguir:

1) Selecione os LOs registrados como disciplina?

Consulta: select|LOs@lom/geral/nível_agregação = disciplina

2) Obtenha os títulos dos LOs registrados como tópico cuja dificuldade é alta

Consulta: select|LOs@lom/geral/nível_agregação = topico and LOs@lom/aspectos_pedagógicos/dificuldade = alta project|LOs@lom/geral/título

- **Navegação**

A forma geral do operador de navegação, simbolizado por λ , é:

$$CR = \lambda i (p) (CE)$$

onde, CE e CR representam a coleção de entrada e a coleção resultante, respectivamente. p contém os predicados que irão compor os relacionamentos dos quais os LOs de CE e CR participam, além de definir as restrições impostas a estes relacionamentos. i indica o papel desempenhado pelos LOs da coleção CR nos relacionamentos, representando o papel de *sujeito* quando o valor de i for igual a s e de *objeto* quando i for igual a o .

O operador λ navega por meio de predicados a partir dos LOs das coleções internas de CE, e para cada um dos LOs segue o(s) caminho(s) estabelecido(s) em p rumo aos LOs resultantes. Os LOs encontrados em trechos intermediários do caminho são desprezados, e para cada LO encontrado pelo trecho final é gerada uma coleção interna da qual o mesmo fará parte exclusivamente. A coleção CR reúne todas essas coleções internas geradas. Os LOs recuperados desempenham o papel de *sujeito* ou *objeto* dos relacionamentos, dependendo do valor estabelecido para i .

A ausência do parâmetro i determina que os LOs da coleção CR desempenharão o papel de *objeto* da associação. A ausência de p determina que nenhum relacionamento deve ser percorrido, logo, $CR = CE$.

A seguir são apresentados alguns exemplos de consultas que utilizam o operador de navegação:

- 1) Quais são os LOs abrangidos pela disciplina banco de dados?

```
Consulta: select|LOs@ lom/geral/titulo = banco de dados and
LOs@ lom/geral/nível_agregação = disciplina
browsing|LOs@( abrange )
```

- 2) Recupere os títulos dos LOs compreendidos ou fundamentados pela disciplina gerenciamento de banco de dados.

```
Consulta: select|LOs@ lom/geral/titulo = gerenciamento de banco de dados and
LOs@ lom/geral/nível_agregação = disciplina
browsing|LOs@( compreende or fundamenta )
project|LOs@lom/geral/titulo
```

Devido à sua complexidade, a operação de navegação pode ser dividida em várias etapas. Algumas delas são:

- Leitura dos predicados de navegação: consiste em obter os predicados especificados na operação λ ;
- Interpretação do predicado obtido: diz respeito à capacidade de selecionar a funcionalidade adequada de acordo com o operador interno encontrado ('^', 'v', '.', '*');
- Precedência entre os operadores internos: determina a ordem na qual os predicados devem ser avaliados;
- Semântica da busca navegacional: implementa a navegação propriamente dita a partir dos predicados, obtendo os objetos da navegação.

Todas as etapas descritas acima necessitam uma atenção especial no desenvolvimento da implementação do operador λ . Todavia, a *precedência entre os operadores internos* e a *semântica da busca navegacional* são os pontos altos da operação, razão pela qual demandam maior explicação, cujos detalhes podem ser encontrados em [Coutinho, 2004].

A MEC ROSA faz parte do módulo de processamento de consultas do sistema proposto. Ela é responsável por encontrar os resultados em cada base de dados e retorná-los ao processamento de consultas, que se encarregará de enviá-los ao *peer* solicitante.

4.3.2.3.3 REESCRITA DE CONSULTAS

A reescrita de consultas representa uma das fases mais importantes do processamento de consultas. Ela consiste em reescrever ou recompor consultas através da reescrita das operações de seleção e/ou de navegação, de acordo com as informações mantidas nos vocabulários controlados local e global, fazendo com que as mesmas possam ser capazes de englobar um universo de dados mais extenso, possibilitando assim, que todas as respostas

possíveis à consulta sejam encontradas, independentemente de como os dados foram semanticamente armazenados.

Desta forma, quando uma consulta é reenviada a um *peer*, o processamento de consultas deste *peer* verificará a possibilidade de reescrevê-la em uma consulta semanticamente mais rica. Para isso, uma análise detalhada das operações que compõem a consulta deve ser realizada. Esta análise permite identificar quais operações, seleção e/ou navegação serão reescritas, ao mesmo tempo em que especifica quais vocabulários controlados global e/ou local serão utilizados.

Assim, o processamento de consultas ao verificar que a consulta possui a operação de seleção e que esta possui o metadado “título”, comparará o metadado “título” da operação de seleção em questão com os demais termos sinônimos persistidos no vocabulário controlado local. Uma vez encontrado, os termos equivalentes são recuperados e a operação de seleção é reescrita com a adição destes novos termos através do operador de disjunção (or). Caso existam outros termos de metadados na operação de seleção, a exemplo do metadado “nível de agregação”, estas devem ser adicionadas, acrescida do operador “or” ou “and” entre elas e localizadas ao final da sentença, obedecendo assim, a *forma normal conjuntiva* utilizada pela operação de seleção da MEC ROSA. O operador utilizado, “or” ou “and”, varia de acordo com a operação de seleção inicialmente definida pelo usuário. Caso o metadado “título” não seja encontrado, a consulta simplesmente não é reescrita.

O processo de reescrita de consultas continua, porém agora para a operação de navegação. Uma vez verificada sua existência, o processamento de consultas compara cada um dos predicados declarados na consulta com os predicados sinônimos persistidos no vocabulário controlado global. Diferentemente do processo de reescrita da operação de seleção, o(s) predicado(s) declarado(s) certamente existirá(ão) no vocabulário controlado global. Isto é possível, pois estes predicados se encontram disponíveis através da interface de geração de consultas do sistema, a qual restringe o universo de predicados possíveis, bastando apenas a sua seleção. Contudo, para cada predicado encontrado no vocabulário controlado global, os respectivos predicados equivalentes são recuperados e reescritos, formando assim, um conjunto de predicados reescritos, os quais são concatenados através do operador de disjunção (or). Este operador, juntamente com o operador de conjunção “and” e o de navegação “.”, são utilizados para fazer a concatenação entre todos estes conjuntos de predicados reescritos. A decisão de qual operador utilizar varia de acordo com a operação de navegação inicialmente definida pelo usuário. Parênteses também são utilizados para auxiliar

a MEC ROSA na identificação de cada um destes conjuntos. Por fim, a reescrita da operação de navegação estará completa quando todos os conjuntos de predicados reescritos forem unidos na mesma sentença.

Ao final do processo de reescrita das operações de seleção e de navegação, o processamento de consultas já pode compor a nova consulta, isto é, reescrevê-la. Uma vez reescrita, a consulta abrangerá todos os metadados e predicados semanticamente relevantes, possibilitando desta forma, que os resultados corretos sejam encontrados e, conseqüentemente, retornados ao *peer* solicitador.

A reescrita da consulta acontece no momento em que esta análise ocorre, isto é, quando da identificação das operações que compõem a consulta. Assim, o processamento de consultas irá reescrevê-la no momento em que a mesma corresponde a uma das situações:

I. Operação de navegação, sem a operação de seleção

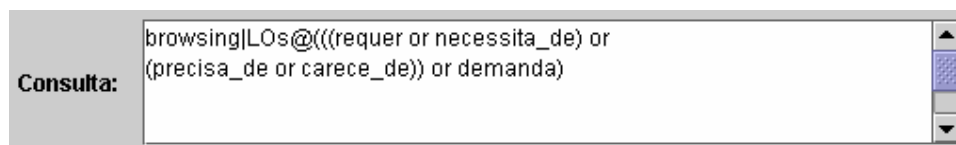
Neste caso, somente a operação de navegação será reescrita. Conseqüentemente, o vocabulário controlado global será utilizado, uma vez que possui, dentre outras coisas, os termos sinônimos aos predicados existentes no sistema, os quais são condizentes e relevantes à operação de navegação. Um pequeno exemplo deste caso pode ser visto a seguir. Suponha a consulta 1, definida como:

- 1) Obtenha os LOs que *requerem* outros LOs.



Consulta: browsing|LOs@(requer)

Esta consulta será reescrita pelo módulo de processamento de consultas em uma nova consulta, acrescida dos predicados equivalentes a *requer*, definida como:



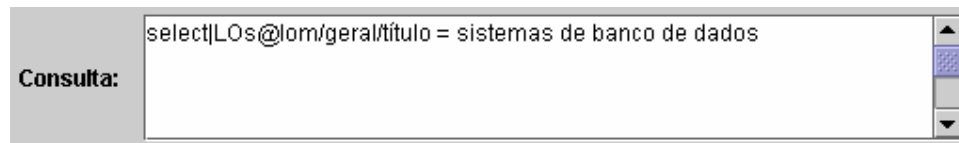
Consulta: browsing|LOs@(((requer or necessita_de) or (precisa_de or carece_de)) or demanda)

II. Operação de seleção, sem a operação de navegação

Neste caso uma análise mais específica deve ser realizada a fim de verificar se a operação de seleção possui em sua declaração o metadado “título”. Caso este não seja

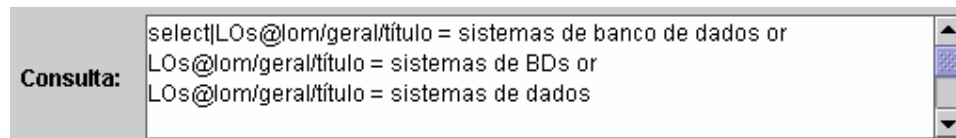
identificado, a operação de seleção não deve ser reescrita. Porém, uma vez encontrado, o vocabulário controlado local, que trata dos termos sinônimos correspondentes aos nomes dos LOs através do metadado “título”, deve ser consultado, e a operação de seleção reescrita. Esta restrição é necessária, pois permite a otimização e maior eficácia do processamento de consultas, uma vez que permite somente reescrever as operações de seleção que são semanticamente possíveis de serem reescritas de acordo com o suporte fornecido pelo vocabulário controlado local. Como visto na seção 4.3.2.1.3, no vocabulário controlado local somente são armazenadas informações condizentes com os nomes dos LOs (título). Outros tipos de metadados, como por exemplo “idiomas” e “nível de agregação”, quando presentes na consulta, não são utilizados na operação de reescrita da consulta, uma vez que estes metadados não fazem parte do vocabulário controlado global. Porém, o enriquecimento do sistema com outros vocabulários é passível de ser realizado e deve ser encarado como um trabalho futuro, o que possibilitaria um maior enriquecimento semântico do processamento de consultas. Todavia, o objetivo se concentrou em validar sua utilização quanto ao metadado mais requisitado na operação de seleção, de forma a agregar maior conteúdo semântico, no caso o metadado “título”. Um pequeno exemplo deste caso pode ser visto a seguir. Suponha a consulta 1, definida como:

- 2) Selecione os LOs cujo título seja *sistemas de banco de dados*.



The screenshot shows a window titled "Consulta:" with a text area containing the following SQL query: `select|LOs@lom/geral/titulo = sistemas de banco de dados`. The text area has a vertical scrollbar on the right side.

Esta consulta será reescrita pelo módulo de processamento de consultas em uma nova consulta, acrescida dos termos equivalentes a *sistemas de banco de dados*, definida como:



The screenshot shows a window titled "Consulta:" with a text area containing the following disjunctive SQL query: `select|LOs@lom/geral/titulo = sistemas de banco de dados or LOs@lom/geral/titulo = sistemas de BDs or LOs@lom/geral/titulo = sistemas de dados`. The text area has a vertical scrollbar on the right side.

III. Operação de navegação e operação de seleção onde o metadado “título” é especificado

Neste caso, ambas as operações serão reescritas. Conseqüentemente, ambos os vocabulários controlados global e local serão utilizados. Um pequeno exemplo deste caso pode ser visto a seguir. Suponha a consulta 1, definida como:

- 3) Selecione os LOs cujo título seja *sistemas de banco de dados* e que *requerem* outros LOs.

```
Consulta: select|LOs@lom/geral/titulo = sistemas de banco de dados  
browsing|LOs@(requer)
```

Esta consulta será reescrita pelo módulo de processamento de consultas em uma nova consulta, acrescida dos predicados equivalentes a *requer* e dos termos equivalentes a *sistemas de banco de dados*, definida como:

```
Consulta: select|LOs@lom/geral/titulo = sistemas de banco de dados or  
LOs@lom/geral/titulo = sistemas de BDs or  
LOs@lom/geral/titulo = sistemas de dados  
browsing|LOs@(((requer or necessita_de) or  
(precisa_de or carece_de)) or demanda)
```

IV. Ausência da operação de navegação e da operação de seleção

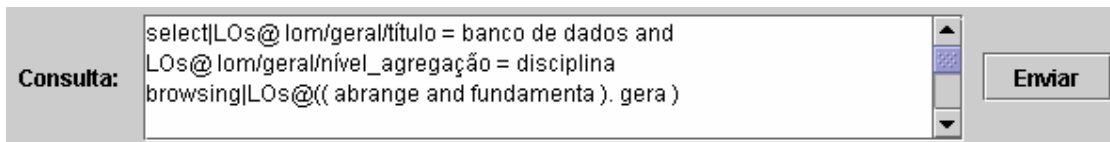
Neste caso a consulta não será reescrita. Conseqüentemente, nenhum dos vocabulários controlados global e local será utilizado. Um pequeno exemplo deste caso pode ser visto a seguir. Suponha a consulta 1, definida como:

- 4) Projete o título e a descrição das anotações de cada LO.

```
Consulta: project|LOs@lom/geral/titulo, LOs@lom/anotações/descrição
```

Contudo, vale a análise de um exemplo complexo. Suponha a consulta 1 submetida a um *peer* que trata do assunto informática.

- 1) Obtenha os títulos dos LOs que são gerados por aqueles que abrangem e fundamentam outros LOs e que possuam “título” igual a *banco de dados* e “nível de agregação” igual a *disciplina*



Nesta consulta, a operação de seleção visa selecionar os LOs que detém o “título” igual a *banco de dados*, ao mesmo tempo em que possuam “nível de agregação” igual a *disciplina*. A operação de navegação por sua vez apresenta 3 predicados: *abrange*, *fundamenta* e *gera*, os quais visam recuperar os LOs que são abrangidos e fundamentados e, descendo um nível na hierarquia, isto é, navegando pelo mapa conceitual, gerados.

Como visto anteriormente, o processamento de consultas primeiramente verificará se a operação de seleção faz parte da consulta. Como isto é um fato, o valor do “título”, *banco de dados*, será comparado aos termos existentes no vocabulário controlado local que trata do assunto informática (ver apêndice 7). Ao encontrá-lo, o processamento de consultas recupera os termos equivalentes para *banco de dados* que estão definidos como *gerenciamento de banco de dados* e *BD*. Estes termos são adicionados ao termo já existente através do operador de disjunção “or”, formando uma sentença parcial. A parte final da operação de seleção inicial, que visa selecionar os LOs de acordo com o “nível de agregação”, é acrescida pelo operador “and” e adicionada ao final da sentença parcial, formado assim uma sentença final e completa, que corresponde à reescrita da operação de seleção, conforme mostra a FIG. 4.23.

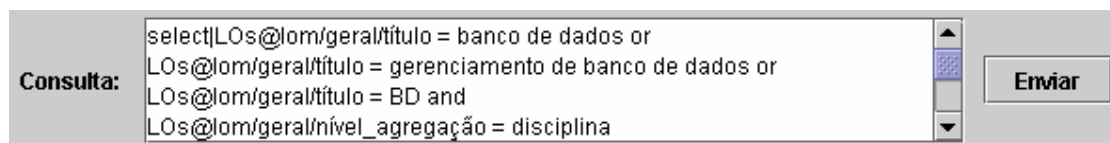


FIG. 4.23 – Reescrita da operação de seleção

O processamento prossegue para tratar a operação de navegação. O primeiro predicado declarado como *abrange* é comparado aos predicados existentes no vocabulário controlado global (ver apêndice 5). Ao encontrá-lo, o processamento de consultas recupera os predicados equivalentes *compreende*, *é constituído de*, *é composto de*, *é formado de*, *possui*, *tem*, *inclui*, *enlaça*, *abarca* e *envolve*. Estes predicados são adicionados ao predicado inicial *abrange* através do operador de disjunção “or”, formando um conjunto de predicados reescritos. O mesmo acontece para o segundo predicado, declarado como *fundamenta*, que também é comparado aos predicados existentes no vocabulário controlado global. Os predicados equivalentes recuperados são identificados como *é pré-requisito de*, *é base para*, *é condição*

para. Estes predicados são unidos ao predicado inicial *fundamenta*, através do operador de disjunção “or”, formando assim, mais um conjunto de predicados reescritos. O mesmo procedimento ocorre para o caso do terceiro predicado, declarado como *gera*. Este também é comparado aos predicados existentes no vocabulário controlado global onde os predicados *cria*, *desenvolve* e *produz* são os predicados equivalentes encontrados. Estes são agregados ao predicado inicial *gera*, através do operador de disjunção “or”, formando mais um conjunto de predicados reescritos. Neste ponto, ao verificar que não existem mais predicados a serem analisados, o processamento de consultas faz a concatenação dos conjuntos de predicados reescritos. No caso, estes operadores de concatenação são reconhecidos como os operadores “and” e “.”, pois a operação de navegação inicial foi definida como *abrange and fundamenta.gera*. Parênteses são adicionados de modo a permitir que a diferenciação na composição dos predicados (*abrange*, *fundamenta* e *gera*) seja identificada pela MEC ROSA. A rescrita completa da operação de navegação pode ser verificada na FIG. 4.24.

