Universidade Federal de Pernambuco – UFPE

Centro de Informática – CIn

Pós-graduação em Ciência da Computação

Roteamento Semântico de Consultas em Sistemas Gerenciadores de Dados P2P

Por:

Crishane Azevedo Freire

Exame de Qualificação e Proposta de Tese

Profa. Ana Carolina Salgado
Orientadora

Profa. Damires Yluska de Souza
Co-Orientadora

Recife, abril de 2012

Resumo

Um PDMS (*Peer Data Management System*) ou sistema gerenciador de dados em ambiente P2P tem como objetivo permitir o compartilhamento de informações e o processamento de consultas através dos pontos (fontes de dados) que compõem a rede. Quando um ponto entra no sistema, ele disponibiliza seu esquema que define os dados a serem compartilhados. A partir de seu esquema, este ponto é associado a um conjunto de outros pontos, chamados vizinhos semânticos, que possuem uma afinidade semântica em termos de metadados. Neste escopo, processar uma consulta em um PDMS significa prover capacidades de responder às consultas sobre uma rede arbitrária de pontos com esquemas compartilhados e um conjunto de mapeamentos entre estes pontos que estabelecem uma vizinhança semântica.

O processamento de consultas é reconhecido como o principal serviço que um PDMS pode prover. Uma das etapas desse processo diz respeito ao roteamento da consulta entre os pontos participantes do sistema. Considerando a quantidade e diversidade de pontos existentes, realizar a seleção de pontos de acordo com a semântica da consulta que possam contribuir com resultados relevantes ao interesse do usuário vem se tornando um grande desafio.

Durante o roteamento de uma consulta, algumas situações, como por exemplo, a indisponibilidade de um ponto, a baixa similaridade semântica entre o ponto e a consulta ou até mesmo um histórico de respostas de baixa qualidade podem se tornar critérios que inviabilizem a seleção de um ponto ou a definição de rotas para a consulta. Neste sentido, o objetivo deste trabalho é propor a definição de um processo para o roteamento de consultas baseado em semântica, onde informações contextuais e critérios de qualidade serão utilizados como forma de melhorar o roteamento de consultas em um PDMS.

Abstract

Peer Data Management Systems (PDMS) are P2P applications which allow data sharing and query answering considering a network of data sources (peers). When a peer enters the system, it exports its data schema that will be shared with the other ones. A connected peer is linked to other peers (called semantic neighbors) by means of mappings, according to the existing semantic similarity, in terms of metadata, between them. In this light, query answering in a PDMS means providing capabilities of answer a query on an arbitrary network of peers, considering the set of mappings among these peers which establish a semantic neighborhood.

Query answering has been recognized as the main service that a PDMS can provide. A key issue in this process regards query routing among a subset of peers. Considering the number and diversity of existing peers in the system, it selects a small subset of relevant peers to route (send) a query according to the topology of the considered setting. In fact, it is still a challenge to accomplish such process in such a way that it provides answers closer to the user interest.

During query routing, some situations, such as peers unavailability, low existing semantic similarity between a target peer and a submitted query or even a low-degree history of answers become important criteria that may forbid the peer selection or the estimated routing paths. In this sense, the goal of this work is to propose a process definition to query routing based on semantics, where contextual information and quality criteria will be used as a way to improve such process in a PDMS.

Sumário

Capítulo 1 - Introdução 1

1.1. Motivação 1

1.2. Definição do Problema 2

1.3. Objetivo 3

1.4. Estrutura do Documento 3

Capítulo 2 - Sistemas P2P e o Gerenciamento de Dados 5

2.1. Sistemas P2P 5

2.2. Peer Data Management Systems (PDMS) 9

2.2.1. Piazza 12

2.2.2. Helios 13

2.2.3. System P 14

2.2.4. Ontozilla 15

2.2.5. SEWASIE 17

2.2.6. SUNRISE 19

2.2.7. ESTEEM 21

2.2.8. Quadro Comparativo 23

2.3. Considerações 25

Capítulo 3 - Ontologia, Contexto e Qualidade 26

3.1. Ontologias 26

3.1.1. Tipos de Ontologias 27

3.1.2. Ontologias em Ambientes Distribuídos 28

3.2. Contexto 29

3.2.1. Sistemas Sensíveis ao Contexto 31

3.2.2. Contexto em Ambientes Distribuídos 34

3.3. Qualidade da Informação 35

3.3.1. Dimensões da Qualidade 35

3.3.2. Qualidade em PDMS 36

3.4. Considerações 38

Capítulo 4 - Estratégias de Busca e Roteamento em PDMS 39

4.1. Roteamento em Sistemas P2P 39

4.1.1. Inundação 41

4.1.2. Controle Central 41
4.1.3. Brokering (Agentes Inteligentes) 42
4.1.4. Super-peer/Peer 42
4.1.5. Tabela Hash Distribuida e Árvore de busca distribuída 43
4.1.6. Semantic Overlay Networks (SON) 44
4.2. Roteamento de Consultas em PDMS 44
4.2.1. Knowledge-Super-Peer (KSP) Network 44
4.2.2. Ontozilla 46
4.2.3. SRI - Semantic Routing Index 48
4.2.4. H-Link 50
4.2.5. OntoSum 52
4.2.6. System P 55
4.2.7. ESTEEM 57
4.2.8. GrouPeer 59
4.2.9. Outros trabalhos 61
4.3. Problemas de Roteamento de Consultas em PDMS 61
4.4. Considerações 67
Capítulo 5 - Proposta da Tese 68
5.1. Contexto e Qualidade no Processo de Roteamento de Consultas 68
5.2. O Processo Semântico de Roteamento de Consultas 71
5.3. O Sistema SPEED 74
5.3.1. Expressões de Mapeamentos e Correspondências 75
5.3.2. Processamento da Consulta 77
5.4. Aplicação do Processo de Roteamento de Consultas no SPEED 78
5.4.1. Algoritmo para o Roteamento Semântico da Consulta 78
5.4.2. Exemplificando o Processo 82
5.5. Contribuições 87
5.6. Metodologia 88
5.7. Cronograma 88
5.7.1. Atividades Realizadas 88
5.7.2. Atividades a serem realizadas 89
Referências Bibliográficas 90

Lista de Figuras

Figura 2.2: Exemplo de rede física e *overlay* [Pires 2007] 6
Figura 2.1: Tipos de sistemas [adaptado de Lin 2004]..... 6
Figura 2.3: Mapeamentos no Piazza [Tatarinov *et al*. 2003] 13
Figura 2.4: Arquitetura da ontologia de um ponto 14
Figura 2.5: Componentes de um ponto no System P [Roth *et al.* 2006] 15
Figura 2.6: Arquitetura do Ontozilla [Joung e Chuang 2009]..... 17
Figura 2.7: Arquitetura SEWAISE [SEWAISE 2011] 19
Figura 2.8:Arquitetura interna do ponto no SUNRISE [Sassatelli 2009]..... 21
Figura 2.9: Conhecimento no ESTEEM [Montanelli *et al.* 2010 23
Figura 3.1 Classificação das ontologias proposta por Guarino [1998] 28
Figura 3.2: Definição de Contexto [Vieira *et al*. 2010]..... 31
Figura 3.3: Sistema Tradicional e Sistema Sensível ao Contexto [Vieira *et al.* 2009]..... 32
Figura 4.1: Roteamento da consulta (KSP) [Ismail *et al.* 2011] 46
Figura 4.2: Esquema do Ontozilla [Joung e Chuang 2009]..... 47
Figura 4.3: Exemplo de referência do trabalho [Mandreolli *et al.* 2007b] 49
Figura 4.4: Interface gráfica do SUNRISE [Mandreolli et al 2007b]..... 50
Figura 4.5: Mecanismo de roteamento H-Link..... 52
Figura 4.6: Topologia da Rede [Li e Vuong 2007] 53
Figura 4.7: Tabela de roteamento *inter-cluster* para o ponto N2 [Li e Vuong 2007]. 54
Figura 4.8: Roteamento da consulta utilizando RDV [Li e Vuong 2007] 55
Figura 4.9: Exemplo de CDT [Aiello *et al.* 2007] 58
Figura 4.10: Propagação de uma consulta no GrouPeer [Kantere *et al.* 2009] 60
Figura 4.11: Procedimento de resposta da consulta no ponto [Kantere *et al.* 2009].... 60
Figura 5.1: Arquitetura do SPEED [Pires 2009]..... 75
Figura 5.2: Expressões de Mapeamentos e Correspondências [Souza 2009]..... 75
Figura 5.3: Vizinhança semântica em uma comunidade 83

Lista de Quadros e Tabelas

Quadro 2.1: Quadro Comparativo entre PDMS 24
Quadro 4.1: Comparação entre as estratégias e os problemas de roteamento 65
Quadro 5.1:Conjunto de critérios de qualidade do ponto..... 71
Quadro 5.2: Definição das variáveis do algoritmo de roteamento semântico..... 79
Quadro 5.3: Conceitos pertencentes às ontologias dos pontos de integração 82
Quadro 5.4:Distribuição das atividades..... 89

Tabela 5.1: Grau de similaridade semântica entre os pontos de integração 82
Tabela 5.2: Valores relacionados à Disponibilidade e Reputação dos pontos..... 83
Tabela 5.3: Valores de Similaridade e de degradação da Consulta..... 85

Capítulo 1

Introdução

Este capítulo apresenta a motivação, definição do problema, objetivos, escopo e a estrutura organizacional desta qualificação e proposta de tese.

1.1. Motivação

Nos últimos anos, as questões relacionadas à distribuição, diversidade e compartilhamento de dados têm sido discutidas sob diferentes pontos de vista. Nesse cenário, a demanda por sistemas que utilizem tecnologias que promovam o acesso fácil a dados distribuídos, heterogêneos ou não, com um nível de abstração sobre o gerenciamento e consulta destas informações, vem se tornando um fator de grande importância para o incremento das pesquisas nessas áreas. Os PDMS são um exemplo de sistema que provê o acesso a fontes distribuídas realizando operações de manipulação de dados, a exemplo da transparência em operações de consultas e gerenciamento dos dados [Halevy *et al.* 2006].

Em um PDMS, os pontos no sistema representam fontes de dados que poderão ser usadas para troca de dados, obtenção de respostas às consultas e compartilhamento de informações. Cada ponto compartilha uma vizinhança semântica estabelecida por meio de mapeamentos semânticos entre os pontos (associações entre pontos que possuem similaridade semântica) [Zhao 2006].

Considerando a natureza dinâmica de um PDMS o uso de informações contextuais (conjunto de elementos que caracterizam uma situação) [Dey 2001] e de critérios de qualidade podem permitir que o processo de seleção de pontos e a definição de rotas

possa ser realizada de forma mais eficiente e dinâmica. Em se tratando do contexto, a partir das informações percebidas (de forma explícita ou implícita) no ambiente, é possível refinar o processo de seleção e encaminhamento da consulta (a exemplo da disponibilidade de um ponto durante a consulta). Da mesma forma, critérios de qualidade podem ser usados para escolha do melhor conjunto de pontos que possam contribuir com resultados relevantes à consulta (por exemplo, considerando o critério de reputação do ponto).

É certo que a utilização destes sistemas depende de técnicas eficazes para encontrar e recuperar dados. Cada ponto precisa decidir, independente dos demais pontos que compõem a rede, para qual dos seus vizinhos semânticos a consulta deverá ser encaminhada. Logo, fazer o roteamento das consultas tomando como base o domínio de conhecimento do ponto e a semântica da consulta de maneira eficiente vem se mostrando um tarefa importante e necessária [Ismail *et al.* 2011].

1.2. Definição do Problema

Para realizar uma consulta em um PDMS, o próprio esquema do ponto é usado para formular a consulta que está sendo submetida, e as respostas à essa consulta podem vir de qualquer outro ponto na rede que esteja conectado por meio de um caminho estabelecido por um mapeamento semântico entre os pontos [Tatarinov e Halevy 2004].

Nesse cenário, o roteamento de uma dada consulta tem como objetivo a seleção e o encaminhamento da mesma aos pontos que possuam fontes de dados relevantes a ela. Nesse trabalho, considerando o escopo de um PDMS, o problema de roteamento pode ser definido da seguinte forma:

Dado um PDMS cujos pontos estão semanticamente relacionados por meio de caminhos de mapeamentos semânticos (constituindo conjuntos de pontos vizinhos semânticos), e uma consulta Q submetida em um ponto qualquer do sistema, o problema está em definir um processo de roteamento de consulta que, tomando como base aspectos semânticos e de qualidade, possa realizar a escolha dos pontos mais adequados à semântica da consulta e identificar as melhores rotas para o seu encaminhamento. Com isso, espera-se conseguir os melhores resultados possíveis,

minimizando o tráfego de informações na rede e reduzindo o tempo de espera da resposta à consulta do usuário.

1.3. Objetivo

A proposta deste trabalho contempla as seguintes atividades:

- Fazer levantamento do estado da arte em sistemas PDMS e suas estratégias de roteamento de consultas;
- Identificar elementos contextuais e critérios de qualidade que possam contribuir no processo de roteamento;
- Especificar e implementar um modelo de representação e uso dos elementos contextuais e dos critérios de qualidade escolhidos;
- Propor um processo semântico para o roteamento de consultas em PDMS
- Implementar o processo semântico de roteamento de consultas
- Realizar experimentos para avaliação do processo proposto.

1.4. Estrutura do Documento

Este documento esta organizado em cinco capítulos, os quais contemplam a qualificação e a proposta de tese, sendo o primeiro de introdução e o último capítulo dedicado à referida proposta.

O Capitulo 2 apresenta o estado da arte sobre sistemas P2P e o gerenciamento de dados nesses ambientes. São apresentadas características, topologias e sistemas existentes na literatura.

O Capitulo 3 apresenta algumas abordagens utilizadas para melhoria de processos em PDMS que estão diretamente relacionadas ao tema desse trabalho. Essas abordagens dizem respeito ao uso de ontologias, contexto e qualidade.

O Capitulo 4 apresenta um levantamento do estado da arte de soluções de roteamento utilizadas em PDMS e dos problemas encontrados nessa área. São apresentadas estratégias de busca e roteamento utilizadas em sistemas P2P e PDMS.

A proposta de tese encontra-se descrita no Capitulo 5. Nela são apresentados os aspectos contextuais e de qualidade a serem utilizados na proposta. É feita uma descrição geral do processo de roteamento semântico proposto, assim como, uma apresentação do sistema SPEED (*Semantic Peer-to-Peer Data Management System*) que será utilizado como ambiente de aplicação para esse trabalho. Em seguida, o processo geral de roteamento é apresentado e sua aplicação no SPEED é descrita. Por fim, são apresentadas as contribuições esperadas nesse trabalho, a metodologia a ser utilizada e um cronograma das atividades realizadas e previstas nesta tese.

Capítulo 2

Sistemas P2P e o Gerenciamento de Dados

O volume e diversidade de dados espalhados pela *internet* e pelas *intranets* têm despertado o interesse quanto à possibilidade de compartilhamento e consulta dessas informações. Os sistemas P2P vêm tornando-se cada vez mais populares nesses ambientes. Sua natureza autônoma, distribuída e descentralizada fez com que o seu uso se tornasse cada vez mais atrativo. Muitos dos sistemas P2P existentes são destinados ao compartilhamento de arquivos, vídeos e músicas. Entretanto, a busca por sistemas que ofereçam soluções ao compartilhamento de dados estruturados e semi-estruturados com consultas mais complexas, baseadas em conteúdo, fez surgir o interesse pelos sistemas de gerenciamento de dados em ambientes P2P (PDMS). Nesse capítulo, é feita uma apresentação geral sobre sistemas P2P, PDMS e suas principais características.

2.1. Sistemas P2P

O termo P2P refere-se ao paradigma da computação distribuída em que cada participante em uma rede de computadores, aqui chamado de ponto (*peer*), compartilha recursos e serviços de uma maneira direta e descentralizada [Bernstein *et al*. 2002]. Cada ponto, nesses sistemas, atua como cliente e servidor simultaneamente. Em um sistema P2P, qualquer ponto que requisita ou fornece algum serviço comunica-se de forma direta sem a intervenção de algum mediador ou coordenador central [Androutsellis-Theotokis e Spinellis, 2004]. Essa ausência de controle central representa uma das características dos sistemas P2P que os diferencia

dos sistemas cliente-servidor, onde apenas o servidor é capaz de realizar a mediação entre os clientes que desejam compartilhar recursos, como mostra a Figura 2.1.

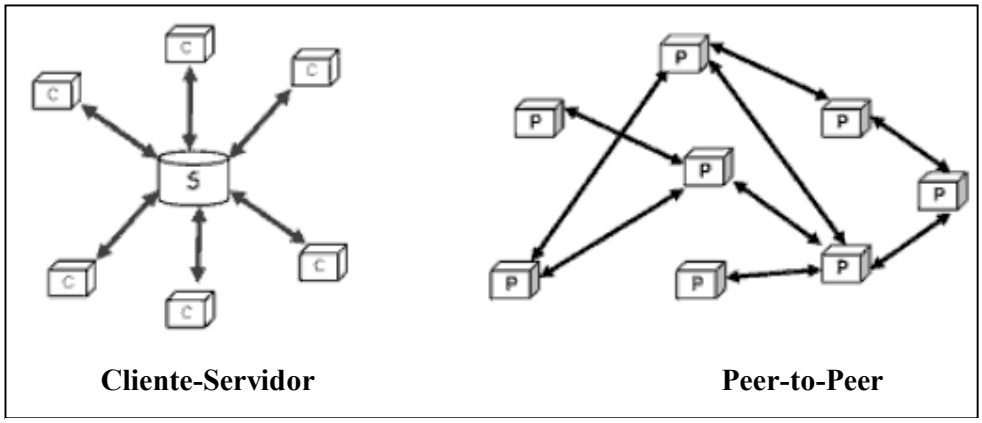


Figura 2.1: Tipos de sistemas [adaptado de Lin 2004]

Os sistemas P2P caracterizam-se não só pela capacidade de descentralização e compartilhamento de recursos. Outras características podem ser alcançadas nesses sistemas [Sassatelli 2009]: **escalabilidade**, crescimento e distribuição de pontos na rede; **autonomia**, onde pontos podem decidir sobre quais recursos compartilhar; **auto-organização**, isto é, a rede se organiza em função da entrada e saída de pontos; **segurança e transparência administrativa**, entre outras.

Em um sistema P2P os pontos comunicam-se por meio de uma rede virtual (lógica), denominada *overlay*, que funciona sobre uma rede física. Logo, a topologia da rede define a forma de comunicação entre os pontos [Doval e O'Mahony 2003, Schlosser *et al.* 2003]. Na Figura 2.2 podemos observar a definição de duas redes *overlays*, formadas por diferentes interconexões de pontos, sobre a infra-estrutura de uma rede física.

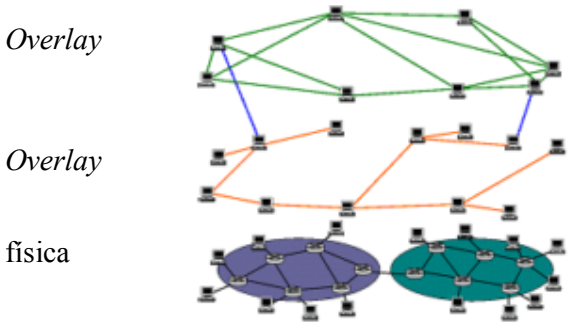


Figura 2.2: Exemplo de rede física e *overlay* [Pires 2007]

Um dos objetivos de um sistema P2P é prover a busca por dados, onde buscar dados significa geralmente encontrar arquivos ou partes deles [Sung *et al.* 2005], através de consultas, geralmente por palavras-chave. Para encontrar os pontos que podem responder à consulta, mecanismos de propagação e encaminhamento da consulta são necessários e suas implementações podem variar conforme a topologia do sistema.

Existem diversas classificações para as topologias P2P [Lv *et al*. 2002, Fiorano 2003, Schollmeier 2001]. De acordo com sua estrutura, as redes P2P podem ser assim classificadas [Theotokis e Spinellis, 2004]:

- Não-estruturada – a rede não possui um servidor central, assim como o controle sobre a topologia e localização de recursos. Um mecanismo de inundação é utilizado para propagar as consultas na rede. Quando um ponto recebe uma consulta, ele a encaminha para seus vizinhos até que o resultado seja alcançado. A propagação da consulta é limitada por um mecanismo de *time-to-live* (TTL). O TTL é um valor indicativo do número máximo de saltos para encaminhamento da consulta entre os pontos vizinhos. Quando o TTL atinge zero (TTL=0), o processo de propagação da consulta é interrompido mesmo que nenhum resultado tenha sido alcançado.
- Estruturada – a localização dos recursos é feita de forma mais eficiente. A identificação e endereço dos recursos normalmente são mantidos em tabelas de roteamento distribuído (*Distributed Hash Tables - DHT*). Sendo assim, consultas baseadas em palavra-chave podem ser realizadas. Uma desvantagem dessa topologia é a dificuldade em manter a estrutura de índices necessária ao roteamento das consultas.
- Fracamente estruturada – essa categoria está entre as topologias estruturadas e não-estruturadas. É caracterizada por não especificar completamente a localização do recurso armazenado, embora essa localização possa ser obtida por meio de seus algoritmos de roteamento.

Quanto ao nível de centralização, as redes P2P podem também ser classificadas da seguinte forma [Theotokis e Spinellis, 2004]:

- Pura – não possui controle central, todos os pontos executam as mesmas tarefas, tanto como cliente quanto como servidor. Por essa razão cada ponto também é chamado de "*servents*", ou seja, *SERvers+cliENTS*. Cada ponto é responsável por manter informações sobre seus recursos podendo, ao receber uma consulta, respondê-la ou encaminhá-la para outros pontos vizinhos com os quais esteja conectado diretamente. Como não existe um controle central dos pontos na rede, as consultas são propagadas por inundação até que o arquivo desejado seja encontrado ou até que a busca seja limitada por um mecanismo de TTL.
- Híbrida – existe um servidor central que mantém a descrição do que cada ponto participante da rede dispõe. No caso de consultas, o papel do servidor é identificar e retornar o endereço do ponto que possui o arquivo ao ponto solicitante. Os sistemas híbridos trabalham com conceito de indexação, típico da arquitetura cliente/servidor e com o conceito de comunicação direta entre pontos para compartilhamento de recursos, típico do paradigma P2P.
- *Super-peer* ou parcialmente centralizada – alguns pontos chamados *super-peers (SP)*, gerenciam uma determinada quantidade de pontos conectados a ele formando um agrupamento de pontos. Esses *super-peers* agem como servidores centrais mantendo a descrição e o endereço dos arquivos armazenados em cada ponto do seu agrupamento. Os *super-peers* que compõem a rede são interligados entre si. Cada ponto submete consultas para seu *super-peer*, e cada *super-peer* pode propagar a consulta para outros *super-peers* vizinhos utilizando-se também de um mecanismo de TTL. Quando um *super-peer* recebe uma consulta, ele verifica os pontos que formam o seu *cluster*. Caso alguma resposta possa ser obtida, o endereço do ponto que contém a resposta é retornado ao *super-peer* que encaminhou a consulta e assim sucessivamente até que o ponto inicial receba o endereço do ponto que contém o recurso. Nesse caso, os pontos conectam entre si e o ponto solicitante recebe o recurso solicitado.

Na próxima seção será feita uma apresentação sobre características dos PDMS e de alguns sistemas encontrados na literatura.

2.2. Peer Data Management Systems (PDMS)

Os *Peer Data Management Systems* (PDMS) foram introduzidos como uma extensão natural dos bancos de dados distribuídos em um ambiente de sistemas P2P [Halevy *et al.* 2003]. Em um PDMS, pontos são fontes de dados heterogêneas, cujo conteúdo está representado por um esquema local associado ao seu domínio de interesse (por exemplo, educação). Por esta razão, os PDMS são também conhecidos como sistemas P2P baseados em esquema (*schema-based P2P systems*) [Sassatelli 2009].

Enquanto existem algumas similaridades entre PDMS e bancos de dados distribuídos, existem também importantes diferenças [Risson e Moors 2006]. Bancos de dados distribuídos não tratam com o crescimento e o dinamismo dos sistemas P2P, onde cada ponto pode sair ou retornar ao sistema a qualquer tempo. Enquanto estes bancos fornecem consultas completas, cada ponto em um PDMS oferece apenas resposta probabilisticamente completa, ou seja, os pontos podem dispor de informações parciais ou até mesmo terem saído da rede o que ocasionaria parcialidade (incompletude) da resposta. A localização dos dados em um banco de dados distribuído é geralmente conhecida enquanto que em um sistema P2P, a localização dos dados é feita por meio de rotas de consultas estabelecidas entre pontos vizinhos até que a fonte de dados seja localizada [Risson e Moors 2006].

Em um PDMS, não existe um esquema global que represente o domínio integrado de conhecimento entre os pontos participantes do sistema. Cada ponto representa uma fonte de dados e exporta seu esquema de dados completo, ou parte dele. Chamado de esquema exportado, esse esquema representa os dados que poderão ser compartilhados com os outros pontos no sistema [Souza 2009]. Mapeamentos são definidos para estabelecer associações entre os elementos que compõem os esquemas dos pontos. Esses mapeamentos são de grande importância para o processamento de consultas no PDMS. Uma consulta submetida em um determinado ponto tanto pode ser respondida pelo próprio ponto quanto propagada aos demais pontos vizinhos. Os mapeamentos permitem que a consulta, durante a propagação, seja reformulada de forma compatível com os diferentes esquemas exportados existentes no sistema. No cenário de um PDMS, as consultas submetidas em um determinado ponto são

normalmente semanticamente mais ricas e mais complexas do que aquelas realizadas por palavras-chave em ambientes P2P [Sassatelli 2009].

O uso de um modelo comum para representação dos esquemas exportados pelos pontos facilita a definição dos mapeamentos e, consequentemente, melhora o processamento da consulta. Alguns exemplos incluem o uso de modelos baseados em esquemas relacionais, XML[1], RDF[2] e ontologias OWL[3] [Euzenat e Shvaiko, 2007]. Entre todos esses modelos, as ontologias têm sido consideradas como uma boa forma para especificar o conteúdo das fontes de dados e, consequentemente, como forma de promover a interoperabilidade em PDMS [Xiao, 2006; Li e Vuong, 2007; Pires, 2009; Montanelli *et al*. 2010].

Em se tratando de questões relacionadas ao gerenciamento de dados em PDMS, o trabalho de Sassatelli [2009] destaca alguns aspectos a serem considerados para prover o sistema com funcionalidades que garantam a integração de dados estruturados, semi-estruturados e semanticamente mais ricos:

- Representação dos dados – o modelo a ser adotado para representação do esquema exportado pelos pontos deve ser capaz de fornecer expressividade semântica dos recursos a serem compartilhados, assim como favorecer o uso de uma linguagem de consulta.
- Compartilhamento de dados – permitir compartilhamento de informações entre os pontos mesmo que as fontes de dados disponíveis possuam diferentes esquemas e representações. O processo de integração, adotado, precisa ser descentralizado e flexível de forma a atender aos requisitos dos PDMS.
- Processamento de Consulta – uma consulta submetida em um determinado ponto pode ser respondida localmente ou ser propagada para os vizinhos diretos daquele ponto. Este é um processo complexo e diversas questões precisam ser abordadas quanto à sua eficiência e eficácia. Em ambientes de compartilhamento de dados distribuídos, uma camada de mediação decompõe as consultas formuladas a partir de um esquema global e as envia para as

[1] http://www.w3.org/xml/
[2] http://www.w3.org/TR/rdf-primer
[3] http://www.w3.org/TR/owl-ref/

fontes de dados distribuídas no sistema. Outra forma de consulta é feita integrando e materializando visões de dados a partir de outras fontes de dados. Entretanto, no cenário P2P, um ponto central que integre resultados pode tornar-se um ponto de falha. Nesse sentido, por questões de crescimento do ambiente e para preservar a autonomia dos pontos, o processamento de consultas deve ser descentralizado e independente de um controle central. Considerando que o número de pontos participantes e a disponibilidade dos dados durante uma consulta podem sofrer variações a qualquer tempo, é necessário que a execução do plano de consulta possa ser feita de forma dinâmica.

