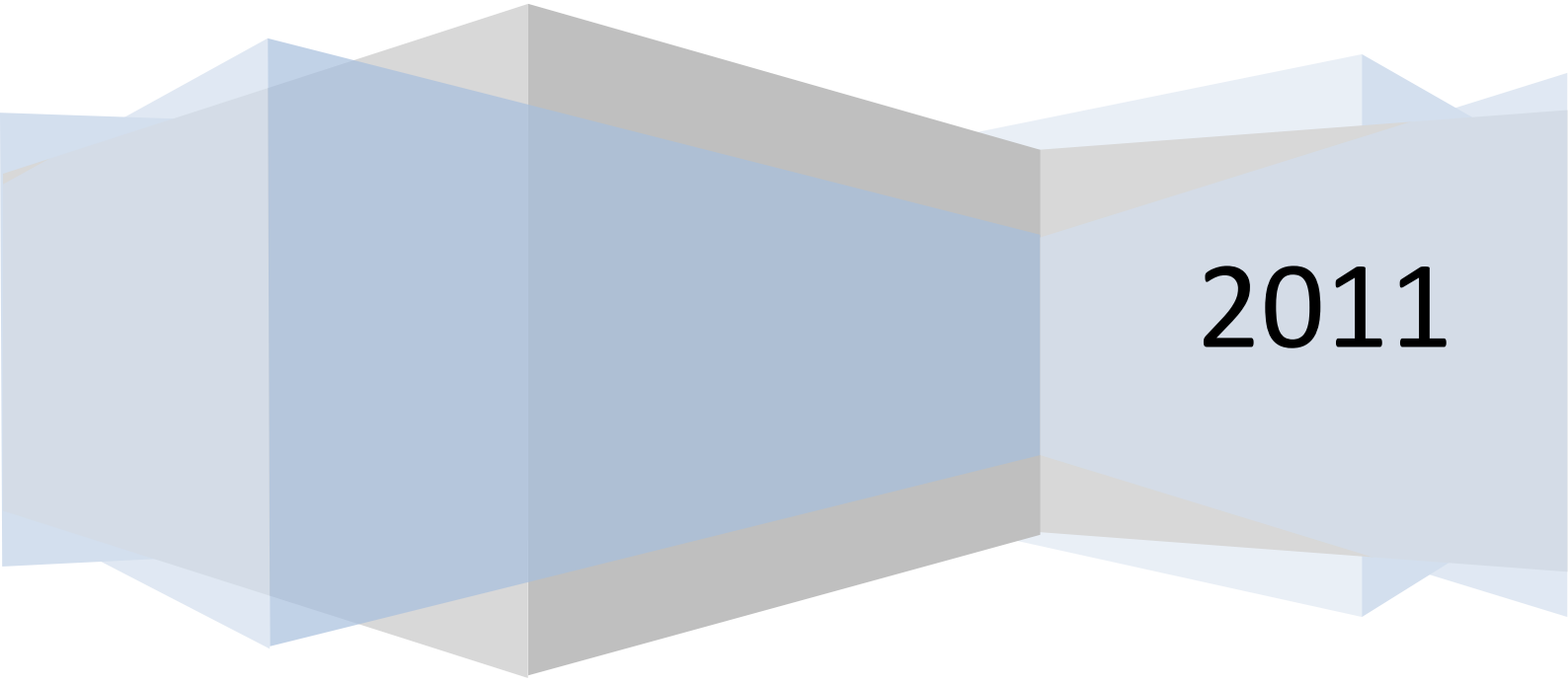
Centro de Informática - UFPE

Qualidade da Informação em PDMS

Trabalho Individual

Bruno Felipe de França Souza



2011

1. Introdução

Com a evolução tecnológica dos computadores, a cada dia são criados mais e mais dados e estes compartilhados entre as pessoas e empresas. Visando facilitar este compartilhamento e armazenamento, foram propostos os bancos de dados, altamente usados na atualidade. Com o passar do tempo, projetos mais robustos de interconexão de bancos de dados deram origem aos bancos de dados distribuídos, várias bases de dados logicamente interligadas. Como consequência dessa arquitetura, problemas com a heterogeneidade das fontes de dados vieram à tona, ou seja, várias fontes com estrutura e dados de diferentes tipos. Visando oferecer aos usuários uma *interface* uniforme para acesso a diferentes fontes de dados por meio de um esquema global e seus esquemas de mediação, soluções como os sistemas de integração de dados [Batista 2008] foram propostos e implementados. Um sistema de integração de dados precisa lidar com a heterogeneidade das fontes de dados, sendo capaz de sobrepor a heterogeneidade de forma a oferecer uma visão integrada dos dados.

Percebeu-se a necessidade de sistemas com mais flexibilidade na troca de informações no tocante a não depender de esquemas de mediação, ou seja, não haver um ponto intermediário entre duas fontes onde é feita a formulação da consulta do usuário. Para fornecer mais flexibilidade ao entrar na rede de compartilhamento, e assim, iniciar a troca de dados foram propostos os sistemas *Peer-to-Peer* (P2P) [Barkai 2000]. O termo *Peer-to-Peer* refere-se ao paradigma de computação distribuída o qual permite uma coleção de pontos (*peers*) compartilharem recursos, serviços e dados de uma maneira descentralizada, por meio de troca direta [Barkai 2000].

Apesar do grande sucesso dos sistemas P2P, os mesmo sofrem em relação à busca de informações semânticas e suas limitações em lidar com a heterogeneidade dos dados. Este próximo passo é representado pelos *Peer Data Management Systems* (PDMS) [Halevy et. al. 2006] que além de serem considerados a evolução dos sistemas de integração, também são a evolução dos sistemas P2P.