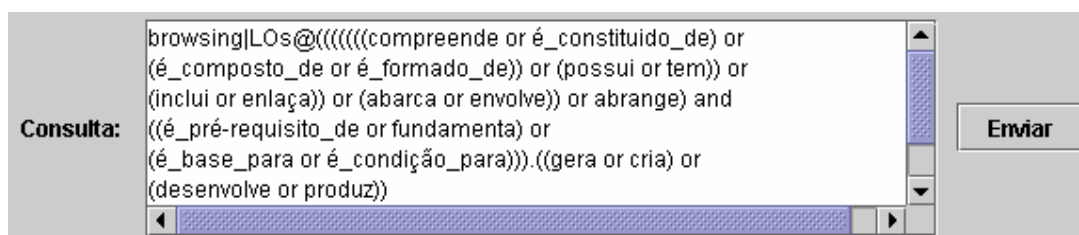


FIG. 4.24 – Reescrita da operação de navegação

Uma vez feita a reescrita das operações de seleção e navegação, o processamento de consultas irá compor a nova consulta, reescrevendo-a de acordo com as novas operações. A FIG. 4.25 exibe a consulta final reescrita.

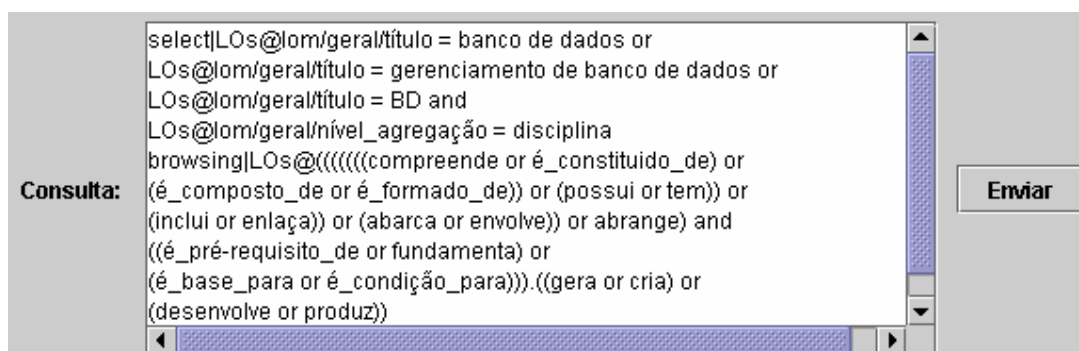


FIG. 4.25 – Consulta final reescrita

Como é possível observar, a reescrita de consultas é uma etapa fundamental, pois permite que as consultas agreguem novos valores semânticos, alcançado assim, todos os resultados existentes. Além de enriquecer o processamento de consultas, faz com que o sistema seja também reconhecido pela sua capacidade semântica.

4.3.2.3.4 ESTRATÉGIA DE PROCESSAMENTO DE CONSULTAS

Esta seção tem como objetivo apresentar a estratégia utilizada pelo processamento de consultas no sistema proposto. Uma vez definidas as etapas de reenvio e reescrita de consultas, torna-se necessário expressar como estas serão integradas e como as consultas serão processadas pela MEC ROSA.

Um dos pontos críticos analisados nesta seção diz respeito ao uso da MEC ROSA. Como discutido na seção 4.3.2.3.2, sabe-se que esta somente é capaz de submeter consultas a uma única base de dados, o que significa que um plano de consultas distribuído e otimizado, a exemplo do desenvolvido em [Nejdl et al., 2003], não pode ser gerado. Este plano forneceria, dentre outras coisas, maior poder de autonomia e controle pelo *peer* que submeteu a consulta, ao mesmo tempo em que diminuiria a necessidade de processamento inteligente de consultas pelos *peers* receptores, aumentando o desempenho geral do sistema, bem como facilitando a sua implementação.

Contudo, a solução adotada aqui consiste em fazer com que cada *peer* processe a consulta de forma semelhante ao *peer* solicitador, permitindo assim que todos tenham a mesma autonomia sobre a consulta no que diz respeito a sua reescrita e processamento, possibilitando a ausência de um plano distribuído. Esta estratégia tornou-se possível, pois em cada *peer* foi inserida uma MEC ROSA, assim como um módulo de processamento de consultas. Cada um destes módulos de processamento de consultas é um componente complexo, isto é, inteligente o bastante não só para processar e reenviar uma simples consulta recebida (o que bastaria no caso de um plano distribuído), mas também para, dentre outras coisas, identificá-la, reescrevê-la e gerenciá-la. Isto possibilita que o melhor caminho seja encontrado, levando-se em consideração todos os fatores externos, a exemplo da identificação do *peer* como *super-peer* ou não, que modificaria o comportamento e trajetória da consulta no sistema proposto.

Um estudo detalhado sobre todos os pontos pertinentes ao processamento de consultas, a exemplo da localização dos *peers* relevantes para respondê-la e sua reescrita, foi realizado de modo a prover a integração entre eles. Este resultou em uma estratégia de processamento de consulta dividida em duas fases segundo dois pontos de vista: submissão da consulta por um usuário e recebimento da consulta de um *peer* ou *super-peer*, a saber:

- **Primeira fase:** consiste em identificar, na submissão ou recebimento de uma consulta, os *peers* e/ou *super-peers* relevantes a respondê-la e que se encontram “on-line”, reenviando-a para estes. Vale lembrar que os *peers* e *super-peers* relevantes são aqueles que tratam do mesmo assunto da consulta enquanto que o metadado “on-line” significa que o *peer* ou o *super-peer* está ativo na rede. Desta forma, do ponto de vista da submissão de uma consulta por um usuário, o módulo de processamento de consultas analisará se o *peer* é um *super-peer* ou não. Caso seja, reenviará a consulta aos seus próprios *peers* e localizará dentre os *super-peers* para os quais aponta, quais são os relevantes para respondê-la, reenviando-lhes a consulta em seguida. Caso contrário, isto é, quando o *peer* não for um *super-peer*, o processamento de consultas simplesmente reenviará a consulta diretamente para o respectivo *super-peer*, que se encarregará de reenviá-la no sistema. Já do ponto de vista do recebimento da consulta de um *peer* ou *super-peer*, o módulo de processamento de consultas também analisará se o *peer* é um *super-peer* ou não. Caso seja, verificará se o *super-peer* remetente pertence ao seu agrupamento. Caso pertença, não será necessário reenviá-la ao demais *super-peers* do agrupamento, pois o *super-peer* remetente a fará. Somente deverá reenviá-la aos seus próprios *peers*. Caso não pertença, deverá fazê-lo, reenviando a consulta aos demais *super-peers* do agrupamento e aos seus próprios *peers*. Em ambos casos, um relógio é acionado no momento em que a consulta é reenviada pelo *peer* ou *super-peer* solicitador. Este serve como tempo limite para espera dos respectivos resultados parciais, conforme será visto detalhadamente na seção 4.3.2.3.4.1, referente ao tempo de espera de resultados. Todavia, todos estes procedimentos são suportados pelo vocabulário controlado de palavras chaves, o qual permite identificar em tempo de execução da consulta o assunto tratado por ela, permitindo assim, que a mesma seja reenviada aos *peers* e/ou *super-peers* correspondentes.

- **Segunda fase:** consiste em reescrever a consulta e processá-la, retornando o resultado ao *peer* ou *super-peer* solicitador. Assim, após a submissão ou recebimento de uma consulta, o *peer* ou *super-peer* a reescreverá. Este processo permite incluir um universo de dados mais extenso, possibilitando que todas as respostas possíveis à consulta possam ser encontradas. Ele é suportado pelo vocabulário controlado global e local, os quais fornecem os conteúdos sinônimos necessários a esta operação. Uma vez reescrita, a consulta é processada pela MEC ROSA, porém, do ponto de vista da submissão de uma consulta por um usuário, o resultado deste processamento é mantido no *cache* do respectivo *peer* ou *super-peer*. Já, do ponto de vista do recebimento da consulta de um *peer* ou *super-peer*, o resultado deve ser encaminhado para o *peer* ou *super-peer* solicitador, permanecendo em *cache* para futura integração. Como visto, o *peer* ou *super-peer* solicitador deve esperá-lo até o tempo limite definido pelo módulo de processamento de consultas. Após este tempo, os dados residentes em *cache* devem ser integrados, conforme será verificado na seção 4.3.2.4, referente à integração dos dados.

A estratégia definida para o processamento de consultas tem como objetivo otimizar o seu funcionamento, fazendo com que as consultas sejam respondidas somente pelos *peers* e *super-peers* relevantes, além de garantir que todos os resultados possíveis sejam alcançados. Assim, além de contribuir para a diminuição do tempo de resposta aos usuários, tem-se um aumento na qualidade dos resultados, representando um ganho no desempenho geral do sistema proposto.

4.3.2.3.4.1 TEMPO DE ESPERA DE RESULTADOS

Uma das dificuldades encontradas na definição da estratégia utilizada pelo processamento de consultas foi identificar o tempo que deveria se aguardar pelas respostas das consultas reenviadas aos *peers* e/ou *super-peers* relevantes. Este assunto é considerado crítico, pois em um ambiente distribuído baseado na arquitetura de *super-peers*, geralmente é difícil controlar para quantos *peers* ou *super-peers* a consulta está sendo reenviada, uma vez que o *peer* ou *super-peer* solicitador não tem autonomia suficiente para gerenciá-la em todo o ambiente. Porém, esse problema não ocorre no ambiente ROSA - P2P.

Como visto na seção 4.3.2.3.4, caso uma consulta trate de um assunto diferente do tratado pelo *peer* ou *super-peer* a qual esta sendo submetida, esta deve ser reenviada ao *super-peer* que trate deste assunto, que se encarregará de reenviá-la aos seus *peers* e demais *super-peers* relevantes. Este caso é típico em ambiente P2P, onde não se sabe para quais outros *peers* e *super-peers* a consulta está sendo reenviada. Contudo, caso a consulta trate do mesmo assunto tratado pelo *peer* ou *super-peer* a qual está sendo submetida, esta deve ser reenviada aos demais *peers* e/ou *super-peers* relevantes. Devido à estrutura de índices SP/SP (seção 4.2.2.8) utilizada pelo sistema proposto, que fornece dentre outras coisas, a quantidade de *super-peers* que trata de um mesmo assunto e o número de *peers* que cada um desses *super-peer* possui, é possível identificar para quantos *peers* a consulta está sendo reenviada. O uso do índice SP/SP auxilia o sistema proposto no processamento de consultas e resolve o problema de não se saber para quantos *peers* e/ou *super-peers* a consulta está sendo reenviada.

Porém, mesmo obtendo a informação correspondente ao número de *peers* e/ou *super-peers* para onde a consulta está sendo reenviada, ainda assim, não é suficiente para resolver o problema referente ao tempo que se deve aguardar pelos resultados, pois o *peer* ou *super-peer* solicitador não pode ficar eternamente aguardando por estes resultados, uma vez que, por eventualidade, um destes *peers* ou *super-peers* pode ter saído da rede. Embora o sistema trate do problema de tolerâncias a falhas, conforme discutido na seção 4.2.3, esta não é uma solução aceitável diante de um ambiente tão dinâmico como o P2P. De fato, este caso poderia se repetir inúmeras vezes, acarretando dentre outras coisas, num tratamento exaustivo, o qual diminuiria o tempo de resposta ao usuário e, conseqüentemente, do desempenho geral do sistema proposto.

Desta forma, com o intuito de solucionar este problema, as informações pertinentes contidas no índice SP/SP são utilizadas para a definição de um tempo de espera de resultados, que deve ser utilizado pelo processamento de consulta do *peer* ou *super-peer* solicitador da consulta. Uma função, denominada TER – Tempo de Espera de Resultados, foi definida para auxiliar na descoberta deste tempo. De fato, esta função equivale ao tempo máximo que o processador de consultas deverá esperar pelos resultados, calculado pela fórmula a seguir:

$$\text{Função TER} = (\sum \text{SP} + \sum \text{P}) \times T$$

Onde,

- SP é a quantidade de *super-peers* “on-line” que tratam de um mesmo assunto, para os quais a consulta será reenviada;
- P é o número de *peers* “on-line” ligados a cada um destes *super-peer*, para os quais a consulta também será reenviada;
- T é o tempo, especificado em minutos e parametrizado pelo administrador do sistema, suficiente para o recebimento de um resultado.

Devido a sua complexidade, algumas variáveis importantes não foram consideradas na definição do valor real do tempo de espera de resultados (função TER), a exemplo do estado da rede e complexidade da consulta. Contudo, estas podem ser futuramente acopladas visando abranger todos estes possíveis parâmetros, contribuindo para o aumento da completude e precisão do seu resultado.

Para um melhor entendimento, suponha um *super-peer* que trate do assunto biologia. Uma vez submetida uma consulta que trate desse assunto, e após identificar e localizar para quais *peers* e/ou *super-peers* esta deve ser reenviada, o processamento de consulta chamará a função TER a fim de definir o tempo máximo que este terá que aguardar pelos resultados. Suponha que existam 3 *super-peers*, cada qual com 10 *peers*. O resultado da função TER será: $3 + 30 \times 0.5$, onde o valor 3 corresponde ao somatório dos *super-peers* relevantes, 30 ao somatório dos *peers* que a estes se encontram conectados e que estão “on-line”, e 0.5 ao tempo especificado em minutos pelo administrador do sistema para o retorno de cada resultado, totalizando assim em 16,5 minutos de espera máxima.

Porém, existe a possibilidade dos resultados serem retornados num tempo inferior ao calculado por TER. Como se sabe a priori para quantos *peers* e/ou *super-peers* a consulta foi reenviada, o processamento de consultas inicia o relógio para a contagem do tempo de espera dos resultados, iniciando também um contador para a contagem dos resultados retornados. Desta forma, caso o número de resultados retornados corresponda ao número de *peers* e/ou *super-peers* para qual a consulta foi reenviada, o processamento de consulta deve parar o relógio referente ao tempo de espera dos resultados e zerá-lo, pois todos os resultados referentes a consulta já foram recebidos. Caso contrário, isto é, enquanto o número de resultados retornados não corresponder ao somatório de *peers* e/ou *super-peers* para qual a consulta foi reenviada, a contagem do tempo de espera de resultados continua sem interrupção. Ao término de TER, serão considerados como resultados somente os resultados até então recebidos. Em ambos os casos, após o término de espera dos resultados, dá-se início

ao processo de integração dos mesmos. Assim, dando seqüência a análise do exemplo citado, como a consulta foi reenviada para 3 *super-peers* e para 30 *peers*, o processamento de consultas deverá aguardar pelo retorno de 33 resultados. Caso o contador de resultados recebidos atinja este número, o relógio que deveria prosseguir até o tempo de 16,5 minutos é parado e zerado. Caso o contador não atinja este número, a contagem do tempo de espera de resultados prossegue até atingir o tempo máximo de 16,5 minutos, obtendo como resultados somente os resultados recebidos até este momento. Desta forma, pode-se verificar que o tempo de espera de resultados é uma estratégia utilizada para garantir que o processamento de consultas não fique inoperante indeterminadamente.

Contudo, cabe apresentar dois casos específicos. O primeiro se refere ao caso onde a consulta trata de um assunto diferente do tratado pelo *peer* ou *super-peer* a qual está sendo submetida, devendo este reenviá-la ao *super-peer* do mesmo assunto, o qual se encarregará de reenviá-la. O segundo caso é identificado quando a consulta é submetida a um *peer* e não a um *super-peer*, devendo este reenviá-la ao seu *super-peer*. Em ambos os casos, o processamento de consultas não saberá a quantidade de *super-peers* e *peers* para onde a consulta será submetida, não podendo desta forma, definir o tempo de espera de resultados. Assim, ao reenviar a consulta ao *super-peer* relevante, o processamento de consulta solicita que este *super-peer* informe a quantidade de *super-peers* e *peers* para quais ele a reenviará. Desta forma, com estas informações torna-se possível identificar o número de resultados que deverão retornar, assim como calcular o tempo que deverá aguardar pelos mesmos.