- Recuperação dos dados – a consulta submetida em um determinado ponto pode precisar obter os dados que estão localizados nos diversos pontos distribuídos no ambiente. Localizar fontes de dados está entre os principais problemas em ambientes P2P. Em alguns sistemas, índices centralizados ou distribuídos são utilizados para armazenar a localização dos dados. Durante as consultas, os índices são consultados para que o ponto, onde o dado desejado está armazenado, possa ser identificado.

Apesar de menos dinâmico do que nos sistemas P2P tradicionais dedicados ao compartilhamento de arquivos, o processo de conectividade em PDMS é considerado de extrema importância, especialmente porque os pontos podem compartilhar dados de domínios de conhecimento distintos [Pires 2007]. Em outras palavras, a tarefa de formação de **comunidades semânticas** está relacionada com a capacidade de agregar dinamicamente os pontos com interesses semelhantes em organizações estruturadas com o objetivo de: i) reduzir a carga da rede, devido ao acúmulo de solicitações por pontos independentes e ii) definir mecanismos de comunicação eficazes para grupos de pontos que compartilham conhecimento de um mesmo domínio de interesse (ou seja, pontos participantes de uma mesma comunidade) [Montanelli 2007]. Cada comunidade deve tratar de um domínio de interesse (também chamado de tópico ou assunto) específico, por exemplo, educação, saúde, geografia. Um interesse pode ser formalizado através de palavras-chave ou ontologias, e deve ser genérico o suficiente para incluir pontos significativos. Entretanto, é fundamental evitar tópicos muito

genéricos que possam ocasionar ineficiência no processamento de consultas. Em um PDMS, como cada ponto representa uma organização individual, pontos diferentes podem exportar diferentes esquemas. Portanto, o agrupamento de pontos deve ser considerado um processo não supervisionado, e a formação de comunidades semânticas deve ser um resultado alcançado a partir da associação semântica entre os pontos [Pires 2009; Montanelli *et al.* 2010; Kantere *et al.* 2008; Lodi *et al.* 2008].

Vários trabalhos têm se dedicado ao desenvolvimento de arquiteturas que permitem o gerenciamento de dados em ambientes P2P. O objetivo desta seção é apresentar de forma geral este cenário, descrevendo algumas das propostas mais relevantes ao tema desse trabalho. Nesse sentido, aspectos relacionados à arquitetura do sistema e suas principais características serão considerados.

2.2.1. **Piazza**

O Piazza representa o exemplo mais conhecido de PDMS [Tatarinov *et al.* 2003; Tatarinov e Halevy. 2004]. Ele surgiu como um projeto de pesquisa da Universidade de Washington com o objetivo de proporcionar a "mediação semântica" entre milhares de pontos com diferentes esquemas em um ambiente P2P. Os esquemas exportados pelos pontos participantes são representados em XML e RDF e as consultas formuladas em XQuery. Sua arquitetura é pura e descentralizada, sem a presença de um esquema global único para integração dos pontos no sistema. Pontos interessados em compartilhar dados devem estabelecer mapeamentos entre seus esquemas. Existem dois tipos de mapeamentos entre esquemas definidos no Piazza (ver Figura 2.3): **descrição do ponto (*Peer Description*)**, que se refere ao mapeamento que relaciona os esquemas exportados por dois ou mais pontos; **descrição de armazenamento (*Storage Description*)**, que se refere ao mapeamento que relaciona o esquema do dado armazenado pelo ponto com seu esquema exportado.

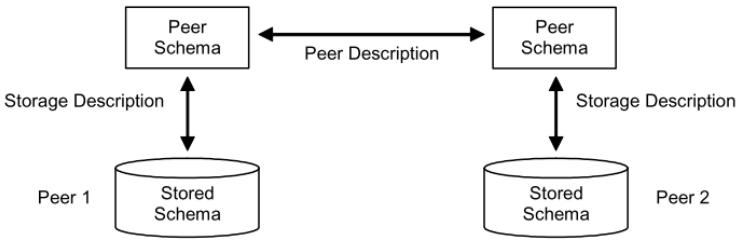


Figura 2.3: Mapeamentos no Piazza [Tatarinov *et al*. 2003]

No Piazza, as consultas são formuladas de acordo com o esquema de preferência do usuário e são encaminhadas para os pontos vizinhos por meio dos mapeamentos semânticos e de um algoritmo de reformulação da consulta. O objetivo é que todos os pontos disponíveis no sistema possam participar do processamento da consulta. Entretanto, um ponto poderá ser excluído desse processamento caso seja identificado que ele não pode contribuir com resultados relevantes à consulta. Para realizar essa seleção, o Piazza oferece um índice centralizado que armazena um sumário dos dados armazenados pelo ponto. Cada ponto participante do sistema é responsável por disponibilizar esse sumário e fazer atualizações periódicas.

2.2.2. **Helios**

O Helios (*Helios EvoLving Interaction-based Ontology knowledge Sharing*) foi desenvolvido pelo ISLab Group da Universidade de Milão como um ambiente de referência para evolução e compartilhamento de conhecimento em sistemas de rede com arquitetura pura [Castano *et al*. 2006]. No Helios, ontologias fornecem uma representação semanticamente rica dos recursos compartilhados. Técnicas de negociação consensual, para auto-organização do sistema e formação de comunidades semânticas são definidas para melhorar o compartilhamento de recursos e promover a colaboração semântica entre pontos que tenham interesse similar.

Cada ponto, nesse ambiente, provê sua própria ontologia e usa um *matcher*, um programa para associação semântica, que identifica a afinidade semântica entre conceitos armazenados nas ontologias dos diversos pontos, denominado H-Match [Castano *et al*. 2004].

A ontologia do ponto (armazenada em um repositório de metadados organizados conforme o H-Model [Castano *et al*. 2004]) é organizada em uma arquitetura de duas camadas (ver Figura 2.4): **conhecimento do conteúdo (*Content Knowledge Layer*)**, que corresponde à descrição dos recursos que o ponto deseja compartilhar na rede; **conhecimento da rede (*Network Knowledge Layer*)**, que se refere ao conhecimento que o ponto tem dos seus vizinhos semânticos (*Neighbor peer*). Na Figura 2.4, o ponto B e D são vizinhos semânticos do ponto que está sendo representado. A camada de conhecimento da rede guarda o conjunto de pontos vizinhos que estão conectados com os conceitos (*Concept*) na camada de conteúdo por meio de links de afinidade semântica (*Affinity Link*).

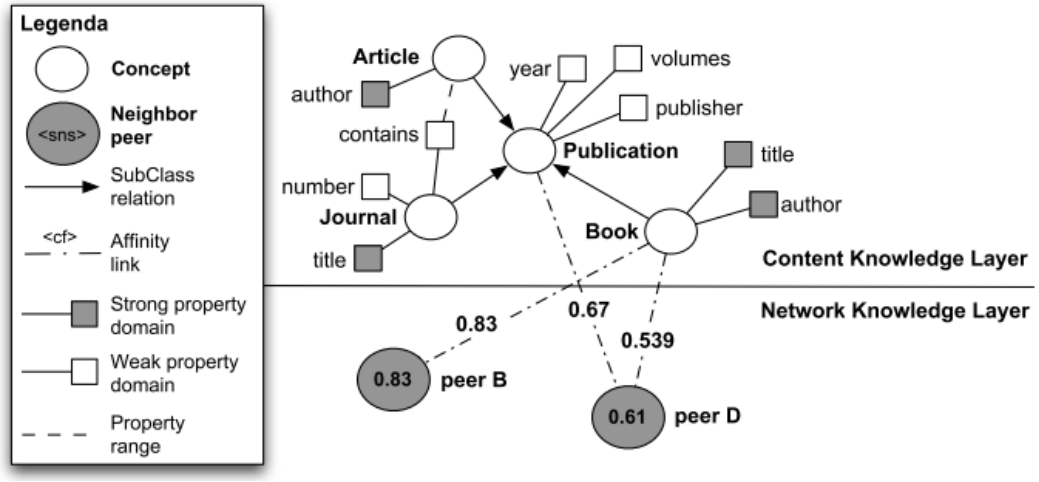


Figura 2.4: Arquitetura da ontologia de um ponto

2.2.3. **System P**

O System P é um PDMS que integra fontes de dados relacionais em uma rede P2P pura e foi desenvolvido para fins experimentais de um projeto que investiga estratégias de consulta baseadas em um modelo de custo e benefício [Roth e Naumann 2005]. O sistema possui um gerador que utiliza um esquema de referência para criar um número aleatório de pontos, fontes locais, esquemas heterogêneos e mapeamentos [Roth e Naumman 2007; Roth *et al.* 2006]. Os dados contidos nos pontos também são gerados automaticamente. No System P, cada ponto (Figura 2.5), consiste de um conjunto de fontes locais conectadas ao esquema do ponto (*Peer schema*) por meio de mapeamentos locais (*local mappings*), e mapeamentos de

pontos (*peer mappings*) entre os esquemas dos diversos pontos participantes do sistema. Qualquer SGBD relacional pode participar do sistema como uma fonte local. O ambiente de comunicação entre os pontos distribuídos é baseado em JXTA[4].

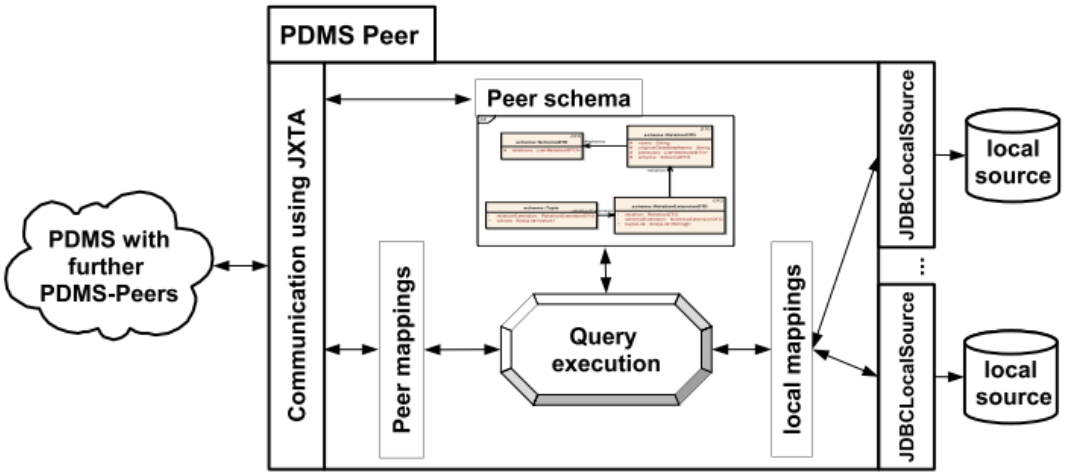


Figura 2.5: Componentes de um ponto no System P [Roth *et al.* 2006]

Os mapeamentos do sistema descrevem as correspondências com as fontes locais e entre esquemas de pontos vizinhos. Os mapeamentos, assim como as consultas, são formulados usando regras datalog [Halevy *et al.* 2003].

Para o processamento de consultas, o sistema utiliza estratégias de poda dos planos de consulta tomando como base um modelo de completude [Roth e Naumann 2005] que estima a alta cardinalidade e riqueza dos resultados, ou seja, conjunto de resultados não nulos.

2.2.4. **Ontozilla**

O Ontozilla combina ontologias e tecnologias P2P com o objetivo de melhorar o desempenho do processo de busca de informações e facilitar o processo de integração e interoperabilidade [Joung e Chuang 2009]. As ontologias são utilizadas para representar informações semânticas no sistema, tais como, relacionamentos, consultas, descrições e anotações de recursos. Nesse sistema, pontos que compõem a rede com mesmo domínio de conhecimento são agrupados em *clusters*, e os relacionamentos entre eles são modelados de acordo com o conhecimento que eles

[4] http://jxta.kenai.com/

representam. Conforme a nomenclatura adotada pelo Ontozilla, pontos com interesses idênticos ou similares são agrupados em SIG (*Special Interest Groups*). Em cada SIG, os pontos empregam algum sistema que hierarquicamente classifica os pontos dentro de classes de interesses. Cada classe representa um *cluster* de pontos que compartilham os mesmos conceitos. A arquitetura da rede é *super-peer* e cada agrupamento (*cluster*) é representado como uma árvore hierárquica (*cluster tree*) [Joung e Chuang 2009].

No Ontozilla, a arquitetura de cada ponto é composta de três camadas [ver Figura 2.6]: mensagem (*message layer)*, comunicação (*communication layer)* e conteúdo (*content layer).* A camada de mensagem contém um módulo de resolução de mensagens responsável por gerenciar as mensagens no sistema. A camada de comunicação contém dois módulos: manutenção da rede (*network maintenance module)* e o módulo de roteamento (*routing module*). O módulo de rede é responsável por manter a topologia da rede, atualizando os relacionamentos entre pontos e mantendo o balanceamento de carga. Todas essas atividades dedicam atenção à tabela de roteamento (*routing table)* e tabela de cache (*cache table*) do sistema. A tabela de roteamento contém ligações semânticas que são registros de relacionamentos entre o ponto e seus vizinhos. A tabela de *cache* armazena os mapeamentos entre classes e o endereço do ponto com o objetivo de melhorar o balanceamento de carga e acelerar o roteamento da consulta. A camada de conteúdo contém um repositório de descrição (*description repository*), que armazena descrições de classes e SIG; e, um repositório de recursos (*resource repository*), responsável por armazenar bases de conhecimento, anotações de recursos e descrição de serviços.

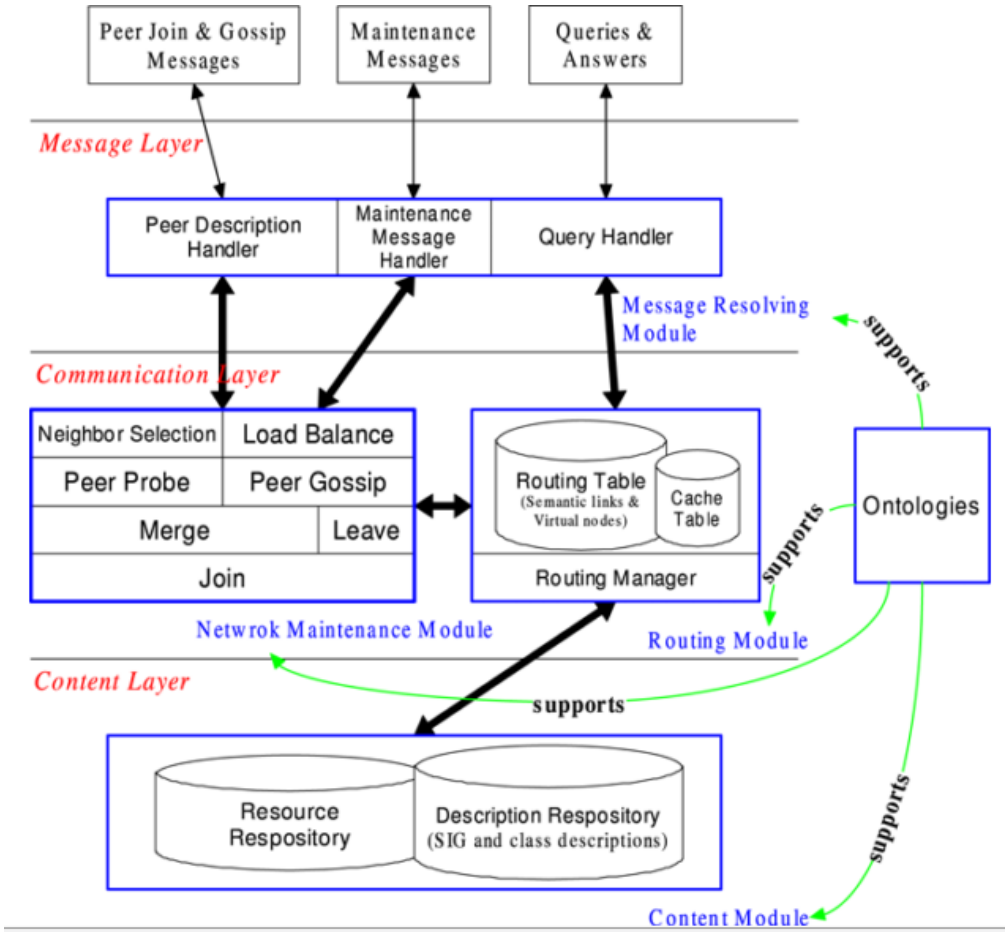


Figura 2.6: Arquitetura do Ontozilla [Joung e Chuang 2009]

Toda essa organização resulta em uma topologia de rede semi-estruturada onde cada ponto descreve seus recursos (*peer descriptions)* utilizando uma linguagem de ontologia (OWL[5]). Periodicamente, a partir dos *peer descriptions* os pontos atualizam suas ligações semânticas, compatibilizando o sistema com a evolução da rede de forma dinâmica.

2.2.5. **SEWASIE**

O projeto SEWASIE (*SEmantic Web and AgentS in Integrated Economies*) é baseado na tecnologia de sistema multi-agentes [Beneventano *et al.* 2007]. Sua arquitetura prevê duas camadas: uma de nível local, onde os pontos (desenvolvidos como agentes mediadores) mantêm uma visão integrada de suas fontes locais; outra,

[5] http://www.w3.org/TR/owl-ref/

no nível de rede, onde agentes (*brokering agents*) mantêm os mapeamentos entre os diferentes pontos [Beneventano *et al.* 2007]. A Figura 2.7 oferece uma ilustração macro da arquitetura do SEWASIE com os seguintes componentes [SEWASIE 2011]:

- SEWASIE *Information Nodes* (*SINodes*) – são sistemas baseados em mediadores que fornecem uma visão virtual das fontes de informação gerenciadas pelo *SINode*. Os *SINodes* utilizam tradutores (*wrapper*) para extrair dados e metadados das fontes. Um construtor de ontologias (*Ontology Builder*) é utilizado para criar uma ontologia integrada de todos os esquemas das fontes, denominado *Global Virtual View* (GVV). Essas ontologias serão futuramente integradas ao *Brokering Agent* para estabelecer uma ligação entre os *SINodes* e a interface do usuário.

- *Brokering Agents* (BA) – responsáveis por integrar as GVV de diferentes SINodes em uma *Brokering Agent Ontology* (BA *Ontology*) e fazer o roteamento das consultas na rede. A ontologia do BA é usada para guiar os *Query Agents* para os *SINodes* que conteêm dados relevantes à consulta. É possível que no sistema existam vários BA, cada um representando um domínio específico. Mapeamentos entre diferentes BA podem ser criados e dessa forma fazer com que novos encaminhamentos de consultas possam ser realizados. Logo, o SEWASIE se apresenta com uma arquitetura *super-peer*, onde os BA agem como *super-peers*, e os SINodes como *peers* de dados.

- *Query Agent* (QA) – após receber a consulta (expressa conforme a ontologia do BA), reescreve a mesma em função do GVV do *SINode* (identificado pelo BA) e faz o envio.

- *User Agent* (UA) – o usuário interage com uma interface *Web* gerenciada pelo UA que disponibiliza uma lista das ontologias dos BA disponíveis para que a consulta possa ser formulada. O UA instancia um QA que traduz, por meio dos BA, a consulta do usuário em um conjunto de consultas que deverão ser realizadas no nível dos *SINodes*. No retorno da consulta, o QA integra as respostas e retorna os dados no formato XML para o UA [Beneventano *et al.* 2007].

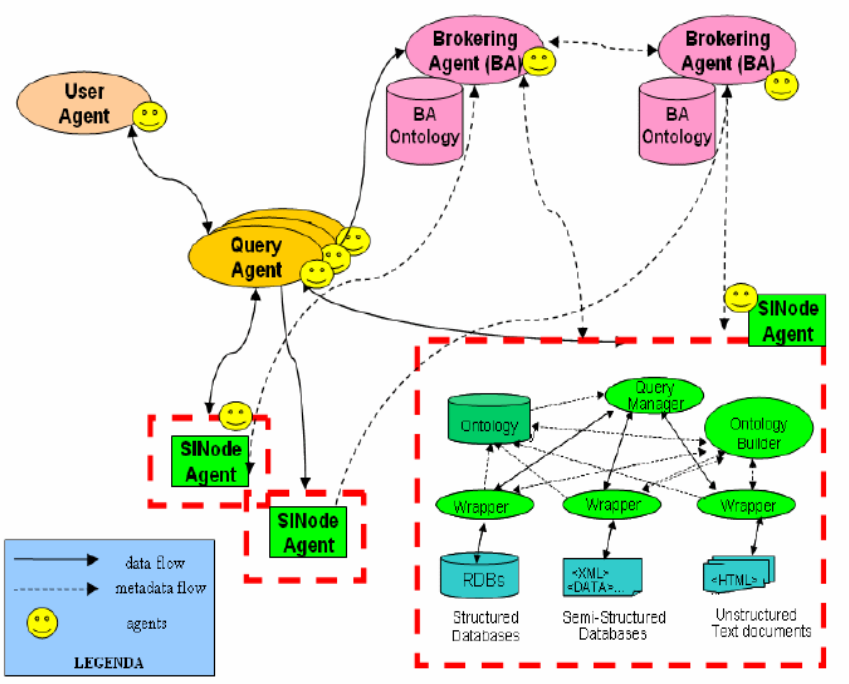


Figura 2.7: Arquitetura SEWAISE [SEWAISE 2011]

No SEWASIE a representação dos esquemas das fontes de dados é feita em ODLI3 [Bergamaschi *et al.* 2001]. A ODLI3 é uma extensão da linguagem orientada a objetos ODL[6] (*Object Definition Language*), que é uma linguagem de especificação utilizada para definir as interfaces de objetos que estão em conformidade com o modelo de objeto definido pela ODMG (*Object Data Management Group*) para extração de informações.

2.2.6. **SUNRISE**

O SUNRISE (*System for Unified Network Routing, Indexing and Semantic Exploration*) é um PDMS que integra fontes de dados heterogêneas utilizando uma rede semântica baseada no agrupamento de pontos de dados com mesmo domínio de interesse [Mandreoli *et al.* 2007b].

A arquitetura do SUNRISE foi desenvolvida para trabalhar, independente do modelo de dados, com qualquer forma de representação dos esquemas e formulação de consultas. Entretanto, as particularidades do modelo de dados utilizado precisam ser consideradas no desenvolvimento dos módulos de software do sistema. A rede é

[6] http://www.odmg.org

organizada em SON de forma que cada ponto tem na sua vizinhança pontos semanticamente relacionados. A similaridade entre os pontos é calculada por uma função que considera a proximidade entre os conceitos exportados pelos pontos.

A organização da rede utiliza algoritmos de agrupamentos. Quando um ponto entra na rede, ele procura no *Access Point Structure* (APS) a rede na qual ele possa fazer parte. O APS é uma estrutura centralizada que mantém uma representação sumarizada de cada SON disponível na rede [Penzo *et al.* 2008].

Além da representação dos esquemas, cada ponto é responsável por manter os mapeamentos semânticos que definem as conexões entre seus vizinhos. A Figura 2.8 mostra os componentes pertencentes à arquitetura de um ponto semântico no SUNRISE: *Semantic Clustering Index* (SCI), usado na fase de construção da rede; *Semantic Routing Index* (SRI), utilizado durante a fase de roteamento de consultas; *Semantic Mapping,* responsável por definir a conexão entre um ponto e seus vizinhos e *XML Schema*. Além das estruturas de dados são módulos de software: *Annotation*, tem como objetivo reduzir a ambiguidade das palavras fornecendo significado às palavras utilizadas no esquema do ponto; *Matching*, responsável por gerar a correspondência semântica entre pontos definido um grau de similaridade semântica entre dois conceitos; *Network Organization*, responsável pelas ações que cada ponto executa durante a criação da rede, sendo formado por dois sub-módulos (*SCI Management e SON Management*); *Query Routing*, responsável por manter as estruturas para encaminhamento das consultas entre os pontos sendo formado por dois sub-módulos (*SRI Management* e *Routing Management*); *Query Reformulation,* utiliza os mapeamentos semânticos gerados pelo Matching para reformular a consulta entre os pontos durante o roteamento.

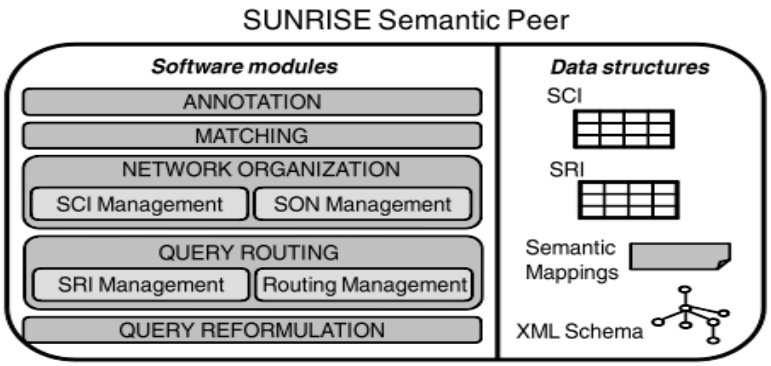


Figura 2.8:Arquitetura interna do ponto no SUNRISE [Sassatelli 2009]

Como apresentado, o SUNRISE oferece funções específicas nos seguintes estágios de um PDMS:

- Construção da rede – técnicas e estruturas de índices para selecionar a melhor SON para que o ponto possa ser integrado, e dentro dela estabelecer o mapeamento com os seus vizinhos semânticos. Oferece também um conjunto de protocolos e algoritmos para gerenciar a evolução e infra-estrutura da rede de maneira incremental.
- Exploração da rede – algoritmos de roteamento e mecanismos de indexação para permitir à seleção de pontos relevantes à consulta.

2.2.7. **ESTEEM**

O ESTEEM (*Emergent Semantics and cooperaTion in multi-knowledgE EnvironMents*) tem como objetivo oferecer uma plataforma baseada em comunidade semântica para integração de dados e serviços em uma rede P2P não estruturada [Montanelli *et al.* 2010].

Uma comunidade ESTEEM surge a partir da formação de um grupo de pontos em torno de um interesse comum, declarado na forma de um manifesto baseado em ontologia (em OWL). A comunidade semântica é definida como uma rede *overlay,* construída no topo de uma infra-estrutura básica chamada *overlay* global (*global overlay*). A ontologia de dados ou serviços do ponto (*peer/service ontology*) é o núcleo do conhecimento de um ponto e provê uma descrição semanticamente rica dos dados/serviços que estão disponíveis para compartilhamento. Essa ontologia é

explorada para avaliar se o ponto dispõe dos recursos solicitados por outro ponto em resposta a uma consulta, assim como para verificar de qual comunidade semântica o ponto poderá fazer parte.