PDMS são sistemas distribuídos, dinâmicos, altamente voláteis e apoiados na arquitetura P2P usado para o compartilhamento de dados. Um PDMS é formado por um conjunto de *peers* autônomos semânticos (chamados sítios, fontes, agentes ou pontos) que guardam informações e são interligados com outros *peers* por meio de mapeamentos. Diferente de um sistema de integração de dados, um PDMS não possui um esquema global único, o qual é responsável por reter as consultas dos usuários e enviá-las de forma concisa às fontes de dados.

Como é da natureza de um PDMS lidar com fontes heterogêneas e autônomas, mapeamentos semânticos entre os *peers* devem existir no intuito de trocar informações e

fornecer bons resultados à consulta submetida pelo usuário. Estes mapeamentos semânticos são provenientes do compartilhamento de esquemas entre os *peers* vizinhos, tais esquemas representam o domínio de atuação de cada *peer* na rede. Durante a execução de uma consulta, um *peer* deve ser capaz de enviar a consulta para seu *peer* vizinho, na intenção de obter mais informações para agregar a esta consulta. . Este operação é chamada de reformulação da consulta e é executada para todas as consultas recursivamente de um vizinho para outro.

Alguns problemas podem surgir no que tange às consultas em um PDMS: falta de disponibilidade dos *peers*, resultados incompletos, tempo de resposta dos *peers* muito alto, inconsistências entre conceitos nos *peers*, entre outros. Qualidade da Informação (QI) [Naumann 2000] na reformulação das consultas pode ser uma técnica bastante oportuna na tentativa de minimizar estes problemas e, consequentemente produzir melhores resultados. Este trabalho tem como objetivo apresentar os PDMS no que diz respeito às arquiteturas, classificação e exemplos. QI também será um tópico abordado juntamente com o seu uso em PDMS. Será mostrado o estado da arte em perda da qualidade durante a reformulação de consultas.

Este documento está organizado da seguinte forma: Seção 2 explica de uma forma geral PDMSs; Seção 3 discorre sobre QI, seus critérios e usos, Seção 4 tem-se a conclusão do trabalho.

2. PDMS

A evolução da tecnologia de banco de dados permitiu a migração de uma arquitetura centralizada para outras distribuídas, federadas, com esquemas globais únicos e para sistemas de integração de dados que oferecem aos usuários uma interface única e transparente para consulta. Como citado em [Halevy et. al. 2006] o aparecimento das arquiteturas P2P levaram os pesquisadores de gerenciamento de dados a unir os benefícios de um sistema P2P com a evolução dos bancos de dados. Dessa união surgiram os *Peer Data Management Systems* (PDMS) que são considerados a evolução dos sistemas de integração de dados [Heese et. al. 2005].

Uma vez que estão apoiados por uma arquitetura P2P, todas as características desta arquitetura são herdadas pelo PDMS. Um exemplo de característica é o dinamismo em que os *peers* saem e entram na rede a qualquer momento. No tocante ao gerenciamento de dados, os PDMS devem tratar os seguintes aspectos [Sung 2005]:

- Localização de dados: *peers* devem ser capazes de identificar e localizar dados armazenados em outros *peers*.

- Processamento da consulta: dada uma consulta realizada em um *peer*, o sistema deverá ser capaz de descobrir os *peers* que podem contribuir com informações relevantes para a consulta e processá-la de forma eficiente.
- Integração de dados: uma vez que uma consulta é submetida e os *peers* que podem atender à mesma são localizados, os dados acessados nos diversos *peers* precisam ser integrados e retornados aos usuários, mesmo que as fontes apresentem diferentes esquemas e representações.
- Consistência de dados: em caso de replicação e uso de *cache,* a consistência dos dados replicados deverá ser preservada com relação aos dados nas fontes de dados.

Na Figura 1 é mostrado um exemplo de PDMS para o domínio de pesquisa. Cada *peer* está conectado logicamente com o outro, por meio de mapeamentos semânticos os quais são criados de acordo com o grau de similaridade entre os esquemas dos *peers*. Os *peers* devem decidir qual caminho tomar ao enviar a consulta para outro *peer*. Estas decisões podem ser com base em árvores de decisões e grafos transversais [Ismail et al. 2010], ontologias [Joung & Chuang 2009, Montanelli 2007, Mandreolli et al. 2007, Li & Vuong 2007], multi-agentes [Beneventano et al. 2007] . O caminho que será dado à consulta remete a um dos grandes problemas em um ambiente PDMS: o roteamento da consulta. Trata-se de fornecer resposta à questão de como responder às consultas postas pelos usuários de forma eficaz e eficiente [Ismail et al. 2010].