Como é possível observar, a utilização da função TER é de suma importância para o processamento de consultas. Seu uso permite que um tempo limite para a espera dos resultados seja definido, possibilitando assim que o processamento de consultas ao término deste tempo, seja capaz de ignorar os resultados ausentes e prosseguir com a integração dos dados, oferecendo assim, um tempo de resposta aceitável ao usuário, embora parcialmente completo.

4.3.2.4 INTEGRAÇÃO DOS DADOS

A integração de dados é a última etapa do sistema de integração de dados do ROSA - P2P. Esta consiste em integrar os dados resultantes de todas as consultas enviadas aos *peers*

e/ou *super-peers* relevantes, possibilitando desta forma, que uma resposta global e correta seja retornada ao usuário.

Como visto na seção anterior, cada consulta deve ser identificada, reescrita e processada pelos *peers* para onde foi submetida. Embora este processo tenha suas desvantagens (ver seção 4.3.2.3.4), este facilita a etapa da integração de dados, uma vez que os problemas existentes nesta etapa passam a ser de cada *peer* e não somente do *peer* solicitador da consulta. Com a minimização destes problemas à granulação de cada *peer*, o processo de integração de dados do sistema proposto se torna muito mais simples, rápido e eficiente, uma vez que os todos *peers* e/ou *super-peers* envolvidos cooperam entre si, no alcance de um objetivo em comum. Esta pode ser comparada a estratégia da divisão e conquista [Celes, Cerqueira e Rangel, 2004], que visa solucionar problemas grandes através da divisão destes em problemas menores, resolvendo-os separadamente e integrando-os posteriormente para a obtenção do resultado final.

Assim, cada um dos resultados parciais retornados pelos *peers* e/ou *super-peers* ao *peer* ou *super-peer* que submeteu a consulta já se encontra livre de qualquer tipo de inconsistência, bastando apenas que o sistema de integração de dados faça a união destes resultados. A FIG. 4.26 apresenta um exemplo de uma consulta submetida por um usuário ao *super-peer* SP_1 . Este, ao verificar que a consulta trata do seu assunto, a executa mantendo o seu próprio resultado parcial em *cache* e a reenvia aos seus *peers* P_0 e P_1 . Estes *peers* também a processam enviando os resultados para o *peer* solicitador (*super-peer* SP_1). Quando os resultados forem todos retornados, ou quando o tempo de espera de resultados estiver terminado, os resultados parciais recebidos são unidos formando o resultado final, que é então enviado como resposta ao usuário.

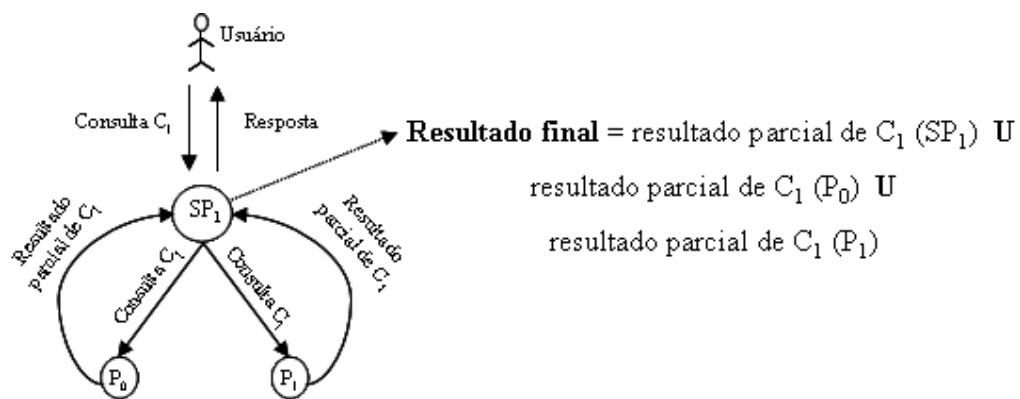


FIG. 4.26 – Exemplo da integração de dados

De fato, a etapa de integração de dados funciona como um complemento da etapa de processamento de consultas. Ela permite que os resultados individuais de cada processamento de consultas sejam unidos formando um único resultado, denominado de resultado final, correspondente à integração de todos os resultados.

Sua utilização e implementação se tornam muito simples em vista do problema a resolver. Contudo, a rapidez e eficiência predominam, fazendo com que esta etapa contribua para otimizar a estratégia do sistema de integração de dados como um todo, aumentando conseqüentemente, o desempenho geral do ROSA - P2P.

4.3.3 CONSIDERAÇÕES FINAIS

O sistema de integração de dados construído para o ROSA - P2P atendeu a todos os pré-requisitos necessários de um sistema de integração de dados *peer-to-peer*, tais como: definição de uma arquitetura específica, de acordo com o objetivo, arquitetura, funcionamento e características do sistema proposto, além da definição de uma estratégia de funcionamento, composta de estruturas semânticas denominadas de vocabulários controlados, que visam embutir significado semântico aos dados, auxiliando durante todo o processo; um serviço de entrega de vocabulários controlados, que permite agilizar atualizações e não permitir inconsistência de versões dos mesmos; uma estratégia bem definida quanto ao processamento de consultas dividida em reenvio de consultas, MEC ROSA (máquina de execução de consultas) e reescrita de consultas; e integração dos dados.

Nesta etapa de integração, trabalhou-se para que todos os problemas relevantes ao sistema proposto fossem resolvidos. Por exemplo, os problemas de caráter semântico, a exemplo das sinonímias e homonímias, foram solucionados através do uso dos vocabulários controlados global e local, os quais permitiram a correta interpretação dos dados residentes nas bases de dados. Outros problemas, tais como os específicos do ambiente P2P, como por exemplo “O que perguntar (WTA - what-to-ask)”, foi resolvido através da utilização do processo de reescrita da consulta, que consiste em reescrevê-la, enriquecendo-a semanticamente, abrangendo assim, todos os resultados semanticamente corretos. O problema de “distribuição/colocação dos dados” também foi solucionado através da estratégia de

agrupamento de *peers* de acordo com características particulares, definidas a priori por assunto e localização, possibilitando assim que dados com estas características comuns ficassem próximos uns dos outros, facilitando sua localização e, conseqüentemente, otimizando todo o processamento de consultas.

O sistema de integração de dados desenvolvido neste trabalho tentou abordar grande parte dos problemas pertinentes a um processo de integração. Procurou fornecer subsídios para que os resultados das consultas pudessem ser devidamente armazenados, identificados, localizados, processados, recuperados e integrados com rapidez, confiabilidade e eficiência.

5. PROTÓTIPO DO SISTEMA PROPOSTO

Este capítulo tem como objetivo apresentar o protótipo do sistema proposto, comentando aspectos sobre a sua implementação, módulos implementados, interfaces de comunicação e um exemplo de utilização do sistema proposto, oferecendo assim uma visão geral do seu desenvolvimento. Visa também exibir a avaliação do mesmo em um ambiente distribuído real, validando o seu funcionamento assim como o trabalho proposto.

O código fonte das principais rotinas se encontra disponível no CD-ROM em anexo.

5.1 DETALHES DA IMPLEMENTAÇÃO

O desenvolvimento do protótipo foi realizado com base nas informações referentes à definição do sistema, havendo porém grande preocupação não apenas com a sua validação, mas também com o uso real do sistema por instituições, usuários e outros alunos que desejem dar seqüência ao desenvolvimento desse sistema. Esta etapa se dividiu em duas fases, a saber:

- ROSA - P2P: refere-se a implementação de cada *peer*. Esta foi dividida em outras duas fases: ambiente P2P e integração de dados. Optou-se em implementar o ambiente P2P antes do módulo de integração de dados, uma vez que o mesmo serve como uma base para o sistema como um todo, visto que define fisicamente a formação da rede P2P assim como a comunicação entre os *peers*;
- Portal ROSA: diz respeito à implementação do portal ROSA (implementação WEB) e de seus serviços – serviço de diretório e serviço de entrega de vocabulários controlados. O portal ROSA foi implementado utilizando a tecnologia de servlets [Servlet], possibilitando assim, que este esteja ativo para todos a todo o momento. Contudo, seus serviços são aplicações JAVA comuns.

O sistema proposto utilizou em sua implementação o software Eclipse [Eclipse] e a linguagem de programação JAVA [JAVA], mais especificamente a plataforma J2EE [J2EE]. O Eclipse, principalmente por ser um software de uso livre cada vez mais utilizado pela

sociedade acadêmica, e a linguagem JAVA, por ser uma linguagem com grandes recursos tecnológicos possuindo assim, os subsídios necessários exigidos na implementação do sistema.

Quanto à programação, foi utilizado o conceito de orientação a objeto (OO) [Lau, 2001]. Este define objetos que são tipos de dados com estruturas e estados, os quais representam um conceito do mundo real. Seu uso se justifica, pois a programação OO possui, dentre outros, o conceito de encapsulamento e herança, que oferece maior legibilidade, além de facilitar a manutenção e favorecer a reutilização.

Para o desenvolvimento WEB foi utilizado o padrão J2EE [Padrão J2EE], especificamente as camadas *Front Controller* e *Dispatcher Helper*. O primeiro por prover um controlador centralizado para gerenciar o processamento de requisições; e o segundo por auxiliar o *Front Controller* no despacho destas requisições. As páginas de apresentação foram desenvolvidas através da tecnologia JSP [JSP].

Optou-se por desenvolver todo o sistema e não instanciá-lo a partir de um framework existente, a exemplo do JXTA [JXTA], uma vez que isto requereria um grande tempo de estudo, ao mesmo tempo em que se desejava controlar e definir toda e qualquer comunicação de forma personalizada. Assim, para prover a comunicação entre os *peers* do sistema foi utilizada a classe *socket* (disponível na plataforma J2EE), que oferece uma abstração das técnicas padrão de programação de soquete TCP [Lemay e Cadenhead, 2003].

5.2 MÓDULOS IMPLEMENTADOS

Os módulos implementados foram classificados quanto ao ambiente P2P e a integração de dados. Desta forma, pode-se destacar dentre os principais relacionados com o ambiente P2P:

- Interoperador P2P: responsável pelas rotinas de conexão e desconexão de *peers* ao sistema e comunicação entre *peers*;
- Atualizador de índices: responsável por manter os índices de roteamento e a tabela de propriedades atualizadas;
- Segurança: responsável pela autenticação dos *peers* e pelos mecanismos de tolerância a falhas.

e dentre os principais relacionados com o ambiente de integração:

- Manutenção de dados: responsável pelo armazenamento, manutenção e recuperação de dados;
- Gerador de consulta: responsável pela geração da consulta segundo parâmetros fornecidos pelo usuário;
- Processamento de consulta: responsável por todo o processamento da consulta. Este consiste das rotinas de reenvio, reescrita e execução da consulta;
- Integrador: responsável pela integração dos resultados parciais retornados;
- Interfaces: oferece os meios para interação com o usuário através de interfaces.

Contudo, o portal ROSA e os serviços de diretório e de entrega de vocabulários controlados também foram implementados. Porém, estes persistem em um na máquina externa ao ambiente P2P, podendo se comunicar com os demais *super-peers* e serem acessados, respectivamente.

5.3 INTERFACES DO USUÁRIO

As interfaces do sistema foram desenvolvidas de modo a permitir e garantir um ambiente amigável de comunicação entre usuários e sistema. As próximas seções apresentam as interfaces, também referenciadas como telas ou páginas, referentes ao ROSA - P2P e ao portal ROSA respectivamente. Os serviços oferecidos pelo sistema não possuem interface, uma vez que são utilizados somente pela aplicação e não por usuários.

5.3.1 INTERFACES DO ROSA - P2P

O ROSA - P2P apresenta três interfaces de comunicação com o usuário. Elas são denominadas respectivamente de tela de abertura, tela de geração de consulta e tela de configuração, conforme se pode ver a seguir.

5.3.1.1 TELA DE ABERTURA

A FIG. 5.1 apresenta a tela de abertura do ROSA - P2P. Ela pode ser dividida em duas partes. A primeira (parte superior da tela) se refere ao processamento de uma consulta e é composta pelos seguintes componentes:

- Botão “Gerar Consulta”: pressionando este botão, o sistema abre uma tela que auxiliará o usuário na formulação e geração da consulta. Esta tela, denominada de tela de geração de consulta será apresentada mais adiante (seção 5.2.1.2);
- Botão “Enviar”: envia a consulta aos *super-peers* e *peers* relevantes aos quais aponta.
- Botão “Nova”: visa limpar o campo “Consulta” assim como o campo “Resposta”;
- Campo “Consulta”: recebe a consulta gerada pela tela de geração de consulta. Este campo não pode ser alterado. Serve somente para dar uma visão da consulta gerada;
- Campo “Resposta”: apresenta os resultados lógicos (LOs lógicos) gerados correspondentes a cada consulta, assim como os resultados resultantes da operação de transitividade. O campo de resposta referente aos resultados físicos (LOs físicos) será visto mais adiante, ainda nesta seção.

A segunda parte (canto inferior da tela) corresponde ao status do sistema. Este enriquece o sistema, uma vez que apresenta ao usuário informações sobre sua configuração atual e facilidades para que o mesmo entenda o que o sistema está fazendo a cada momento. Ele é composto pelos seguintes componentes:

- Campo “*Super-peer*”: indica se o *peer* é um *super-peer* ou não;
- Campo “Assunto”: apresenta o assunto tratado pelo *peer* ou *super-peer*;
- Campo “Localização”: exhibe o país de origem do *peer* ou *super-peer*;
- Campo “QTD *super-peers*”: este campo só é ativado no caso em que o *peer* é um *super-peer*. Ele mostra a quantidade de *super-peers* ligados ao *super-peer* em questão;
- Área “Cliente”: exhibe as tarefas realizadas pelo lado cliente do *peer* ou *super-peer*, tais como: passos para a conexão com o *super-peer*, envio de consultas, etc;
- Área “Servidor”: exhibe as tarefas realizadas pelo lado servidor do *peer* ou *super-peer*, tais como: se o mesmo está ativo, recebimento de consultas, envio de respostas, etc.

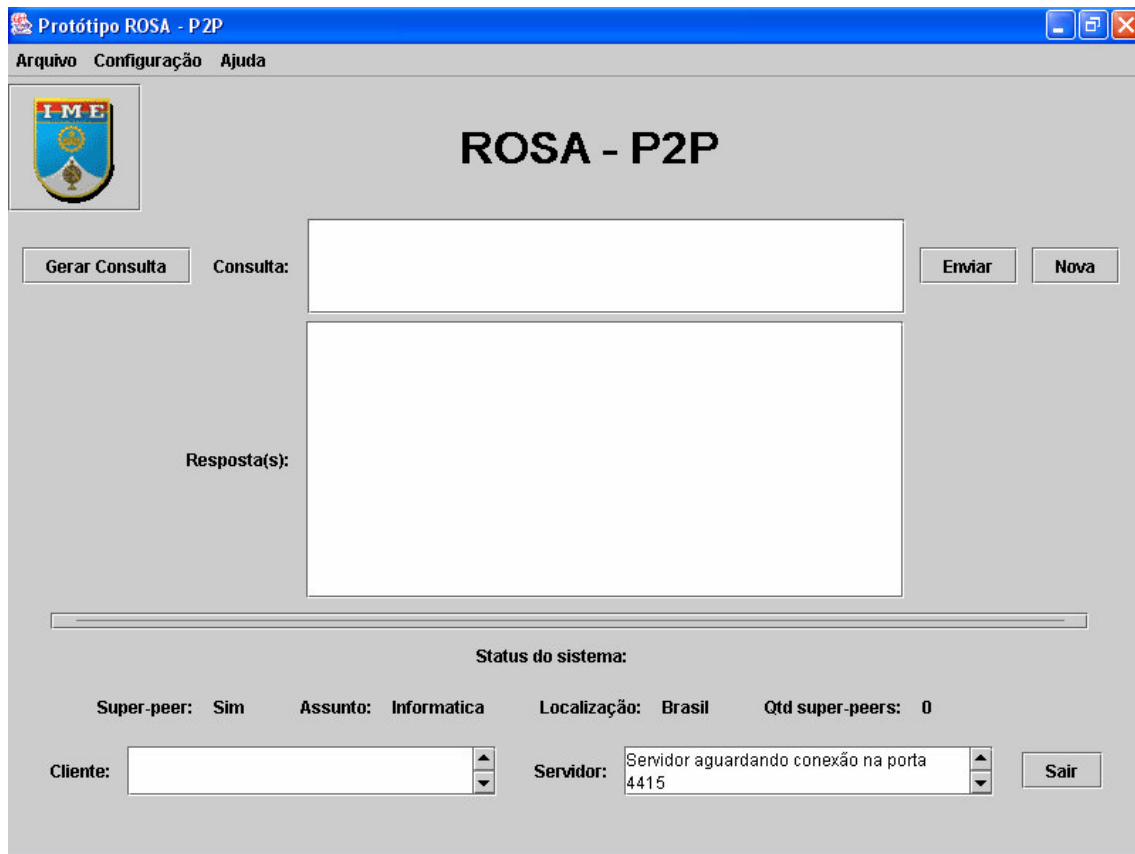


FIG. 5.1 – Tela de abertura do ROSA - P2P

Existem mais dois componentes: o botão “Sair”, que fecha o sistema, e um menu, composto por:

- Arquivo: apresenta os botões “Gerar Consulta”, “Enviar”, “Nova” e “Sair”. É uma outra opção para acessá-los;
- Configuração: oferece o botão para configurar o sistema, o qual leva a tela de configuração. Esta tela será discutida na seção 5.2.1.3;
- Ajuda: Apresenta dois botões. O botão de ajuda, que oferece explicações para as possíveis dúvidas dos usuários e o botão “Sobre”, que fornece informações técnicas sobre o sistema, tais como desenvolvedor e versão.