De acordo com [Bianchini *et al.* 2009], as descrições de serviços definidas pela ontologia de serviço são obtidas por meio de uma ontologia de mensagem de serviço *(Service Message Ontology - SMO)* , cujos conceitos são usados para fornecer semântica aos parâmetros de serviço de entrada/saída, e uma ontologia de funcionalidade de serviços (*Service Functionality Ontology - SFO*), cujos conceitos são utilizados para adicionar semântica a funcionalidades de serviços (operações). Outras funcionalidades do ponto incluem o contexto corrente (*current context*), responsável por descrever o perfil do ponto, seus interesses, situação e coordenadas espacial/temporal; e perfil de confiança e qualidade dos dados (*data quality and trust profile*), relacionado ao cálculo de métricas de qualidade dos dados exportados para compartilhamento com os outros pontos. As métricas utilizadas referem-se àquelas normalmente definidas para qualidade de dados: completude, consistência de formato, acurácia e consistência interna [Batini e Scannapieco, 2006].

O manifesto, associado à comunidade, permite ao ponto mover-se do seu espaço de conhecimento (*peer knowledge space*) para o espaço de conhecimento coletivo (*collective knowledge space*), onde ele atua como membro de várias comunidades armazenando parte do conhecimento de cada grupo. A Figura 2.9 representa o espaço de conhecimento dos pontos no ESTEEM.

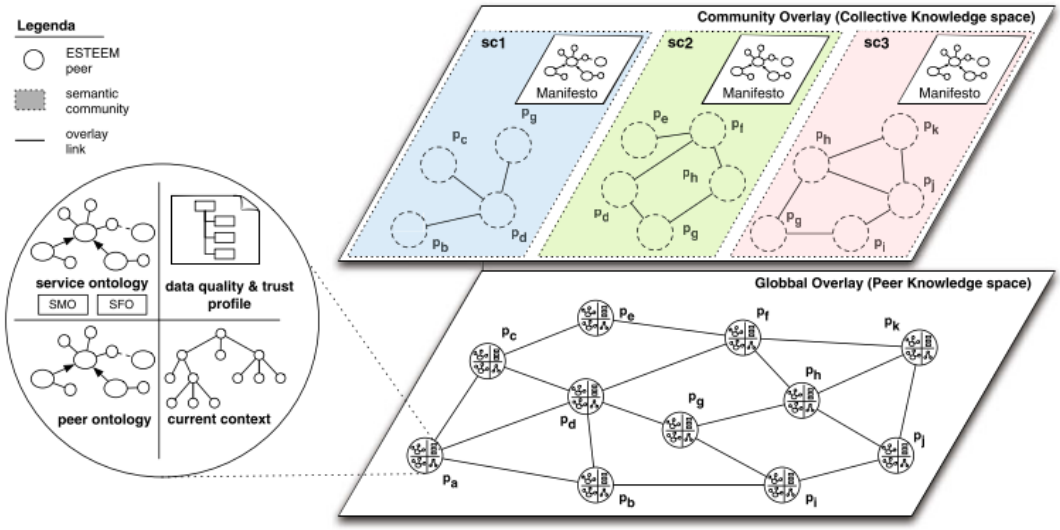


Figura 2.9: Conhecimento no ESTEEM [Montanelli *et al.* 2010

2.2.8. **Quadro Comparativo**

Nesse capítulo os principais PDMS relacionados ao tema desse trabalho foram apresentados. O Quadro 2.1 apresenta um comparativo das principais características observadas em cada PDMS. O significado de cada característica é apresentado a seguir:

- PDMS – nome do PDMS;
- Arquitetura – qual a arquitetura utilizada pelo PDMS;
- Representação dos esquemas – formato de representação dos esquemas das fontes de dados pertencentes a cada ponto no PDMS
- Conectividade – como ocorre a ligação entre as fontes de dados dos diversos pontos do PDMS
- Outras características – informações adicionais e relevantes ao comparativo entre os trabalhos.

Quadro 2.1: Quadro Comparativo entre PDMS

PDMS	Arquitetura	Representação dos esquemas	Conectividade	Outras características
Piazza	Pura	XML, RDF	Mapeamento entre esquemas	Linguagem de consulta XQuery
Helios	Pura	H-Model (RDF, OWL e UML)	Mapeamento entre ontologias; comunidades semânticas;	Utiliza técnicas de negociação consensual para formação de comunidades
System P	Pura	Relacional	Mapeamento local; mapeamento entre pontos	Desenvolvido para fins experimentais de um projeto que investiga estratégias de consulta baseadas em um modelo de custo e benefício
Ontozilla	Super-Peer	OWL	Ligações semânticas; SIG (*Special Interest Groups*) como *cluster* semântico	Agrupamento é representado como uma árvore hierárquica (*cluster tree*)
SEWASIE	Super-Peer	ODLI3	Global virtual view (GVV) – ontologia integrada	Tecnologia de sistema multi-agentes
SUNRISE	Pura	Sem restrição (ontologia, relacional, XML)	Mapeamentos semânticos; SON; *cluster* semântico	*Access Point Structure* (APS) *Semantic Clustering Index* (SCI)
ESTEEM	Pura	Ontologia	Mapeamento entre ontologias; comunidade semântica	Gerenciamento do contexto e da qualidade/confiança.

2.3. Considerações

Esse capítulo apresentou os sistemas P2P e suas principais características. Em seguida, analisou alguns PDMS considerando aspectos como arquitetura, forma de representação dos esquemas e conectividade. Em análise a alguns trabalhos, foi observado que a formação de agrupamentos semânticos em um PDMS pode ser utilizada como alternativa para tornar mais eficiente o processamento da consulta reduzindo o espaço de busca a ser consultado durante o roteamento da consulta. Outra característica importante é o uso de contexto, critérios de qualidade e proveniência de dados. Tais aspectos permitem a geração de soluções semanticamente mais ricas e tornam o sistema e seus processos mais adaptáveis conforme os resultados obtidos durante as interações entre os pontos.

O próximo capítulo apresenta conceitos sobre ontologia, contexto e qualidade e sua aplicação em PDMS.

Capítulo 3

Ontologia, Contexto e Qualidade

Nesse capítulo é feita uma apresentação de algumas abordagens utilizadas em PDMS que estão diretamente relacionadas ao tema desse trabalho. Essas abordagens dizem respeito ao uso de ontologias, contexto e qualidade.

3.1. Ontologias

Ontologia é um tema de pesquisa bastante conhecido em várias áreas tais como engenharia do conhecimento, processamento de linguagem natural, sistemas de informação cooperativos e integração de dados. Elas provêem uma compreensão comum e compartilhada de um conhecimento que pode ser comunicado entre pessoas e sistemas distribuídos e heterogêneos. Uma ontologia fornece uma conceituação (isto é, uma meta-informação) que descreve a semântica dos dados, e é utilizada de forma a facilitar o reuso e o compartilhamento de conhecimento [Fensel 2001].

Uma ontologia pode ser tratada a partir de um conjunto de componentes assim definidos [Gruber 1993, Gruber 1995, Noy e McGuinness 2001, Maedche 2002]:

- Conceito (classe) – é a representação relacionada a um conjunto conceitual de elementos semelhantes. Exemplo: Veículo pode ser a representação de um conceito relacionado aos sub-conceitos carro e motocicleta.
- Propriedade – é definida como um relacionamento entre dois elementos. O primeiro elemento deve ser um conceito que indica o domínio da relação. O segundo deve ser um conceito com quem se tem a relação

(*range*). Por exemplo, *condutor* estabelece um relacionamento entre os conceitos *Pessoa* e *veículo*.

- Instância – um objeto é uma instância de um conceito se ele é membro do conjunto indicado pelo conceito. Por exemplo, Ana é uma instância de Pessoa.
- Axiomas – são sentenças que são sempre verdadeiras. São usadas para definir restrições de domínio ou gerar novos fatos sobre a ontologia. Por exemplo, um axioma seria afirmar que toda pessoa tem uma mãe.

Sendo teorias formais sobre certo domínio de discurso, as ontologias, em geral, requerem uma linguagem lógica formal para expressá-las. Algumas linguagens podem ser citadas, entre elas: RDF[7]/RDFS[8] (*Resource Description Framework/RDF Schema*), OWL[9] (*Ontology Web Language*) e XOL[10] (*Ontology Exchange Language*).

3.1.1. **Tipos de Ontologias**

De acordo com o seu nível de generalidade as ontologias podem ser classificadas da seguinte forma [Guarino 1998]:

- Ontologia *Top-level* – descreve conceitos gerais como espaço, tempo, objetos, eventos, ações, entre outros. Tais conceitos são independentes de um domínio ou problema em particular e poderiam ser reutilizados na criação de novas ontologias.
- Ontologia de Domínio – descreve o vocabulário relacionado a um domínio genérico, como medicina ou automobilismo, pela especialização dos conceitos presentes na ontologia *top-level*.
- Ontologia de Tarefa – descreve o vocabulário relacionado a uma tarefa ou atividade genérica, como diagnóstico ou vendas, pela especialização dos conceitos presentes na ontologia *top-level*.

[7] http://www.w3.org/TR/rdf-primer
[8] http://www.w3.org/TR/rdf-schema
[9] http://www.w3.org/2004/OWL/
[10] http://www.ai.sri.com/pkarp/xol/

- Ontologia de aplicação – descreve conceitos dependentes de um domínio ou tarefa em particular. Estes conceitos frequentemente correspondem a papéis desempenhados por uma entidade de domínio durante a execução de certa atividade.

A Figura 3.1 mostra a relação de dependência entre os tipos de ontologias definidos. As setas representam relacionamentos de especialização.

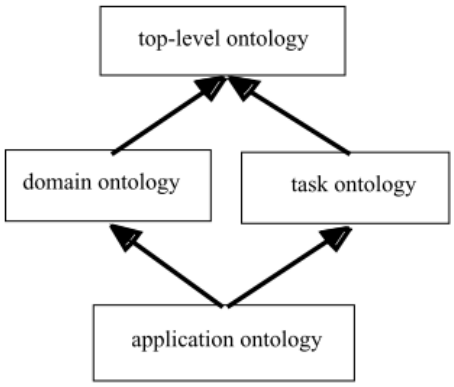


Figura 3.1 Classificação das ontologias proposta por Guarino [1998]

3.1.2. **Ontologias em Ambientes Distribuídos**

Com as ontologias é possível dar significado aos dados favorecendo a integração e consequentemente o gerenciamento de dados em ambientes distribuídos [Pires *et al.* 2011]. Sendo assim, nesses ambientes, as ontologias têm sido usadas para alguns propósitos, incluindo [Xiao 2006]: (i) **representação de metadados**: cada fonte de dados é representada por uma ontologia local; (ii) **conceituação global**: uma ontologia global pode ser usada para prover uma visão conceitual sobre os esquemas heterogêneos das fontes; (iii) **suporte às consultas de alto nível**: dada uma ontologia global, os usuários podem formular consultas sem conhecimento específico das diferentes fontes de dados.

Quando múltiplas ontologias são simultaneamente usadas, elas podem sofrer diferentes tipos de heterogeneidades. Essas heterogeneidades podem ser classificadas da seguinte forma [Euzenat e Shvaiko 2007]: Sintática (quando as ontologias estão representadas em diferentes formalismos, por exemplo, OWL e F-Logic,); Terminológica (quando ocorrem variações em nomes usados para se referir às

mesmas entidades em ontologias diferentes, por exemplo, *Paper vs Artigo);* Conceitual (heterogeneidade semântica, diferenças encontradas na modelagem de um mesmo domínio de interesse) e Semiótica (diferenças relacionadas a como as entidades são interpretadas pelas pessoas).

Para tratar com a heterogeneidade, são utilizados processos de associação ou correspondências entre os elementos das ontologias. *Matching* é o processo de localização de relacionamentos ou correspondências entre elementos de diferentes ontologias, e alinhamento (*alignment*) é o conjunto de correspondências entre duas ou mais ontologias, ou seja, a saída do processo de *matching* [Souza 2009]. As pesquisas em ambientes distribuídos também têm considerado o uso de ontologias como uma forma de dar ao sistema um domínio de referência. Nesse caso, uma ontologia de domínio pode ser usada como referência semântica ou conhecimento de base (*background knowledge*) para melhorar os processos de *matching* [Souza *et al.* 2011]. Outras pesquisas incluem: o desenvolvimento de OPDMS (*Ontology Peer Data Manegement System*), onde ontologias são usadas nas questões relacionadas ao gerenciamento de dados, como representação de esquemas, agrupamentos e mapeamentos [Pires 2009, Kantere *et al* 2009]; processos de *matching*, usado para estabelecer as associações entre ontologias [Mazak *et al.* 2010; Pires *et al.* 2009] e no processamento da consulta [Souza *et al.* 2009, Sassatelli 2009].

É certo que os especialistas de um dado domínio são as pessoas mais indicadas para definir os conceitos-chave de uma determinada área de conhecimento. Mas, mesmo assim, deve-se considerar que a semântica de um termo pode variar de um contexto para outro, de um lugar para outro e mesmo de uma pessoa para outra [Cantele *et al.* 2004]

3.2. Contexto

O conceito de contexto tem sido objeto de investigação há vários anos em algumas comunidades científicas, como Linguística e Psicologia Cognitiva. Na comunidade de Ciência da Computação, os estudos sobre o tema são mais recentes, porém pode-se observar importantes contribuições para o seu entendimento e formalização, particularmente em trabalhos da área de Inteligência Artificial [Calvi *et al.* 2005].

Várias definições podem ser encontradas na literatura, entretanto a mais referenciada é aquela apresentada por Dey [2000]:

> *"Contexto é qualquer informação que pode ser usada para caracterizar uma situação de uma entidade. Uma entidade é uma pessoa, um lugar, ou um objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação."*

É importante destacar que a definição de Dey é bastante ampla, e, dependendo da área de aplicação e do uso de contexto, as definições podem se tornar mais específicas, uma vez que o contexto está sempre relacionado a um domínio, um foco ou um ponto de vista [Vieira 2008].

Segundo Brézillon [1999], há um consenso sobre o fato de que o contexto é indissociável da sua utilização. O contexto é considerado como um espaço de conhecimento compartilhado, que é explorado repetidamente pelos participantes na interação. Além disso, o contexto está sempre relacionado a um foco, onde um foco pode ser um passo na execução de uma tarefa ou uma decisão. É ele, o foco, que permite determinar quais elementos deveriam ser instanciados e usados para compor o contexto.

Vieira *et al.* [2010] define contexto com base nas definições de Dey e Brézillon, e faz uma clara distinção entre os conceitos de contexto e elemento contextual (CE). **Elemento contextual** é qualquer pedaço da informação que permite ao agente caracterizar entidades em um domínio. O **contexto** é o conjunto de elementos contextuais instanciados necessários à tarefa que está sendo executada na interação entre um agente e uma aplicação. Um agente pode ser um indivíduo ou um software. Mais do que isso, os elementos que compõem um contexto têm um relacionamento relevante com a tarefa que o agente está executando. Pode-se observar que o elemento contextual é estável e pode ser definido em tempo de projeto, enquanto que o contexto é dinâmico e deve ser construído em tempo de execução, quando uma interação ocorre. A Figura 3.2 ilustra essa definição de contexto.

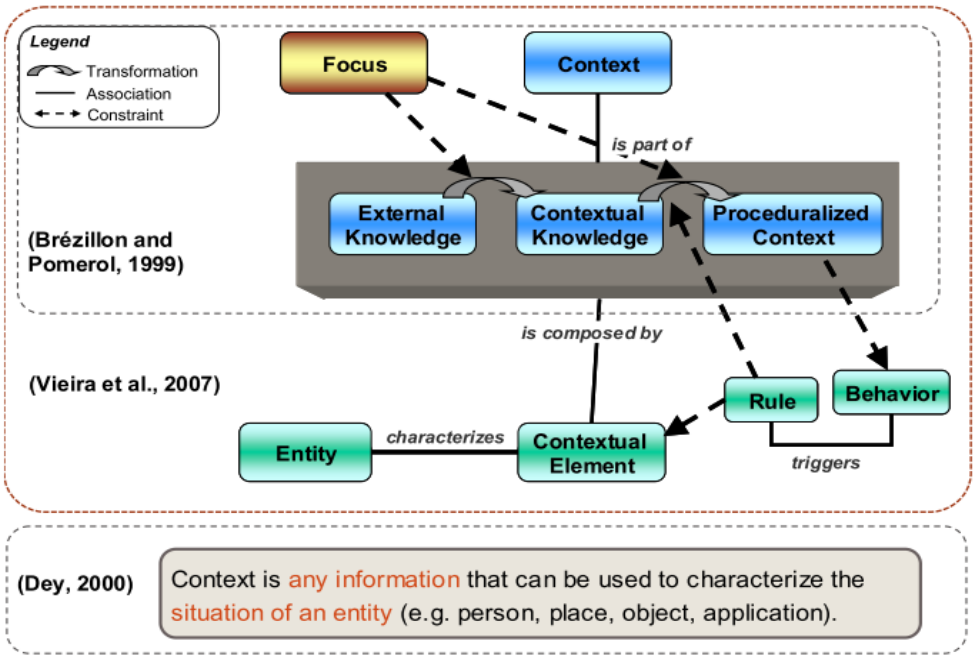


Figura 3.2: Definição de Contexto [Vieira *et al*. 2010]

Apesar das várias definições, é possível observar que [Vieira 2008]: o contexto só existe quando relacionado à outra entidade (tarefa, agente ou interação); o contexto é um conjunto de itens (conceitos, regras e proposições) associados a uma entidade; e um item é considerado parte de um contexto somente se ele é usado como suporte ao problema que está sendo analisado pelo sistema.

Contexto é o conhecimento que permite definir o que é ou não relevante em uma dada situação, tornando-se uma importante ferramenta de apoio à comunicação entre os sistemas e seus usuários. Compreendendo o contexto, o sistema pode se adaptar e mudar seu comportamento, ou seja, sua seqüência de ações, o estilo das interações e o tipo da informação fornecida aos usuários, em circunstâncias diversas [Salgado *et al.* 2009].

3.2.1. **Sistemas Sensíveis ao Contexto**

Sistemas sensíveis ao contexto (*context-aware systems)* são capazes de adaptar suas operações para o contexto corrente, sem a intervenção explícita do usuário aumentando assim sua utilidade e eficácia [Baldauf *et al.* 2007]. Dey e Abowd (2001) definiram os sistemas sensíveis ao contexto como aqueles que "*utilizam o contexto*

para fornecer informações e/ou serviços relevantes para o usuário, onde relevância depende da tarefa do usuário".

A Figura 3.3 mostra dois tipos de sistemas: os tradicionais e aqueles sensíveis ao contexto. Os sistemas tradicionais são aqueles que agem levando em consideração apenas as solicitações e informações fornecidas explicitamente pelos usuários (Figura 3.3a). Os sistemas sensíveis ao contexto consideram além daquelas informações explícitas fornecidas pelos usuários, as armazenadas em uma base de conhecimento contextual, as inferidas por meio de raciocínio, e aquelas percebidas a partir do ambiente, conforme mostra a Figura 3.3b. Com base nessas informações contextuais, o sistema tem condições de identificar e otimizar fluxos de interação, recuperar históricos, determinar ações e adaptações. Como resultado, é possível prover serviços mais relevantes e direcionados ao que o usuário necessita naquele momento [Vieira *et al.* 2009].

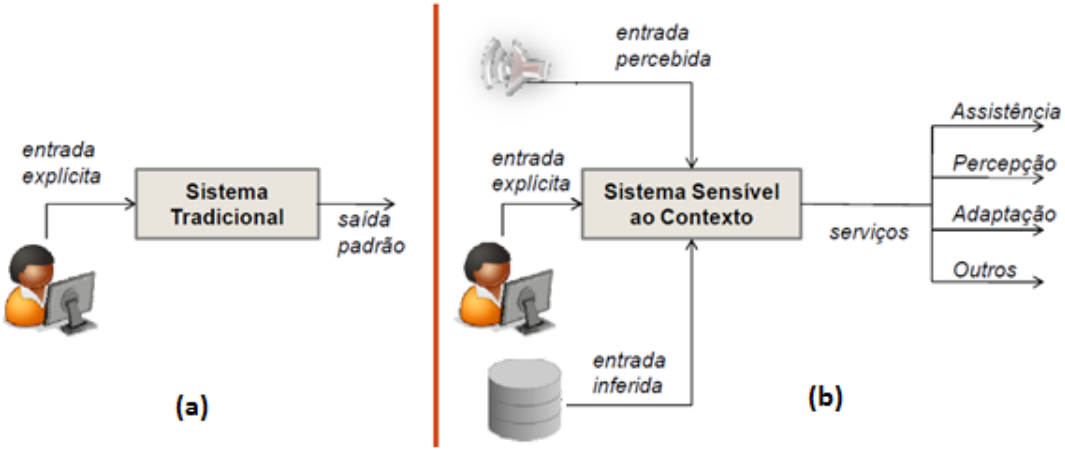


Figura 3.3: Sistema Tradicional e Sistema Sensível ao Contexto [Vieira *et al.* 2009]

O desenvolvimento de sistemas e de processos mais inteligentes e com maior riqueza semântica está diretamente relacionado ao uso do contexto. Serviços mais relevantes podem ser disponibilizados de acordo com as funcionalidades de assistência ao usuário, adaptação, percepção do contexto entre outros (Figura 3.3b).

Apesar das diferentes visões sobre contexto, uma questão é comum à comunidade. Para desenvolver sistemas sensíveis ao contexto é necessário considerar questões relacionadas ao gerenciamento da informação contextual. Em Souza [2009] é

apresentado um conjunto de tarefas que normalmente são realizadas por um serviço de gerenciamento de contexto:

- Aquisição (*Acquisition*) – a qualidade dos serviços sensíveis ao contexto depende da qualidade da informação coletada das fontes de contexto. Dados de contexto podem ser capturados de quatro maneiras: (i) a partir de sensores físicos (GPS, microfones e outros); (ii) sensores lógicos (agentes inteligentes ou serviços capazes de coletar contexto); (iii) entrada explícita (preferências definidas pelo usuário); (iv) fontes estáticas (perfis de usuários ou informações armazenadas em um banco de dados).
- Representação (*Representation*) – embora não exista um modelo padrão para representar a informação contextual, é senso comum que a representação do contexto vem se tornando uma necessidade na maioria dos domínios de aplicação. Algumas pesquisas têm trabalhado com um número considerável de técnicas de representação de contexto, tais como grafos de contexto [Brézillon 2003], ontologias [Souza *et al.* 2008] e Mapas de tópicos [Power 2003].
- Raciocínio (*Reasoning*) – assim que um dado de contexto é obtido ele deve ser interpretado para permitir que o sistema reaja ou se adapte de maneira adequada. Dessa forma, mecanismos de raciocínio podem ser usados para processar a informação contextual, isto é, deduzindo o contexto implícito de alto-nível a partir do contexto explícito de baixo nível.
- Armazenamento (*Storage*) – quase todas as partes de uma informação contextual supostamente estão armazenadas de forma que possam ser recuperadas posteriormente. Entretanto, em algumas situações, se o raciocínio do sistema depende do valor corrente de sensores, então o contexto não precisa ser armazenado.
- Compartilhamento (*Sharing*) – questões relacionadas à privacidade e segurança devem ser consideradas em um ambiente de muitos usuários e aplicações onde ocorra a necessidade de compartilhamento do contexto.
- Reusabilidade (*Reusability*) – contexto pode ser reusado no futuro. Isso evita novos processos de aquisição do mesmo.

- Evolução (*Evolution*) – é necessário que algum mecanismo possa ser usado para planejar mudanças do contexto quando as aplicações sofrerem alterações.
- Responsabilidade (*Accountability*) – permitir que os usuários possam tomar decisões baseadas no contexto. O serviço de gerenciamento do contexto deverá prover *feedback* ao usuário e manter o controle em casos de conflito de interesses [Bellotti e Edwards 2001].

3.2.2. **Contexto em Ambientes Distribuídos**

Em ambientes distribuídos e dinâmicos, é possível fazer uso do contexto para vários processos, incluindo: no processamento da consulta, no processo de integração de esquemas e na identificação de recursos para compartilhamento disponíveis no ambiente.

No trabalho de Souza [2009], para cada consulta submetida no ambiente, todo o processo de reformulação da consulta pode ser adaptado ou enriquecido por meio do uso de contexto. A reformulação da consulta pode sofrer mudanças dependendo do contexto que envolve a consulta (sua semântica), dos pontos (disponibilidade), das correspondências semânticas (seus diferentes tipos), existentes entre os esquemas dos pontos, e dos usuários (preferências). Logo, três tipos de contexto são analisados: (i) o contexto dos usuários, representado pelo conjunto de preferências definidas por eles; (ii) o contexto da consulta, adquirido a partir da identificação de sua semântica (incluindo conceitos, propriedades e operadores) e seu modo de reformulação de consulta.

O contexto também pode ser usado para melhorar o processo de integração dos dados como, por exemplo, na reconciliação de esquemas [Belian 2008]. No processo de reconciliação de esquemas, normalmente algumas tarefas são consideradas [Belian 2008]: (i) as rotinas de pré-processamento que traduzem os esquemas em um formato comum; (ii) a comparação de esquemas para estabelecer o significado dos elementos do esquema, produzindo mapeamentos entre esquemas; (iii) o *merging (*integração dos esquemas*)* e as tarefas de reestruturação dos elementos correspondentes ao esquema integrado. Nesse processo os nomes dos elementos

podem ter diferentes significados dependendo do contexto com os quais eles estejam relacionados.

No trabalho de Montanelli e seu grupo [Montanelli *et al.* 2010] é oferecida uma plataforma baseada em comunidade semântica para integração de dados e serviços em uma rede P2P não estruturada. Neste, o gerenciamento de contexto é utilizado para criação de perfis de acordo com o comportamento dos pontos e para a filtragem dos recursos disponíveis conforme as preferências e contexto corrente dos pontos.

3.3. Qualidade da Informação

O crescimento do número de fontes de informação disponíveis na web e a natureza acessível desta informação por um conjunto diversificado de usuários fez surgir o interesse pelos aspectos relacionados à Qualidade da Informação (QI) nesse ambiente [Arazy e Kopak, 2011].

Em seu uso, os dados podem apresentar baixa qualidade tanto por não refletirem a realidade ou por serem mal utilizados e mal entendidos pelos usuários. Mesmo dados precisos, se não estiverem disponíveis e dispostos em tempo hábil para sua utilização por parte do usuário interessado, serão de pouco valor [Batista 2008].

3.3.1. **Dimensões da Qualidade**

A QI está associada ao conjunto de critérios ou dimensões utilizados para indicar o grau de qualidade global associados à informação no sistema [Pipino *et al.* 2002]. Existe uma definição comum na área de que a QI está relacionada à sua "adequacão ao uso" (*fitness for use*), ou seja, a informação é considerada adequada ao uso na perspectiva das necessidades do usuário quando está sendo utilizada [Wang e Strong 1996].

A QI é um aspecto multidimensional baseado em um conjunto de critérios ou dimensões. Uma das classificações mais referenciadas é apresentada no trabalho de Wang e Strong [1996]. Várias dimensões de qualidade foram analisadas a partir da perspectiva do usuário e agrupadas em quatro classes:

(i) intrínseca – diz respeito à qualidade do dado. Inclui os seguintes critérios: credibilidade, acurácia, objetividade, reputação.

(ii) contextual – qualidade do dado relacionado à tarefa. Inclui os seguintes critérios: valor agregado, relevância, temporalidade, completude, quantidade apropriada.

(iii) representação – está relacionada ao formato e significado do dado. Inclui os seguintes critérios: interpretação, facilidade de entendimento, representação concisa, representação consistente.