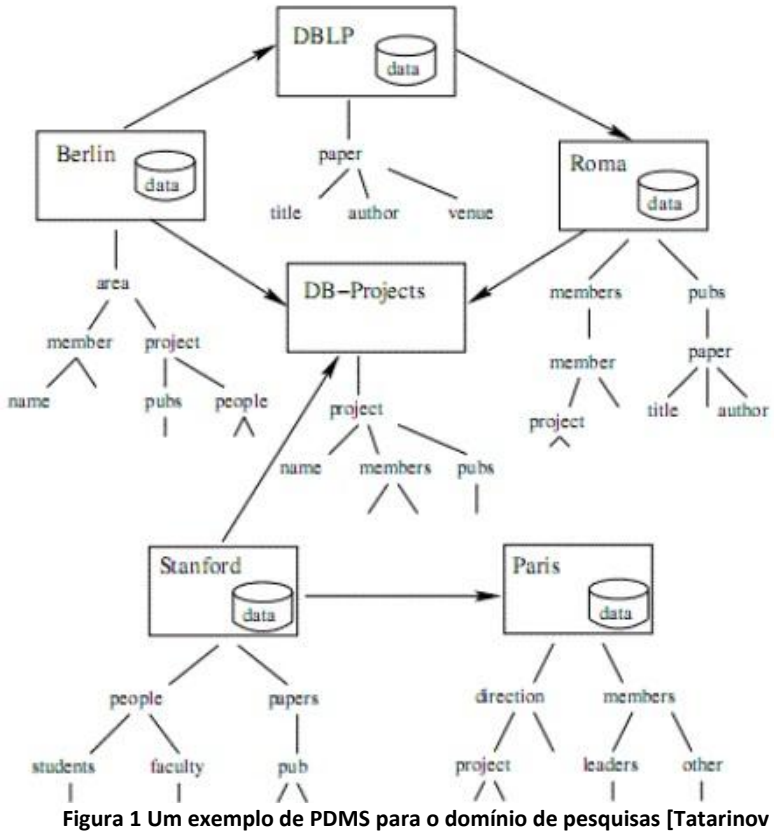


Figura 1 Um exemplo de PDMS para o domínio de pesquisas [Tatarinov e Halevy 2004].

2.1Vantagens dos PDMS

Em [Tatarinov & Halevy 2004] são apresentadas as principais vantagens no uso de um PDMS, a saber:

Peers podem compartilhar dados em vários domínios sem a necessidade de um esquema de mediação, visto que, em um ambiente PDMS, *peers* são ligados entre si por meio de mapeamentos diretos e através deles o compartilhamento de dados acontece.

Um *peer* pode fornecer um mapeamento para o *peer* mais conveniente (e.g. similar) que já esteja no PDMS. Um *peer* cujo esquema reflete um domínio de área médica pode, por meio da similaridade com os dados que comporta, estabelecer um mapeamento com outros *peers* do domínio da área médica.

Um *peer* pode formular uma consulta usando seu próprio esquema, sem ter a necessidade de conhecer outros esquemas. Na execução de uma consulta o *peer* primeiramente executa uma consulta localmente e, para isso, não é preciso conhecer os esquemas dos seus *peers* vizinhos.

Algumas outras vantagens que podem ser citadas são [Pires 2009]:

Compartilhamento de dados estruturados e semi-estruturados: diferentemente de sistemas P2P que compartilham dados não-estruturados i.e., imagens, arquivos de áudio, um PDMS pode compartilhar dados estruturados i.e., dados vindos de banco de dados, ou mesmo dados semi-estruturados i.e., XML[1].

Ausência de um esquema global único: o sistema não é responsável por manter um esquema global único, utilizado em sistemas de integração de dados para executar a tarefa de integração dos esquemas das fontes de dados.

Consultas elaboradas de acordo com o esquema do *peer*: cada *peer* fornece seu esquema para compartilhamento de dados e a consulta é submetida sobre este esquema. Ou seja, não é preciso formular consultas por meio de um esquema de mediação único.

Pouca administração: as fontes de dados são bastante autônomas, podendo entrar e sair a qualquer momento do sistema.

Inexistência de um ponto único de falha: em várias arquiteturas inexiste o papel de uma entidade responsável por gerenciar o sistema, de tal forma que sua ausência torne o sistema inoperante.

[1] Extensible Markup Language, W3C. http://www.w3.org/XML/

Replicação de dados: os dados podem ser replicados em vários *peers* e não apenas em um repositório de dados, como realizado na abordagem materializada *(data warehouse*) de sistemas de integração de dados;

Semântica dos dados: como os dados são estruturados e/ou semi-estruturados, estes detêm um alto poder de significado;

Uso de ontologias: ontologias podem aumentar o poder de expressividade de um domínio e ajudar a melhorar problemas em PDMSs como, por exemplo, agrupamento semântico de *peers* onde o objetivo é agrupar, em uma comunidade, *peers* que possuem domínios semelhantes [Kantere et. al., 2009].

2.2 Classificação dos PDMS

Existem diferentes arquiteturas possíveis para sistemas P2P bem como para um PDMS. Por simplicidade, consideramos três classes principais de arquitetura: não-estruturada (*unstructured*), estruturada (*structured*) e *super-peer*. PDMS não-estruturados e estruturados também são chamados de "puros" ao passo que *super-peer* são chamados híbridos. Sistemas "puros" consideram todos os *peers* iguais, ou seja, nenhum possui ou fornece funcionalidades especiais.

Em um PDMS não-estruturado, cada *peer* pode se comunicar diretamente com o seu vizinho. A Figura 2 mostra um PDMS não-estruturado, cada *peer*, por exemplo o *peer* Brasil, pode se comunicar de forma direta com seus vizinhos França, Israel e EUA. O método de busca por informação é bastante simples: o *peer* que faz uma consulta inicia um processo de *flooding* (inundação) [Valduriez & Pacitti, 2004] em toda a rede, através do envio da consulta aos *peers* vizinhos. Os *peers* vizinhos, por sua vez, enviam as consultas para outros *peers* vizinhos e assim, sucessivamente. Este tipo de roteamento por *flooding* não é escalável quando se tem uma grande quantidade de *peers* pelo fato da consulta ser enviada para muitos *peers*, deteriorando assim o desempenho da rede. Além disso, a incompletude das informações pode ser bastante alta, visto que nem todos os *peers* compartilham dados que interessam à consulta do usuário e *peers* podem estar fora da rede no momento da consulta [Valduriez & Pacitti, 2004]. Uma vez que, comumente, em um PDMS, todos os *peers* são iguais em relação às funções que desempenham no sistema e capazes de replicar dados, a tolerância a falhas é bastante alta.