Porém, uma vez que a interface de abertura apresenta os resultados aos usuários, algumas variações de seus componentes são apresentadas de acordo com o conteúdo dos resultados retornados. A FIG. 5.2 apresenta a tela de abertura referente à exibição dos resultados segundo a variação mais completa, que compõe todos os componentes adicionais possíveis.

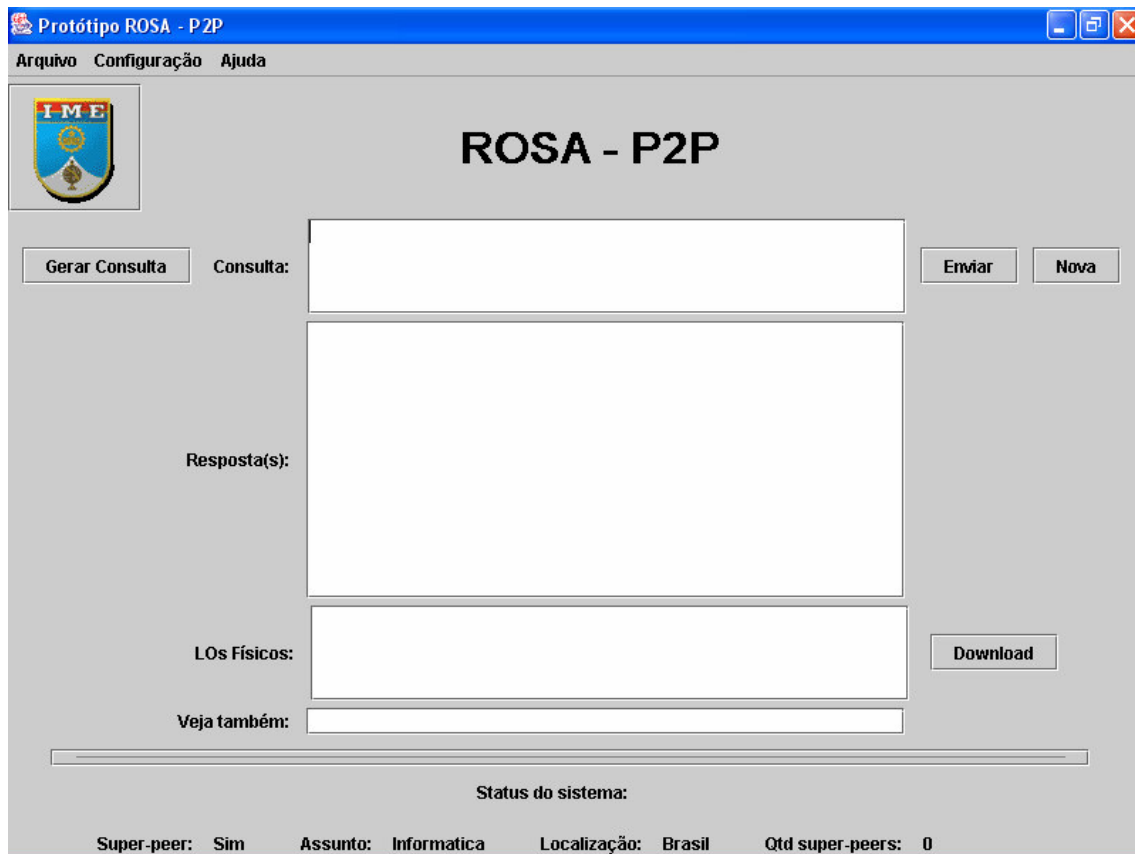


FIG. 5.2 – Tela de exibição de resultados

A diferença consiste da adição de mais três componentes, definidos como:

- Campo “LOs Físicos”: apresenta os LOs físicos retornados como resultado, possibilitando que o usuário selecione-os para *download*;
- Botão “Download”: Pressionando este botão, o usuário executa a download do arquivo (LO físico) selecionado no campo “LOs Físicos”. Todos os arquivos “baixados” são armazenados na pasta download situada no “path” c:\ROSA - P2P;
- Campo “Veja também”: apresenta os termos associados ao termo “título” da consulta.

5.3.1.2 TELA DE GERAÇÃO DE CONSULTA

Esta tela pode ser vista na FIG. 5.3. Ela auxilia o usuário na formulação e conseqüente geração da consulta. Houve necessidade de desenvolvê-la, visto a complexidade na sintaxe da

MEC ROSA. Ela é formada na sua maioria por diversas caixas de seleção, as quais dependem uma das outras para estarem ativas. Seus componentes são:

- Caixa de seleção “Operação”: apresenta as opções “select”, “browsing” e “project”, as quais determinam quais outras caixas de seleção serão ativadas e quais serão seus valores. Ativam também alguns componentes correlacionados. Assim, se o usuário selecionar a opção “select”, as caixas de seleção “Metadado”, “Atributo” e “Op”, assim como o campo “Valor” serão ativados, uma vez que oferecem as opções para a formulação desta operação. As demais caixas devem ser desativadas. Caso a opção selecionada seja “browsing”, somente as caixas de seleção referente aos predicados e operadores devem ser ativadas. As demais, casos estejam ativadas, devem ser desativadas. Por último, caso seja selecionada a operação “project”, somente deve ser ativada as caixas de seleção “Metadado” e “Atributo”;
- Caixa de seleção “Metadado”: apresenta os metadados disponíveis no sistema, de acordo com os descritores LOM. Porém, de modo a facilitar a implementação do protótipo, optou-se por utilizar somente um subconjunto destes descritores;
- Caixa de seleção “Atributo”: exibe um nível abaixo do metadado selecionado na caixa de seleção “Metadado”, atuando desta forma, como um filtro. Ex. O metadado “aspectos técnicos” inclui os metadados “formato”, “tamanho”, “localização” e “requisitos”. Assim, caso este seja selecionado na caixa de seleção “Metadado”, a caixa de seleção “Atributo” exibirá os metadados que são incluídos por ele;
- Campo “Valor”: recebe o valor determinado para o metadado;
- Botão “Op”: apresenta os operadores “and” e “or”, os quais têm o papel de compor a operação de seleção;
- Caixa de seleção “Predicado”: exibe os predicados definidos no sistema;
- Caixa de seleção “Operador”: exibe os operadores “and”, “or” ou “.”, os quais têm o papel de compor a operação de navegação;
- Botão “Adicionar Operação”: adiciona a operação na área “Formulação da consulta”. Devido a possibilidade de cada operação poder ser muito extensa e não se saber quantas caixas de seleção serão necessárias, optou-se por compor cada uma das operações em pequenas partes. Desta forma, para cada parte composta, o usuário a adiciona à área de formulação de consulta que se encarrega de colocá-la junto à operação correta. Por exemplo, a consulta “selectlOs@lom/aspectos_pedagógicos/ dificuldade = alta and lOs@ lom/geral/idioma = Francês” seria primeiramente composta por

“selectLOs@lom/aspectos_pedagógicos/dificuldade = alta and” e depois por “LOs@lom/geral/idioma = Francês”;

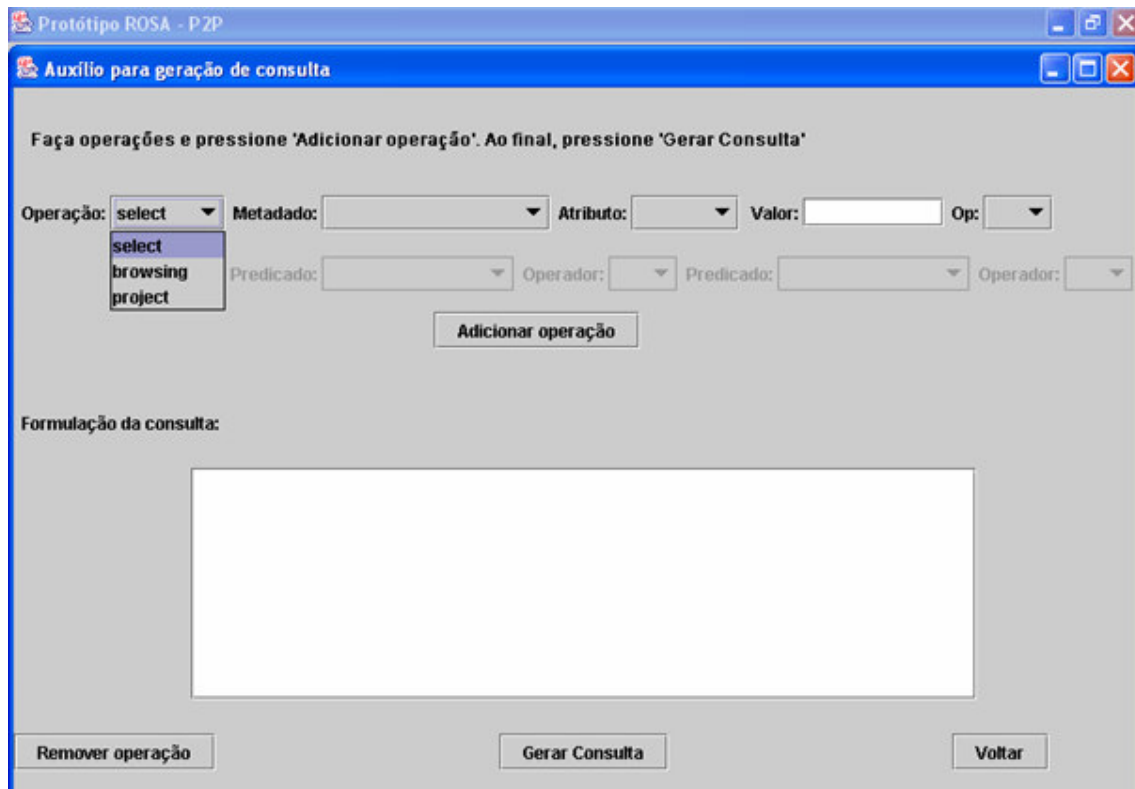


FIG. 5.3 – Tela de geração da consulta do ROSA - P2P

- Área “Formulação da consulta”: recebe as partes das operações e as monta corretamente;
- Botão “Remover operação”: remove uma operação já inserida na área “Formulação da consulta”;
- Botão “Gerar consulta”: finaliza a geração da consulta, reenviando-a para a campo “Consulta” da tela de abertura;
- Botão “Voltar”: volta para a tela anterior, desconsiderando todas as mudanças realizadas.

5.3.1.3 TELA DE CONFIGURAÇÃO

A tela de configuração é de extrema importância para o correto funcionamento do sistema. Aparece automaticamente para o usuário na primeira vez que o sistema é utilizado e visa obter informações cruciais para a ativação/inclusão do *peer* no sistema, envio de consulta, eleição de *super-peers*, dentre outras. Esta tela pode ser vista na FIG. 5.4 e é composta pelos seguintes componentes:

- Campo “Nome”: recebe o nome do *peer* ou *super-peer*. Este campo tem como objetivo identificar o *peer* ou *super-peer* e, desta forma, anexar esta informação nos resultados retornados a um *peer* ou *super-peer* solicitador da consulta;

Configuração do sistema

Informe os dados abaixo e pressione OK.

Nome:

Tipo: **Instituicao de ensino** **Particular**

Deseja ser um super-peer: **Sim** **Nao**

Assunto:

Localização:

Características Físicas:

Estabilidade: **Baixo** **Medio** **Alto**

Largura de banda: **Baixo** **Medio** **Alto**

Poder de Processamento: **Baixo** **Medio** **Alto**

Capacidade de armazenamento: **Baixo** **Medio** **Alto**

FIG. 5.4 – Tela de configuração do ROSA - P2P

- Botão de opção “Tipo”: recebe a informação se o *peer* ou *super-peer* é uma instituição de ensino ou é um usuário particular;
- Botão de opção “Deseja ser um *super-peer*”: recebe a informação se o *peer* deseja ser um *super-peer* ou não. Caso ele seja, pode optar por não ser mais;
- Caixa de seleção “assunto”: recebe a informação do assunto tratado pelo *peer* ou *super-peer*;
- Caixa de seleção “Localização”: recebe a informação sobre a localização de origem do *peer* ou *super-peer*;
- Caixa de checagem “Estabilidade”: obtém o nível de estabilidade do *peer* ou *super-peer*: baixo, médio ou alto;
- Caixa de checagem “Largura de banda”: obtém a o nível de largura de banda do *peer* ou *super-peer*: baixo, médio ou alto;
- Caixa de checagem “Poder de processamento”: obtém o nível do poder de processamento do *peer* ou *super-peer*: baixo, médio ou alto;
- Caixa de checagem “Capacidade de armazenamento”: obtém o nível de capacidade de armazenamento do *peer* ou *super-peer*: baixo, médio ou alto.

5.3.2 INTERFACES DO PORTAL ROSA

O portal ROSA apresenta três interfaces de comunicação com o usuário, que na verdade correspondem a páginas JSP. Elas são denominadas de página inicial, página principal e página de geração de consulta. A página referente à configuração do sistema não é necessária, uma vez que o portal ROSA não faz parte da rede P2P e atua somente como um ponto de entrada de consultas para aqueles que não possuem o ROSA - P2P.

Devido a semelhança de apresentação do portal ROSA com o ROSA - P2P, a página referente à geração de consulta não será apresentada. Inclusive, a tela principal já transparece esta semelhança, conforme será visto adiante.

5.3.2.1 PÁGINA INICIAL

Conforme mostra a FIG. 5.5, esta página tem como objetivo indicar para o usuário que ele encontrou o portal ROSA. Ela possui um texto de boas vindas e o botão “Entrar”, que tem o papel de acionar o *servlet*, ativando uma seção e dando permissão ao usuário para passar para a tela principal.