(iv) acessibilidade - está relacionada à disponibilidade do dado. Inclui os seguintes critérios: acessibilidade, segurança de acesso.

Como a qualidade é uma condição dependente de seu uso, é necessário que possam ser estabelecidos mecanismos de medição desses critérios. Alguns critérios podem ser medidos de forma direta, quantitativamente como, por exemplo, o critério de temporalidade. No entanto, outros critérios são subjetivos a exemplo da relevância cuja medição depende da satisfação do usuário. Sendo assim, para cada critério é preciso estabelecer formas de como avaliar ou medir seus resultados.

3.3.2. **Qualidade em PDMS**

Uma característica importante de um PDMS diz respeito a sua dinamicidade. Pontos podem entrar e sair da rede a qualquer momento. Logo, as relações de confiança (*trust*), identificação e classificação dessas fontes de dados se apresentam como fatores de grande relevância para o sistema. Em um PDMS, a QI das respostas das consultas depende não somente da qualidade dos dados de uma fonte de dados em particular (ponto), mas também da qualidade dos mapeamentos entre os pontos vizinhos [Yatskevich *et al.* 2006]. Pontos podem armazenar dados de baixa qualidade, e dados podem estar desatualizados, errados, incompletos ou ter procedência duvidosa [Heese *et al.* 2005].

Em um PDMS, existem (no mínimo) três fatores que podem influenciar a resposta para uma dada consulta do usuário, assim como a sua qualidade [Zaihrayeu 2006]:

- Rede – pontos podem modificar os dados de suas fontes, de seus esquemas, redefinir mapeamentos, e novos pontos podem entrar ou sair

do sistema a qualquer tempo. Nesse cenário, a mesma consulta submetida em um dado ponto, mas em outro instante, poderá fornecer diferentes respostas de diferente qualidade.

- Ponto – mapeamentos são estabelecidos de várias formas entre pontos. Portanto, a mesma consulta submetida no mesmo instante, por diferentes pontos, resultará em diversos grafos de propagação da consulta. Consequentemente, os resultados poderão ser diferentes e de diferente qualidade.
- Consulta – diferentes consultas submetidas no mesmo ponto poderão resultar em diferentes grafos de propagação de consulta e assim, produzir resultados variados de qualidade diversa.

Alguns trabalhos na área também destacam a necessidade de obter e medir critérios de qualidade como forma de melhorar os seus processos.

O trabalho de Zaihrayeu [2006] faz uma discussão sobre a aplicação de critérios de qualidade em sistemas P2P. Ele mostra como a natureza dos sistemas P2P agrega novas dimensões para a medição dos parâmetros de qualidade. Para o autor, nesse tipo de ambiente, os usuários não podem esperar receber respostas corretas e completas. É provável que as respostas venham de forma incompleta e parcialmente incorreta. Nesse sentido, uma consulta pode não necessitar da melhor resposta possível, mas simplesmente de alguma resposta. Esse tipo de resposta tem sido considerado uma resposta "*good-enough*". Tal conceito está relacionado à definição de que uma resposta será suficientemente boa se ela serve ao seu propósito dada a quantidade de esforço feito para obtê-la.

Zhuge *et al*. [2005] apresenta um método automático de descoberta de *links* semânticos que estabelece uma correspondência entre dois pontos. O método utiliza uma medida de similaridade dos pontos baseada em semântica para o roteamento eficiente da consulta, e algoritmos de mapeamento de esquemas para reformulação da consulta. Se existe inconsistência dos dados, o sistema usa um método de qualidade dos pontos (QoP). Esse método utiliza uma pontuação da qualidade percebida pelo usuário tais como número de resultados retornados, tempo de

respostas, sobrecarga de tráfego e precisão nos fluxos de dados retornados. Os dados retornados por pontos com alta QoP são considerados mais consistentes. Finalmente, o ponto de início da consulta irá combinar dados relevantes e, em seguida, dar uma visão uniforme dos resultados para os usuários.

O ESTEEM [Montanelli *et al* 2010] é uma plataforma para colaboração semântica em ambiente P2P que utiliza o sistema DaQuinCIS [Scannapieco *et al.* 2004] para manter informações sobre a qualidade e confiabilidade dos dados. O DaQuinCIS é uma arquitetura para gerenciamento da qualidade dos dados em sistema de informação cooperativos. No ESTEEM, metadados de qualidade e de confiabilidade podem ser associados com os dados exportados pelo ponto. Esses metadados representam medidas de qualidade correspondentes às dimensões mais comuns definidas para qualidade dos dados: completude, consistência de formato, precisão e consistência interna [Batini e Scannapieco 2006].

3.4. Considerações

Este capítulo apresentou conceitos sobre ontologias, contexto e qualidade da informação. Para cada conceito, foi considerada a sua aplicação em PDMS.

No caso das ontologias, foi destacado o seu uso como forma de estabelecer uma representação comum e compartilhada entre sistemas distribuídos com fontes de dados heterogêneas, ajudando a superar os problemas de interoperabilidade semântica e heterogeneidade dos dados e, por conseqüência, facilitando a integração dos esquemas e processamento da consulta.

Quanto ao uso do contexto, foram relacionados trabalhos que tratam da melhoria e enriquecimento semântico de alguns processos em um PDMS, a exemplo da reformulação da consulta, integração de esquemas e identificação de recurso com base no contexto do ponto.

Quanto à QI, é fato que, em se tratando de PDMS, critérios de qualidade poderiam agregar confiança ao sistema e tornar melhor os seus processos internos (por exemplo, roteamento de consultas), estabelecendo formas para medição da qualidade dos dados, dos pontos e das respostas obtidas durante o processamento da consulta.

O próximo capítulo apresenta um levantamento do estado da arte em soluções de roteamento para PDMS.

Capítulo 4

Estratégias de Busca e Roteamento em PDMS

Em um PDMS, devido a não existência de um ponto de centralização de todo conhecimento distribuído e disponível nesses sistemas, torna-se importante observar as estratégias adotadas para o roteamento eficiente das consultas. Cada ponto precisa decidir independente dos demais pontos que compõem a rede, para qual dos seus vizinhos a consulta deverá ser encaminhada. Essa decisão normalmente é feita tomando como base apenas o conhecimento local disponível no ponto.

Nesse capítulo, será realizada uma apresentação de estratégias de roteamento de consultas em PDMS. Para melhor fundamentar, inicialmente serão apresentadas soluções comuns de roteamento em sistemas P2P e, finalizando, serão relacionados os problemas correspondentes ao roteamento de consultas em PDMS.

4.1. Roteamento em Sistemas P2P

Os sistemas P2P adotam uma abordagem descentralizada quanto ao gerenciamento de recursos em uma rede. Distribuindo o armazenamento e o processamento por meio de pontos autônomos, eles podem escalar sem a necessidade de poderosos servidores. Em se tratando da consulta de informações, toda aplicação deve ter uma maneira de selecionar os pontos relevantes que podem responder a uma determinada consulta submetida em um ponto qualquer, mesmo que essa maneira não seja a melhor solução. Essa é uma forma de evitar que as consultas postas nesses ambientes acabem por congestionar toda a rede.

Em redes P2P não estruturadas, os dados estão distribuídos aleatoriamente, cada ponto conhece seu ponto vizinho, mas não sabe qual o recurso que ele dispõe. O

roteamento da consulta é feito tipicamente a partir de mecanismos de inundação, onde a consulta é encaminhada para todos os pontos que compõem a rede como forma de distribuição da consulta. Ao final, os resultados encontrados são retornados ao ponto solicitante, ou seja, de onde partiu a consulta. Nesse mecanismo, para evitar o congestionamento da rede, as consultas são enviadas a um número limitado de pontos. Sendo assim, um dos principais problemas em redes não estruturadas está relacionado ao seu crescimento e incompletude dos resultados alcançados pelas consultas em virtude do mecanismo de roteamento adotado. Alguns exemplos de sistemas P2P que utilizam redes não estruturadas incluem o Gnutella [Gnutella 2011], KaZaA [Kazaa 2011], e FreeHaven [Akbarinia *et al.* 2007].

De forma contrária, as redes estruturadas surgiram como uma proposta de resolver os problemas de crescimento das redes não estruturadas [Akbarinia *et al*. 2007]. A localização dos dados assim como os seus mapeamentos é representada na forma de tabelas de roteamento distribuídas. Sendo assim, durante o processamento de uma consulta submetida em um determinado ponto, um índice é utilizado para fazer o roteamento dessa consulta para os pontos relevantes. Essa estrutura pode envolver vários graus de coordenação central ou de conhecimento global disposto, por exemplo, em *super-peers*. Exemplos de sistemas em redes P2P estruturadas incluem, Chord [Stoica *et al.* 2001], CAN [Ratnasamy *et al*. 2001], Tapestry [Zhao *et al.* 2004], Pastry [Rowstron e Druschel 2001], Freenet [Clarke *et al*. 2002] e P-Grid [Aberer *et al.* 2003a].

Ao contrário das consultas em bancos de dados tradicionais, a maioria das consultas em sistemas P2P não é exaustiva. Por exemplo, quando um usuário inicia uma consulta de uma música, ele não está interessado em cada instância daquela música. Semelhante a uma pesquisa na web, a maioria dos usuários fica satisfeita com um pequeno subconjunto de todas as correspondências encontradas [Crespo e Garcia-Molina 2003]. Muitos sistemas fazem uso de abordagens que combinam diversas técnicas e estratégias. A seguir são apresentadas algumas delas [Haase *et al.* 2008].

4.1.1. **Inundação**

Nessa estratégia, as mensagens são enviadas a todos os pontos na rede, ou seja, um ponto encaminha a consulta para seus pontos vizinhos, que por sua vez encaminham para seus pontos vizinhos e assim sucessivamente. Para evitar que as mensagens fiquem navegando indefinidamente, é definido um tempo de vida – *time to live* (TTL) – medido em saltos, ou seja, em número de vezes que a consulta foi encaminhada [Costa 2009]. O objetivo é que pontos possam encaminhar sua consulta aos demais pontos de forma que um número suficiente de repostas seja alcançado ou até que um determinado número de encaminhamentos (saltos) seja realizado. Essa abordagem tem como aspecto negativo o fato de que cada consulta pode retornar uma grande quantidade de dados o que consequentemente ocasionaria uma sobrecarga na rede. Outro aspecto está relacionado à incompletude das respostas mesmo que a informação esteja disponível na rede. Ou seja, durante o processo de roteamento, é possível que a consulta atinja o número máximo de saltos (encaminhamentos) estimados antes mesmo de obter o dado desejado.

Com o objetivo de melhorar o desempenho da técnica de inundação, algumas pesquisas têm indicado o uso do algoritmo de colônia de formigas [Ciglaric e Vidmar 2006; Michlmayr 2006] na otimização dos caminhos que devem receber a consulta. Colônia de formigas, também conhecido como ACO – *Ant Colony Optimization System*, é um algoritmo de otimização baseado no comportamento das formigas e suas colônias na natureza. Sua utilização permite identificar quais caminhos têm maior probabilidade de retornar bons resultados. Outra característica importante deste algoritmo é a capacidade de absorver mudanças no grafo de roteamento dinamicamente. Tal característica habilita esta técnica para o uso em sistema de roteamento em redes de computadores dinâmicas, como as redes *peer-to-peer* [Costa 2009].

4.1.2. **Controle Central**

O sistema mantém a centralização de todas as informações disponíveis nos pontos que compõem a rede. Quando recebe uma solicitação, o diretório central identifica, a partir de um índice, o ponto mais adequado a fornecer a resposta e a comunicação

entre eles é então estabelecida. Em pequenas organizações, essa abordagem pode trabalhar muito bem porque a rede, por ser pequena e estável, apresenta pouco processamento de consulta e atualizações. Em grandes redes, problemas de escalabilidade devem surgir uma vez que o número de solicitações aumenta em virtude do aumento de usuários e pontos. Conseqüentemente, o espaço para armazenamento e o gerenciamento central ficam comprometidos [Kamienski *et al.* 2005]. Esse modelo foi popularizado pelo NAPSTER [2010].

4.1.3. **Brokering (Agentes Inteligentes)**

A comunidade de sistemas multi-agentes sugere o conceito de "*broker agents*" onde tecnologias relacionadas a agentes e ontologias são utilizadas como forma de desenvolver mecanismos de consulta avançados para acesso inteligente às fontes de dados espalhadas na rede.

O projeto InfoSleuth [Bayardo *et al.* 1997] através do uso dos agentes *brokering* utiliza a representação da ontologia exportada em LDL (*Logical Data Language*) para fazer o encaminhamento da consulta de forma inteligente. O mecanismo dedutivo de LDL ajuda a determinar a consistência das restrições na consulta do usuário e daquelas exportadas pelo Agente de recurso que por sua vez determina a relevância das informações que gerencia.

4.1.4. **Super-peer/Peer**

Desenvolvida pela comunidade de pesquisa *peer-to-peer*, essa técnica oferece bons resultados para compartilhamento de informações. Como sistemas cliente-servidor, alguns pontos, ou *super-peers*, agem como servidores dedicados a alguns outros pontos e podem executar funções complexas tais como indexação, processamento de consulta, controle de acesso e gerenciamento de metadados. Um dos exemplos mais conhecidos desse modelo é o KaZaa cujos pontos agem voluntariamente como *super-peers* que mantêm grandes tabelas de roteamento onde estão armazenadas informações sobre o conteúdo de outros pontos [Haase *et al.* 2008]. Essa abordagem não deixa de ser uma forma de centralização do sistema embora seja melhor que a solução de inundação.

O trabalho de Nejdl *et al.* [2003] mostra como esta abordagem baseada em esquema pode ser usada para criar SOC (*Semantic Overlay Clusters*) [Löser *et al.* 2003] em uma rede científica *peer-to-peer* (P2P) com um pequeno conjunto de atributos de metadados que descrevem os documentos dispostos na rede. Para evitar inundação das consultas, foram introduzidos índices de roteamento entre os *super-peers* que permitem identificar os pontos que contêm respostas relevantes a partir do armazenamento dos metadados usados em cada ponto.

4.1.5. **Tabela Hash Distribuida e Árvore de busca distribuída**

As *Distributed Hash Tables* (DHT) são baseadas na idéia de roteamento por conteúdo para aqueles pontos cujo identificador mais se aproxima do identificador do conteúdo desejado. Esta técnica parte do princípio de que todos os pontos têm a mesma função *hash* para associar identificadores únicos para qualquer tipo de conteúdo como, por exemplo, documentos, músicas, URL ou palavras. Alguns algoritmos como Chord (Stoica, 2001), Pastry (Druschel, 2001) e CAN (Ratnasamy, 2001) implementam esse modelo. Uma desvantagem desse modelo é o alto custo, devido às freqüentes mudanças na rede em função da dinâmica relacionada à entrada e saída de pontos.

Em se tratando de árvores de busca distribuída, o P-Grid [Aberer *et al.* 2003a] é uma árvore de busca binária virtual que utiliza cópias dos pontos na rede e usa algoritmos randômicos para acesso e busca. Chaves de busca são representadas de forma binária e distribuídas através dos pontos. Cada ponto armazena parte da árvore total e sua posição é representada na forma de uma seqüência de *bits* binários. Essa seqüência representa o subconjunto das informações contidas na árvore total pelo qual o ponto é responsável. Logo, para cada *bit* em sua seqüência, um ponto armazena o endereço de no mínimo outro ponto para garantir que embora algum ponto esteja fora da rede outros pontos poderão ser responsáveis pelo mesmo caminho.

4.1.6. **Semantic Overlay Networks (SON)**

Nessa estratégia, é definida uma rede virtual pelo agrupamento de nós com ligações semânticas. Durante a conexão dos pontos, o conteúdo é classificado de acordo com a semântica associada, podendo, inclusive, ocorrer sobreposição de redes. Os pontos com conteúdo semanticamente equivalentes são agrupados em *clusters* (cada *cluster* forma uma SON) [Crespo e Garcia-Molina 2003].

As consultas executadas em um sistema que utiliza SON são enviadas apenas aos agrupamentos semânticos relacionados ao tema da consulta, ignorando os pontos que estejam fora do tema [Costa 2009].

4.2. Roteamento de Consultas em PDMS

Para obter melhores resultados no roteamento de consultas, alguns PDMS organizam seus pontos de acordo com a similaridade semântica entre eles, formando agrupamentos semânticos. Dessa forma, o espaço inicial para roteamento da consulta é reduzido considerando apenas sua vizinhança semântica. Nessa seção apresentamos algumas estratégias de roteamento de consulta aplicadas a esses PDMS.

4.2.1. **Knowledge-Super-Peer (KSP) Network**

Ismail e seu grupo [Ismail *et al.* 2011] propõem uma topologia *super-peer* para a criação de comunidades semânticas e utilizam um método baseado em árvore de decisão para selecionar os pontos relevantes em uma dada consulta; algoritmos baseados em hipergrafos transversais também são utilizados para o roteamento da consulta. Segundo os autores, a vantagem desse modelo está relacionada à robustez do roteamento da consulta e às questões relacionadas ao crescimento da rede, privacidade dos dados e natureza dinâmica da rede *overlay* (pontos podem sair e outros podem juntar-se à rede a qualquer tempo).

O sistema proposto, nesse trabalho, é um sistema híbrido P2P baseado na organização de pontos em torno de *super-peers* conforme similaridade de conteúdo. Cada *super-peer* (SP) também está conectado ao *Knowledge-super-peer* (KSP), um mecanismo que especifica os *super-peers* que tenham pontos com dados relevantes

que possam ser respondidos com o mínimo de tarefas de consulta e, por conseqüência, fornecem um melhor tempo de resposta.

Os pontos KSP têm um melhor poder computacional e maior largura de banda. Eles são responsáveis por encaminhar as consultas, a partir do uso de índices, para *super-peers* relevantes, permitindo não somente reduzir o esforço de compilação das consultas, mas também prevenir que as consultas sejam espalhadas por toda a rede. Cada ponto KSP é representado por uma árvore de decisão, construída a partir das consultas processadas pelo próprio ponto.

O agrupamento de super-peers a partir do seu domínio semântico leva à construção explícita de comunidades onde cada uma é representada por um conjunto de super-peers e cada super-peer pode pertencer a mais de um agrupamento. Nesse caso, o conjunto de agrupamentos constitui um conjunto de hipergrafos, onde cada nó representa uma comunidade. O algoritmo utilizado produz um agrupamento entre super-peers que possuem histórico de sucesso em resposta a alguns itens correspondentes aos componentes de uma consulta. A Figura 4.1 exemplifica os conjuntos de rotas em um hipergrafo de *super-peers* { {SP_1, SP_2, SP_6}, {SP_1, SP_6, SP_8} e {SP_3, SP_7, SP_8, SP_{10}}}. Cada agrupamento tem no mínimo um *super-peer* em comum usado para encontrar a transversalidade mínima entre os agrupamentos. Os *super-peers* em comum são usados para encaminhar as consultas para outros agrupamentos.

Na Figura 4.1 assumimos que uma consulta Q1 é submetida em P1, e que o processo de roteamento da consulta se dará da seguinte forma:

1. O *super-peer* responsável por P_1 é identificado, no caso, SP_1
2. SP_1 envia a consulta para cada *super-peer* de sua rota transversal {SP_1, SP_2, SP_6}
3. Cada SP que recebe a consulta encaminha para seus pontos aptos a responder a consulta
4. No final, o retorno da consulta irá corresponder ao conjunto de pontos relevantes e seus respectivos SP ((P_2:SP_1),(P_{11}:SP_8)...).

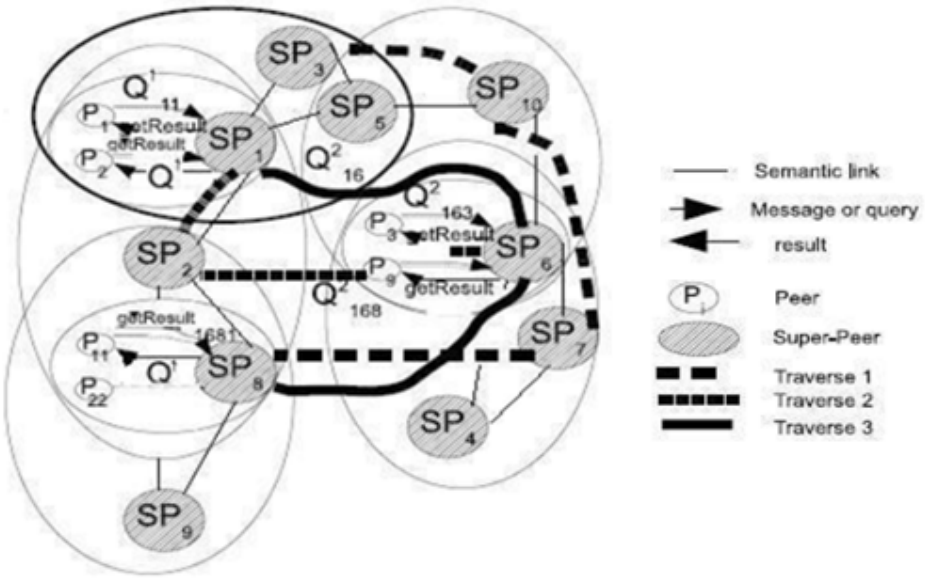


Figura 4.1: Roteamento da consulta (KSP) [Ismail *et al.* 2011]

4.2.2. **Ontozilla**

No Ontozilla, os pontos que compõem a rede com mesmo domínio de conhecimento são agrupados em *clusters*, e os relacionamentos entre eles são modelados de acordo com o conhecimento que eles representam [Joung e Chuang 2009]. Pontos com interesses idênticos ou similares são agrupados em SIG (*Special Interest Groups*). Em cada SIG, os pontos empregam algum sistema de classificação para agrupar seus interesses em classes, onde cada classe representa um *cluster,* representado como uma árvore hierárquica (*cluster tree*), de pontos que abrigam o mesmo conceito. Cada *link*, na árvore*,* que representa uma ligação entre um ponto pai e outro filho é representado por relacionamento de subclasse "*is subclass of*" (*is-a*). SIG são descrições conhecidas por todos os pontos na rede. Cada classe tem uma descrição contendo o nome da classe, o nome SIG, a hierarquia de classificação e algumas anotações da classe. As descrições de SIG e classes são representadas com linguagens de ontologia. Logo, os relacionamentos entre SIG e/ou classes podem ser inferidos com uma ontologia pré-definida, ou usando algumas técnicas de correspondência semântica onde se possa medir a similaridade entre os SIG ou classes de forma a determinar se dois pontos abrigam o mesmo assunto.

Considerando a descrição do ponto, algumas ligações (*links*) semânticas podem ser estabelecidas entre os pontos para facilitar o roteamento. Um ponto *x* que tem um

link com um ponto *y,* na prática *x* mantém a descrição de *y* em sua tabela de roteamento, fazendo com que *x* passe a ter uma referência direta com *y.* Alguns *links* no Ontozilla são classificados da seguinte forma:

- SIG *links* - são usados para conectar pontos de diferentes SIGs.
- Partner *links* – estabelecem relacionamento cooperativo entre pontos. Por exemplo, se um ponto *x* frequentemente consulta informações do ponto *y*, é interessante que *x* mantenha um *partner link* com *y*.
- Twin links – são usados para conectar pontos em um mesmo *cluster*.
- Parent links and child links – estabelece a conexão entre classe e sub-classe em um *cluster-tree*

Para ilustrar, a Figura 4.2 mostra um esquema do Ontozilla. Cada agrupamento representado na figura por linhas pontilhadas que cercam *clusters* representa um SIG. Os *links* semânticos entre os pontos (dentro de um *cluster)* estabelecem caminhos para que as consultas possam ser roteadas de forma eficiente aos pontos que contenham as informações solicitadas na consulta. O encaminhamento de consultas é limitado por TTL.

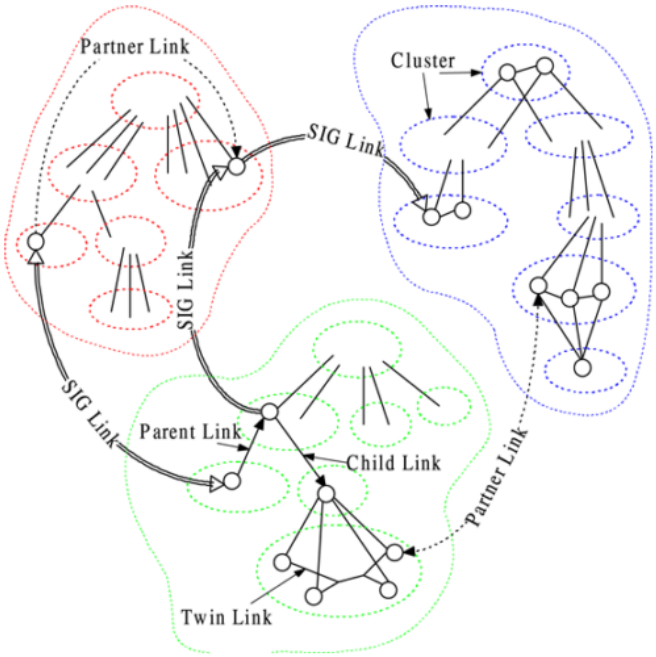


Figura 4.2: Esquema do Ontozilla [Joung e Chuang 2009]

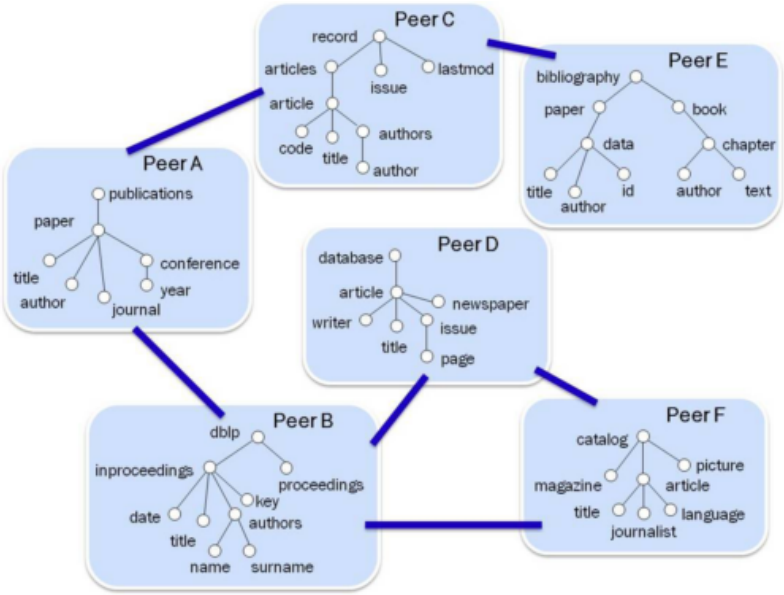
O roteamento no Ontozilla é feito conforme o esquema de roteamento intra-SIG e inter-SIG. O **roteamento intra-SIG** é feito de forma direta. Se um ponto recebe uma consulta que solicita um recurso pertencente a sua super-classe, ele encaminha a consulta para o *cluster* mais superior (*parent links*). Se a consulta é para recursos pertencentes a sua subclasse, ele encaminha a consulta para o *cluster* mais inferior (*child links*). Finalmente, se a consulta é para os recursos pertencentes a sua própria classe, ele transmite a consulta ao seu *cluster* através de ligações individuais (*twin links*). Se mesmo assim, o ponto que fez a consulta não estiver satisfeito com o número de respostas, ele pode fazer uma nova consulta agora envolvendo as subclasses ou super-classes das classes que já foram consultadas. Fazendo com que o processo se repita. O **roteamento inter-SIG** está relacionado ao pedido de recurso fora do SIG do próprio ponto. Quando um ponto necessita de um recurso ele procura em sua tabela de roteamento e encaminha a consulta para o ponto correspondente por meio de ligações SIG. Se a tabela de roteamento não contém qualquer ponto correspondente, ele transmite um *SIG inquire messages*, que são mensagens utilizadas no processo de junção de SIG, dentro de um número de saltos estimados, para que seja feita uma busca do novo SIG. Caso o SIG seja encontrado, a consulta é enviada para ele e, então, é roteada conforme o processo intra-SIG.