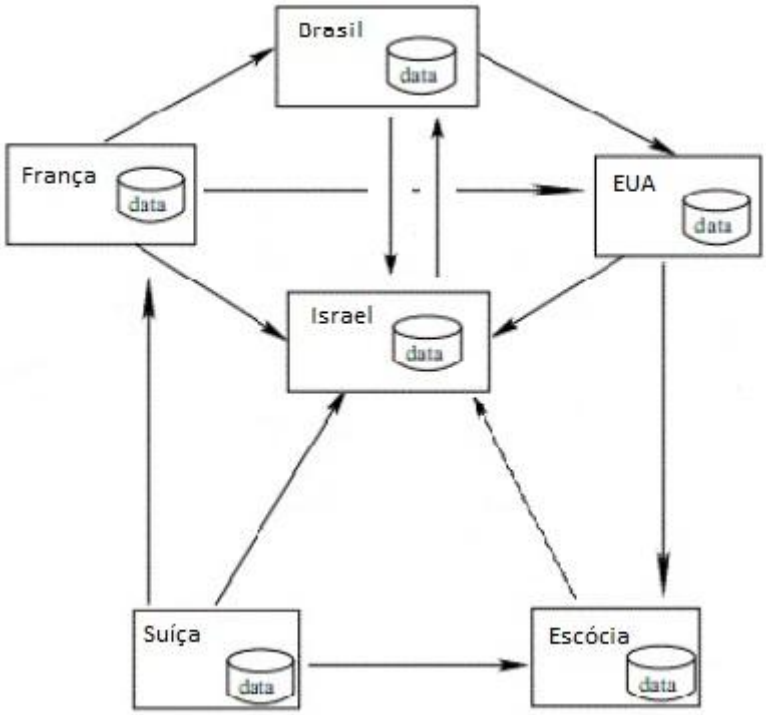


Figura 2 Um PDMS não-estruturado.

Estudos sobre como melhorar o desempenho dos sistemas baseados em arquiteturas não-estruturadas levaram ao desenvolvimento de sistemas com arquiteturas estruturadas tendo como alicerce as *Distributed Hash Table* (DHT) ilustrada na Figura 2.

Exemplos de PDMS com arquitetura estruturada são CAN [Ratnasamy et. al. 2001] e o CHORD [Stoica et. al. 2001]. Um sistema DHT fornece uma tabela *hash* com funções do tipo: *put(chave,valor)* e *get(chave)*, onde *chave* é comumente um nome de arquivo e cada *peer* é responsável pelo armazenamento do *valor* (conteúdo do arquivo). DHT é bastante utilizado em sistemas P2P no intuito de localizar arquivos por meio de casamento de palavras-chave. Em um PDMS que compartilha dados semânticos, uma DHT pode ser utilizada para auxiliar *peers* que possuem interesses em comum a encontrar um ao outro e construir comunidades semânticas [Pires, 2009].

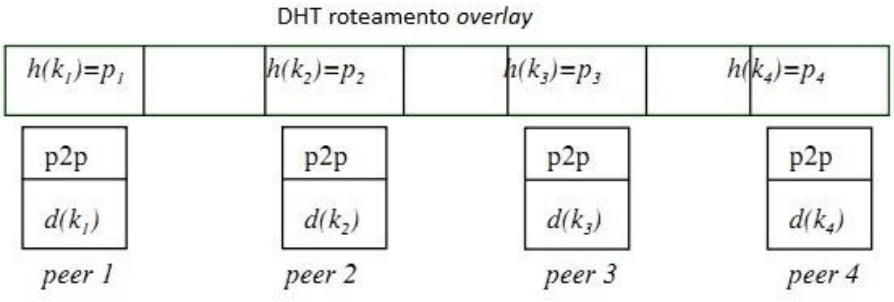


Figura 3 Uma rede DHT [Valduriez & Pacitti, 2004].

Na Figura 3, podemos notar a presença de uma rede *overlay* (rede virtual) criada acima dos *peers*, responsável por armazenar a tabela *hash* definida pela DHT. Cada *peer* compartilha seus dados sendo identificados por uma função *hash*, então cada arquivo é localizado através da chamada *h(chave) = Peer Origem*, por exempo, $h(K_1) = P_1$ resultando no arquivo armazenado no *peer* 1.

PDMS baseados em arquiteturas *super-peer* (Figura 4) são aqueles em que nem todos os *peers* são iguais, ou seja, alguns *peers* - os *super-peers* - atuam como servidores dedicados para outros *peers* e podem executar as funções de: indexação, processamento de consulta, controle de acesso e gerenciamento de metadados [Valduriez & Pacitti, 2004]. Tendo um *peer* com capacidades diferenciadas de outros *peers*, o sistema apresenta uma arquitetura similar à arquitetura cliente-servidor, com todos os problemas de ter um servidor central como, por exemplo, falha no servidor de dados tornando o sistema inoperante e sobrecarga de requisições feitas por clientes ao servidor. Se alguma falha acontecer em um *super-peer* e o deixar inoperante, o sistema deve ser capaz de eleger outro *peer* para assumir o papel e se tornar o novo *super-peer*. Logo, uma arquitetura de *super-peer* redundante que combina vantagens tanto de sistemas centralizados quanto de sistemas distribuídos é considerada a mais apropriada para sistemas que precisam de computação distribuída [Fioriano, 2003].