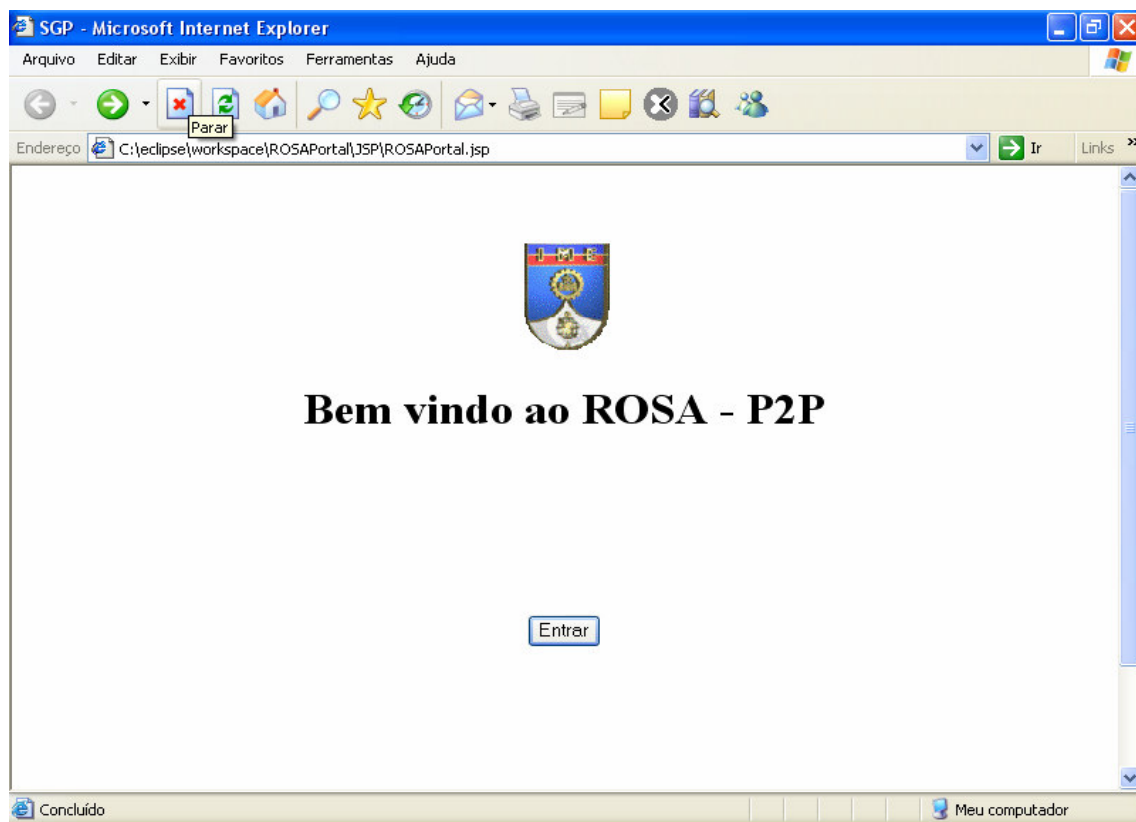


FIG. 5.5 – Página inicial do portal ROSA

5.3.2.2 PÁGINA PRINCIPAL

Seus componentes possuem a mesma funcionalidade dos existentes na tela de abertura do ROSA - P2P, com os botões para a geração da consulta e envio da mesma, assim como os campos referentes à consulta e aos resultados, conforme exibe a FIG. 5.6.

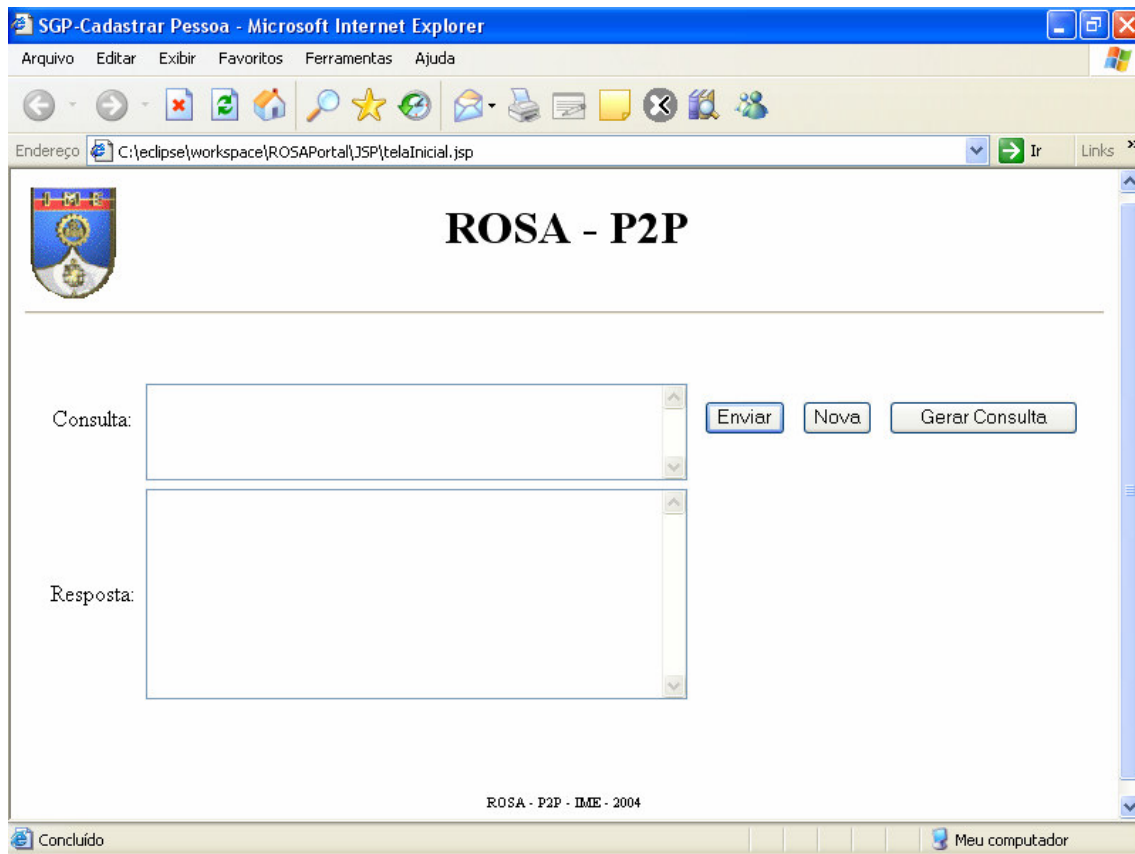


FIG. 5.6 – Página principal do portal ROSA

5.4 EXEMPLO DE USO

Uma vez apresentada as interfaces do sistema proposto, faz-se necessário a apresentação de um exemplo de sua utilização. Este se utilizará das interfaces do ROSA - P2P, valendo o mesmo para as interfaces referentes ao portal ROSA. Considera-se que o sistema já esteja devidamente instalado e o *peer* apto a enviar e receber consultas. Sendo assim, este exemplo visa mostrar a submissão de uma consulta e a apresentação dos respectivos resultados através das interfaces do ROSA - P2P.

Partindo da premissa de que o sistema foi inicializado e todas as rotinas de inicialização completadas com sucesso, tais como a conexão com um *super-peer* ou sua eleição como *super-peer*, assim como a atualização de dados nos índices SP/P e/ou SP/SP, o usuário se deparará com a tela de abertura onde o status referente ao servidor será “Servidor aguardando

conexão na porta 4415”, conforme mostra a FIG. 5.1. A partir deste momento, o *peer* já está ativo no sistema e apto a enviar e receber consultas. Assim, para enviar uma consulta, o usuário deve pressionar o botão “Gerar consulta”. Este abrirá a tela de geração de consulta, que auxiliará o usuário na formulação e geração da mesma. Desta forma, levando-se em conta os mapas conceituais parciais exibidos nas FIG. 5.7 e 5.8, suponha que o usuário deseja submeter a seguinte consulta ao sistema:

Consulta - Exiba os títulos dos LOs (físicos e/ou lógicos) que são formados pela disciplina *banco de dados distribuídos*.

Mapa Conceitual Parcial do IME-RJ – Assunto Informática

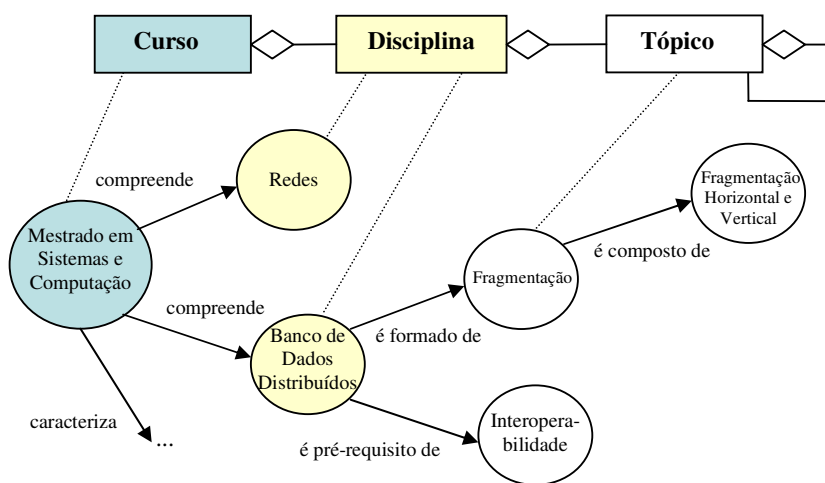


FIG. 5.7 – Mapa conceitual parcial do IME-RJ

Mapa Conceitual Parcial da PUC-RJ – Assunto Informática

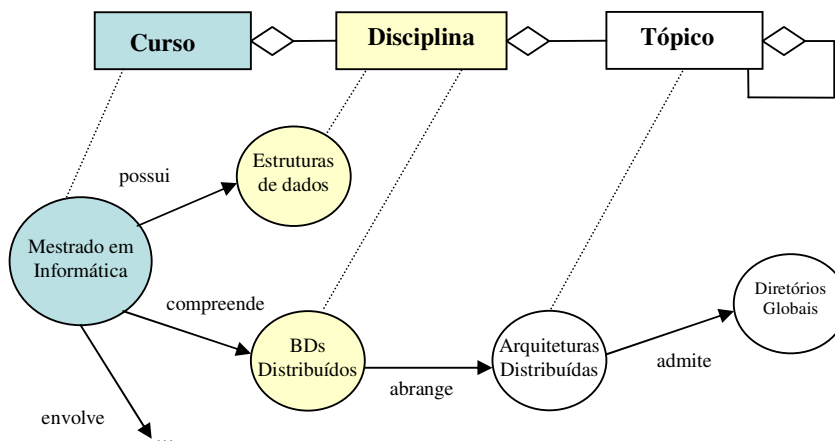


FIG. 5.8 – Mapa conceitual parcial da PUC-RJ

Para gerar esta consulta, o usuário deve formulá-la selecionando as operações correspondentes, uma de cada vez no campo “Formulação da consulta”. A ordem na qual estas operações são formuladas é irrelevante ao sistema, pois à medida em que são adicionadas, as mesmas são posicionadas de modo que possam ser corretamente interpretadas pela MEC ROSA. Esta ordem consiste em primeiro definir a operação de seleção, seguida pela de navegação e por último a de projeção.

Sendo assim, uma das maneiras possíveis seria a seleção da operação de seleção “select|LOs@ lom/geral/título = banco de dados distribuídos” seguida pela operação de navegação browsing|LOs@(é_formado_por) e finalizado com a de projeção project|LOs@lom/geral/título, conforme mostra a FIG. 5.9.

Protótipo ROSA - P2P

Auxílio para geração de consulta

Faça operações e pressione 'Adicionar operação'. Ao final, pressione 'Gerar Consulta'

Operação: project Metadado: Atributo: Valor: Op:

Predicado: Operador: Predicado: Operador:

Adicionar operação

Formulação da consulta:

```
select|LOs@ lom/geral/título = banco de dados distribuídos
browsing|LOs@( é_formado_por )
project|LOs@lom/geral/título
```

Remover operação Gerar Consulta Voltar

Status do sistema:

FIG. 5.9 – formulação de consulta (exemplo)

Terminada a formulação da consulta, o usuário deve pressionar o botão “Gerar Consulta”, que finaliza sua construção e a envia ao campo “Consulta” da tela de abertura.

De volta à tela de abertura, o usuário deve enviar a consulta. Para isso, basta pressionar o botão “Enviar”. Neste momento, a consulta será reescrita e executada pelo *peer*. Este também a reenviará, na sua formula inicial, aos *peers* e/ou *super-peers* relevantes. O contador de resultados e do tempo de espera de resultados é iniciado. Na medida em que os resultados vão sendo recebidos, são armazenados em *cache*. Ao fim da chegada de todos os resultados, ou ao término do tempo de espera de resultados, estes são integrados e exibidos ao usuário.

Conforme apresentado na FIG. 5.10, o *peer* identificado como IME-RJ retornou como resultado os LOs lógicos *fragmentação* e *fragmentação horizontal e vertical*. Este último através da propriedade transitiva. Também retornou o LO físico *tudo sobre BDs distribuído*, que embora não se encontre representado no mapa conceitual, existe fisicamente na base de dados (vide seção 3.3). Já o *peer* identificado como PUC-RJ retornou somente o LO lógico *arquiteturas distribuídas*. Estes foram os únicos resultados retornados.



FIG. 5.10 – Resultado da consulta

Contudo, caso o usuário deseje fazer o “download” do arquivo físico *tudo sobre BDs distribuídos*, basta selecioná-lo e pressionar o botão “Download”. Também é apresentado no campo “Veja também”, os termos genéricos a *banco de dados distribuídos*. Este indica ao usuário que a consulta refeita com a adição desse termo pode fazer com que outros resultados relevantes possam ser encontrados.

5.5 TESTES E AVALIAÇÃO DO SISTEMA PROPOSTO

A etapa de testes e avaliação do sistema proposto tem como objetivo testar o sistema em um ambiente acadêmico distribuído, focalizando todos os pontos pertinentes ao seu correto funcionamento, tais como as funcionalidades referentes ao ambiente P2P e ao processo de integração de dados. Estes possibilitaram a avaliação do sistema sob o ponto de vista da formação da rede P2P, comunicação entre os *peers*, atualizações dos índices de roteamento e processo de integração dos dados, levando-se em conta o tempo e a correção dos resultados retornados aos usuários.

Os testes foram realizados em um laboratório cedido pelo LNCC – Laboratório Nacional de Computação Científica [LNCC]. As características de hardware referentes ao ambiente distribuído utilizado e à instanciação dos *peers* foram as seguintes:

- Ambiente distribuído constituído por seis máquinas conectadas em rede;
- Cada máquina dotada de um processador Pentium 4 com tecnologia HT – hyper-threading [Intel] e 2Gb de memória;
- Velocidade da rede de 1Gb/s;
- *Peers* participantes trataram do assunto de Informática, Medicina e Direito, cada um com um vocabulário local específico;
- Cada *peer* possuiu uma base de dados distinta, correspondente a um mapa conceitual.

Os testes constituíram-se primeiramente da formação da rede P2P pelos *peers* participantes. Para tal, uma máquina foi escolhida para suportar o portal ROSA e os serviços de diretório e de entrega de vocabulários controlados. Uma vez que estes serviços foram ativados, *peers* foram sendo instalados até a completa formação da rede. Como exemplo, uma

das redes formadas constituiu-se de dois *super-peers* e um *peer* de Informática, um *super-peer* e um *peer* de Medicina e um *super-peer* de Direito.

Durante o processo de formação da rede, funcionalidades do ambiente P2P foram sendo testadas. Corresponderam ao acesso aos serviços oferecidos, comunicação entre os *peers* e *super-peers*, atualizações dos índices de roteamentos e tabela de propriedades, formação correta da rede segundo as características de assunto e país de origem, dentre outras.

Finalizado os testes referentes à formação e funcionamento do ambiente P2P, deu-se início aos testes referentes ao sistema de integração de dados. Consultas com diferentes conotações semânticas e distintos níveis de complexidade foram submetidas ao sistema visando abranger todos os possíveis conflitos e dificuldades existentes. Alguns delas se encontram definidas a seguir, caracterizadas segundo sua finalidade e importância no contexto dos testes realizados.

1) Exiba os LOs fundamentados por aqueles que geram outros LOs e que possuam “título” igual a *bioinformática* e “idioma” igual a *português*

Características: Por tratar de uma instância da área de *bioinformática*, esta consulta será submetida à *peers* que tratem tanto do assunto informática quanto de medicina. Além disso, navegará 2 níveis na hierarquia segundo seu mapa conceitual (gera -> fundamenta).

2) Obtenha os LOs físicos que tratem do assunto *direito* localizados em instituições *brasileiras* com tamanho máximo de *1024kbytes*

Características: Esta consulta verifica o retorno e apresentação de LOs físicos no sistema, segundo especificações de alguns metadados.

3) Exiba os títulos e as descrições dos LOs que são base para outros LOs os quais são pré-requisito ou compreendidos por aqueles que requerem outros LOS e que possuam “título” igual a *Internet* e “nível de agregação” igual a *disciplina* com dificuldade *alta*.

Características: Esta consulta atesta se o problema referente ao conflito de papéis é realmente solucionado pelo sistema. Neste contexto, somente os LOs que possuem título igual a *Internet* e nível de agregação igual a *disciplina* devem ser retornados. Assim, caso um *peer* possua um LO cujo “título” seja igual a *Internet* e “nível de agregação” igual a *tópico*, este não deve ser considerado. Além disso, esta consulta navega 2 níveis na

hierarquia, considerando 2 predicados para o primeiro nível (é pré-requisito -> é base para) (compreende -> é base para).

Em todos os casos, elas foram corretamente reenviadas somente aos *peers* relevantes, isto é, aos que tratavam do mesmo assunto e que estavam “on-line”, reescrita com o suporte do vocabulário controlado e executada pela MEC ROSA. Os respectivos resultados foram retornados ao *peer* solicitador em um tempo médio de 0.25s por *peer*, sendo corretamente integrados e exibidos ao usuário.