A eficiência e corretude do roteamento da consulta estão baseadas na confiança das ligações semânticas entre os pontos. Entretanto, os pontos registrados na tabela de roteamento podem estar indisponíveis, considerando que os pontos podem entrar ou sair da rede de forma imprevisível. Além disso, devido às falhas e ao atraso na propagação das mudanças, as visões dos pontos na rede podem estar inconsistentes. É importante, então, destacar a necessidade de mecanismos que melhorem essa instabilidade ocasionada na rede.

4.2.3. **SRI - Semantic Routing Index**

No trabalho de Mandreolli e seu grupo [Mandreolli *et al.* 2007b], o PDMS SUNRISE usa índices de roteamento semânticos (SRI) [Mandreolli *et al.* 2007a, Mandreolli *et al.* 2006] como forma de tornar eficiente o processamento da consulta em seu sistema. Cada ponto que compõe a rede mantém um resumo das informações a respeito do

grau de similaridade semântica entre os conceitos armazenados entre um ponto e seus respectivos vizinhos. Essa informação é mantida em uma estrutura de dado local denominada SRI (*Semantic Routing Index*). Sendo assim, um ponto **p** que tenha **n** vizinhos e **m** conceitos em seu esquema, armazena um SRI estruturado como uma matriz com **m** colunas e **n+1** linhas, onde a primeira linha refere-se ao conhecimento sobre o esquema local do peer **p**, como mostra a letra (b) da Figura 4.3 que utilizou como cenário o PDMS representado na letra (a) da mesma figura. É possível assim, que cada ponto sintetize, para cada conceito de seu esquema, a aproximação semântica das sub-redes acessíveis a partir de seus vizinhos e, assim, forneça uma informação sobre a relevância dos dados que podem ser alcançados em cada trajeto a ser escolhido.



(a) Cenário de demonstração de um PDMS

PeerA SRI	paper	title	author	...
PeerB	0.51	0.49	0.37	...
PeerC	0.81	0.86	0.66	...

(b) Parte do Índice de Roteamento Semântico (SRI) do ponto A

Figura 4.3: Exemplo de referência do trabalho [Mandreolli *et al.* 2007b]

O SUNRISE permite ao usuário, por meio de uma interface gráfica, explorar os caminhos mais promissores durante uma busca. Nessa interface, o usuário pode

indicar o cenário inicial de consulta: o ponto e o conceito, a condição de parada e a estratégia de roteamento. São condições de parada: (a) máximo de saltos (TTL) (b) satisfação do objetivo (uma medida de qualidade dos caminhos a serem explorados). Com relação às estratégias de roteamento, o usuário poderá escolher entre randômico, baseado no mapeamento semântico (explorando apenas os vizinhos do mapeamento) e baseado em SRI (utilizando os índices semânticos). Na Figura 4.4, o ponto A é o ponto inicial da consulta, o conceito solicitado é *paper* e a estratégia de roteamento baseada em SRI. O SRI do ponto A indica que a direção mais promissora é o ponto C. Logo, o ponto C é escolhido e o processo é iniciado, o ponto C torna-se o ponto corrente e o conceito é atualizado conforme o mapeamento dos esquemas. O conceito *paper* do ponto A torna-se o conceito *article* para o ponto C. Em seguida, o usuário pode continuar a consulta ou finalizar.

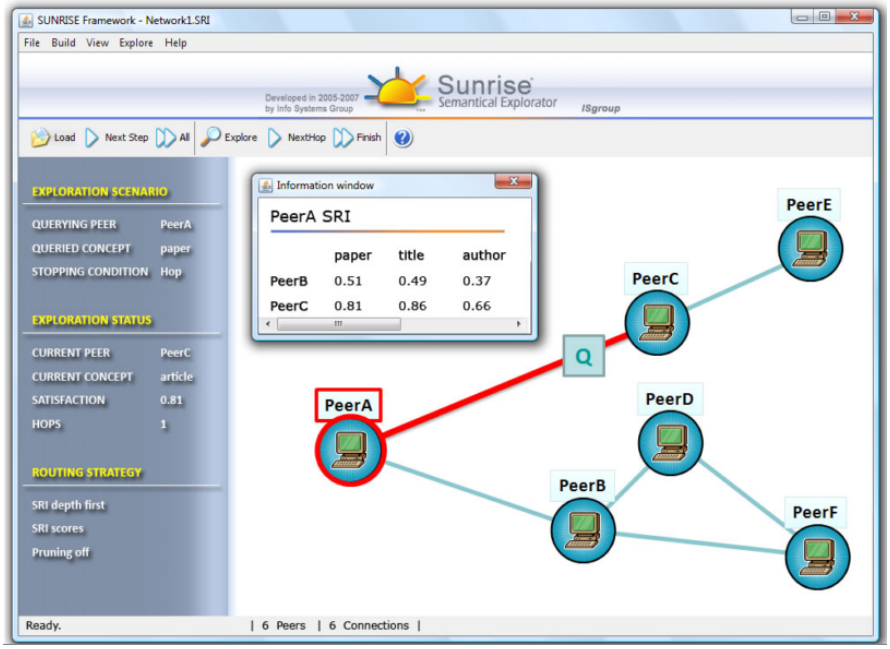


Figura 4.4: Interface gráfica do SUNRISE [Mandreolli *et al* 2007b]

4.2.4. **H-Link**

O H-link é um mecanismo de roteamento semântico para sistemas P2P cuja principal característica está no uso de ontologias para representação do conhecimento dos pontos, de técnicas de correspondência de ontologias para seleção de pontos semanticamente similares e gerenciamento independente pelos pontos de sua própria

ontologia [Montanelli 2007]. A idéia chave do H-Link é explorar os resultados das interações durante a fase de descoberta de conhecimento para treinar o comportamento do mecanismo de roteamento. Para alcançar esse objetivo, os pontos são conectados por meio de medidas de confidência baseadas em técnicas de correspondência que acompanham a afinidade semântica entre os conteúdos dos diferentes pontos. Logo, os pontos são organizados em uma SON (*Semantic Overlay Network*) cujos pontos têm conhecimento similar e estão interligados como vizinhos semânticos.

A ontologia do ponto no H-Link provê uma descrição formal do contexto do ponto em termos de: (1) o conhecimento local dos recursos do ponto que serão compartilhados com os outros pontos, e (2) o conhecimento do ponto sobre a rede que é o conhecimento do ponto sobre seus vizinhos semânticos progressivamente descobertos ao longo das interações que ocorrem na rede. Importante destacar que a definição de contexto nesse trabalho está relacionada à representação formal do conhecimento do ponto (por exemplo, uma ontologia) sobre os recursos compartilhados.

Para ilustrar esse mecanismo, o autor propõe um exemplo, Figura 4.5, onde cada ponto é independente e se uniu ao sistema por meio de sua própria ontologia. Suponha que o ponto A está interessado em localizar outros pontos que possuam recursos semanticamente relacionados ao domínio de **publicação**. Para isso, o ponto A formula uma consulta Q_1 e submete ao sistema contendo os conceitos de interesse *Publication* e *Book* com suas respectivas propriedades *year* e *author*. Ao receber a consulta Q_1 os pontos B, C e D usam o gerador de correspondência semântica (H-Match) [Castano *et al.* 2004] para comparar a descrição dos conceitos na consulta com a ontologia do ponto. Conforme o resultado desse processo, os pontos B e D enviam ao ponto A uma lista ordenada dos conceitos encontrados, e, para cada entrada, o cálculo do valor de afinidade semântica (SA). Como se pode observar na Figura 4.5, apenas os pontos B e D retornam valores enquanto C não responde ao ponto A por não possuir conceitos correspondentes.

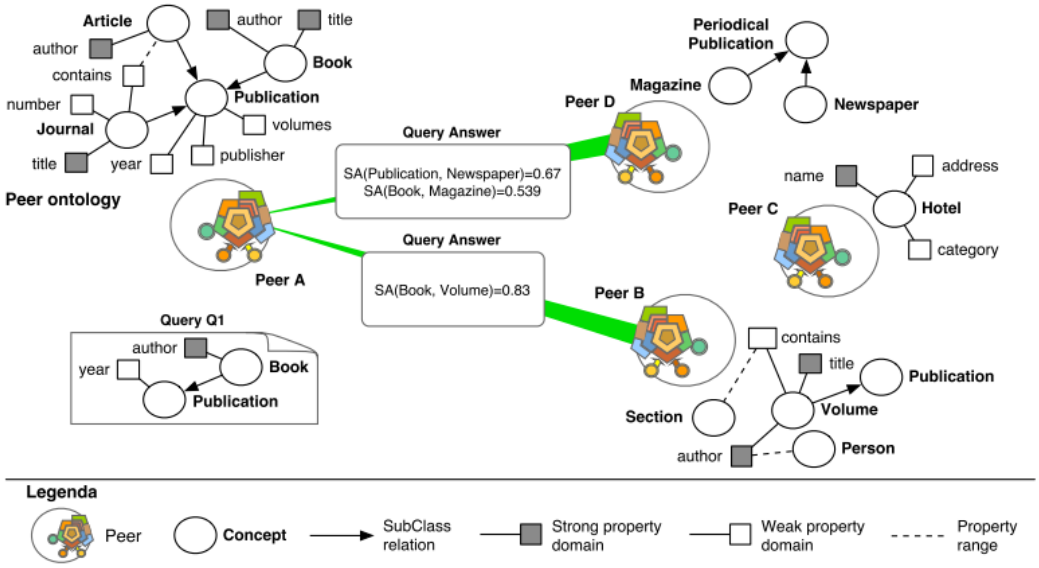


Figura 4.5: Mecanismo de roteamento H-Link

É importante destacar que a resposta fornecida ao ponto A poderá ser utilizada em interações futuras. Quando consultas similares forem feitas, esses resultados contribuirão no processo de identificação dos melhores pontos disponíveis na rede, aptos a responder à consulta [Castano e Montanelli 2008].

Outro critério utilizado no H-Link é o sistema de distribuição de crédito [Montanelli e Castano 2008]. Nesse sistema, o ponto ao encaminhar a consulta para outro ponto define um número de créditos que corresponde ao número de respostas esperadas pelo ponto. Dessa forma o número de crédito é decrementado sempre que a consulta é roteada para um novo ponto. Ao final, se não existir crédito disponível o mecanismo de propagação da consulta é interrompido.

4.2.5. **OntoSum**

É um mecanismo de roteamento de consultas em sistemas P2P baseado na utilização de um índice da sumarização da ontologia dos pontos que compõe a rede [Li e Vuong 2007]. Esse mecanismo assume que cada ponto pode usar sua própria ontologia para descrever o conhecimento relativo aos seus recursos e que a topologia da rede será ajustada de acordo com as propriedades ontológicas de cada ponto. Uma estratégia de indexação permite encaminhar as consultas apenas aos pontos semanticamente relacionados. A organização dos pontos na rede se apresenta da

seguinte forma: pontos com conteúdo similar formam um grande domínio e dentro desse domínio os pontos podem agrupar-se em *clusters* se eles compartilharem da mesma ontologia conforme pode-se observar na Figura 4.6.

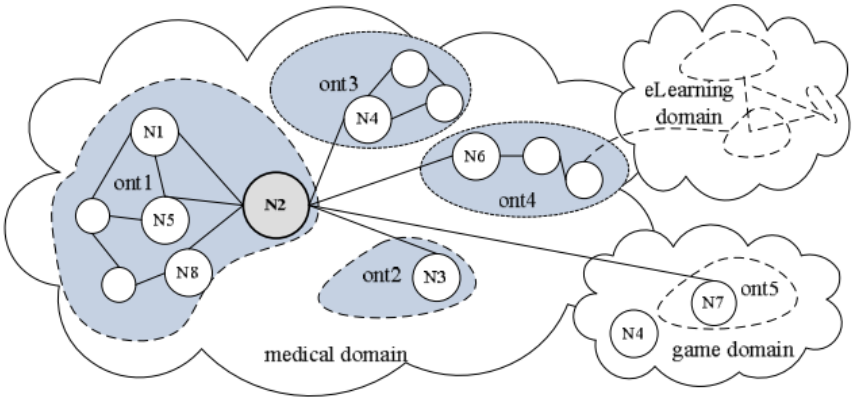


Figura 4.6: Topologia da Rede [Li e Vuong 2007]

Na Figura 4.6, todos os pontos do domínio *medical* estão interessados nos diferentes aspectos dos recursos médicos, e cada um dos pontos pode estar usando diferentes ontologias para descrever seus recursos. Entretanto, se eles compartilham o mesmo interesse, eles estarão conectados uns aos outros. No mesmo domínio *medical*, observamos que os pontos N1, N2, N5 e N8 estão agrupados porque utilizam a mesma ontologia ont1. Nessa rede estruturada multi-nível, por exemplo, um ponto B vizinho de A pode distinguir três tipos de vizinhos baseado na sua similaridade semântica: (1) vizinho com distância zero (ou vizinho de mesma ontologia, vizinho *intra-cluster*), se sim(A,B)=1 (similaridade semântica entre A e B é igual a 1); (2) vizinho com curta distância (ou semanticamente relacionado), se sim(A,B)≥t (0<t<1 é o limite semântico de A); (3) vizinho com longa distância (ou vizinhos não relacionados semanticamente), se sim(A,B)<t. O princípio básico é permitir que um ponto sempre consiga achar os vizinhos mais próximos, mesmo aqueles que estão em distâncias maiores alcançáveis apenas por outros *clusters*.

Para o roteamento eficiente das consultas, são propostos nesse trabalho dois esquemas de roteamento: roteamento *inter-cluster*, para rápida localização dos clusters relacionados semanticamente; e roteamento *intra-cluster*, para localização dos recursos que satisfazem às restrições da consulta. Para construir tais esquemas, duas tabelas de roteamento são mantidas em cada ponto: tabela de roteamento *inter-*

cluster (*inter-table*) e tabela de roteamento intra-cluster (*intra-table*). Essas tabelas de roteamento mantêm um conhecimento mais refinado sobre os vizinhos. Essa é uma das características dessa estratégia de roteamento: a consulta inicialmente, caminha sobre a rede, e após atingir o domínio de destino amplia esse domínio e investiga as propriedades que possam indicar seus vizinhos semânticos.

A Figura 4.7, mostra a representação de uma tabela de roteamento *inter-cluster* para o ponto N2. Os pontos N3, N4, e N6 são vizinhos de curta distância de N2. O ponto N7 é um vizinho de longa distancia que tem um domínio semântico não relacionado a N2. Os vetores com assinatura de vizinhos semânticos são comprimidos em um *Bloom filter* [Bloom 1970] como seqüências de 0s e 1s. A última coluna da tabela armazena os mapeamentos inter-ontologias entre N2 e os outros vizinhos semanticamente relacionados.

neighbor	semantic similarity	compressed signature vector	Inter-ontology mappings
N3	0.8	ont2 [1001010…]	*Ca=Ca',P1=P1'...*
N4	0.7	ont3 [0111010…]	*Cm⊃Cm',P2⊃P2'*
N6	0.6	ont4 [1100010…]	*Ct⊂Ct'* ...
N7	0	ont5 [0001010…]	none

Figura 4.7: Tabela de roteamento *inter-cluster* para o ponto N2 [Li e Vuong 2007].

Após o *cluster* ter sido identificado por meio da tabela de roteamento *inter-cluster*, a tabela *intra-cluster* será usada para encaminhar a consulta dentro do *cluster*. Essa tabela inclui um sumário dos recursos sobre os vizinhos dentro do mesmo *cluster*, que podem ser localizados a partir de determinado ponto.

O roteamento da consulta dentro do *cluster* é baseado em um algoritmo de roteamento sobre um vetor de distância de recurso (RDV). Cada ponto mantém uma tabela de índices de recursos com informações sobre a distância de cada recurso (em número de saltos). Quando um ponto recebe uma consulta, o algoritmo escolhe a rota mais próxima e encaminha a consulta. Um exemplo pode ser observado na Figura 4.8. O ponto A recebe uma consulta para o recurso que está mapeado para as posições 3 e 6. Ele checa sua tabela de roteamento e acha 2 casamentos: o ponto C com 2 saltos; e o ponto D com 3 saltos. De acordo com o algoritmo, o caminho mais curto será o

escolhido, no caso, pelo ponto C. A consulta é encaminhada para C que de forma similar repetirá o processo e encaminhará para E onde a consulta é respondida.

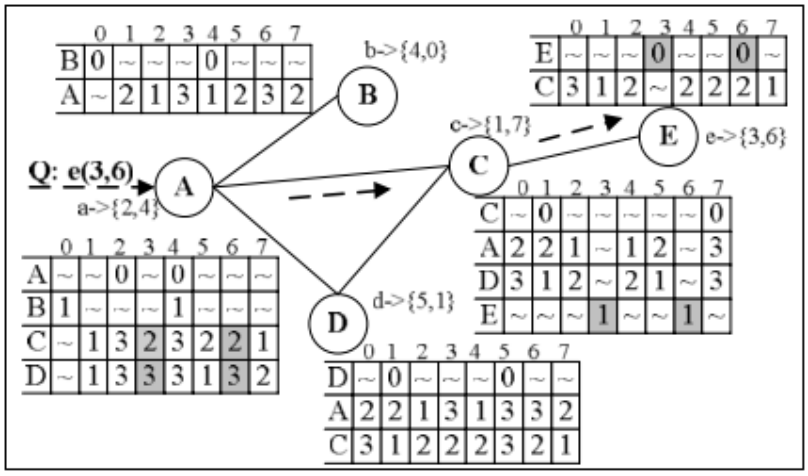


Figura 4.8: Roteamento da consulta utilizando RDV [Li e Vuong 2007]

4.2.6. **System P**

A estratégia adotada no System P está voltada para a completude da consulta a partir da redução de custo dos planos de consulta [Roth e Naumann 2007]. O principal objetivo é encaminhar a consulta apenas para aqueles pontos que possam fornecer resultados. Para obter essa decisão, cada ponto classifica todos os seus vizinhos de acordo com o potencial de resultados a serem retornados baseados em seus respectivos mapeamentos [Roth e Naumann 2007].

A contribuição do mapeamento para uma consulta é determinada pelo cálculo da completude do plano de consulta local considerando ou não os mapeamentos. O modelo de completude envolve duas dimensões: cobertura e densidade. A cobertura descreve a proporção do tamanho de um conjunto de tuplas em função do total do número de tuplas armazenadas no PDMS. A medida aplica-se tanto para os dados armazenados como para o conjunto de tuplas retornadas como resultado a uma consulta. No caso das tuplas retornadas, a medida é baseada no número total de tuplas que preenchem os requisitos da consulta. A dimensão densidade, por outro lado, descreve o número de valores de atributo para cada resultado em relação aos atributos da consulta. É esperado que alguns atributos retornem com valores nulos criando assim tuplas com resultados incompletos de baixa densidade.

Dessa forma, o algoritmo de cálculo da completude revela o impacto da perda de informação de um mapeamento sobre os resultados da consulta de um ponto. Baseado nessa comparação, planos de consulta com estratégias de orçamento (*budget*) são definidas objetivando estabelecer um limite de roteamento (a exemplo do mecanismo de TTL) para mapeamentos alternativos entre pontos.

O planejamento da consulta dirigida à completude é baseado em um orçamento prévio. Basicamente, um ponto que recebe uma consulta deve classificar diferentes planos de consulta local de acordo com o seu potencial de dados a serem retornados e mediante um orçamento (estabelecido em função dos resultados estimados), podá-los, ou seja, retirá-los da lista de classificação gerada. Esta estimativa é obtida por meio de histogramas multi-dimensionais [Roth e Naumann 2005].

Existem dois tipos de estratégias de orçamento definidas: *weight* e *Greedy*:

- *Weight* – nessa estratégia, o ponto considera o peso de contribuição dos mapeamentos de acordo com a sua vizinhança. Nesse caso, o ponto distribui um orçamento na proporção inversa à contribuição da informação perdida em determinado mapeamento. Por exemplo, se o uso de determinado mapeamento provoca uma grande perda de informação, o orçamento será menor em comparação a outros mapeamentos que produzam resultados melhores. Essa estratégia prefere explorar a vizinhança direta de um ponto (com alta relevância semântica e menor perda de informação) em vez de tentar alcançar vizinhos indiretos, ou seja, mais distantes. Essa estratégia apresenta como problema o fato de que o orçamento estabelecido inicialmente pode levar os pontos que sejam obrigados a gastar o orçamento com mapeamentos que possuem alta perda de informação.

- *Greedy* – para maximizar a exploração de mapeamentos essa estratégia destina todo o orçamento para o mapeamento que apresenta melhores resultados, ou seja, menor perda de informação. Essa estratégia promete um retorno de mais dados, para orçamentos pequenos, comparado ao tamanho do espaço de busca.

É importante observar que ambas as estratégias são fortemente dependentes da razão entre o valor total do orçamento, o tamanho do espaço de busca e a média do número de mapeamentos existentes no PDMS.

4.2.7. **ESTEEM**

Como discutido no Capítulo 2, do ponto de vista de uma rede, o ESTEEM é uma rede P2P organizada em SON formando comunidades semânticas [Montanelli *et al.* 2010]. Cada ponto nesse ambiente é caracterizado por uma **ontologia do ponto** (descrição dos dados a serem compartilhados), uma **ontologia de serviço** (descrição dos serviços a serem compartilhados), um **contexto corrente** (descreve o perfil do ponto, seu interesse, situação e coordenadas espacial/temporal), **perfil de confiança** (determinado com base no número de queixas disparadas por outros pontos da comunidade, para o qual o ponto tenha sido um provedor de um determinado dado sobre uma transação específica) e de **qualidade dos dados** (computação de métricas de qualidade sobre os dados exportados pelo ponto). São critérios utilizados para os dados exportados pelo ponto: *column completeness*, *format consistency*, *accuracy* e *internal consistency* [Batini e Scannapieco 2006]. Em se tratando dos aspectos de qualidade, cada ponto tem a possibilidade de associar metadados de qualidade para o dado exportado.

No ESTEEM, um modelo de contexto geral, a *Context Dimension Tree* (CDT) (Figura 4.9) representa todos os possíveis contextos passíveis de ocorrer em determinada situação. Baseado nesse modelo, o ponto que inicia a formação de uma comunidade semântica poderá associar a CDT a um contexto desejável para a comunidade. Assim, a CDT de um ponto expressa algumas perspectivas (dimensões) que determinam qual porção do dado é interessante em diferentes situações. Por exemplo, *actor*, *situatio*n e *interest topic* são algumas das dimensões mais comuns que podem guiar a seleção de informação ou serviços relevantes à consulta [Montanelli *et al.* 2010].

Uma sub-árvore da CDT determina uma porção do conjunto de dados, especificado como uma visão. Essa visão representa os dados que são relevantes quando o contexto correspondente torna-se corrente. Na Figura 4.9, pode-se observar um exemplo da modelagem CDT para uma aplicação na área médica.

Considerando a Figura 4.9, suponha a situação em que um médico de um hospital da África Central está interessado em saber sobre estruturas disponíveis para doenças contagiosas em sua região. O objetivo desse médico é encontrar dados e serviços relevantes para o tratamento de pacientes com malária. A área cinza na figura representa o contexto corrente do ponto de submissão da consulta para formação da comunidade, onde os valores em tempo de execução para os parâmetros **id_name** e **reg_name** são "Malária" e "África Central".

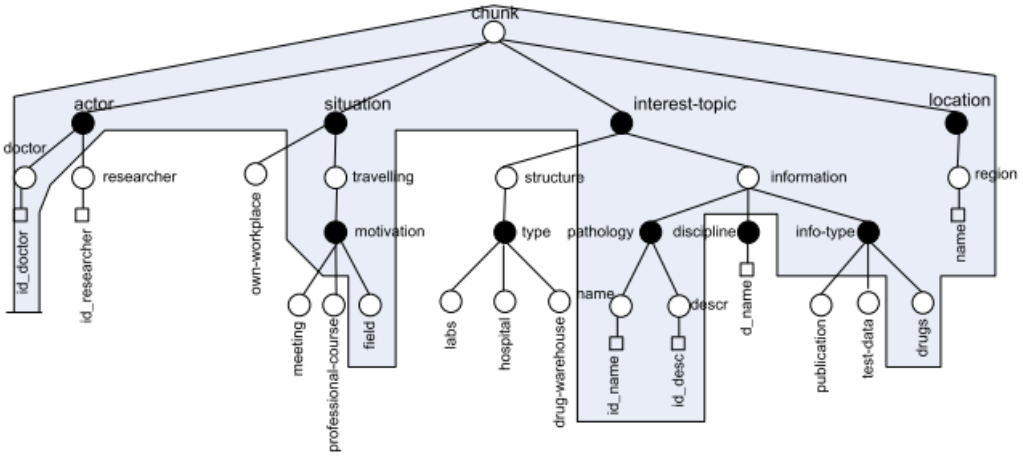


Figura 4.9: Exemplo de CDT [Aiello *et al.* 2007]

O ponto de origem da consulta submete seu contexto corrente para seus vizinhos semânticos que por sua vez utilizam um *matching* de contexto para calcular a similaridade entre o contexto recebido e o local. Os pontos com contexto similar respondem ao ponto que submeteu o CDT inicial. Como consequência, vizinhanças semânticas são estabelecidas por ligações entre os pontos similares. Em geral, as solicitações orientadas a contexto podem antecipar solicitações de dados ou serviços e ajudar os pontos na especificação de consultas mais refinadas.

No ESTEEM, o módulo de roteamento semântico (Seção 2.1.7 desse trabalho) tem a responsabilidade de fazer a seleção dos pontos que deverão receber a consulta em uma base semântica (a comunidade) [Montanelli *et al.* 2010]. Esse procedimento é realizado através da identificação dos pontos que tenham maior probabilidade de fornecer resultados de acordo com a consulta. O H-Link [Montanelli 2007] é o mecanismo usado para implementação do mecanismo de roteamento semântico no

ESTEEM. Entretanto, além do mecanismo de crédito do H-link, um TTL é implementado para evitar ciclos e propagação ilimitada na rede durante a busca de serviços [Bianchini *et al.* 2009].

4.2.8. **GrouPeer**

O GrouPeer é um sistema desenvolvido para permitir a avaliação precisa de consultas por meio de um processo automático de criação, manutenção e agrupamento de grupos semânticos similares em uma rede P2P não estruturada [Kantere *et al.* 2009].

O método de descoberta de pontos remotos e relevantes é propagar, ao longo do caminho percorrido por uma consulta, tanto a consulta original submetida, como também as sucessivas versões que vão sendo reescritas. Neste caso, os pontos receberão um conjunto de consultas e decidirão qual a versão que irão responder. Para realizar essa tarefa, cada ponto possui um mecanismo de reescrita para reescrever as consultas expressas em seus esquemas baseados nos respectivos conhecimentos. Uma ferramenta é utilizada pelos pontos para compreender e traduzir as consultas, ou parte delas, expressas nos esquemas para os quais os mapeamentos não estão disponíveis.