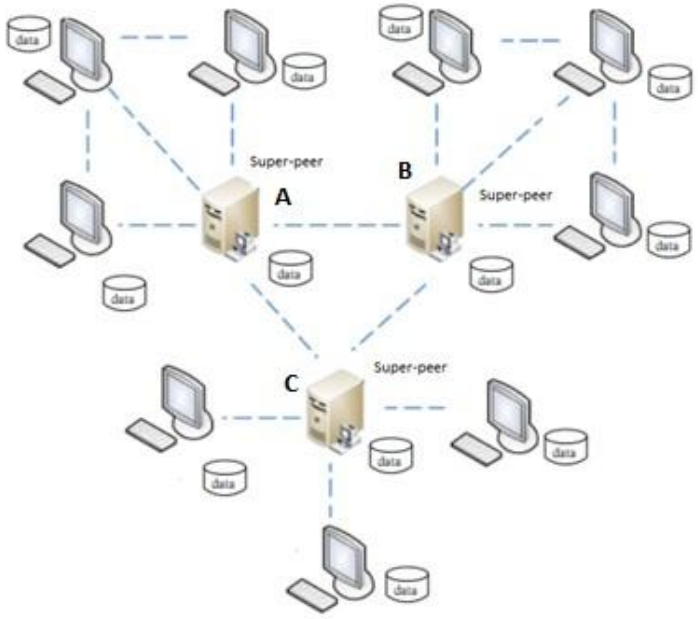


Figura 4 Um PDMS com arquitetura *super-peer*, adaptado de [Silva, 2011].

Na Figura 4, nota-se a presença de três *super-peer*, A, B e C onde cada um é responsável por um grupo (*cluster*) de *peers*. É importante notar que os mapeamentos são feitos entre os *super-peer* e não entre os *peers* em si.

2.3 Processamento da Consulta

Tradicionalmente, o processamento da consulta em ambientes de compartilhamento de dados distribuídos é implementado pelo uso de uma camada de mediação responsável por decompor as consultas formuladas em um esquema global e enviar as sub-consultas para varias fontes de dados dispersas, ou mesmo integrando e materializando dados de várias fontes e dados em um lugar central no estilo *data warehouse*.

Todavia, em um cenário onde a arquitetura P2P é levada em consideração, tal componente central o qual prover conhecimento consolidado bem como coordena a execução das consultas, pode eventualmente, se tornar um gargalo ou um ponto único de falha. Deste modo, por questões de escalabilidade e preservação da autonomia consultas em ambientes P2P devem ser descentralizadas e livres de coordenação. Além do mais, devido à natureza autônoma de cada *peer* o plano da execução da consulta deve ser feito dinâmicamente, uma vez que o número de *peers* participantes e dos dados disponíveis podem mudar a qualquer momento.

Em um PDMS, *peers* são interligados por meio de mapeamentos semânticos existentes entre eles, assim compartilhando seus dados. Os mapeamentos são gerados por meio da identificação das correspondências entre os esquemas. Este processo tem como objetivo identificar conceitos similares entre os esquemas dos *peers* e assim gerar um mapeamento entre eles.

A Figura 5 apresenta dois esquemas de *peers* hipotéticos representando informações de hospitais no modelo relacional [Codd, 1970]. Podemos observar que em ambos os esquemas temos algo em comum, a relação *Cama*, as outras relações são semanticamente equivalentes, embora representadas de formas diferentes. Existe um processo chamado *matching*, o qual é responsável por identificar similaridades entre os domínios dos *peers*, ou seja, por meio deste processo é possível saber que *Cama* representada no *P1* é equivalente à *Cama* representada no *P2*, da mesma forma que *Empregado* e *Paciente* no P1 são equivalentes à *Funcionário* e *Usuário* no *P2* respectivamente. Ao final do *matching*, as correspondências (*C1*, *C2*, *C3*) como mostra a Figura 5, são geradas entre cada conceito dos esquemas. Este processo de *matching* pode ainda ser enriquecido semanticamente com o uso

de ontologias [Russell & Norvig, 1995] que explicam os termos de cada esquema e como eles estão relacionados.

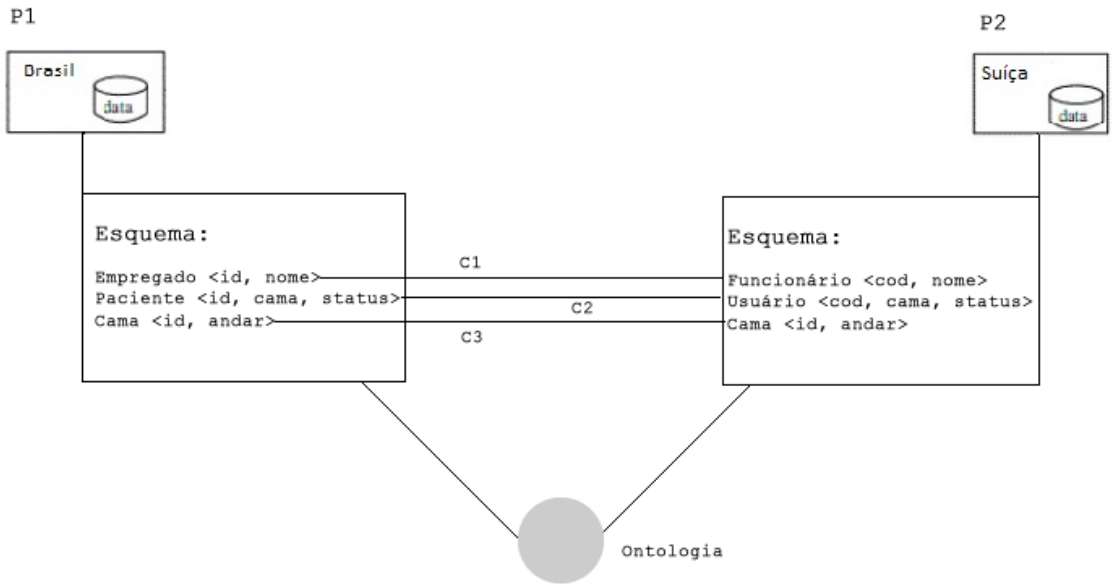


Figura 5 Um processo de *matching* entre dois esquemas de *peers* apoiados por uma ontologia.