5.6 CONSIDERAÇÕES FINAIS

A implementação do protótipo utilizou-se de tecnologias atuais, tais quais o software Eclipse e a linguagem de programação JAVA. Estas possibilitam, dentre outras coisas, facilidade de manutenção e maior compatibilidade com os demais módulos e sistemas.

Quanto aos testes realizados, embora tenha sido utilizada uma rede física de alta velocidade com os *peers* localizados na mesma rede, o tempo médio de 0.25s por resposta nos surpreendeu. Acredita-se que quando os *peers* estiverem espalhados geograficamente, o tempo de resposta por consulta não ultrapassará o tempo médio de 0.5s, isto é, o dobro do tempo obtido nos testes realizados na mesma rede. Esta margem é atribuída às variantes externas, tais como o meio físico utilizado e o tempo de percurso da informação. Apesar deste tempo ser de caráter estimativo, testes suplementares e simulações são necessários para a sua comprovação. Estes servirão também para a avaliação da estabilidade do protocolo de comunicação desenvolvido, previstos como trabalho futuro.

O sistema se mostrou capaz de comprovar a especificação teórica desenvolvida no escopo desse trabalho, estando apto a ser utilizado pelas instituições e usuários interessados. Os testes, além de demonstrarem a ausência de *bugs* e inconsistências, explicitaram um tempo de resposta satisfatório em ambiente distribuído. Este agrega maior valor ao sistema, caracterizando-se também como um sistema gerenciador de banco de dados P2P com um excelente desempenho na execução de consultas.

6. CONCLUSÃO

A tecnologia P2P vem se destacando como uma das grandes tecnologias de sistemas distribuídos da atualidade. Como é possível verificar, esta oferece benefícios com um baixo custo operacional, tais como o compartilhamento de recursos, que pode ser mais objetivamente referenciado como o compartilhamento de serviços e conteúdos. Embora outras tecnologias, a exemplo de GRID [Foster, Kesselman e Tuecke, 2001], venham também tomando grande posição de destaque na literatura, estas possivelmente não se encaixariam aos moldes e características da realidade do sistema ROSA, onde existe a individualização e autonomia de cada sistema, sem falar no custo de infra-estrutura que uma arquitetura GRID requer.

O objetivo desta dissertação foi transformar o sistema ROSA em um sistema P2P, gerando o ROSA - P2P, viabilizando a integração de objetos de aprendizagem em ambiente distribuído. Dentre as arquiteturas P2P existentes, optou-se por utilizar a arquitetura baseada em *super-peers*, visto que esta possui redução de tempo e largura de banda para pesquisa, estratégias contra falhas do sistema, gerenciamento através dos *super-peers* e um índice aceitável de confiabilidade e escalabilidade, se encaixando e favorecendo na estratégia definida para o ROSA - P2P.

Esta estratégia, embora baseada em alguns aspectos quanto ao uso dos índices de roteamento SP/P e SP/SP, reescrita das consultas e localização de *peers* relevantes às consultas de alguns sistemas P2P existentes, possui características bem específicas e definidas, as quais se destacam, dentre outras coisas, pela evolução e aprimoramento das utilizadas por estes sistemas, a exemplo do agrupamento de *peers* utilizado pelo sistema Edutela, que no ROSA - P2P agrupa os *super-peers* ao invés dos *peers*, favorecendo a legibilidade da rede e, principalmente, o processamento de consultas

Quanto ao sistema de integração de dados construído para o ROSA - P2P, este atendeu a todos os pré-requisitos necessários de um sistema de integração de dados P2P, possuindo uma arquitetura bem definida, baseada no objetivo, arquitetura, funcionamento e características do ROSA - P2P, além de uma estratégia de integração semântica composta de estruturas denominadas de vocabulários controlados, que visam embutir significado semântico aos dados, auxiliando durante todo o processo; uma estratégia bem definida quanto ao processamento de consultas dividida em reenvio de consultas, MEC ROSA e reescrita de consultas; e integração dos dados.

O módulo de integração de dados se preocupou também em solucionar todos os problemas relevantes ao sistema proposto, a exemplo das sinonímias e homonímias, que foram solucionados através do uso dos vocabulários controlados global e local, os quais permitiram a correta interpretação dos dados residentes nas bases de dados. Procurou também fornecer subsídios para que os resultados das consultas pudessem ser devidamente armazenados, identificados, localizados, processados, recuperados e integrados com rapidez, confiabilidade e eficiência.

A tabela 6.1 apresenta algumas características do ROSA - P2P que podem ser comparadas às características dos demais sistemas apresentadas na seção 2.5.5 – tabela 2.3. Pode-se apontar como principal característica, o controle semântico dos dados através do uso de vocabulários controlados, destacando a complexidade do sistema de integração em todos os aspectos. Contudo, pode-se ainda ressaltar que o sistema de integração de dados utilizado pelo sistema ROSA - P2P se sobressai, uma vez que o uso do modelo de dados ROSA na integração de dados embute uma semântica mais rica na sua representação, além de contemplar a integração de dados numa arquitetura de *super-peers*, refletindo a tendência das pesquisas atuais na área.

Tabela 6.1 – Características do ROSA - P2P

Sistema de integração	Arquitetura P2P	Representação de dados	Ferramenta semântica	Mapeamento de esquemas	Processamento de consultas	Linguagem de consulta	Particularidades
ROSA - P2P	<i>Super-peers</i>	Modelo de dados ROSA (extensão do RDF)	Metadados e Vocabulários Controlados (global, local e de palavras-chaves)	Não possui. A integração ocorre entre as próprias instâncias	Reenvio e reescrita de consultas segundo os vocabulários controlados. Auxílio dos índices de roteamento	MEC - ROSA	Identificação da consulta em tempo de execução. Estratégia baseada nos valores contidos nos vocabulários controlados

O estudo de alguns dos principais sistemas P2P atuais foi de suma importância no desenvolvimento do ROSA - P2P, sugerindo e incentivando idéias até então não observadas.

Finalmente, pode-se considerar que o objetivo do trabalho tenha sido alcançado, através da evolução do ROSA de um sistema local para um sistema distribuído P2P, oferecendo, a partir deste momento, respostas globais às consultas dos usuários. Este foi validado com uma

implementação e testes exaustivos. Muito embora estes tenham sido realizados num ambiente com um número reduzido de *peers*, os resultados apresentaram um tempo de resposta satisfatório, face ao número de domínios de conhecimento testados, e serviram para a aprovação e liberação do sistema às instituições e usuários interessados.

6.1 CONTRIBUIÇÕES

Este trabalho contemplou o estudo e utilização da tecnologia P2P como instrumento capaz de evoluir o sistema ROSA atual em um sistema ROSA distribuído – ROSA - P2P, atribuindo novas funcionalidades e características até então não passíveis de serem realizadas e/ou observadas.

Desta forma, as principais contribuições desta dissertação foram:

- Estudo comparativo entre os principais sistemas P2P da atualidade, incluindo o ROSA - P2P, focalizando suas principais funcionalidade e características;
- Criação de um ambiente P2P servindo de estrutura básica para que o ROSA - P2P pudesse ser construído. Este contempla uma arquitetura consistente e estratégias relevantes para o gerenciamento de *peers* nesse ambiente;
- Criação de um sistema de integração de dados com características semânticas. Este visa abordar grande parte dos problemas pertinentes a um processo de integração e fornece os subsídios para que os resultados das consultas possam ser devidamente armazenados, identificados, localizados, processados, recuperados e integrados com rapidez, confiabilidade e eficiência;
- Evolução do sistema ROSA atual para um sistema distribuído – ROSA - P2P, possibilitando a interoperabilidade e compartilhamento de dados entre eles, oferecendo assim, respostas globais às consultas dos usuários;
- Implementação e testes do sistema ROSA - P2P através da utilização de tecnologias recentes, podendo-se destacar a linguagem JAVA, assim como o uso de servlet e JSP.

6.2 SUGESTÕES PARA TRABALHOS FUTUROS

Esta seção aponta novos caminhos que podem dar seguimento à pesquisa desenvolvida nesta dissertação. A seguir, são apresentadas algumas sugestões para trabalhos futuros:

- Dar continuidade a um trabalho de avaliação do ambiente com base numa plataforma mais robusta, e com um número relevante de domínios diversificados para consultas;
- Simulação do protocolo de comunicação em diferentes topologias de redes e quantidade de *peers*. Este demonstraria a real estabilidade do sistema;
- Enriquecimento do sistema com outros vocabulários controlados;
- Classificação de assuntos possíveis no sistema. Este seria realizado através da demanda dos usuários e gerenciado pelo administrador do sistema;
- Atualização dos vocabulários controlados. Este também seria gerenciado pelo administrador do sistema;
- Geração de um plano de consulta distribuído otimizado, o que possibilitaria um maior enriquecimento semântico e autonomia do processamento de consultas.
- Tratar o ROSA - P2P como um serviço Web, possibilitando a criação de outras aplicações também viabilizadas como serviços P2P, os quais favoreceriam uma melhor utilização e funcionamento do sistema.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- ABERER, K. et al. **P-Grid: A Self-organizing Structured P2P System**. SIGMOD Record, setembro/2003.
- ARENAS, M. et al. **The Hyperion Project: From Data Integration to Data Coordination** - *ACM SIGMOD Record*, 32(3):53-58, 2003.
- AYRES, F. **QEEF - uma máquina de execução de consultas extensível**. 2003. Tese (Doutorado em Ciências) – PUC-RIO - Pontifícia Universidade Católica do Rio de Janeiro, 2003.
- BATISTA, M. **Otimização de acesso em um sistema de integração de dados através do uso de caching e materialização de dados**. (Dissertação de Mestrado) - Universidade Federal de Pernambuco, Recife, abril/2003.
- BERNERS-LEE, T. et al. **The semantic web: a new form of web content that is meaningful to computers will unleash a revolution of new possibilities**. Revista Scientific American, maio/2001. Disponível em <http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html>
- BERTOSSI, L., BRAVO, L. **Query answering in data exchange *peer-to-peer* systems**. In Proc. of the 10th Int. Conf. on Database Theory, 2004.
- BRITO, G. A. D. D., MOURA, A. M. de C. **Fundamentos da Tecnologia *Peer-to-peer***. Relatório Técnico nº RT103/SE9/SET04. IME - Instituto Militar de Engenharia, Rio de Janeiro, setembro/2004.
- BRITO, G. A. D. D., MOURA, A. M. de C. **Integração de dados: dos sistemas tradicionais a sistemas P2P**. Relatório Técnico nº RT112/SE9/ABR05. IME - Instituto Militar de Engenharia, Rio de Janeiro, abril 2005.
- BUCCELLA, A., CECHICH, A. **An Ontology Approach to Data Integration**. JCS&T Vol. 3 N° 2 - outubro/2003.
- CALVANESE, D. et al. **What to ask to a *peer*: Ontology-query reformulation**. In Proc. of the 9th Int. Conf. on Principles Knowledge Representation and Reasoning, 2004.
- CELES, W. et al. **Introdução a Estruturas de Dados - com técnicas de programação em C**. Editora Campus, 2004.

- CORCHO, O., LOPEZ, M., PEREZ, A. **Methodologies, tools and languages for building ontologies. Where is their meeting point?** Elsevier Science B.V., outubro/2002.
- COSTA, F. **Uma abordagem baseada em lógica para representação e busca de LOs** - Tese de Mestrado, IME - Instituto Militar de Engenharia - Rio de Janeiro, 2005.
- COUTINHO, F. **Processamento de consultas sobre o modelo de dados ROSA** - Tese de Mestrado, IME - Instituto Militar de Engenharia - Rio de Janeiro, maio/2004.
- COUTINHO, F., PORTO, F. **Query Processing in ROSA Data Motel**. XIX Simpósio Brasileiro de Banco de Dados, Brasília, outubro/2004.
- ECLIPSE. **WEB site oficial** - <http://www.eclipse.org>
- EDUTELLA. **WEB site oficial** - <http://edutella.jxta.org>
- ELMASRI, R., NAVATHE, S. **Sistemas de Banco de Dados - Fundamentos e Aplicações**. Terceira edição. Editora LTC – 2002
- FERNANDEZ, A. **Representação e acesso a Objetos de Aprendizagem no sistema ROSA: uma abordagem em Topic Maps** - Tese de Mestrado, IME - Instituto Militar de Engenharia - Rio de Janeiro, 2004.
- FOSTER, L., KESSELMAN, C., TUECKE, S. **The Anatomy of the Grid: Enabling Scalable Virtual Organizations**. Int'l J. High-Performance Computing Applications, vol. 15, no. 3, pp. 200-222. 2001.
- GNUTELLA. **Projeto Gnutella**. Disponível em <http://www.file-sharing-page.de/gnutella.htm>. [Capturado em 09 jun 2004].
- GRIBBLE, S. et al. **What can databases do for peer-to-peer?** In Proc. of the 4th Int. Workshop on the Web and Databases, 2001.
- GRUBER, T. R. **A translation approach to portable ontology specification**. Knowledge Acquisition, v. 5, n. 2, p. 199-200, 1993.
- GUARINO, N. **Formal Ontologies and Information Systems**. In: First International Conference, Anais... Trento: IOS Press, Trento, 1998.
- GOMES, H. **Manual de elaboração de tesauros monolíngues**. Brasília, Programa Nacional de Bibliotecas das Instituições de Ensino Superior, 1990. 78 p.
- HALEVY, A. et al. **Piazza: Data Management Infrastructure for Semantic Web**

Applications. In WWW, 2003.

HTML. **WEB site oficial** - <http://www.w3.org/MarkUp/>

IANA. **WEB site oficial** - <http://www.iana.org/>

INTEL. **WEB site oficial** - <http://www.intel.com>

IPV6. **WEB site oficial** - www.ipv6.org

J2EE. **WEB site oficial** - <http://java.sun.com/j2ee/>

JAVA. **WEB site oficial** - <http://java.sun.com>

JAVASCRIPT. **Tutorial JavaScript.** Disponível em <http://www.w3schools.com/js/default.asp/> [Capturado em 15 jan 2005].

JSP. **WEB site** - <http://java.sun.com/products/jsp/>

JXTA. **WEB site oficial** - <http://www.jxta.org>

KAON. **WEB site oficial** - <http://kaon.semanticweb.org>

KAZAA. **WEB site oficial** - www.kazaalite.tk

KEMENTSIETSIDIS, A. et al. **Managing data mappings in the Hyperion Project.** In ICDE, pages 732-734, 2003.

LAU, Y. **The Art of Objects: Object-Oriented Design and Architecture.** Editora Addison-Wesley, 2001.

LEMAY, L. CADENHEAD, R. **Aprenda em 21 dias. JAVA2: professional reference.** 3ª edição. Editora Elsevier, 2003

LI, C. et al. **RACCOON: A Peer-Based System for Data Integration and Sharing.** Proceedings of the 20th International Conference on Data Engineering, 2004.

LNCC. **Laboratório Nacional de Computação Científica** - <http://www.lncc.br>

BAWA, M. et al. **Peer-to-peer - Research at Stanford.** SIGMOD Record, 32(3), setembro/2003

MCGUINNESS, D. **Ontologies Come of Age.** In D. Fensel, J. Hendler, H. Lieberman e W. Wahlster, editors. Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential. MIT Press, 2003.