Reformulações sucessivas da consulta podem produzir versões distorcidas da consulta original. Se a cadeia de mapeamentos utilizados na reescrita é pobre em informação relevante para a consulta (ou seja, partes da consulta não podem ser reformuladas com precisão), isso pode resultar em uma rápida degradação da consulta em poucos saltos do roteamento.

O GrouPeer mantém uma lista dos atributos eliminados durante as reformulações para que eles possam ser utilizados com os esquemas dos próximos pontos, para os quais a consulta tenha sido roteada. No sistema, pontos individualmente decidem se respondem a determinada consulta que tenha sido reescrita ou a sua versão original.

O método utiliza o encaminhamento de consultas limitado por parâmetros de TTL que são utilizados para interromper o processo de roteamento da consulta. Durante o roteamento a partir de cada ponto, a identificação de vizinhos relevantes é feita a

partir da avaliação de similaridade entre o esquema do ponto e dos atributos da consulta. Isto implica dizer que um ponto envia uma consulta apenas para os vizinhos cujos esquemas têm o maior valor de similaridade em relação à consulta.

A Figura 4.10 mostra um processo de propagação de uma consulta no GrouPeer. Com base na figura, observa-se que o ponto P2 recebe a consulta original Qorig e a consulta reescrita Qsr_M1,2 , resultante do mapeamento M1,2, entre P1 e P2, e assim sucessivamente entre os demais pontos.

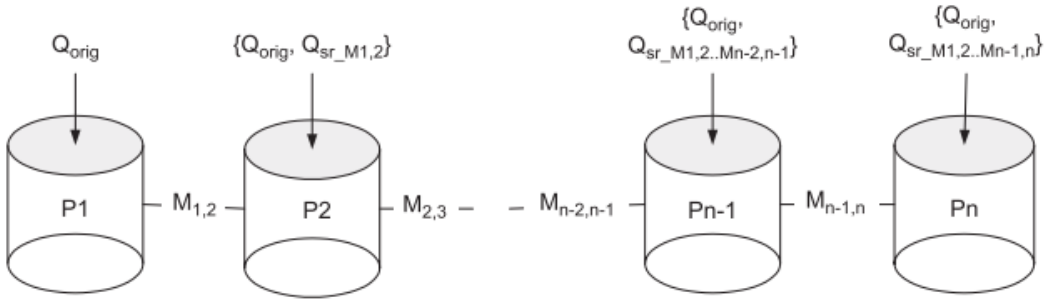


Figura 4.10: Propagação de uma consulta no GrouPeer [Kantere *et al.* 2009]

No ponto destino, as duas consultas, a original (Qorig) e a previamente reescrita (Qsr_previous), serão novamente reescritas de acordo com os mapeamentos desse ponto. A Qorig será rescrita para Qar e a Qsr_previous para Qsra (Figura 4.11).

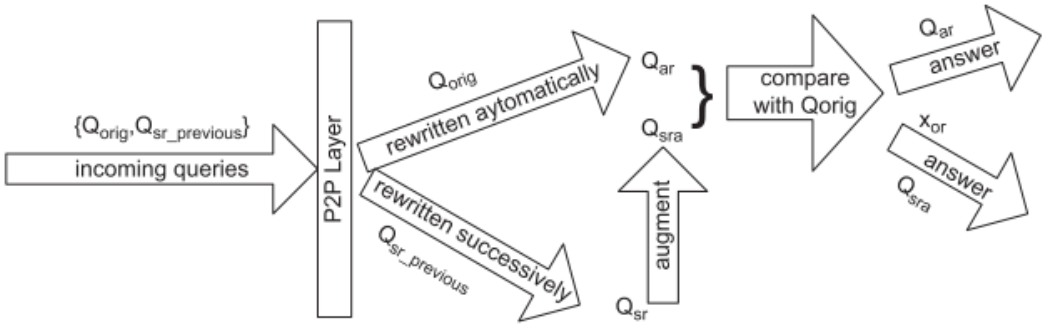


Figura 4.11: Procedimento de resposta da consulta no ponto [Kantere *et al.* 2009]

Após reescritas, as consultas Qar e Qsra são comparadas com Qorig, por meio de uma função de similaridade descrita em [Kantere *et al.* 2009]. Depois dessa comparação, a consulta que tiver o maior grau de similaridade com a consulta original, será a consulta a ser respondida pelo ponto.

Ao receber a resposta da consulta, o ponto que submeteu a consulta original indica seu grau de satisfação em função da resposta recebida. Essa avaliação em torno da versão utilizada para a resposta será enviada, ao ponto que a respondeu.

Finalmente, a cada repetição desse processo e com base nas avaliações recebidas, o ponto que submeteu a consulta original pode decidir que ele tem interesse comum com o ponto remoto e convidá-lo a estabelecer novos mapeamentos entre eles. O GrouPeer utiliza parâmetros de TTL para limitar o encaminhamento da consulta na rede.

4.2.9. **Outros trabalhos**

Outras estratégias não abordadas nesse capítulo de forma mais detalhada trazem propostas também interessantes para o roteamento semântico em sistemas P2P, como o REMINDIN' [Staab *et al.* 2004] que provê um mecanismo de roteamento baseado em ontologias e seleção de pontos a partir da memorização dos resultados de sucesso obtidos nas respostas fornecidas por outros pontos. O NeuroGrid [Joseph 2002], onde cada ponto mantém uma base de conhecimento que contém associações entre pontos e palavras-chave. Tal base é utilizada durante o processo de busca de recursos na rede. O QSummary [Hose *et al* 2006] utiliza índice de roteamento combinando estruturas em árvore e histogramas para tornar mais eficiente o processo de busca, reduzindo o custo de execução, e propondo uma estratégia de manutenção eficiente desses índices.

4.3. Problemas de Roteamento de Consultas em PDMS

Com a ausência de um esquema global as estratégias de roteamento têm que focar no roteamento eficiente da consulta para somente aqueles pontos que possam melhor contribuir com o resultado final dessa consulta. Em geral, cada estratégia existente apresenta aspectos diferenciados quanto à forma de roteamento de consultas em seus ambientes.

Em um PDMS, cada ponto possui um esquema que representa seu domínio de conhecimento, mapeamentos semânticos estabelecem a ligação entre os esquemas de dois pontos semanticamente relacionados. Para realizar uma consulta, o esquema

local do ponto é utilizado para sua formulação, e as respostas de cada consulta podem surgir de qualquer ponto na rede que tenha uma conexão a partir dos mapeamentos.

Rotear de maneira eficiente uma consulta em um PDMS significa reduzir o espaço de busca de uma consulta, a partir da escolha seletiva de pontos relevantes a semântica dessa consulta, obtendo as melhores respostas possíveis dentro do menor intervalo de tempo, sem causar sobrecarga no tráfego de informações na rede.

Neste sentido, serão elencados alguns problemas relacionados ao roteamento de consultas em um PDMS:

P1. Quantidade e diversidade de fontes disponíveis no sistema para consulta esse problema gera dificuldades para localização e seleção de pontos que possam contribuir com resultados à consulta. Pontos com afinidade semântica podem estar dispostos em pontos diversos e distantes uns dos outros. Em redes não estruturadas a técnica de *flooding* é utilizada para que a consulta seja encaminhada na rede para todos os pontos a partir do ponto que originou a consulta. Essa ação além de gerar um aumento do tráfego de informações na rede pode impedir que pontos com informações relevantes à consulta não sejam consultados em função do limite de TTL.

P2. Seleção dos pontos que podem responder à consulta – nesse aspecto, o problema está em estabelecer critérios que possam ser utilizados para identificar a relevância do ponto em relação à consulta, evitando assim, a seleção de pontos que possuam baixa similaridade semântica com a consulta.

P3. Estabelecer um mecanismo de interrupção no processo de roteamento – identificar um critério que cause a interrupção do mecanismo de roteamento evitando que a consulta seja encaminhada infinitamente e/ou sem chances de obter resultados realmente relevantes à consulta em questão.

Com base nesse cenário, será feita uma análise considerando a forma como os trabalhos apresentados nessa seção vêm tratando cada um dos problemas descritos anteriormente.

Quanto ao problema **P1** (Quantidade e diversidade de fontes disponíveis no sistema para consulta), a formação de agrupamentos semânticos a partir da afinidade semântica entre os pontos no sistema vêm sendo utilizada como uma alternativa para diminuir esse problema. Sendo assim, pontos semanticamente relacionados farão parte de um mesmo agrupamento. Em Ismail *et al.* [2011], o agrupamento formado com *super-peers*, restringe ainda mais o escopo da consulta uma vez que os *super-peers* mantêm um controle mais específico dos pontos que estão sob sua responsabilidade e reduzem de forma significativa a carga na rede. Os demais trabalhos fazem a formação de agrupamentos apenas em torno de pontos dispostos em uma topologia de rede pura.

Em se tratando do problema **P2** (Seleção dos pontos que podem responder à consulta), os trabalhos analisados utilizam como critério de seleção de pontos os elementos da consulta, restringindo assim o número de pontos a serem consultados durante o roteamento. Em geral, as estratégias analisadas adotaram índices semânticos na identificação dos pontos na rede. Essa solução tem se mostrado uma solução mais comum, mesmo que implementados de forma diferenciada como árvores [Ismail *et al.* 2011; Joung e Chuang 2009; Mandreolli *et al.* 2007a;], *Bloom Filter* [Li e Vuong 2007] ou RDV(*Resource Distance Vector*) [Li e Vuong 2007]. Outras adotam funções de similaridade para identificação dos vizinhos mais relevantes aos elementos da consulta [Kantere *et al.* 2009; Montanelli 2007; Montanelli *et al.* 2010], mapeamentos semânticos e critérios de qualidade (completude das respostas) [Roth e Naumann 2007].

Em relação ao problema **P3** (Estabelecer um mecanismo de interrupção no processo de roteamento), Joung e Chuang [2009] e Montanelli *et al.* [2010] utilizam um mecanismo de TTL para limitar o número de encaminhamento da consulta na rede; outros, como o trabalho do System P [Roth e Naumann 2006], utiliza um plano de orçamento (*budget*) em função do cálculo da completude das respostas. No H-Link [Montanelli 2007] e ESTEEM [Montanelli *et al.* 2010] é estabelecido um sistema baseado em crédito que corresponde ao número de respostas esperadas pelo ponto. Nos outros trabalhos não foi identificado um mecanismo de interrupção do roteamento.

O Quadro 4.1 mostra uma síntese da analise comparativa realizada entre as estratégias apresentadas nessa seção e os problemas descritos anteriormente. O significado de cada coluna do quadro é descrito a seguir:

- **Estratégia** – nome da estratégia
- **P1** – descreve o tipo de agrupamento utilizado na estratégia para resolver o problema P1 (Quantidade e diversidade de fontes disponíveis no sistema para consulta)
- **P2** – descreve a forma utilizada para resolver o problema P2 (Seleção dos pontos que podem responder a consulta).
- **Consulta** – forma como utiliza os elementos da consulta para selecionar pontos relevantes na rede.
- **Contexto** – descreve os elementos contextuais utilizados para seleção de pontos no roteamento.
- **Qualidade** – descreve os critérios de qualidade utilizados para seleção de pontos.
- **P3** - descreve o mecanismo utilizado pela estratégia para resolver o problema P3 (Estabelecer um mecanismo de interrupção no processo de roteamento).
- **Informações complementares** – outras informações relevantes da estratégia.

Quadro 4.1: Comparação entre as estratégias e os problemas de roteamento

Estratégia	P1	P2			P3	Informações complementares
		Consulta	Contexto	Qualidade		
KSP [ISMAIL *et al*. 2011;]	Super-peer	Índices semânticos	Não usa	Não usa	Não identificado	Agrupamento entre *super-peers* que possuem histórico de sucesso em resposta a alguns itens correspondentes aos componentes de uma consulta
Ontozilla [Joung e Chuang 2009]	peer	Índices semânticos	Não usa	Não usa	TTL	Tabela de roteamento (links semânticos e *peers* virtuais)
SRI [Mandreolli et al. 2007a]-	peer	Índices semânticos; mapeamentos semânticos	Não usa	Não usa	TTL; satisfação do objetivo	Randômico (obedecendo os mapeamentos semânticos) Índices de roteamento; satisfação do objetivo (uma medida de qualidade dos caminhos a serem explorados)
H-Link [Montanelli 2007	peer	Função de similaridade	Não usa	Não usa	baseado em crédito	Ontologia do ponto organizada em duas camadas: *network knowledge layer* e *content knowledge layer.;*contexto está relacionado ao conhecimento do ponto sobre os recursos compartilhados
OntoSum [Li e Vuong 2007]	peer	Índices semânticos	Não usa	Não usa	TTL	Roteamento *Inter-cluster* e *Intra-cluster;* Bloom filter, RDV(*Resource Distance Vector*)

System P	peer	Mapeamento semântico	Não usa	Completude da resposta	Completude da resposta	O modelo de completude envolve duas dimensões: cobertura e densidade
ESTEEM	peer	Função de similaridade	Não usa	Não usa	TTL e mecanismo de crédito	Contexto (CDT) e qualidade dos dados são utilizados na formação dos agrupamentos
GrouPeer	peer	Função de similaridade	Não usa	Degradação da Consulta	TTL	Mantém uma lista dos atributos eliminados durante as reformulações para que eles possam ser utilizados com os esquemas dos próximos pontos

4.4. Considerações

É fato que um dos grandes desafios em soluções para sistemas P2P está em tornar eficiente o roteamento da consulta. Rotear uma consulta nesses ambientes trata diretamente com questões do tipo: (1) localização de conteúdo, isto é, decidir para quais outros pontos a consulta deve ser encaminhada de forma a responder com eficiência e eficácia; (2) sistemas que enviam todas as consultas (algoritmos de inundação) para todos os pontos sofrem com questões de eficiência e escalabilidade; (3) consultas podem ter qualquer forma e, conforme o ponto onde tenha sido submetida, é preciso conhecer e confiar nos vizinhos semânticos para os quais serão encaminhadas; (4) tempo de resposta; (5) crescimento e dinamismo da rede (entrada e saída de pontos). E mais, em soluções de roteamento que utilizam índices, é preciso garantir a manutenção das tabelas, e considerar que até mesmo os processos de balanceamento e formação da rede podem ter impacto direto no mecanismo de roteamento utilizado.

Dessa forma, as pesquisas relacionadas a mecanismos de roteamento vêm se tornando um grande desafio. É preciso investigar as soluções de roteamento semântico aplicadas aos sistemas P2P. Além disso, pode-se verificar uma grande necessidade em pesquisar métricas que possam ser utilizadas para medir a relevância, o contexto e a qualidade de um ponto para execução de determinada consulta.

Capítulo 5

Proposta da Tese

Nesse capitulo será feita a apresentação da proposta deste trabalho com base nos conceitos e problemas descritos nos capítulos anteriores. Para tal, na Seção 5.1, serão realizadas algumas considerações sobre o uso do contexto e da qualidade para o processo de roteamento. Na Seção 5.2, será apresentada uma visão geral do processo de roteamento de consulta baseado em semântica. Na Seção 5.3, será apresentado o SPEED, sistema a ser utilizado como ambiente de aplicação, e na Seção 5.4 será feita uma descrição do algoritmo do processo de roteamento e sua exemplificação no SPEED. Por fim, nas seções seguintes, serão apresentadas as contribuições, metodologia e cronograma de realização para as atividades planejadas.

5.1. Contexto e Qualidade no Processo de Roteamento de Consultas

Neste trabalho, o uso de contexto será considerado em três perspectivas: contexto do usuário, contexto da consulta e contexto do ambiente. Em se tratando do contexto do usuário, poderão ser consideras as informações relacionadas ao perfil do usuário e suas preferências. No caso do contexto da consulta, a análise da perda semântica da consulta ocorrida durante a sua reformulação (reescrita da consulta de um ponto origem para um ponto destino tomando como base os esquemas exportados por cada um dos pontos). No caso do contexto do ambiente, elementos contextuais como disponibilidade e localização do ponto poderão ser considerados. Neste sentido, qualquer informação contextual considerada relevante ao processo deverá ser especificada e implementada para uso como critério adicional durante a seleção de pontos.

Neste trabalho, o uso de contexto será considerado em três perspectivas: contexto do usuário (por exemplo, perfil do usuário e suas preferências), contexto da consulta (por exemplo, a análise da perda semântica ocorrida durante a reformulação da consulta (reescrita da consulta de um esquema ponto para outro durante o roteamento) e contexto do ambiente (por exemplo, disponibilidade e localização). As informações contextuais consideradas relevantes ao processo deverão ser especificadas e implementadas para uso como critério adicional durante a seleção de pontos.

Quanto aos aspectos relacionados à qualidade da informação (QI), foi iniciado um processo de levantamento, definição e classificação de critérios de qualidade tomando como base os seguintes elementos de um PDMS [Freire *et al.* 2012]: os pontos, o esquema do ponto/ontologia (no caso dos PDMS que utilizam ontologia para representar seus esquemas [Pires 2009; Xiao 2006]), os mapeamentos, os dados e a resposta das consultas.

Nesse levantamento, foi observado que um grande número de abordagens tem apontado na direção da necessidade de obter e avaliar a QI com o objetivo de melhorar seus processos [Zaihrayeu 2006, Zhuge *et al.* 2005, Löser *et al.* 2003, Aberer *et al*. 2003b, Roth e Naumann 2005, Herschel e Heese 2005, Heese *et al.* 2005, Karnstedt *et. al.* 2008, Kantere *et al.* 2009, Montanelli *et al.* 2010, Green *et al.* 2010]. No entanto, apenas alguns trabalhos realmente utilizam QI de forma completa [Löser *et al.* 2003, Roth e Naumann 2005, Herschel e Heese 2005, Heese *et al.* 2005, Karnstedt *et. al.* 2008, Montanelli *et al.* 2010, Green *et al.* 2010], ou seja, indicando como tais informações podem ser obtidas, medidas e aplicadas.

Como resultado da análise realizada em Freire [2012], foi estabelecido um conjunto de critérios de qualidade agrupados por elementos de um PDMS. Considerando que o processo de roteamento deverá fazer uso dos aspectos relacionados à qualidade do ponto para redução do espaço de busca, neste capítulo, será dada apenas a definição do elemento ponto e dos critérios de qualidade correspondentes a este elemento, indicando como os mesmos podem ser medidos.

Um **ponto**, em um PDMS, é um elemento que representa uma fonte de dados autônoma que exporta seu esquema de dados ou apenas parte dele. Dessa forma,

cada ponto expressa e responde às consultas com base no esquema exportado. São critérios de qualidade do ponto:

- *Conectividade* – refere-se ao grau de disponibilidade do ponto em um dado intervalo de tempo. Pode ser medido considerando as estatísticas de conexão do ponto na rede, monitorando o número de vezes em que o ponto ficou indisponível ou até mesmo o percentual de vezes que o ponto esteve acessível.
- *Frequência de acesso* – refere-se ao número de acessos ao ponto em um determinado período. Pode ser medido a partir da razão entre o número de acessos ao ponto considerando um intervalo de tempo específico.
- *Tempo de atualização* – refere-se ao tempo decorrido desde a última atualização dos dados no ponto. Pode ser medido considerando a relação entre a última atualização e a data atual.
- *Reputação* – refere-se ao grau de importância obtido pela informação ou conteúdo da fonte armazenada no ponto. Pode ser medida em função do cálculo da média de consultas respondidas pelo ponto.
- *Relevância* – refere-se ao fator de adequação dos dados às respostas dos usuários. Este critério é subjetivo e dependente do usuário. Pode ser medido em função do *feedback* do usuário em relação aos resultados obtidos em uma consulta. A princípio pode-se considerar que o usuário informaria ao sistema a ocorrência de respostas inadequadas à consulta.
- *Confiança* – refere-se ao nível de confiança nos dados armazenados no ponto. Pode ser medida a partir de uma função que calcule a distorção entre os valores correspondentes à reputação e à relevância do ponto.

O Quadro 5.1 apresenta um resumo do conjunto de critérios de qualidade relacionados ao ponto e suas respectivas formas de medição.

Quadro 5.1:Conjunto de critérios de qualidade do ponto

Critério de Qualidade	Forma de Medição
Conectividade	Estatísticas de conexão do ponto na rede
Frequência de acesso	Número de acessos ao ponto em um determinado intervalo de tempo
Tempo de atualização	Relação entre a última atualização e data atual
Reputação	Cálculo da média de consultas respondidas pelo ponto;
Relevância	*Feedback* do usuário em relação aos resultados daquele ponto
Confiança	Relação entre a reputação do ponto e a sua relevância

5.2. O Processo Semântico de Roteamento de Consultas

Na definição de um processo para o roteamento de consultas é necessário estabelecer uma arquitetura de referência. Nesse sentido, para este processo, será considerado um PDMS formado por um conjunto de pontos semanticamente relacionados e organizado em uma rede pura e não estruturada. Nessa arquitetura, são pontos vizinhos aqueles relacionados entre si por meio de mapeamentos semânticos estabelecidos a partir da afinidade semântica entre os pontos.

Um dos principais problemas relacionados ao roteamento é a seleção de pontos que possam contribuir com respostas à consulta. Neste sentido, serão utilizados critérios baseados na medição do nível de similaridade semântica entre a consulta e o ponto, no uso de informações contextuais e de critérios de qualidade.

Para medir o nível de similaridade semântica entre a consulta e o ponto (para um PDMS cujos esquemas dos pontos são representados por ontologias), será necessário gerar a representação da consulta como uma ontologia e realizar a comparação entre a ontologia da consulta e a ontologia que representa o esquema do ponto a ser consultado. Depois de ter sido gerado o grau de similaridade, será preciso estabelecer um indicador de limite semântico (*semantic threshold*) que possa ser utilizado como referência no momento de indicar a relevância semântica do ponto para a consulta.

Para as informações contextuais e critérios de qualidade será necessário definir um modelo que represente e armazene tais informações para uso no processo.

Para que a consulta não seja encaminhada indefinidamente, será utilizado um mecanismo de interrupção do roteamento que considere a avaliação conjunta do valor de TTL (em unidade de tempo) e do nível de degradação da consulta [Delveroudis e Lekeas 2007]. Esta solução se baseia no fato de que adotar exclusivamente o TTL pode levar a situações em que pontos relevantes à consulta não sejam consultados (em virtude da interrupção ocasionada pelo TTL) ou que a consulta continue a ser encaminhada mesmo tendo perdido parte da sua semântica durante o processo de reformulação.

Durante o roteamento das consultas é importante que cada ponto possa aprender com o processo, para que, em outras interações, possam encaminhar a consulta para pontos que tenham fornecido "bons resultados" em consultas anteriores. Entretanto é preciso estabelecer critérios para medir os "bons resultados" de forma que pontos que retornam, apesar da quantidade, dados irrelevantes, não sejam considerados fortes candidatos em uma seleção posterior.

Diante desse cenário, a partir do ponto de submissão de uma determinada consulta no sistema, o processo semântico de roteamento de consultas deverá seguir as seguintes etapas:

1. No ponto onde se encontra a consulta, gerar uma lista de pontos candidatos (LPC) formada a partir do conjunto dos vizinhos semântico desse mesmo ponto.
2. Para cada ponto da LPC fazer uma verificação da similaridade semântica entre a consulta e o ponto. Neste momento, informações contextuais e critérios de qualidade serão considerados para auxiliar no processo de seleção dos pontos da seguinte forma:

   2.1. informações contextuais

   - **Usuário** – perfil do usuário (localização, tipo de usuário)
   - **Consulta** – de acordo com a semântica da consulta, o valor de degradação da mesma, calculado durante a reformulação entre dois pontos, será

utilizado como informação contextual e como critério para interrupção do roteamento. Dessa forma, caso ocorra alta perda semântica da consulta durante a reformulação essa não deverá ser encaminhada e, consequentemente, não será reformulada para um novo ponto.

- **Ambiente** – disponibilidade do ponto (situação do ponto no instante da consulta).

2.2. Critérios de qualidade

Verificar os critérios de qualidade relacionados ao **Ponto:** conectividade, frequência de acesso, tempo de atualização, reputação, relevância e confiança. O objetivo é obter medidas com base nesses critérios que possam ser utilizadas como referência para seleção do ponto.

3. Aplicados os critérios para seleção dos pontos, uma Lista de Pontos Relevantes (LPR) é formada.
4. A consulta é executada localmente, os valores para interrupção do roteamento são verificados (no caso será utilizado um valor de TTL e de degradação da consulta). Não acontecendo a interrupção, a consulta é reformulada, calculado o seu valor de degradação e encaminhada para os pontos da LPR. Cada ponto que recebe a consulta repete os passos 1, 2, 3 e 4.
5. Quando ocorrer a interrupção, as respostas são integradas e retornam ao ponto que encaminhou a consulta, no sentido inverso do caminho percorrido. Esse retorno é feito, até que cheguem ao ponto inicial de submissão da consulta original. Nesse ponto as respostas são integradas e apresentadas ao usuário.
6. O usuário poderá fornecer um *feedback* relacionado à sua satisfação com os resultados obtidos na sua consulta. Dessa forma, medidas de qualidade poderão ser estabelecidas e mantidas para uso em novas interações, permitindo que cada ponto melhore o seu conhecimento em relação a sua vizinhança semântica.

Para exemplificar, o processo descrito será instanciado no ambiente do SPEED, um PDMS que atende aos requisitos da arquitetura especificada para o processo. Para tal,

a próxima seção apresenta uma descrição desse sistema, sua arquitetura e principais características.

5.3. O Sistema SPEED

O SPEED (*Semantic PEEr Data Management System*) [Pires 2009] é um PDMS que adota uma abordagem semântica baseada em ontologias e informações contextuais [Souza 2009] com o propósito de prover soluções para problemas críticos de gerenciamento de dados em sistemas P2P, como conectividade, mapeamentos e processamento de consultas.

A Figura 5.1 mostra a arquitetura do SPEED. No sistema, existem três tipos de pontos: **ponto de dados**, **ponto de integração** e **ponto semântico**. Um **ponto de dados** representa uma fonte de dados (estruturada ou semi-estruturada) que compartilha dados com outros pontos de dados no sistema. Por exemplo, na Figura 5.1, são pontos de dados: $\mathbf{I_1D_1}$ e $\mathbf{I_1D_2}$. Pontos de dados são agrupados de acordo com seu domínio de conhecimento em *clusters* semânticos.

Cada ponto possui uma **ontologia local (OL)** que descreve o esquema exportado pelo ponto. Cada *cluster* semântico possui um ponto de integração que é responsável por tarefas como indexação de metadados, processamento de consultas e integração dos dados. **Pontos de integração** são pontos de dados com alta disponibilidade e poder computacional [Pires 2009]. A ontologia mantida pelo ponto de integração é denominada **ontologia do cluster (CLO)** e é gerada pela fusão (*merging)* das **OL** que representam os esquemas exportados pelos pontos de dados e ponto de integração.

Pontos de integração se comunicam com um **ponto semântico** que é responsável por armazenar e oferecer uma **ontologia da comunidade (OC)** que contém elementos de um domínio específico como, por exemplo, educação, saúde ou geografia. Quando um ponto de dados entra no sistema, o **ponto semântico** identifica um *cluster* apropriado onde o ponto deve ser conectado. Na Figura 5.1, o ponto $\mathbf{S_1}$ é um exemplo de ponto semântico. Desta maneira, uma **comunidade semântica** é composta de *clusters* que possuem interesses semânticos similares [Pires 2009].