Cada *peer* no PDMS publica um esquema e este descreve o conteúdo do *peer*. Neste cenário, quando um usuário submete uma consulta ao sistema, esta pode ser respondida pelo *peer* de origem da consulta ou por outros que fazem parte da rede. Todavia, se um *peer* passar a consulta para outro *peer* na rede, acontece um processo chamado de reformulação da consulta, ou seja, a consulta deve ser reescrita de um esquema para outro esquema.

Devido à natureza de um PDMS ser bastante dinâmica e volátil, *peers* podem sair e entrar na rede a qualquer momento. Com isso, o PDMS tem que auto-organizar os mapeamentos existentes no intuito de manter relações com *peers* que compartilham interesses em comum [Souza, 2009]. Como consequência desta facilidade em entrar e sair do sistema podemos citar: perda semântica em relação à reformulação da consulta uma vez que *peers* que atendam bem uma determinada consulta podem não estar disponíveis na próxima vez; caminhos extensos podem surgir por conta de um *peer* não se encontrar mais na rede e por isso a consulta pode percorrer um caminho mais longo em busca de outro *peer* e; tempo de resposta alto por consequência dos caminhos extensos.

Em uma arquitetura pura, um processamento de consulta tem como vantagem a escalabilidade. Tomando como base que qualquer *peer* pode conectar-se ou desconectar-se da rede sem nenhuma interferência com os demais *peers,* uma consulta em execução não será prejudicada por conta de um *peer* que saiu da rede, uma vez que outros *peers* podem ser

consultados. Outra vantagem refere-se à inexistência de um ponto de falha, pois as respostas são geralmente encontradas, mesmo que nem todos os *peers* estejam conectados, a não ser que a consulta retorne um resultado vazio. Como desvantagens, o algoritmo pode percorrer caminhos não ótimos sem recuperar respostas ou seguir por caminhos redundantes obtendo respostas desnecessárias e, assim, o tempo de resposta pode vir a ser bastante alto.

No caso de uma arquitetura *super-peer* o processamento da consulta apresenta um desempenho melhor, se comparado à arquitetura pura, uma vez que no *super-peer*, os *peers* são agrupados em comunidades que possuem informações do domínio em comum, facilitando assim o roteamento da consulta. Devido aos *peers* estarem agrupados, existe uma unidade controladora responsável por integrar os resultados das consultas, recuperados do seu grupo de *peers*. Já as desvantagens estão em os *super-peers* poderem se tornar pontos críticos de falhas e, ao mesmo tempo, degradarem o desempenho da resposta da consulta visto que o *super-peer*, geralmente, é o responsável pelo armazenamento do conhecimento do domínio que é compartilhado por cada um dos seus grupos de *peers*.

2.4 Roteamento da Consulta

PDMS são constituídos por *n peers* autônomos e distribuídos semanticamente na rede, compartilhando informações por meio de seus mapeamentos. Um dos principais problemas que surgem em tal arquitetura é: como explorar tais mapeamentos com o objetivo de responder às consultas postas em cada *peer* de forma eficiente [Staab et. al., 2004]. A utilização de sistemas baseados em arquiteturas P2P depende de técnicas eficazes para encontrar e recuperar dados, porém, o roteamento de consultas com base no conteúdo de maneira eficiente é um problema desafiador em redes P2P [Ismail et. al., 2010].

Em um PDMS aliado ao processamento de consulta está o roteamento da mesma. Considerando a não existência de um ponto central para a retenção do conhecimento dos vários *peers*, torna-se importante observar as estratégias adotadas para o roteamento eficiente dessas consultas.

Alguns trabalhos buscam tornar eficiente a forma de recuperação de informações em um PDMS a partir da seleção do melhor caminho semântico durante a consulta [Ismail et. al., 2010], [Montanelli & Castano, 2008], [Juang & Chuang, 2009].

No trabalho de [Freire, 2010] a autora levanta algumas questões relacionadas ao roteamento de consultas (1) como localizar conteúdo relevante, decidir para quais outros pontos a consulta deve ser encaminhada de forma a responder com eficiência e eficácia; (2)

sistemas que usam algoritmos de inundação sofrem com questões de eficiência e escalabilidade; (3) *peers* devem confiar nos vizinhos e conhecer seus esquemas; (4) tempo de resposta, tentar minimizar ao máximo; (5) crescimento e dinamismo da rede (entrada e saída de *peers* e *super-peers*).

O estudo e, possivelmente o uso de critérios de QI pode ser um fator bastante positivo para a medição da qualidade de um *peer* e do mapeamento entre eles, contribuindo assim para um melhor roteamento da consulta.

2.5 Exemplos de PDMS

Atualmente, existem diversas propostas de PDMSs, porém cada um com suas particularidades diferenciadas por suas arquiteturas. Nesta seção serão apresentados alguns PDMSs.

2.5.1 Piazza

Piazza é um PDMS que fornece uma infraestrutura para aplicações baseadas na Web Semântica [Halevy et. al., 2003a].

Desenvolvido pela Universidade de Washington e da Pensilvânia, o Piazza foi concebido com o intuito de ser um PDMS escalonável em um ambiente heterogêneo e distribuído.

2.5.1.1 Arquitetura

O sistema assume que usuários participantes interessados em compartilhar seus dados, devem estar dispostos a definir mapeamentos entre seus esquemas, ou seja, os mapeamentos não são feitos de forma automática sendo assim necessário que o usuário identifique mapeamentos entre o esquema que está exportando e o esquema ao qual irá se conectar, conforme ilustra a Figura 6. Na Figura 6 as linhas pontilhadas mostram os mapeamentos feitos pelos usuários. Cada *peer* compartilha seus dados na forma de relações armazenadas. O *peer* define o seu esquema de forma que outros *peers* possam acessar suas relações armazenadas através deste esquema. Além disso, o *peer* mantém dois tipos de mapeamentos de esquemas: o primeiro refere-se às relações armazenadas e ao esquema do próprio *peer* e o segundo, refere-se ao mapeamento entre o esquema do *peer* com os esquemas de seus vizinhos [Halevy et al., 2003b].