- MONTEIRO, C. **Sistemas *peer-to-peer* - Estudo e Implementação**. Projeto final - Universidade Federal da Bahia - Instituto de Matemática - Curso de Ciência da Computação - Departamento de Ciência da Computação, Salvador/Bahia – abril/2003
- MOURA, A. **A Web Semântica: Fundamentos, Tecnologias e Tendências**. Tutorial. XVII Simpósio Brasileiro de Banco de Dados. Rio Grande do Sul, outubro/2002.
- MOURA, A., TANAKA, A., VIEIRA, A. **Ferramenta para Extração de Ontologias a Partir de Bancos de Dados Relacionais**. CLEI 2002 - XXVIII Conferência Latino americana de Estudos de Informática - Montevidéu - Uruguai, novembro/2002.
- NAPSTER. **WEB site oficial** - <http://www.napster.com>
- NEJDL, W. et al. **A P2p Networking Infrastructure Based On Rdf**. In 11th International WWW Conference – maio/2002.
- NEJDL, W. et al. **Design issues and challenges for rdf- and schema-based *peer-to-peer* systems**. ACM SIGMOD Records. 2003.
- NISO PRESS. **National Information Standards Organization. Understanding Metadata**. Copyright © 2004 National Information Standards Organization. ISBN: 1-880124-62-9. Disponível em: www.niso.org [Capturado em 12 dez 2004]
- NOY, N., MCGUINNESS, D. **Ontology Development 101: A Guide to Creating Your First Ontology**. Relatório Técnico KSL-01-05, Knowledge Systems Laboratory, março/2001.
- OOI, B., SHU, Y., TAN, K. **Relational Data Sharing in *Peer*-based Data Management Systems**. ACM SIGMOD Record, 32(3), setembro 2003.
- ORAM, A. ***Peer-to-peer: o poder transformador das redes ponto a ponto***. Editora Berkeley, São Paulo – SP – Brasil, 2001
- OWL. **WEB Site oficial** - <http://www.w3.org/2004/OWL/>
- PADRÃO J2EE. **WEB Site** - <http://java.sun.com/blueprints/corej2eepatterns/>
- PORTO, F., MOURA, A. et al. **ROSA: A Data Model and Query Language for e-Learning Objects**. I PGL de Pesquisa em Banco de Dados para E-Learning - PUC, RJ – abril/2003.
- PORTO, F., MOURA, A., FERNANDEZ, A. et al. **ROSA/e-learning - Repositório de**

- Objetos com Acesso Semântico para e-Learning.** Agosto/2003. Disponível em <http://www.des.ime.eb.br/~Rosa/> [capturado em 10 jan 2004]
- PORTO, F., MOURA, A., SILVA, F. **ROSA: a Repository of Objects with Semantic Access for e-Learning.** 8th International Database Engineering & Applications Symposium - IDEAS '04 – Coimbra, Portugal - julho 2004.
- PRESSMAN, R. **Engenharia de software.** 5ª edição. Editora Mcgraw-hill, junho/2002.
- PROTEGE. **WEB site oficial** - <http://protege.stanford.edu>
- RDF. **WEB site oficial** - <http://www.w3.org/RDF/>
- RDFS. **WEB site oficial** - <http://www.w3.org/TR/rdf-schema/>
- RUZZI, M. **Data Integration: state of the art, new issues and research plan.** Dipartimento di Informatica e Sistemistica. outubro/2004. Disponível em: www.dis.uniroma1.it/~dottorato/db/relazioni/
- SCHLOSSER, M. et al. **HyperCuP - Hypercubes, Ontologies and Efficient Search on P2P Networks.** In International Workshop on Agents and *Peer-to-peer* Computing, Itália - julho/2002.
- SCHOLLMEIER, R. **A Definition of peer-to-peer Networking for the Classification of peer-to-peer Architectures and Applications.** IEEE Internet Computing, 2002
- SERVLET. **WEB site** - <http://java.sun.com/products/servlet/>
- SINGH, M. **Peering at Peer-to-peer.** Computing. IEEE Internet Computing, Vol. 5, N^o 1. janeiro/2001
- TAN, K. et al. **A selfconfigurable peer-to-peer system.** In Proceedings of the 18th International Conference on Data Engineering. abril/2002.
- TATARINOV, I. et al. **The Piazza Peer Data Management Project.** ACM SIGMOD Record, Vol. 32 – setembro/2003.
- TATBUL, N. **Data Integration Services.** Disponível em: www.cs.brown.edu/people/koa/biblio.html [Capturado em 12 jan 2005]
- TAVARES, Y. **Um gerenciador de meta-esquemas no suporte a mediadores numa arquitetura para interoperabilidade entre sistemas de bancos de dados.** Dissertação de Mestrado – IME - Instituto Militar de Engenharia, Rio de Janeiro, agosto/1999.

- THOMAS, L., SUCHTER, S., RIFKIN, A. **Developing *Peer-to-peer* Applications on the Internet: the Distributed Editor, SimulEdit.** Dr. Dobb's Journal #281, pp. 76-81, janeiro/1998
- TOLEDO, A. **Portais Corporativos: uma ferramenta estratégica de apoio à gestão do conhecimento.** Tese de Mestrado, UFRJ, outubro/2002.
- TRANTAFILLOU, P. et al. **Towards High Performance Peer-to-peer Content and Resource Sharing Systems.** Conference on Innovative Data Systems Research (CIDR), janeiro/2003
- VALDURIEZ, P., ÖZSU, M. **Princípios de Sistemas de Banco de Dados Distribuídos.** Editora Campus, 2001.
- VDOVJAK, R., HOUBEN, G. **RDF Based Architecture for Semantic Integration of Heterogeneous Information Sources.** In International Workshop on Information Integration on the Web, abril/2001.
- W3C. **WEB site oficial** - <http://www.w3c.org>
- XML. **WEB site oficial** - <http://www.w3c.org/XML>
- XMLSCHEMA. **WEB site oficial** - <http://www.w3.org/XML/Schema>
- XQUERY. **WEB site oficial** - <http://www.w3.org/XML/Query>
- YIN, Z. et al. **Study of Metadata for Advanced Multimedia Learning Objects.** CCECE 2003 – CCGEI 2003, 0-7802-7781-8/03/\$17.00 © 2003 IEEE. Montreal, maio/2003.
- ZHU, Y., WANG, H., HU, Y. **A *Super-peer* Based Lookup in Highly Structured *Peer-to-peer* Networks.** In Proceedings of Parallel and Distributed Computing Systems, 2003.

8. APÊNDICES

8.1 APÊNDICE 1 – DTD da Tabela de Propriedades

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Peer (propriedades+)>
<!ELEMENT propriedades (IPAtual, IPAntigo, IPSuperpeer, assunto, localizacao, nome, tipo, superpeer,
desejaSerSuperpeer, qtdSuperpeers, caracteristicasFisicas+)>
<!ELEMENT IPAtual (#PCDATA)>
<!ELEMENT IPAntigo (#PCDATA)>
<!ELEMENT IPSuperpeer (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
<!ELEMENT localizacao (#PCDATA)>
<!ELEMENT nome (#PCDATA)>
<!ELEMENT tipo (#PCDATA)>
<!ELEMENT superpeer (#PCDATA)>
<!ELEMENT desejaSerSuperpeer (#PCDATA)>
<!ELEMENT qtdSuperpeers (#PCDATA)>
<!ELEMENT caracteristicasFisicas (estabilidade, larguraBanda, poderProcessamento, capacidadeArmazenamento)>
<!ELEMENT estabilidade (#PCDATA)>
<!ELEMENT larguraBanda (#PCDATA)>
<!ELEMENT poderProcessamento (#PCDATA)>
<!ELEMENT capacidadeArmazenamento (#PCDATA)>
```

8.2 APÊNDICE 2 – DTD do Índice de Roteamento SP/P

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Peers (peer+, qtdPeers)>
<!ELEMENT peer (IP, assunto, localizacao, tipo, desejaSerSuperpeer, status, caracteristicasFisicas)*>
<!ELEMENT IP (#PCDATA)>
<!ELEMENT assunto (#PCDATA)>
<!ELEMENT localizacao (#PCDATA)>
<!ELEMENT tipo (#PCDATA)>
<!ELEMENT desejaSerSuperpeer (#PCDATA)>
<!ELEMENT status (#PCDATA)>
<!ELEMENT caracteristicasFisicas (estabilidade, larguraBanda, poderProcessamento, capacidadeArmazenamento)>
<!ELEMENT estabilidade (#PCDATA)>
<!ELEMENT larguraBanda (#PCDATA)>
<!ELEMENT poderProcessamento (#PCDATA)>
<!ELEMENT capacidadeArmazenamento (#PCDATA)>
<!ELEMENT qtdPeers (#PCDATA)>
```

8.3 APÊNDICE 3 – DTD do Índice de Roteamento SP/SP

```
<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT Peer (superpeer+, qtdSuperpeers, qtdSuperpeersMesmoAssunto)>  
<!ELEMENT superpeer (IP, assunto, localizacao, status)>  
<!ELEMENT IP (#PCDATA)>  
<!ELEMENT assunto (#PCDATA)>  
<!ELEMENT localizacao (#PCDATA)>  
<!ELEMENT status (#PCDATA)>  
<!ELEMENT qtdSuperpeers (#PCDATA)>  
<!ELEMENT qtdSuperpeersMesmoAssunto (#PCDATA)>
```

8.4 APÊNDICE 4 – XMLSchema do Vocabulário Controlado Global

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="TermosGenericos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="predicado" type="predicadoType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="predicadoType">
    <xs:sequence>
      <xs:element ref="nome"/>
      <xs:element ref="tipoDePredicado" minOccurs="0"/>
      <xs:element ref="transitivo" minOccurs="0"/>
      <xs:element ref="inversoDe" minOccurs="0"/>
      <xs:element ref="recursivo" minOccurs="0"/>
      <xs:element ref="simetrico" minOccurs="0"/>
      <xs:element ref="use" minOccurs="0"/>
      <xs:element name="equivalente" type="equivalenteType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="equivalenteType">
    <xs:sequence>
      <xs:element ref="nomeEquivalente" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="inversoDe" type="xs:string"/>
  <xs:element name="nome" type="xs:string"/>
  <xs:element name="recursivo" type="xs:boolean"/>
  <xs:element name="tipoDePredicado" type="xs:string"/>
  <xs:element name="transitivo" type="xs:boolean"/>
  <xs:element name="simetrico" type="xs:boolean"/>
  <xs:element name="use" type="xs:string"/>
  <xs:element name="nomeEquivalente" type="xs:string"/>
</xs:schema>
```

8.5 APÊNDICE 5 – Instância em XML do Vocabulário Controlado Global

```
<?xml version="1.0" encoding="UTF-8"?>
<TermosGenericos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Gabriel\IME\Estudo da tese\VC\implementacao\OntoGlobal.xsd">
  <?predicado - É PRÉ-REQUISITO DE?>
    <predicado>
      <nome>é_pré-requisito_de</nome>
      <tipoDePredicado>associação</tipoDePredicado>
      <transitivo>true</transitivo>
      <inversoDe>requer</inversoDe>
      <recursivo>true</recursivo>
      <equivalente>
        <nomeEquivalente>fundamenta</nomeEquivalente>
        <nomeEquivalente>é_base_para</nomeEquivalente>
        <nomeEquivalente>é_condição_para</nomeEquivalente>
      </equivalente>
    </predicado>
    <predicado>
      <nome>fundamenta</nome>
      <use>é_pré-requisito_de</use>
    </predicado>
    <predicado>
      <nome>é_base_para</nome>
      <use>é_pré-requisito_de</use>
    </predicado>
    <predicado>
      <nome>é_condição_para</nome>
      <use>é_pré-requisito_de</use>
    </predicado>
  <?predicado - REQUER?>
    <predicado>
      <nome>requer</nome>
      <tipoDePredicado>associação</tipoDePredicado>
      <transitivo>true</transitivo>
      <inversoDe>é_pré-requisito_de</inversoDe>
      <recursivo>true</recursivo>
      <equivalente>
        <nomeEquivalente>necessita_de</nomeEquivalente>
        <nomeEquivalente>precisa_de</nomeEquivalente>
        <nomeEquivalente>carece_de</nomeEquivalente>
        <nomeEquivalente>demanda</nomeEquivalente>
      </equivalente>
    </predicado>
    <predicado>
      <nome>necessita_de</nome>
      <use>requer</use>
    </predicado>
    <predicado>
      <nome>precisa_de</nome>
      <use>requer</use>
    </predicado>
    <predicado>
      <nome>carece_de</nome>
      <use>requer</use>
    </predicado>
    <predicado>
      <nome>demanda</nome>
      <use>requer</use>
    </predicado>
  <?predicado - IMPLICA?>
    <predicado>
      <nome>implica</nome>
      <tipoDePredicado>associação</tipoDePredicado>
      <transitivo>true</transitivo>
      <inversoDe>é_implicado_por</inversoDe>
      <recursivo>true</recursivo>
    </predicado>
  </TermosGenericos>

```

```

<equivalente>
  <nomeEquivalente>segue</nomeEquivalente>
  <nomeEquivalente>aponta</nomeEquivalente>
  <nomeEquivalente>origina</nomeEquivalente>
  <nomeEquivalente>determina</nomeEquivalente>
  <nomeEquivalente>deriva</nomeEquivalente>
  <nomeEquivalente>resulta</nomeEquivalente>
  <nomeEquivalente>provê</nomeEquivalente>
  <nomeEquivalente>define</nomeEquivalente>
  <nomeEquivalente>especifica</nomeEquivalente>
  <nomeEquivalente>estabelece</nomeEquivalente>
</equivalente>
</predicado>
<predicado>
  <nome>segue</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>aponta</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>origina</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>determina</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>deriva</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>resulta</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>provê</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>define</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>especifica</nome>
  <use>implica</use>
</predicado>
<predicado>
  <nome>estabelece</nome>
  <use>implica</use>
</predicado>
<?predicado - GERA?>
<predicado>
  <nome>gera</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>false</transitivo>
  <inversoDe>é_gerado_por</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>cria</nomeEquivalente>
    <nomeEquivalente>desenvolve</nomeEquivalente>
    <nomeEquivalente>produz</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>cria</nome>
  <use>gera</use>
</predicado>

```

```

<predicado>
  <nome>desenvolve</nome>
  <use>gera</use>
</predicado>
<predicado>
  <nome>produz</nome>
  <use>gera</use>
</predicado>
<?predicado - INFLUENCIA?>
<predicado>
  <nome>influencia</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>true</transitivo>
  <inversoDe>é_influenciado_por</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>influi</nomeEquivalente>
    <nomeEquivalente>inspira</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>influi</nome>
  <use>influencia</use>
</predicado>
<predicado>
  <nome>inspira</nome>
  <use>influencia</use>
</predicado>
<?predicado - PERMITE?>
<predicado>
  <nome>permite</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>true</transitivo>
  <inversoDe>é_permitido_por</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>admite</nomeEquivalente>
    <nomeEquivalente>licencia</nomeEquivalente>
    <nomeEquivalente>aceita</nomeEquivalente>
    <nomeEquivalente>deixa</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>admite</nome>
  <use>permite</use>
</predicado>
<predicado>
  <nome>licencia</nome>
  <use>permite</use>
</predicado>
<predicado>
  <nome>aceita</nome>
  <use>permite</use>
</predicado>
<predicado>
  <nome>deixa</nome>
  <use>permite</use>
</predicado>
<?predicado - SE DISTINGUE DE?>
<predicado>
  <nome>se_distingue_de</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>true</transitivo>
  <inversoDe>equivale_à</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>se_diferencia_de</nomeEquivalente>
    <nomeEquivalente>se_difere_de</nomeEquivalente>
    <nomeEquivalente>é_distinto_de</nomeEquivalente>
  </equivalente>

```



```

</predicado>
<predicado>
  <nome>se_diferencia_de</nome>
  <use>se_distingue_de</use>
</predicado>
<predicado>
  <nome>se_difere_de</nome>
  <use>se_distingue_de</use>
</predicado>
<predicado>
  <nome>é_distinto_de</nome>
  <use>se_distingue_de</use>
</predicado>
<?predicado - USA?>
<predicado>
  <nome>usa</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>false</transitivo>
  <inversoDe>é_usado_por</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>utiliza</nomeEquivalente>
    <nomeEquivalente>usufrui</nomeEquivalente>
    <nomeEquivalente>aproveita</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>utiliza</nome>
  <use>usa</use>
</predicado>
<predicado>
  <nome>usufrui</nome>
  <use>usa</use>
</predicado>
<predicado>
  <nome>aproveita</nome>
  <use>usa</use>
</predicado>
<?predicado - EQUIVALE À?>
<predicado>
  <nome>equivale_à</nome>
  <tipoDePredicado>associação</tipoDePredicado>
  <transitivo>>true</transitivo>
  <inversoDe>se_distingue_de</inversoDe>
  <recursivo>>true</recursivo>
  <equivalente>
    <nomeEquivalente>é_mesma_que</nomeEquivalente>
    <nomeEquivalente>se_assemelha_à</nomeEquivalente>
    <nomeEquivalente>se_compara_à</nomeEquivalente>
    <nomeEquivalente>se_igual_à</nomeEquivalente>
    <nomeEquivalente>se equipara_à</nomeEquivalente>
    <nomeEquivalente>nivela-se_à</nomeEquivalente>
    <nomeEquivalente>é_igual_à</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>é_mesma_que</nome>
  <use>equivale_à</use>
</predicado>
<predicado>
  <nome>se_assemelha_à</nome>
  <use>equivale_à</use>
</predicado>
<predicado>
  <nome>se_compara_à</nome>
  <use>equivale_à</use>
</predicado>
<predicado>
  <nome>se_igual_à</nome>
  <use>equivale_à</use>

```