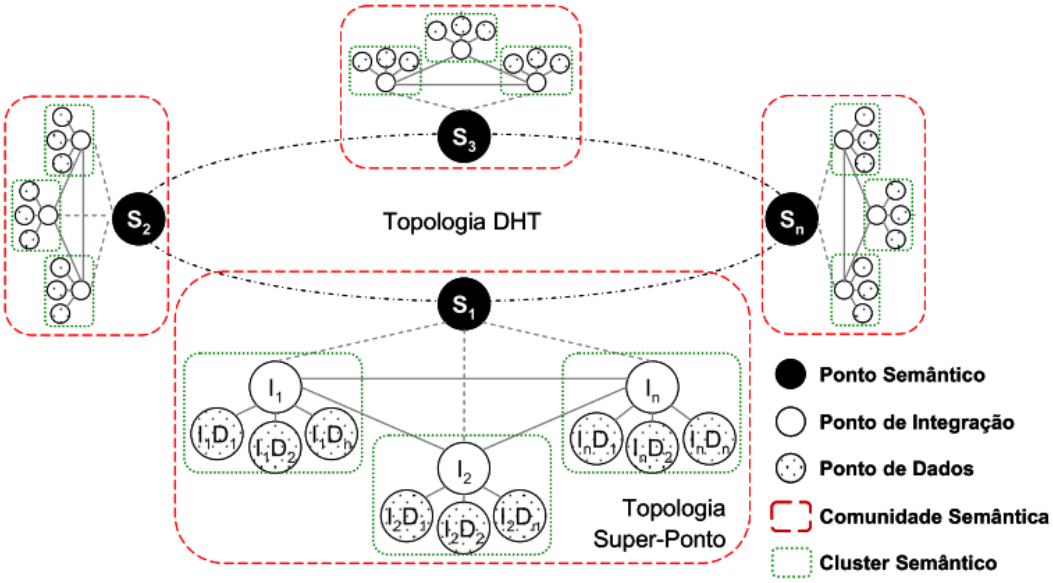


Figura 5.1: Arquitetura do SPEED [Pires 2009]

5.3.1. **Expressões de Mapeamentos e Correspondências**

O SPEED pode ser definido como uma tripla < {P}, {C}, {M}> onde, {P} representa o conjunto de pontos no sistema, {C} as correspondências semânticas entre os pontos e {M} um conjunto de expressões de mapeamentos dentro do *cluster*, como mostra a Figura 5.2.

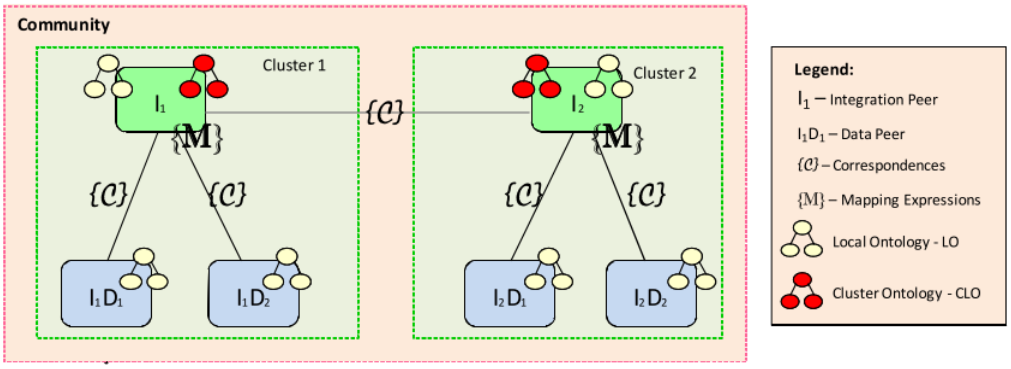


Figura 5.2: Expressões de Mapeamentos e Correspondências [Souza 2009]

Cada correspondência em {C} é definida de forma direcional a partir do conceito ou propriedade da CLO para o conceito ou propriedade da LO. Dentro de um *cluster*,

expressões de mapeamentos {M} são definidas entre um conceito da ontologia e as visões sobre os pontos de dados [Souza 2009].

Para definição das correspondências semânticas, o SPEED usa uma ontologia de domínio como *background knowledge*. As ontologias de domínio (*domain ontology - DO*) contêm conceitos e propriedades de um domínio de conhecimento específico, como, por exemplo, educação ou saúde. [Pires *et al.* 2011].

Inicialmente, os conceitos e propriedade das duas ontologias são mapeados para conceitos e propriedades equivalentes na DO. Dessa forma, as correspondências semânticas são inferidas baseadas nos relacionamentos existentes entre os elementos da DO.

Para especificar as correspondências entre ontologias são considerados os seguintes aspectos baseados na DO [Souza 2009]: o conhecimento semântico encontrado na DO; se os conceitos das ontologias compartilham super-conceitos na DO; se esses super-conceitos são diferentes da raiz; e a profundidade dos conceitos medidos em nós.

Considerando esses aspectos, uma correspondência semântica é definida como uma das seguintes expressões [Souza 2009]:

1. O_i:x $\equiv$ O_i:y, é uma correspondência *isEquivalentTo*
2. O_i:x $\sqsubset$ O_i:y, é uma correspondência *isSubConceptOf*
3. O_i:x $\sqsupset$ O_i:y, é uma correspondência *isSuperConceptOf*
4. O_i:x $\rhd$ O_i:y, é uma correspondência *isPartOf*
5. O_i:x $\lhd$ O_i:y, é uma correspondência *isWholeOf*
6. O_i:x $\approx$ O_i:y, é uma correspondência *isCloseOf*
7. O_i:x $\perp$ O_i:y, é uma correspondência *isDisjointWith*

Onde *x* e *y* são elementos (conceito ou propriedade) pertencentes às ontologias O_1 e O_2 que representam respectivamente ontologias de pontos vizinhos semanticamente relacionados.

5.3.2. **Processamento da Consulta**

Quando um usuário submete uma consulta em um determinado ponto, elementos contextuais são analisados com o objetivo de identificar qual dado é relevante para aquela situação específica do usuário [Souza 2009].

A informação contextual, no SPEED, é armazenada em um ontologia de contexto denominada CODI (*Contextual Ontology for Data Integration*) [Souza *et al.* 2008] que permite ao sistema organizar o conhecimento (isto é, seus elementos contextuais), reconhecer condições e auxiliar nas respostas às consultas.

Considerando o conjunto de correspondências definidas, é possível executar dois tipos de reformulação da consulta: a **reformulação exata**, que considera apenas as correspondências de equivalência (*isEquivalentTo*); a **reformulação enriquecida**, que é o resultado de todas os outros tipos de correpondências (*isSubConceptOf*, *isSuperConceptOf*, *isPartOf*, *isWholeOf*, *isCloseTo*, *isDisjointWith*). A existência de outras correspondências semânticas, diferentes da correspondência de equivalência, evita a produção de reformulações de consultas com resultados vazios e permite o enriquecimento desse processo [Souza 2009].

No processo de reformulação da consulta, dois tipos de contexto são utilizados: o **contexto do usuário** e o **contexto da consulta** [Souza 2009].

O **contexto do usuário** é obtido quando ele define suas preferências relativas à estratégia de reformulação da consulta. Essas preferências envolvem quatro variáveis que indicam o que deve ser considerado quando a consulta (Q) for enriquecida na reformulação: *Aproximação* (indica a inclusão de conceitos que são próximos dos conceitos em Q); *Especialização* (indica a inclusão de conceitos que são sub-conceitos de alguns conceitos de Q); *Generalização* (indica a inclusão de conceitos que são super-conceitos de alguns conceitos de Q) e *Composição* (indica a inclusão de conceitos que são parte-de ou todo-de alguns conceitos de Q).

O **contexto da consulta** é obtido por meio da análise de sua semântica (por exemplo, operadores utilizados) e por meio do modo de reformulação da consulta (estabelecido pelo usuário). No primeiro caso, os conceitos e construtores da consulta são identificados. O segundo, diz respeito à forma como o algoritmo de reformulação

irá operar: no modo expandido, onde ambas as reformulações exata e enriquecida serão realizadas; ou restrita, onde a prioridade é produzir uma reformulação exata. Nesse último caso, uma reformulação enriquecida também poderá ser realizada se a reformulação exata não produzir resultados.

O processo descrito na Seção 5.2, a ser instanciado no SPEED, irá considerar o roteamento da consulta no nível dos pontos de integração (*inter-cluster*) por se tratar de um ambiente formado por pontos semanticamente relacionados (através das correspondências semânticas) organizados em uma rede P2P pura e não estruturada. A próxima seção descreve a aplicação do processo no SPEED. Para tal, apresenta um algoritmo instanciado a partir do processo descrito, exemplificando sua aplicação no SPEED.

5.4. Aplicação do Processo de Roteamento de Consultas no SPEED

No SPEED, uma consulta submetida em um ponto de integração terá disponível todo o conjunto de dados que representa aquele *cluster* semântico. Nele, o processo de reformulação da consulta entre dois pontos (um origem e outro destino) utiliza o contexto de duas formas: contexto do usuário (quando o usuário define suas preferências relativas à estratégia de reformulação da consulta), contexto da consulta (análise da semântica da consulta e modo de reformulação da consulta) [Souza 2009]. Dessa forma, qualquer processo de roteamento que venha a ser aplicado no SPEED deverá preservar e garantir a propagação das variáveis de enriquecimento e do modo de execução de reformulação durante o encaminhamento da consulta por entre os pontos de integração no sistema.

5.4.1. **Algoritmo para o Roteamento Semântico da Consulta**

Essa seção apresenta uma primeira versão do algoritmo a ser aplicado no SPEED. Serão considerados, por simplicidade, como elemento contextual e critério de qualidade apenas as informações relacionadas à disponibilidade do ponto e reputação do ponto, respectivamente.

O Quadro 5.2 mostra a definição de cada variável utilizada e, na sequência, a descrição desse algoritmo.

Quadro 5.2: Definição das variáveis do algoritmo de roteamento semântico

Variáveis	Definição
DC	Degradação da consulta
Disp	Disponibilidade do ponto
LimDC	Limite aceitável de degradação da consulta
LimRep	Limite aceitável para reputação do ponto
LPC	Lista de Pontos de integração Candidatos
LPR	Lista de Pontos de integração Relevantes
LRC	Lista de Resultados da Consulta
MODE	Modo de reformulação (exata, enriquecida)
$\mathbf{O_{PIcorrente}}$	Ontologia do ponto de integração corrente
$\mathbf{O_{PIdestino}}$	Ontologia do ponto de integração destino
$\mathbf{O_{Q}}$	Ontologia de representação da consulta
$\mathbf{PI_{corrente}}$	Ponto de integração corrente
$\mathbf{PI_{destino}}$	Ponto de integração destino
$\mathbf{PI_{origem}}$	Ponto de integração de onde partiu a consulta
Pref_usr	Preferências na reformulação (aproximação, especialização, generalização e composição
Q	Consulta
Qr	Consulta reformulada
Rep	Reputação do ponto
resposta_consulta	Resposta da consulta
resultado	Resultado retornado pelo algoritmo
result_integrado	Resultado das respostas das consultas integrado
Sem_threshold	Indicador de limite para avaliação semântica (S*emantic threshold)*
SimCP	Grau de similaridade entre a consulta e o ponto
TTL	*Time-to-Live*, refere-se ao número de encaminhamentos da consulta

```
1.  Algoritmo Roteamento_Semântico (Q, PI_origem, PI_destino, TTL): result_integrado
2.  Inicio
3.      PI_corrente = PI_destino;
4.      Cria_inicializa (LRC, LPC, LPR) ;
5.      resposta_consulta = executa_consulta (Q, PI_corrente) ;
6.      armazena_resposta (resposta_consulta, LRC) ;
7.      LPC = { PI_destino ∈ vizinhos_semânticos(PI_corrente) } ;
8.      Para cada PI_destino ∈ LPC faça
9.      Inicio
10.         SimCP = calcula_similaridade (O_Q, O_PIdestino) ;
11.         Se  (SimCP > Sem_threshold) e  (contexto(PI_destino, Disp) = Sim) e
                qualidade (PI_destino, Rep) > LimRep)
12.          Então   Inclua (PI_destino, LPR);
13.     Fim
14.     Para cada PI_destino ∈ LPR faça
15.     Início
16.        Qr = SemRef (Q, PI_corrente, PI_destino, Co[O_corrente,O_PIdestino], MODE, Pref_usr);
17.        DC = calcula_degradação_consulta (Q, Qr);
18.        Se (TTL > 0)  e  (DC  <  LimDC) então
19.        Inicio
20.            resultado = Roteamento_Semântico (Qr, PI_corrente, PI_destino, TTL-1) ;
21.            armazena_resposta (resultado, LRC) ;
22.        Fim
23.     Fim
24.     Integra_resultado (LRC, result_integrado) ;
25.     retorna_resultado (result_integrado) ;
26. Fim.
```

O algoritmo ***Roteamento_Semântico*** tem quatro parâmetros de entrada: a consulta (**Q**), o ponto de onde se veio a consulta (PI_{origem}), o ponto para onde vai a consulta ($PI_{destino}$) e o valor de **TTL** para interrupção do roteamento. Para cada chamada do algoritmo, ao final, o resultado da execução da consulta naquele ponto é integrado considerando a execução local da consulta mais os resultados integrados de cada vizinho semântico. A partir da linha 3, o algoritmo executa as seguintes instruções: faz $PI_{corrente}$ igual a $PI_{destino}$, cria as listas utilizadas por cada ponto a media que recebe uma consulta (LPC,LPR,LRC) (linha 4), executa a consulta no ponto de integração corrente (ponto que está recebendo a consulta) (linha 5) e guarda o resultado na LRC (Lista de Resultados da Consulta) do ponto (linha 6). Em seguida, gera a lista de pontos candidatos (LPC) em função da vizinhança semântica do ponto de integração corrente (linha 7).

Da linha 8 a 13, para cada ponto de integração destino ($PI_{destino}$) que pertença a LPC, calcula a similaridade semântica entre a ontologia da consulta e a ontologia do ponto (linha 10); verifica se $PI_{destino}$ está disponível, se tem reputação aceitável e se a sua similaridade semântica com a consulta está acima de limite definido em S*em_threshold* (linha 11). Caso tais condições sejam verdadeiras, é feita a inclusão de $PI_{destino}$ na Lista de Pontos relevantes (LPR) (linha 12).

Da linha 14 a 23, para cada ponto de integração da LPR (considerados ponto destino ($PI_{destino}$) para envio da consulta) (linha 14), é gerada a reformulação da consulta utilizando o módulo *SemRef* [Souza 2009] com parâmetros que indicam a consulta (Q), o ponto de integração corrente ($PI_{corrente}$, de onde está partindo a consulta), o ponto de integração destino ($PI_{destino}$, para onde vai a consulta), conjunto de correspondências semânticas entre as ontologias dos pontos de integração corrente e destino, modo de reformulação da consulta (exata ou enriquecida) e preferências do usuário para a reformulação (aproximação, especialização, generalização ou composição) (linha 16).

Na linha 17, o valor de degradação (DC) é calculado considerando a perda semântica ocorrida durante a reformulação de Q (consulta original) em Qr (consulta reformulada). Na linha 18, os critérios para interrupção do roteamento são analisados, e, caso ocorra TTL maior do que zero e se DC estiver dentro do limite aceitável para degradação da consulta, o algoritmo será novamente instanciado para o ponto de integração que pertença a LPR, com os seguintes parâmetros: a consulta reformulada (Qr) passa a ser a consulta original (Q), o ponto de integração corrente passa a ser o ponto de origem da consulta, o ponto de integração destino passa a ser o ponto corrente e o TTL é decrementado de 1 (linha 20). O resultado retornado pelo algoritmo instanciado no ponto é armazenado na LRC (linha 21).

Na linha 24, após a LPR ter sido percorrida, os resultados da LRC (resultado da execução local da consulta mais os resultados integrados devolvidos por cada ponto da LPR) são integrados e retornados.

5.4.2. **Exemplificando o Processo**

A exemplificação do processo de acordo com o algoritmo definido anteriormente será feita em um cenário no domínio de educação. Inicialmente, valores hipotéticos serão definidos para configuração do ambiente.

Para as variáveis TTL, LimDC e Sem_threshold serão utilizados os valores 3, 0.8 e 0.5, respectivamente. O Quadro 5.3, relaciona parte dos conceitos presentes na ontologia de cada ponto de integração (PI_n) que tenham alguma correspondência com os conceitos que serão consultados neste exemplo.

Quadro 5.3: Conceitos pertencentes às ontologias dos pontos de integração

Pontos	Ontologia do ponto (conceitos relacionados a consulta do exemplo)
PI_1	{ Article, Book, ConferencePaper, Course, Department}
PI_2	{ ConferencePaper, Course, Department }
PI_3	{ Article, Book, ConferencePaper, Course, Department}
PI_4	{ Article, Book, ConferencePaper, Course}
PI_5	{ ConferencePaper, Course, Department }
PI_6	{ Book, Course}
PI_7	{ Book, ConferencePaper, Course, Department}
PI_8	{ Course}

A Tabela 5.1 mostra o grau de similaridade semântica (valor que indica a similaridade entre duas ontologias) existente entre os pontos de integração e, que por sua vez, definem a vizinhança semântica a ser considerada para cada ponto no SPEED que faça parte da comunidade de educação.

Tabela 5.1: Grau de similaridade semântica entre os pontos de integração

	PI_1	**PI_2**	**PI_3**	**PI_4**	**PI_5**	**PI_6**	**PI_7**	**PI_8**
PI_1		0.8	0.6	0.85				
PI_2	0.8				0.75	0.6		0.65
PI_3	0.6						0.8	
PI_4	0.85							
PI_5		0.75						0.6
PI_6		0.7					0.75	
PI_7			0.8			0.75		
PI_8		0.65			0.6			

Quanto ao uso de critérios de qualidade e informações contextuais, a Tabela 5.2 mostra os valores correspondentes à *Disponibilidade* (informação contextual) e à *Reputação* (critério de qualidade do ponto) no instante de submissão da consulta para todos os pontos na rede. Para esse exemplo, pontos com valor de reputação inferior a 0.5 não deverão compor a LPR por se tratarem de pontos inadequados à consulta.

Tabela 5.2: Valores relacionados à Disponibilidade e Reputação dos pontos

	PI_1	PI_2	PI_3	PI_4	PI_5	PI_6	PI_7	PI_8
Disponibilidade	Sim	Sim	Sim	Não	Sim	Sim	Sim	Sim
Reputação	0.85	0.8	0.88	0.8	0.3	0.9	0.7	0.8

Com base nas informações para configuração do ambiente apresentadas anteriormente, a Figura 5.3 mostra como ficaria a representação dos pontos na rede do SPEED. Nesse caso, estamos considerando apenas a visualização no nível dos pontos de integração na comunidade educação.

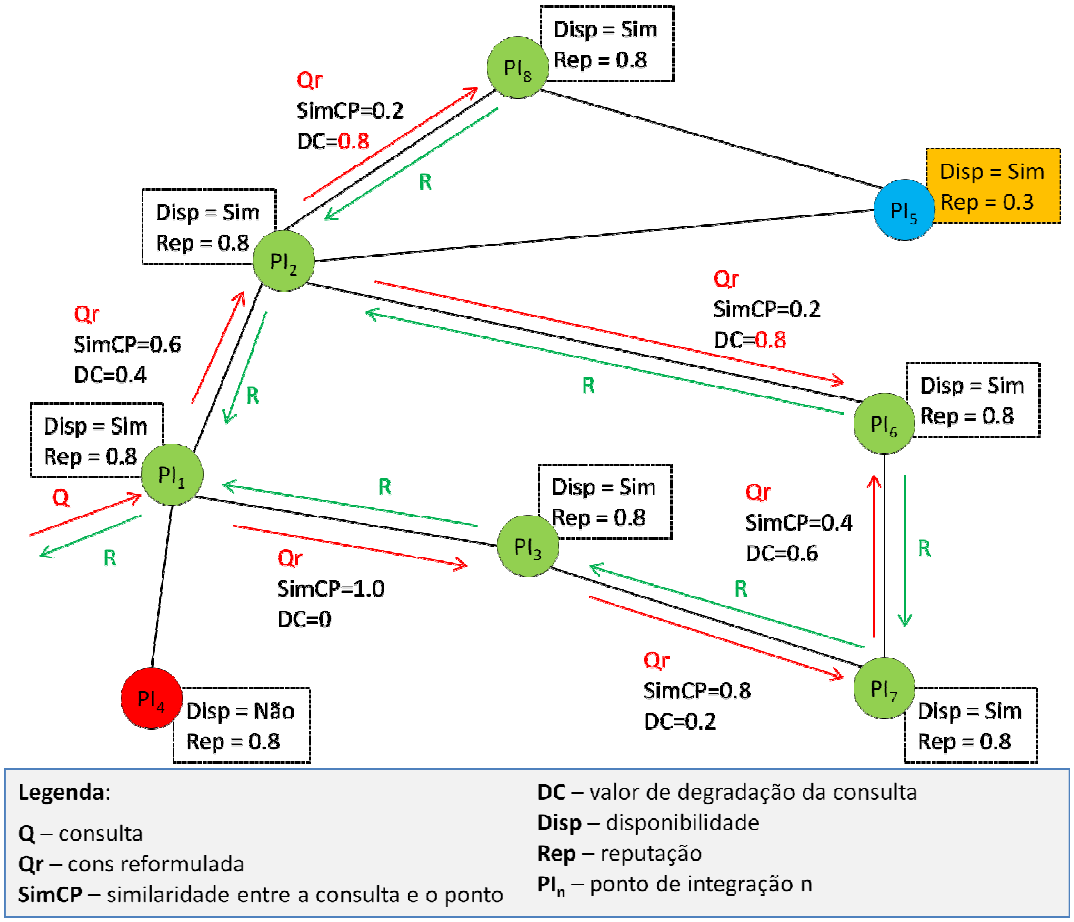


Figura 5.3: Vizinhança semântica em uma comunidade

Dado o cenário, considere que a consulta Q tenha sido formulada em PI_1 envolvendo os seguintes conceitos: (Article, Book, ConferencePaper, Course, Department).

Na Figura 5.3, **Q** representa a consulta, **Qr** a consulta reformulada de acordo com os vizinhos semânticos entre os pontos de origem e destino da consulta e **R**, a resposta da consulta. O grau de similaridade entre a consulta e o ponto para o qual a consulta será encaminhada ($\mathbf{PI_{destino}}$) é indicado por **SimCP** e foi estimado considerando o percentual de conceitos equivalentes entre a consulta e a ontologia do ponto. **DC** indica o grau de degradação da consulta após a reformulação e foi estimado, considerando o percentual de conceitos não equivalentes, identificados durante a reformulação da consulta do ponto origem para um ponto destino. Na Figura 5.3, existe um quadro relacionado a cada ponto, com **Disp** e **Rep** que representam respectivamente a disponibilidade (informação contextual) e a reputação do ponto (critério de qualidade relativo ao ponto) no instante da consulta.

A Tabela 5.3 mostra os graus de similaridade entre a consulta e o ponto (**SimCP**), e o grau de degradação da consulta (**DC**) apresentados na Figura 5.3. Cada valor representado na tabela corresponde aos valores (hipotéticos) obtidos à medida que a consulta chega em um ponto e, em seguida, é reformulada e encaminhada para um ponto destino. Por exemplo, o valor de **DC** igual a 0.8, para a linha com valor $\mathbf{PI_2}$ e coluna $\mathbf{PI_6}$, refere-se à degradação da consulta em função da reformulação ocorrida entre $\mathbf{PI_2}$ e $\mathbf{PI_6}$, que deve ser usado como mecanismo de interrupção do roteamento no ponto $\mathbf{PI_6}$.

Tabela 5.3: Valores de Similaridade e de degradação da Consulta

			Ponto destino da consulta							
			PI_1	PI_2	PI_3	PI_4	PI_5	PI_6	PI_7	PI_8
Ponto origem da consulta	PI_1	SimCP	1.0	0.6	1.0	0.8				
		DC	0	0.4	0	0.2				
	PI_2	SimCP					0.6	0.2		0.2
		DC					0.4	**0.8**		**0.8**
	PI_3	SimCP							0.8	
		DC							0.2	
	PI_4	SimCP								
		DC								
	PI_5	SimCP								0.2
		DC								**0.8**
	PI_6	SimCP							0.4	
		DC							0.6	
	PI_7	SimCP								0.4
		DC								0.6
	PI_8	SimCP					0.2			
		DC					**0.8**			

Para a consulta submetida em **PI_1** são identificados os vizinhos semânticos (**PI_2, PI_3, PI_4**) com os respectivos graus de similaridade semântica (**0.8, 0.6, 0.85**) que formam a **LPC** (*Lista de Pontos Candidatos*) do ponto. Para o roteamento é importante observar que embora **PI_3** tenha grau de similaridade semântica com **PI_1** inferior ao grau entre **PI_1** e **PI_2**, sua relevância de acordo com a semântica da consulta é superior a **PI_2**, ou seja, **PI_3** tem o valor de **SimCP** igual a **1.0** enquanto que em **PI_2** o valor de **SimCP** é igual a **0.6**. Identificada a **LPC** é preciso gerar a **LPR** (*Lista de Pontos Relevantes*) considerando agora, os de similaridade semântica entre a consulta e o ponto, aspectos contextuais e de qualidade. Como **PI_4** encontra-se indisponível (**Disp=Não**), não existem restrições relacionadas ao critério de qualidade reputação do ponto e todos os pontos possuem **SimCP** superior a 0.5 (*Sem_threshold*),), a **LPR** passa a ser formada apenas pelos pontos (**PI_2** e **PI_3**).

Após formação da **LPR** em **PI_1**, a consulta é reformulada (**Qr**), preservando e garantindo o contexto definido em sua submissão e, encaminhada para cada um dos pontos da **LPR** (**PI_2** e **PI_3**). Cada ponto que recebe a consulta deverá executar a consulta localmente, armazenar em sua LRC (Lista de Resultados da Consulta), encaminhar para os seus pontos vizinhos, armazenar os resultados retornados por seus vizinhos em sua LRC, integrar os resultados da LRC e retornar o seu resultado para

o ponto que enviou a consulta. Antes do envio da consulta para os pontos da LPR, o valor de **TTL** (*Time-To-Live*) e de degradação da consulta (**DC**) deverão ser verificados para que se possa identificar o critério de interrupção do roteamento. Para o **TTL**, verificar se o mesmo atingiu o valor **0** (em números de saltos). No caso da degradação da consulta, se o valor de **DC** for maior ou igual a **0.8**, o roteamento deverá ser interrompido naquele ponto. Se o valor de **DC** for menor do que **0.8**, novamente formar a **LPC** a partir daquele ponto, em seguida, verificar os aspectos contextuais e de qualidade de cada ponto que compõe a **LPC**, calcular o valor de **SimCP** e gerar a **LPR**. A verificação dos aspectos contextuais e de qualidade, para o exemplo, está relacionada à identificação de pontos disponíveis (informação contextual) e de reputação superior a **0.5** (critério de qualidade). A interrupção do roteamento em cada ponto significa que será preciso integrar os resultados da LRC naquele ponto e retornar o resultado integrado para o ponto que enviou a consulta.

Quando a consulta chega em $\mathbf{PI_3}$ é executada localmente, seu resultado armazenado na LRC do ponto $\mathbf{PI_3}$, os valores de **TTL** e **DC** são verificados e, caso não ocorra interrupção do roteamento, a consulta é reformulada e encaminhada para $\mathbf{PI_7}$. O ponto $\mathbf{PI_7}$, novamente, executa, armazena em sua LRC, verifica **TTL** e **DC**, reformula e encaminhada para $\mathbf{PI_6}$ que executa a consulta e armazena em sua LRC. Ao chegar em $\mathbf{PI_6}$, o valor de **TTL** atinge o limite estimado (**TTL=0**) e o roteamento nesse ponto é então interrompido. A cada execução local da consulta os valores da LRC são integrados e retornados ao ponto que enviou a consulta no sentido inverso do caminho percorrido até atingir o ponto inicial de submissão da consulta. Em todos os casos considerando a etapa de integração da LRC de cada ponto e envio dos resultados ao ponto que enviou a consulta .