O Piazza suporta compartilhamento de dados em XML/RDF para suporte a aplicações da Web Semântica. Portanto, o esquema do *peer* é descrito usando XML Schema ou ontologias na linguagem OWL para dados XML e RDF respectivamente. Além disso, a linguagem para mapeamentos e para consultas é baseada em XQuery.

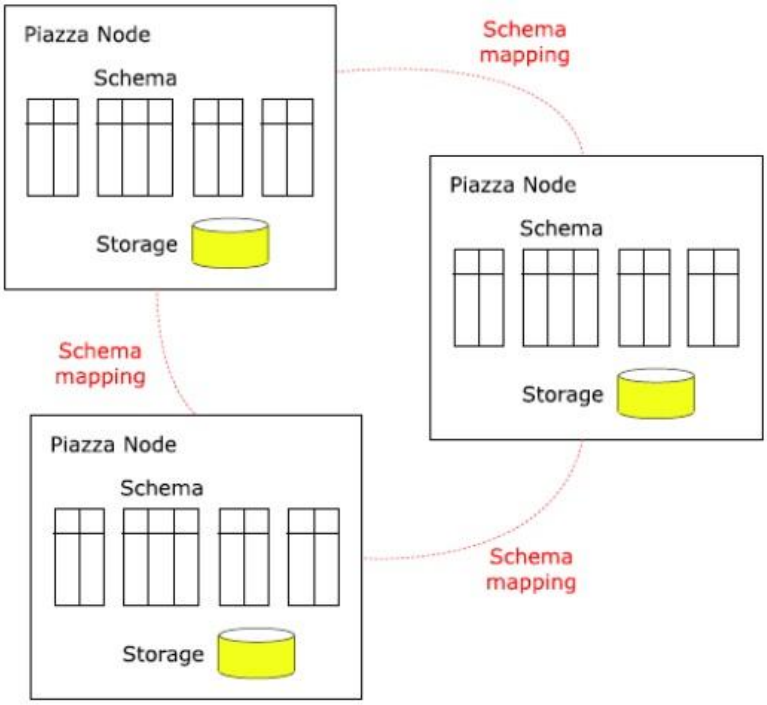


Figura 6 Arquitetura do Piazza [VU et. al., 2010]

O Piazza possui uma arquitetura descentralizada, inexistindo assim a presença de um esquema global único. O conjunto de mapeamentos definidos por este sistema visa definir a semântica do mesmo. Cada mapeamento é implementado através de um gráfico arbitrário de esquemas conectados, sendo alguns destes esquemas definidos virtualmente para propósitos de consulta e de mapeamento. E, para a construção dos mapeamentos, o Piazza baseia-se no uso conjunto de heurísticas e algoritmos a fim de resolver problemas semânticos. Estes por sua vez, são baseados em técnicas de aprendizagem de máquina e na exploração de experiências anteriores, onde informações sobre mapeamentos válidos já existentes são utilizados para o mapeamento de novos esquemas.

2.5.1.2 Processamento da Consulta

A principal meta no processamento de consulta do Piazza é ser capaz de responder consultas colocadas em qualquer *peer*, fazendo uso de dados relevantes para a consulta [Halevy et. al., 2003b]. Para isto, o algoritmo usado pelo Piazza realiza as seguintes tarefas: dada uma rede de *peers* com dados XML, um conjunto de mapeamentos semânticos entre os *peers*, e uma consulta sobre o esquema de um dado *peer*, de forma eficiente, forneça todas as *respostas corretas* possíveis que podem ser obtidas por meio de uma consulta. Respostas corretas são aqueles resultados que são *garantidos* de estar no esquema lógico do *peer*, em outras palavras, são resultados consistentes e que existem na fonte de dados.

Em um nível de abstração alto, o algoritmo funciona a partir de uma consulta *Q* colocada sobre o esquema de um *peer P*. Primeiramente são usadas as descrições dos dados armazenados em *P*, (i.e. os mapeamentos que descrevem quais dados estão atualmente armazenados em *P*), para reescrever a consulta *Q* em uma consulta *Q'* sobre os dados armazenados em *P*. O próximo passo é considerar os vizinhos semânticos de *P*, i.e., todos os *peers* que têm relação com o esquema de *P* definidos pelos mapeamentos semânticos. Estes mapeamentos são usados para expandir a reformulação da consulta *Q* para uma consulta *Q"* sobre os vizinhos de *P*. Por sua vez, *Q"* é expandida, assim como *Q'*, sobre os dados armazenados em *P*, mas também sobre os dados armazenados dos seus vizinhos; com isso é feito uma união com *Q'*, eliminando assim qualquer tipo de redundância. Este processo é feito de forma recursiva, seguindo todos os mapeamentos entre os esquemas dos *peers*, até não haver mais caminhos úteis a percorrer.

2.5.2 SPEED

O SPEED (Semantic *Peer*-to-*peer* Data Management System) é um PDMS desenvolvido na Universidade Federal de Pernambuco que possui uma topologia de rede mista: DHT, *super-peer* e não estruturada [Pires, 2009]. Além disso, utiliza semântica como base para o desenvolvimento e gerenciamento de seus serviços e representa os esquemas das fontes ligadas aos *peers* através de ontologias.

Com relação à semântica, o SPEED realiza o agrupamento de *peers* que possuem similaridades semânticas, com o objetivo de facilitar o mapeamento semântico entre os *peers* e, como conseqüência, facilitar o processamento de consulta sobre o grande volume de fontes de dados.