```

</predicado>
<predicado>
  <nome>se equipara_à</nome>
  <use>equivale_à</use>
</predicado>
<predicado>
  <nome>nivela-se_à</nome>
  <use>equivale_à</use>
</predicado>
<predicado>
  <nome>é_igual_à</nome>
  <use>equivale_à</use>
</predicado>
<?predicado - COMPREENDE?>
<predicado>
  <nome>compreende</nome>
  <tipoDePredicado>agregação</tipoDePredicado>
  <transitivo>true</transitivo>
  <inversoDe>é_parte_de</inversoDe>
  <recursivo>true</recursivo>
  <equivalente>
    <nomeEquivalente>é_constituído_de</nomeEquivalente>
    <nomeEquivalente>é_composto_de</nomeEquivalente>
    <nomeEquivalente>é_formado_de</nomeEquivalente>
    <nomeEquivalente>possui</nomeEquivalente>
    <nomeEquivalente>tem</nomeEquivalente>
    <nomeEquivalente>inclui</nomeEquivalente>
    <nomeEquivalente>enlaça</nomeEquivalente>
    <nomeEquivalente>abarca</nomeEquivalente>
    <nomeEquivalente>envolve</nomeEquivalente>
    <nomeEquivalente>abrange</nomeEquivalente>
  </equivalente>
</predicado>
<predicado>
  <nome>abrange</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>é_constituído_de</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>é_composto_de</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>é_formado_de</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>possui</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>tem</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>inclui</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>enlaça</nome>
  <use>compreende</use>
</predicado>
<predicado>
  <nome>abarca</nome>
  <use>compreende</use>
</predicado>
<predicado>

```

```

    <nome>envolve</nome>
    <use>compreende</use>
</predicado>
<?predicado - É PARTE DE?>
<predicado>
    <nome>é_parte_de</nome>
    <tipoDePredicado>agregação</tipoDePredicado>
    <transitivo>>true</transitivo>
    <inversoDe>compreende</inversoDe>
    <recursivo>>true</recursivo>
    <equivalente>
        <nomeEquivalente>é_abrangido_por</nomeEquivalente>
        <nomeEquivalente>é_membro_de</nomeEquivalente>
        <nomeEquivalente>é_compreendido_por</nomeEquivalente>
        <nomeEquivalente>se_constitui_de</nomeEquivalente>
        <nomeEquivalente>compõe</nomeEquivalente>
        <nomeEquivalente>forma</nomeEquivalente>
        <nomeEquivalente>é_composto_de</nomeEquivalente>
        <nomeEquivalente>é_possuído_por</nomeEquivalente>
        <nomeEquivalente>é_tido_por</nomeEquivalente>
        <nomeEquivalente>é_enlaçado_por</nomeEquivalente>
        <nomeEquivalente>é_abarcado_por</nomeEquivalente>
        <nomeEquivalente>é_envolvido_por</nomeEquivalente>
    </equivalente>
</predicado>
<predicado>
    <nome>é_abrangido_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_membro_de</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_compreendido_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>se_constitui_de</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>compõe</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>forma</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_composto_de</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_possuído_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_tido_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_enlaçado_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>
    <nome>é_abarcado_por</nome>
    <use>é_parte_de</use>
</predicado>
<predicado>

```

```
<nome>é_envolvido_por</nome>  
<use>é_parte_de</use>  
</predicado>  
</TermosGenericos>
```

8.6 APÊNDICE 6 – XMLSchema do Vocabulário Controlado Local

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="TermosEspecificos">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Assunto"/>
        <xs:element name="LO" type="LOType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="LOType">
    <xs:sequence>
      <xs:element ref="nivelAgregacao" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="nome"/>
      <xs:element ref="use" minOccurs="0"/>
      <xs:element name="equivalente" type="equivalenteType" minOccurs="0"/>
      <xs:element name="associado" type="associadoType" minOccurs="0"/>
      <xs:element name="idiomas" type="idiomaType" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="nivelAgregacao">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="curso"/>
        <xs:enumeration value="disciplina"/>
        <xs:enumeration value="tópico"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:complexType name="equivalenteType">
    <xs:sequence>
      <xs:element ref="nomeEquivalente" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="associadoType">
    <xs:sequence>
      <xs:element ref="nomeAssociado" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="idiomaType">
    <xs:sequence>
      <xs:element name="idioma" type="idioma" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="idioma">
    <xs:sequence>
      <xs:element ref="nomeldioma" maxOccurs="unbounded"/>
      <xs:element ref="termoldioma" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Assunto" type="xs:string"/>
  <xs:element name="nome" type="xs:string"/>
  <xs:element name="nomeEquivalente" type="xs:string"/>
  <xs:element name="nomeAssociado" type="xs:string"/>
  <xs:element name="use" type="xs:string"/>
  <xs:element name="nomeldioma" type="xs:string"/>
  <xs:element name="termoldioma" type="xs:string"/>
</xs:schema>
```

8.7 APÊNDICE 7 – Instância em XML do Vocabulário Controlado Local de Informática

OBS.: Devido à sua extensão, este apêndice apresenta somente uma parte do vocabulário controlado local de informática. Porém, o mesmo em sua composição integral se encontra disponível no CD-ROM em anexo.

```
<?xml version="1.0" encoding="UTF-8"?>
<TermosEspecificos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Gabriel\IME\Estudo da tese\VC\implementacao\OntoInformatica.xsd">
  <Assunto>Informática</Assunto>
  <?LO - ACCESS METHODS?>
  <LO>
    <nome>access methods</nome>
    <use>método de acesso </use>
  </LO>
  <?LO - ACTIVE DATABASE?>
  <LO>
    <nome>active database</nome>
    <use>banco de dados ativo</use>
  </LO>
  <?LO - B TREE?>
  <LO>
    <nome>B tree</nome>
    <use>arvore B</use>
  </LO>
  <?LO - B+ TREE?>
  <LO>
    <nome>B+ tree</nome>
    <use>arvore B+</use>
  </LO>
  <?LO - ATTRIBUTE?>
  <LO>
    <nome>attribute</nome>
    <use>atributo</use>
  </LO>
  <?LO - AUTHORIZATION?>
  <LO>
    <nome>authorization</nome>
    <use>autorização</use>
  </LO>
  <?LO - DATABASE SYSTEMS ?>
  <LO>
    <nome>database systems</nome>
    <use>sistemas de banco de dados</use>
  </LO>
  <?LO - ADMDB?>
  <LO>
    <nome>ADMDB</nome>
    <use>administração de banco de dados</use>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
        <termoldioma>DBADM</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - ADMINISTRAÇÃO de BD?>
  <LO>
    <nome>administração de BD</nome>
    <use>administração de banco de dados</use>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
```

```

        <termoldioma>DB administration</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - ADMINISTRAÇÃO DE BANCO DE DADOS?>
  <LO>
    <nivelAgregacao>curso</nivelAgregacao>
    <nivelAgregacao>disciplina</nivelAgregacao>
    <nivelAgregacao>tópico</nivelAgregacao>
    <nome>administração de banco de dados</nome>
    <equivalente>
      <nomeEquivalente>ADMBD</nomeEquivalente>
      <nomeEquivalente>administração de BD</nomeEquivalente>
    </equivalente>
    <associado>
      <nomeAssociado>recuperação de dados</nomeAssociado>
    </associado>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
        <termoldioma>database administration</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - APLICAÇÕES de BD?>
  <LO>
    <nome>aplicações de BD</nome>
    <use>aplicações de banco de dados</use>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
        <termoldioma>DB application</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - APLICAÇÕES DE BANCO DE DADOS?>
  <LO>
    <nivelAgregacao>curso</nivelAgregacao>
    <nivelAgregacao>disciplina</nivelAgregacao>
    <nivelAgregacao>tópico</nivelAgregacao>
    <nome>aplicações de banco de dados</nome>
    <equivalente>
      <nomeEquivalente>aplicações de BD</nomeEquivalente>
    </equivalente>
    <associado>
      <nomeAssociado>gerenciamento de banco de dados</nomeAssociado>
    </associado>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
        <termoldioma>database application</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - ARQUITETURA de BD?>
  <LO>
    <nome>arquitetura de BD</nome>
    <use>arquitetura de banco de dados </use>
    <idiomas>
      <idioma>
        <nomeldioma>Inglês</nomeldioma>
        <termoldioma>BD architecture</termoldioma>
      </idioma>
    </idiomas>
  </LO>
  <?LO - ARQUITETURA DE BANCO DE DADOS?>
  <LO>
    <nivelAgregacao>curso</nivelAgregacao>
    <nivelAgregacao>disciplina</nivelAgregacao>
    <nivelAgregacao>tópico</nivelAgregacao>

```

```

<nome>arquitetura de banco de dados</nome>
<equivalente>
  <nomeEquivalente>arquitetura de BD</nomeEquivalente>
</equivalente>
<idiomas>
  <idioma>
    <nomeldioma>Inglês</nomeldioma>
    <termoldioma>database architecture</termoldioma>
  </idioma>
</idiomas>
</LO>
<?LO - ÁRVORE B?>
<LO>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>arvore B</nome>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>B tree</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - ÁRVORE B+?>
<LO>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>arvore B+</nome>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>B+ tree</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - ATRIBUTO?>
<LO>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>atributo</nome>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>attribute</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - AUTORIZAÇÃO?>
<LO>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>autorização</nome>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>authorization</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - BANCO DE DADOS?>
<LO>
  <nome>banco de dados</nome>
  <use>gerenciamento de banco de dados </use>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>database</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - BD?>
<LO>
  <nome>BD</nome>
  <use>gerenciamento de banco de dados</use>

```



```

<idiomas>
  <idioma>
    <nomeldioma>Inglês</nomeldioma>
    <termoldioma>DB</termoldioma>
  </idioma>
</idiomas>
</LO>
<?LO - GERENCIAMENTO DE BANCO DE DADOS?>
<LO>
  <nivelAgregacao>curso</nivelAgregacao>
  <nivelAgregacao>disciplina</nivelAgregacao>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>gerenciamento de banco de dados</nome>
  <equivalente>
    <nomeEquivalente>BD</nomeEquivalente>
    <nomeEquivalente>banco de dados</nomeEquivalente>
  </equivalente>
  <associado>
    <nomeAssociado>aplicações de banco de dados</nomeAssociado>
    <nomeAssociado>sistemas de banco de dados</nomeAssociado>
  </associado>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>database management</termoldioma>
    </idioma>
    <idioma>
      <nomeldioma>Francês</nomeldioma>
      <termoldioma>base de données</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - BANCO DE DADOS ATIVOS?>
<LO>
  <nivelAgregacao>curso</nivelAgregacao>
  <nivelAgregacao>disciplina</nivelAgregacao>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>banco de dados ativos</nome>
  <equivalente>
    <nomeEquivalente>BD ativos</nomeEquivalente>
    <nomeEquivalente>BD ativos baseados em regras</nomeEquivalente>
  </equivalente>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>active databases</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - BANCO DE DADOS BASEADOS EM REGRAS ?>
<LO>
  <nome>banco de dados baseados em regras</nome>
  <use>banco de dados ativos</use>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>database based on rules</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - SISTEMAS DE BDs?>
<LO>
  <nome>sistemas de BDs</nome>
  <use>sistemas de banco de dados</use>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>DBs systems</termoldioma>
    </idioma>
  </idiomas>

```

```

</LO>
<?LO - SISTEMAS DE BANCO DE DADOS ?>
<LO>
  <nivelAgregacao>curso</nivelAgregacao>
  <nivelAgregacao>disciplina</nivelAgregacao>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>sistemas de banco de dados</nome>
  <equivalente>
    <nomeEquivalente>sistemas de BDs</nomeEquivalente>
    <nomeEquivalente>sistemas de dados</nomeEquivalente>
  </equivalente>
  <associado>
    <nomeAssociado>gerenciamento de banco de dados</nomeAssociado>
  </associado>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>database systems</termoldioma>
    </idioma>
  </idiomas>
</LO>
<?LO - BANCO DE DADOS DISTRIBUÍDOS ?>
<LO>
  <nivelAgregacao>curso</nivelAgregacao>
  <nivelAgregacao>disciplina</nivelAgregacao>
  <nivelAgregacao>tópico</nivelAgregacao>
  <nome>banco de dados distribuídos</nome>
  <equivalente>
    <nomeEquivalente>BDs distribuídos</nomeEquivalente>
  </equivalente>
  <associado>
    <nomeAssociado>integração de banco de dados</nomeAssociado>
  </associado>
  <idiomas>
    <idioma>
      <nomeldioma>Inglês</nomeldioma>
      <termoldioma>distributed database</termoldioma>
    </idioma>
  </idiomas>
</LO>
</TermosEspecificos>
<?fim?>

```

8.8 APÊNDICE 8 – XMLSchema do Vocabulário Controlado de Palavras Chaves

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="Termos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="palavra_chave" type="palavra_chaveType" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="assuntoType">
    <xs:sequence>
      <xs:element ref="nomeAssunto"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="nome" type="xs:string"/>
  <xs:element name="nomeAssunto">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="Informatica"/>
        <xs:enumeration value="Medicina"/>
        <xs:enumeration value="Direito"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:complexType name="palavra_chaveType">
    <xs:sequence>
      <xs:element ref="nome" maxOccurs="unbounded"/>
      <xs:element name="assunto" type="assuntoType"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

8.9 APÊNDICE 9 – Instância em XML do Vocabulário Controlado de Palavras Chaves

OBS.: Devido à sua extensão, este apêndice apresenta somente uma parte do vocabulário controlado local de palavras chaves. Porém, o mesmo em sua composição integral se encontra disponível no CD-ROM em anexo.

```
<?xml version="1.0" encoding="UTF-8"?>
<Termos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Gabriel\IME\Estudo da tese\VC\Implementacao\ontoPC.xsd">
  <palavra_chave>
    <nome>informatica</nome>
    <nome>BD</nome>
    <nome>BDs</nome>
    <nome>banco de dados</nome>
    <nome>banco</nome>
    <nome>dados</nome>
    <nome>rede</nome>
    <nome>rede de computadores</nome>
    <nome>software</nome>
    <nome>programa</nome>
    <nome>linguagem</nome>
    <nome>bio</nome>
    <nome>bioinformatica</nome>
    <nome>data</nome>
    <nome>database</nome>
    <nome>network</nome>
    <nome>gerenciamento de banco de dados</nome>
    <assunto>
      <nomeAssunto>Informatica</nomeAssunto>
    </assunto>
  </palavra_chave>
  <palavra_chave>
    <nome>medicina</nome>
    <nome>biologia</nome>
    <nome>bioinformatica</nome>
    <nome>cirurgia</nome>
    <nome>pediatria</nome>
    <nome>ortopedia</nome>
    <nome>cardiologia</nome>
    <nome>imunologia</nome>
    <nome>cuidados</nome>
    <assunto>
      <nomeAssunto>Medicina</nomeAssunto>
    </assunto>
  </palavra_chave>
  <palavra_chave>
    <nome>direito</nome>
    <nome>leis</nome>
    <nome>normas</nome>
    <nome>direito na informatica</nome>
    <nome>informatica</nome>
    <nome>direito na medicina</nome>
    <nome>medicina</nome>
    <assunto>
      <nomeAssunto>Direito</nomeAssunto>
    </assunto>
  </palavra_chave>
</Termos>
```

8.10 APÊNDICE 10 – DTD da Estrutura de Identificação e Localização dos Vocabulários Controlados

```
<?xml version="1.0" encoding="UTF-8"?>  
<!ELEMENT Vocabularios (vocabulario+)>  
<!ELEMENT vocabulario ( tipo, assunto?, nomeArquivo, localizacao)>  
<!ELEMENT tipo (#PCDATA)>  
<!ELEMENT assunto (#PCDATA)>  
<!ELEMENT nomeArquivo (#PCDATA)>  
<!ELEMENT localizacao (#PCDATA)>
```

8.11 APÊNDICE 11 – Instância em XML da Estrutura de Identificação e Localização dos Vocabulários Controlados

```
<?xml version="1.0" encoding="UTF-8"?>
<Vocabularios>
  <vocabulario>
    <tipo>global</tipo>
    <nomeArquivo>VocGlobal</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/global</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>local</tipo>
    <assunto>Informatica</assunto>
    <nomeArquivo>VocInformatica</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/local</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>local</tipo>
    <assunto>Medicina</assunto>
    <nomeArquivo>VocMedicina</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/local</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>local</tipo>
    <assunto>Direito</assunto>
    <nomeArquivo>VocDireito</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/local</localizacao>
  </vocabulario>
  <vocabulario>
    <tipo>palavra_chaves</tipo>
    <nomeArquivo>VocPC</nomeArquivo>
    <localizacao>c:/portalROSA/SEVC/vocabularios/PC</localizacao>
  </vocabulario>
</Vocabularios>
```

Livros Grátis

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

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

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