A consulta submetida em $\mathbf{PI_1}$, como descrito no exemplo gerou uma **LPR** com dois pontos o $\mathbf{PI_2}$ e $\mathbf{PI_3}$. Finalizado o acompanhamento do processo em $\mathbf{PI_3}$, será analisado, de agora em diante, o roteamento a partir do ponto $\mathbf{PI_2}$.

Ao chegar em $\mathbf{PI_2}$, a consulta é executada localmente, seu resultado armazenado em sua LRC e os valores de **TTL** e **DC** verificados. Em seguida, são identificados os pontos vizinhos que deverão formar a **LPC** ($\mathbf{PI_8}$, $\mathbf{PI_5}$, $\mathbf{PI_6}$). Entretanto, $\mathbf{PI_5}$ apesar de disponível, não atende ao critério de escolha de pontos baseado em sua reputação, no

caso, inferior a **0.5**. Nesse caso, a **LPR** gerada será formada pelos pontos $\mathbf{PI_8}$ e $\mathbf{PI_6}$. Para cada ponto da **LPR** a consulta é reformulada e roteada. Em $\mathbf{PI_8}$, executada localmente e, apesar de não ter atingido o valor de **TTL** estimado é verificado que a consulta após reformulação sofreu uma degradação de **0.8**, o que consequentemente deverá gerar uma interrupção do roteamento. Da mesma forma ocorre para a consulta encaminhada para $\mathbf{PI_6}$.

5.5. Contribuições

A principal contribuição desta tese é a definição de modelos, técnicas e algoritmos que façam uso de informações contextuais e de critérios de qualidade, para o roteamento semântico de consultas em um PDMS. Nesse sentido, as contribuições esperadas deste trabalho podem ser descritas como segue:

- **Identificação de pontos baseada na similaridade semântica entre a consulta e o ponto**

  Especificar e implementar algoritmos para conversão da consulta em uma representação de ontologia e posterior verificação do grau de similaridade semântica entre a ontologia que representa a consulta e a ontologia do ponto. Estabelecer um indicador para avaliação semântica (S*emantic threshold*).

- **Especificação e implementação de um modelo para representação do contexto e da qualidade**

  Especificar e implementar um modelo que permita manter, representar e avaliar as informações de contexto e critérios de qualidade relacionados ao sistema para serem utilizadas durante o roteamento.

- **Especificação e implementação de um mecanismo de interrupção do roteamento baseado em semântica**

  Definir um indicador baseado em semântica que possa ser utilizado em conjunto com mecanismos de TTL para controle do roteamento da consulta. Nesse caso o nível de degradação da consulta poderá ser utilizado, em conjunto com TTL, como critério de parada durante o encaminhamento da consulta.

5.6. Metodologia

Serão realizadas pesquisas, especificação, desenvolvimento e testes de algoritmos. Para validar o trabalho serão utilizados testes de aplicação com base nos requisitos de qualidade e contexto. A metodologia a ser seguida deverá contemplar as seguintes etapas:

- **1ª Etapa** – Identificar e definir um modelo de representação e armazenamento para os elementos contextuais e de qualidade.
- **2ª Etapa** – Definir e implementar o processo de roteamento de consultas baseado em semântica.
- **3ª Etapa** – Validar Resultados
- **4ª Etapa** – Escrita da tese
- **5ª Etapa** – Preparação e defesa da tese.

5.7. Cronograma

Apresentamos nesta seção o cronograma para a realização do trabalho proposto. Neste cronograma encontram-se os prazos dados para cada uma das atividades já realizadas e para aquelas que ainda serão desenvolvidas. As atividades serão distribuídas considerando o período de Julho de 2009 à Junho de 2013.

5.7.1. **Atividades Realizadas**

I. Créditos em disciplinas
II. Estudo sobre aspectos semânticos (ontologia e contexto) e de qualidade da informação e sua aplicação em PDMS
III. Estudo sobre estratégias de roteamento de consultas em sistemas P2P
IV. Estudo sobre estratégias de roteamento de consultas baseada em semântica em PDMS com ligações semânticas entre pontos
V. Preparação para a Qualificação e Proposta de Tese
VI. Defesa da Qualificação e Proposta de Tese

5.7.2. **Atividades a serem realizadas**

VII. Definição do processo de roteamento de consultas baseada em semântica

VIII. Especificação e implementação de um modelo para armazenar informações contextuais e critérios de qualidade para uso na processo de roteamento

IX. Preparação do ambiente para realização dos experimentos

X. Implementação do processo de roteamento especificado

XI. Testes e avaliação do processo

XII. Escrita de artigos científicos

XIII. Escrita da tese

XIV. Defesa da tese

O Quadro 5.4, a seguir, ilustra a distribuição das atividades ao longo deste doutoramento. Para facilitar a visualização do quadro será usada a seguinte legenda: (R) - para as atividades realizadas e (SR) – para as atividades a serem realizadas.

Quadro 5.4:Distribuição das atividades

Atividades	**2009**											
	Jan	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**	**Jul**	**Ago**	**Set**	**Out**	**Nov**	**Dez**
I							R	R	R	R	R	R
	2010											
I	R	R	R	R	R	R	R	R	R	R	R	R
II								R	R	R	R	R
	2011											
II	R	R	R									
III				R	R	R						
IV						R	R	R	R			
V									R	R	R	R
XII					R	R						
	2012											
VI	R	R	R	SR								
VII				SR	SR	SR	SR	SR	SR	SR		
VIII					SR	SR	SR	SR	SR	SR		
IX										SR	SR	SR
X											SR	SR
XII	R	R	SR	SR	SR	SR	SR	SR	SR	SR	SR	SR
	2013											
XI	SR	SR	SR									
XII	SR	SR	SR	SR	SR	SR						
XIII	SR	SR	SR	SR	SR							
XIV						SR						

Referências Bibliográficas

Aberer K., Cudré-Mauroux P., Datta A., Despotovic Z., Hauswirth M., Punceva M., Schmidt R. (2003a). P-Grid: a self-organizing structured P2P system". ACM SIGMOD Record, v. 32, n. 3, p. 29-33

Aberer K., Cudre-Mauroux P., Hauswirth M. (2003b). The Chatty Web: Emergent Semantics Through Gossiping, in: WWW, p. 197-206.

Aiello C., Baldoni R., Bianchini D., Bolchini C., Bonomi S., Castano S., Curino C. A. (2007). The ESTEEM Architecture for Emergent Semantics and Cooperation in MultiKnowledge Environments. Relatório Técnico. MIUR PRIN Esteem Project. Università Degli Studi Di Milano, Italy.

Akbarinia R., Pacitti, E., Valduriez P.(2007). Query processing in P2P systems. Relatório Técnico de Pesquisa. n. 6112, INRIA – Institute National de Recherche en Informatique et en Automatique – ISSN 0249-6399

Androutsellis-Theotokis S., Spinellis D.(2004). A survey of peer-to-peer content distribution technologies. ACM Computer Survey, ACM, New York, NY, USA, v. 36, n. 4, p. 335–371.

Arazy O., Kopak R. (2011). On the Measurability of Information Quality. Journal of the American Society for Information Science, v. 62, n. 1, p. 89-99.

Baldauf M., Dustdar S., Rosemberg F., (2007). A Survey On Context-Aware Systems. International Journal of Ad Hoc and Ubiquitous Computing, v. 2, n. 4, p. 63-277, Inderscience Publishers.

Batini C., Scannapieco M. (2006). Data Quality: Concepts, Methods, and Techniques. (Chapther 2). Springer, 2006.

Batista M. C. (2008). Schema Quality Analysis in a Data Integration System. Tese de doutorado, Universidade Federal de Pernambuco, Brasil.

Bayardo R. J., Bohrer W., Brice R. S., Cichocki A., Fowler J., Helal A., Kashyap V., Ksiezyk T., Martin G., Nodine M. H., Rashid M., Rusinkiewicz M., Shea R., Unnikrishnan C., Unruh A., Woelk D. (1997). Infosleuth: semantic integration of information in open

and dynamic environments. In: Peckham J (ed) SIGMOD 1997, Proceedings ACMSIGMOD international conference on management of data, USA, ACM Press, p. 195– 206.

Belian R. B. (2008). A Context-based Name Resolution Approach for Semantic Schema Integration. Tese de doutorado. Universidade Federal de Pernambuco.

Bellotti V., Edwards K. (2001). Intelligibility and Accountability: Human Considerations in Context-Aware Systems. Human Computer Interaction, v. 16, n. 2-4, p. 193-212

Beneventano D., Bergamaschi S., Guerra F., Vincini M. (2007). The SEWASIE Network of Mediator Agents for Semantic Search. Journal of Universal Computer Science, v. 13, n. 12, p.1936-1969.

Bergamaschi S., Castano S., Beneventano D., Vincini M. (2001). Retrieving grating data from multiple sources: the MOMIS approach. Data Knowledge. Eng. 36, p. 215– 249.

Bernstein P., Giunchiglia F., Kementsietsidis A., Mylopoulos J., Serafini L., Zaihrayeu I. (2002). Data Management for Peer-to- Peer Computing: A Vision. In Proc. of the 5th International Workshop on the Web and Databases, Wisconsin, USA.

Bianchini D., De Antonellis V., Melchiori M. (2009). P2P-SDSD: On-the-fly service-based collaboration in distributed systems. Int Journal of Metadata, Semantics and Ontologies, v. 5, n. 3, p. 222–237.

Bloom B. (1970). Space/time trade-offs in hash coding with allowable errors. Commun. ACM, v. 13, n. 7, p. 422–426, 1970.

Brézillon P. (1999). Context in Artificial Intelligence: IA Survey of the Literature. Computer&Artificial Intelligence, v. 18, p. 321-340.

Brézillon P. (2003): Context Dynamic and Explanation in Contextual Graphs. Proceedings of the 4th International and Interdisciplinary Conference, CONTEXT 2003, USA, p. 94-106.

Calvi C., Pessoa R., Filho J. (2005). Um Interpretador de Contexto para Plataformas de Serviços Context-Aware, In Anais do XXXII SEMISH, XXV Congresso da SBC, São Leopoldo, RS.

Cantele R., Adamatti D., Grigas M., Sichman J. (2004). Reengenharia e ontologias: análise e aplicação. In Anais do I Workshop de Rede Semântica (WWS2004), Brasília.

Castano S., Montanelli S., (2008). Semantically routing queries in peer-based systems: the H-Link approach. In Jounal The Knowledge Engineering Review, v. 23, p. 51-72.

Castano S., Ferrara A., Montanelli S. (2006). Web Semantics and Ontology, chapter Dynamic Knowledge Discovery in Open, Distributed and Multi-Ontology Systems: Techniques and Applications. Idea Group.

Castano S., Ferrara A., Montanelli S., Racca G. (2004). Semantic Information Interoperability in Open Networked Systems. In Proc. of the Int. Conference on Semantics of a Networked World (ICSNW 2004), Paris, France.

Ciglaric M., Vidmar T. (2006). Ant-inspired query routing performance in dynamic peer-to-peer networks. Parallel and Distributed Processing Symposium, 20th International, IEEE Computer Society, p. 287.

Clarke I., Miller S., Hong T. W., Sandberg O., Wiley B. (2002). Protecting free expression online with Freenet. IEEE Internet Computing, v. 6, n. 1, p. 40-49.

Costa L. R. (2009). Roteamento de consultas em bancos de dados peer-to-peer utilizando colônias de formigas e ontologias. Dissertação de Mestrado. Universidade Estadual Paulista. São José do Rio Preto, Brasil.

Crespo A., Garcia-Molina H. (2003). Semantic Overlay Networks for P2P Systems. In Proceedings of the 29th VLDB Conference, Germany.

Delveroudis Y., Lekeas, P. V. (2007). Managing Semantic Loss during Query Reformulation in Peer Data Management Systems. Computer Engineering, p. 61-66.

Dey A. K. (2000). Providing architectural support for building context-aware applications. Ph.D. Thesis, Georgia Institute of Technology.

Dey A. K., Salber D., Abowd G. D. (2001). A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications, Human Computer Interaction Journal, v. 16, n. Special Issue on Context-Aware Computing, p. 97-166.

Doval D., O'Mahony D. (2003) Overlay Networks: A Scalable Alternative for P2P. IEEE Internet

Computing, v. 7, n. 4, p. 79-82.

Euzenat J., Shvaiko P. (2007). Ontology matching. Springer Publishing Company, Incorporated. ISBN 3642080553 9783642080555

Fensel D. (2001). Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce. ISBN: 3540416021. Ed. Springer.

Fiorano. (2003). Super-Peer Architectures for Distributed Computing. White Paper, Fiorano Software, Inc. Disponivel em http://www.fiorano.com/whitepapers/superpeer.pdf.

Freire C., Souza B. F. F., Souza D., Batista M. C. M., Salgado A. C. (2012). Information Quality Criteria for Peer Data Management Systems – A Survey. Sumetido em 15/02/2012. Journal Information System. Special Issue on Data Quality.

Gnutella (2011). The Gnutella System web site http://www.gnutella2.com/. Último acesso em outubro de 2011.

Green T.J., Ives Z.G., Tannen V. (2010). Provenance in ORCHESTRA. In Genomics, p. 1-8.

Gruber T. (1993). A Translation Approach to Portable Ontology specification. In Knowledge Acquisition, v. 5, n. 2, p. 199-220.

Gruber T. (1995). Towards principles for the design of ontologies used for knowledge sharing. In International Journal of Human and Computer Studies, v.43, n.5/6, p. 907-928.

Guarino N. (1998). Formal Ontology and Information Systems. In Proc. of FOIS'98, Trento, Italy, p. 3-15.

Haase P., Siebes R., Harmelen F. V. (2008). Expertise-Based Peer Selection. In Journal of Knowledge and Information Sytems., v. 1, p.75-107.

Halevy A. Y., Ives Z., Suciu D., Tatarinov I. (2003). Schema mediation in peer data management systems. In Proc. Of the Int. Conf. on Data Engineering (ICDE2003), p. 505–516.

Halevy A., Rajarama A., Ordille J. (2006). Data Integration: The Teenage Years. Proceedings of the 32nd International Conference on Very large data bases, p. 9-16. Seoul, Korea.

Heese R., Herschel S., Naumann F., Roth A. (2005). Self-extending Peer Data Management. In Proceedings of The German Conference on Datenbanksysteme in Business,

Technologie und Web, v. 65 of LNI. GI.

Herschel S., Heese R. (2005). Humboldt Discoverer: A Semantic P2P index for PDMS. In Proceedings of the International Workshop Data Integration and the Semantic Web. Porto, Portugal. Disponivel em http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.503&rep=rep1&type=pdf.

Hose K., Klan D., Sattler K. (2006). Distributed Data Summaries for Approximate Query Processing in PDMS. In Proc of 10th International Database Engineering and Applications Symposium (IDEAS'06), Delhi, India

Ismail A., Quafafou M., Durand N., Nachouki G. Hajjar M. (2011). Queries Routing In Super-Peer-Based System : Simulation and Evaluation. In Journal of Emerging Technologies in Web Intelligence, v. 3, n. 3.

Joseph S. (2002). NeuroGrid: Semantically Routing Queries in peer-to-peer Networks, in Proc. of the Int. Workshop on peer-to-peer Computing , Pisa, Italy.

Joung Y, Chuang F. (2009). OntoZilla: An ontology-based, semi-structured, and evolutionary peer-to-peer network for information systems and services. in Journal Future Generation Computer Systems, v. 25, n. 1, p. 53-63.

Kamienski C., Souto E., Rocha J., Domingues M., Callado A., Sadok D. (2005). Colaboração na Internet e a Tecnologia Peer-to-Peer. In XXV Congresso da Sociedade Brasileira de Computação – SBC2005, São Leopoldo, RS.

Kantere V., Tsoumakos D., Sellis T., Roussopoulos N. (2009). GrouPeer: Dynamic Clustering of P2P Databases. Information Systems Journal, v. 34, n. 1, p. 62 – 86.

Karnstedt M., Sattler K., HaB M., Hauswirth M., Sapkota B., Schmidt R. (2008). Approximating Query Completeness by Predicting the Number of Answers in DHT-based Web Applications, in: 10th ACM WIDM, p. 71-78.

Kazaa (2011). The Kazaa System web site, http://www.kazaa.com/us/index.htm. Último acesso em outubro de 2011.

Li J., Vuong S. (2007). OntSum : A Semantic Query Routing Scheme in P2P Networks Based on Concise Ontology Indexing. In Proc of 21st International Conference on Advanced

Networking and Applications(AINA'07), p. 94-101, Ontario, Canada.

Lodi S., Mandreoli F., Martoglia R., Penzo W., Sassatelli S. (2008). Semantic Peer: Here are the Neighbors You Want!. Proceedings of the 11th International Conference on Extending Database Technology (EDBT '08), p. 26-37, Nantes, France.

Löser A., Naumann F., Siberski W., Nejdl W., Thaden U. (2003). Semantic Overlay Clusters within Super-Peer Networks. In Proc. of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing in Conjunction with the VLDB 2003, Berlin, Germany.

Lv Q., Cao P., Cohen E., Li K., and Shenker S. (2002). Search and Replication in Unstructured Peer-to-Peer Networks. In Proc. of the 16th ACM International Conference on Supercomputing (ICS'02), New York, USA.

Maedche A. (2002). Ontology Learning for the Semantic Web. Kluwer Academic Publishers.

Mandreoli F., Martoglia R., Penzo W., Sassatelli S. (2007a). SRI @ work : Efficient and Effective Routing Strategies in a PDMS. In Proc. of Web Information Systems Engineering – WISE 2007 8th International Conference on Web Information Systems Engineering, p. 285-297, Nancy, France.

Mandreoli F., Martoglia R., Penzo W., Sassatelli S. (2006). Semantic Query Routing Experiences in a PDMS. SWAP 2006 - Semantic Web Applications and Perspectives, Proceedings of the 3rd Italian Semantic Web Workshop, Scuola Normale Superiore, Pisa, Italy.

Mandreoli F., Martoglia R., Penzo W., Sassatelli S., Villani G. (2007b). SUNRISE: Exploring PDMS Networks with Semantic Routing Indexes. In 4th European Semantic Web Conference, Innsbruck, Austria.

Mazak A., Schandl B. e Lanzenberger M.(2010). align++ A Heuristic-based Method for Approximating the Mismatch-at-Risk in Schema-based Ontology Alignment. In Proc of the International Conference on Knowledge Engineering and Ontology Development, p. 17-26.

Michlmayr E. (2006). Ant algorithms for search in unstructured peer-to-peer networks. In: ICDEW '06: Proceedings of the 22nd International Conference on Data Engineering

Workshops. USA: IEEE Computer Society, p. 142–146.

Montanelli S. (2007). Emergent Communities for Semantic Collaboration in Multi-Knowledge Environments : Methods and Techniques. Ph.D. Thesis, Università Degli Studi Di Milano, Italy.

Montanelli S., Bianchini D., Aiello C., Baldoni R., Bolchini C., Bonomi S., Castano S., Catarci T., Antonellis V., Ferrara A., Melchiori M., Quintarelli E., Scannapieco M., Schreiber F., Tanca L. (2010). The ESTEEM platform: enabling P2P semantic collaboration through emerging collective knowledge, Journal of Intelligent Information Systems. v. 36, n. 2.

Montanelli S., Castano S. (2008). Semantically routing queries in peer-based systems: the H-Link approach. The Knowledge Engineering Review, v. 23, n.1, p. 51-72. Cambridge University Press

NAPSTER. (2010). Último acesso em novembro de 2010, disponível em http://www.napster.com

Nejdl W, Wolpers M, Siberski W, Schmitz C, Schlosser M, Brunkhorst I, Loser A (2003). Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In Proceedings of the 12th international world wide web conference, Budapest, Hungary, cite- seer.nj.nec.com/nejdl02superpeerbased.html

Noy N., McGuinness D. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory, Technical Report KSL-01-05 and Stanford Medical Informatics, Technical Report SMI-2001- 0880.

Penzo W., Lodi S., Mandreoli F., Martoglia R., Sassatelli S. (2008). Semantic peer, here are the neighbors you want!. In Proceedings of the 11th international conference on Extending database technology Advances in database technology - EDBT '08. New York, New York, USA: ACM Press. doi:10.1145/1353343.1353351.

Pipino L. L, Lee Y., Wang R. (2002). Data Quality Assessment, Communications of the ACM, v. 45, n. 4, p. 211-218.

Pires C. (2007). Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semantica. Trabalho de Qualificacao e Proposta de Tese. Universidade Federal de

Pernambuco.Recife, Brasil.

Pires C. (2009). Ontology-based Clustering in a peer Data Management System. Tese de Doutorado. Universidade Federal de Pernambuco. Recife, Brasil.

Pires C. E., Souza D., Kedad Z., Bouzeghoub M., Salgado A. C. (2009). Using Semantics in Peer Data Management Systems. In Knowledge Creation Diffusion Utilization, p. 2579-2582.

Pires C. E., Souza D., Lóscio B., Belian R., Tedesco P., Salgado A. C. (2011). Using Ontologies to Enhance Data Management in Distributed Environments. In Ibero American Meeting on Ontological Research, Gramado, Brazil.

Power R. (2003). Topic Maps for Context Management. In International Symposium on Information and Communication Technologies (ISICT 2003), p. 199-204.

Ratnasamy S., Francis P., Handley M., Karp R., Shenker S. (2001). A scalable content-addressable network. ACM SIGCOMM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications, p. 161-172, INRIA.

Risson J., Moors T. (2006). Survey of Research towards Robust Peer-to-Peer Networks. The International Journal of Computer and Telecommunications Networking archive, vol. 50, pp. 1-36. New York, USA.

Roth A., Naumann F.(2007). System P : Completeness-driven Query Answering in Peer Data Management Systems. In Business, Technologie and web (BTW'07), p. 1-4, Aachen, Germany.

Roth A., Naumann F, ubner T. H., Schweigert M. (2006). System P: Query Answering in PDMS under Limited Resources. In Proc. of the Workshop on Information Integration on the Web (IIWeb).

Roth A., Naumann F. (2005). Benefit and Cost of Query Answering in PDMS. In Proc. of the Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P).

Rowstron A., Druschel p. (2001). Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems. IFIP/ACM Int. Conf. on Distributed Systems

Platforms (Middleware), p. 329-350.

Salgado A. C., Santoro F. M., Borges M. R. S., Araujo R. M. , Vieira V. (2009). Gerência de Contexto e suas aplicações. In Colibri Colloquium, 2009, Bento Gonçalves. XXIX Congresso da Sociedade Brasileira de Computação, Porto Alegre.

Sassatelli S. (2009). Query Processing in a PDMS. PhD. Thesis. Universit` a degli Studi di MODENA e REGGIO EMILIA.

Scannapieco M., Virgillito A., Marchetti C., Mecella M., and Baldoni R.. (2004). The DaQuinCIS architecture: A platform for exchanging and improving data quality in cooperative information systems. Information Systems, v. 29, n. 7, p. 551–582.

Schlosser M., Sintek M., Decker S., Nejdl W. (2003) HyperCuP - Hypercubes, Ontologies and Efficient Search on P2P Networks. In Proc. of the 1st Workshop on Agents and P2P Computing, Bologna, Italy.

Schollmeier, R. (2001) "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications". In Proc. of the 1st International Conference on Peer-to-Peer Computing (P2P '01), p.101- 102, Lingkoping, Sweden.

SEWAISE. (2011). SEmantic Webs and AgentS in Integrated Economies. Último acesso em 3 de março de 2011, disponível em http://www.sewaise.org

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

Souza D., Arruda T., Salgado A. C., Tedesco P., and Kedad, Z. (2009). Using Semantics to Enhance Query Reformulation in Dynamic Environments. In the 13th East European Conference on Advances in Databases and Information Systems (ADBIS'09), Riga, Latvia.

Souza D., Belian R., Salgado A. C., Tedesco P. (2008). Towards a Context Ontology to Enhance Data Integration Processes. In Proceedings of the 4th Workshop on Ontologies-based Techniques for DataBases in Information Systems and Knowledge Systems (ODBIS). VLDB '08, Auckland, New Zealand.

Souza D., Pires C., Kedad Z., Tedesco P., Salgado A. (2011). A Semantic-based Approach for

Data Management in a P2P System. In. Journal on Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS). v. 6790, p. 56-86.

Staab S., Tempich C., Wranik A. (2004). REMINDIN': Semantic Query Routing in peer to-peer Networks based on Social Metaphors. In Proc. of the 13th Int. conference on World Wide Web (WWW 2004), New York,USA.

Stoica i., Morris r., Karger D.R., Kaashoek M.F., Balakrishnan H. (2001). "Chord: a scalable peer-to-peer lookup service for internet applications". ACM Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), p. 149-160.

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

Tatarinov I., Halevy A. (2004). Efficient query reformulation in peer-data management systems. In Proc. of SIGMOD Conference, p. 539-550.

Tatarinov I., Ives Z., Madhavan J., Halevy A., Suciu D., Dalvi N., Dong X., Kadiyska Y., Miklau G., Mork P. (2003). The Piazza Peer Data Management Project. In Proc. of the ACM SIGMOD Record, v. 32, n. 3, p. 47-52.

Theotokis S. A., Spinellis D. (2004). A survey of peer-to-peer content distribution technologies. ACM Computing Survey, v. 36, n.4, p. 335–371.

Vieira V. (2008). CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems, Tese de Doutorado, Centro de Informática – UFPE, Brasil

Vieira V., Tedesco P. , Salgado A. C. (2010). Designing Context-Sensitive Systems : An Integrated Approach. In Expert Systems with Applications.

Vieira V., Tedesco P., Salgado A. C. (2009). A Process for the Design of Context-Sensitive Systems, In Proc. of the 13th International Conference on Computer Supported Cooperative Work in Design (CSCWD'09), p. 143-148, Santiago, Chile.

Wang R. Y., Strong D. M. (1996). Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of Management Information Systems, v. 12, n. 4, p. 5-33.

Xiao H. (2006). Query processing for heterogeneous data integration using ontologies. PhD Thesis in Computer Science. University of Illinois at Chicago.

Yatskevich M., Giunchiglia F., McNeill F., Shvaiko P. (2006). OpenKnowledge Deliverable 3.3: A methodology for ontology matching quality evaluation. Available at http://www.cisa.informatics.ed.ac.uk/OK/Deliverables/D3.3. Último acesso em Maio de 2011.

Zaihrayeu I. (2006). Towards Peer-to-Peer Information Management Systems. PhD Thesis, DIT – University of Trento.

Zhao J. (2006). Schema Mediation and Query Processing in Peer Data Management Systems. Master Thesis, The University Of British Columbia.

Zhao B. Y., Huang L., Stribling J., Rhea S.C., Joseph A. D., Kubia-TOWICZ J.D. (2004). Tapestry: a resilient global-scale overlay for service deployment. In IEEE Journal on Selected Areas in Communications (JSAC), v. 22, n. 1, p. 41-53.

Zhuge H., Liu J., Feng L., Sun X., He C. (2005), "Query Routing in a Peer-To-Peer Semantic Link Network. In Computational Intelligence, v. 21, n. 2, p. 197- 216.