2.5.2.1 Arquitetura

A Figura 7 ilustra a arquitetura do SPEED. Nela podemos ressaltar [Pires, 2009]:

Pontos de dados: *peers* que possuem dados a serem compartilhados com outros *peers*.

Pontos de integração: *peers* responsáveis por processamento de consulta, indexação de metadados e integração de dados. Pontos de integração são pontos de dados com alta disponibilidade e poder computacional, além de nomearem seus respectivos *clusters* semânticos.

Cluster semântico: está associado a informações de domínio comuns entre pontos de dados e possui um ponto de integração.

Ponto semântico: *peer* que armazena e oferece uma ontologia padrão de um domínio específico (por exemplo, "saúde", "educação", "engenharia", etc.) e nomeia suas

comunidades semânticas correspondentes. Apenas um ponto semântico é permitido por comunidade semântica.

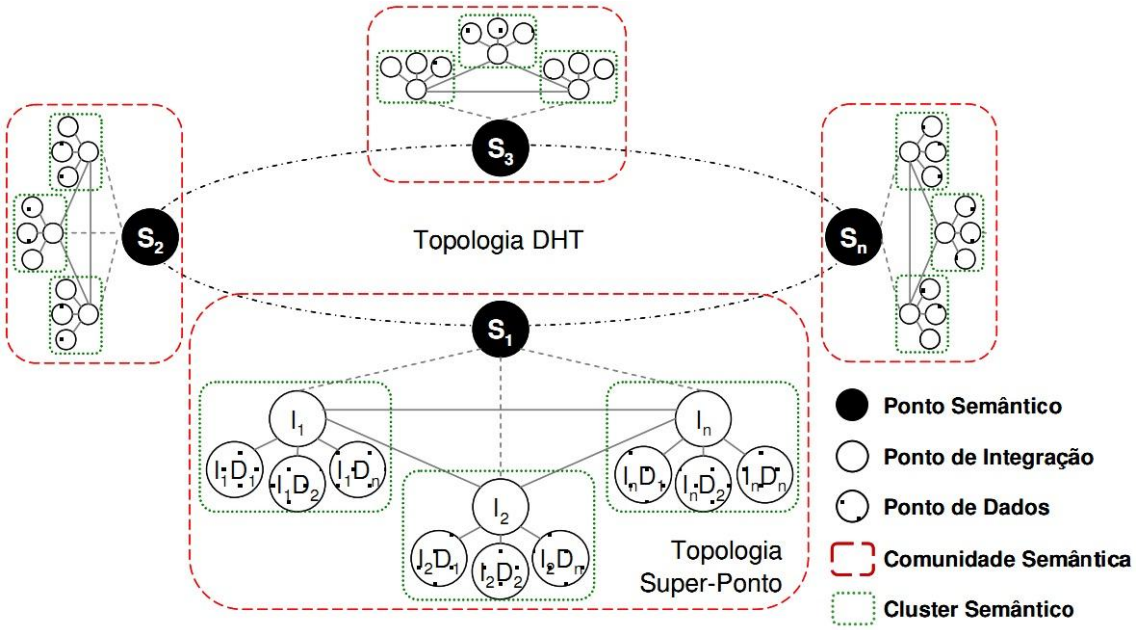


Figura 7 Arquitetura do SPEED [Pires, 2009].

Comunidade semântica: é construída por meio da composição de *clusters* de um mesmo domínioque possuem interesses e semânticas em comuns. Ao entrar no sistema, um ponto de dados é selecionado, por um ponto semântico para entrar em um *cluster* semântico apropriado, onde o ponto de dados deve estar conectado.

Topologia de *super-peer*: com o objetivo de fragmentar a rede em porções mais fáceis de gerenciar, ou seja, uma modularização da rede, cada *super-peer* agrupa um conjunto de *peers*. O conceito de *cluster* alia-se ao de *super-peer,* já que são criados grupos de *peers* similares com o intuito de melhorar o sistema de uma forma geral. Ou seja, melhorar aspectos como: resposta à consulta, tempo de resposta, roteamento da consulta e redundância dos dados. Com o emprego desta topologia, existe uma maior exploração da heterogeneidade dos pontos participantes, pois, a priori, é conhecido cada esquema do *peer* participante, facilitando, por exemplo, o trabalho do *super-peer* em indicar à qual *cluster* um determinado *peer* deve fazer parte já que o *super-peer* armazena conceitos dos *peers* do *cluster* que é responsável.

Topologia DHT: utilizada para auxiliar *peers* que possuem interesses comuns a encontrar um ao outro e construir comunidades semânticas. Essa topologia é composta pelos pontos

semânticos com boa largura de banda e que permanecem na rede por longos períodos de tempo.

2.5.2.2 Processamento da Consulta

No SPEED, uma consulta submetida por um *peer* é reformulada de acordo com o esquema exportado por este *peer*, e traduzida na linguagem de consulta SPARQL[2]. A consulta além de ser processada naquele ponto, é encaminhada até o ponto de integração daquele *cluster*. O ponto de integração identifica os pontos de dados capazes de responder àquela consulta, ao mesmo tempo em que a consulta é propagada para outros pontos de integração da mesma comunidade [Pires 2009, Neves 2008]. Os *peers* de integração se responsabilizam por integrar os resultados retornados pelos *peers* de dados e de integração. Vale a pena salientar que os pontos semânticos (*super-peers*) não participam do processamento da consulta, desta forma, se uma consulta for submetida a uma determinada comunidade semântica, a consulta não é encaminhada para outras comunidades.

2.5.3 *Peer*DB

Desenvolvido pela Universidade de Singapura, o *Peer*DB é um PDMS implementado sobre a plataforma Best*Peer* [NG et al., 2002], baseado na topologia pura não-estruturada, onde cada *peer* possui dados que são gerenciados por um SGBD relacional, proporcionando buscas por conteúdo. Os dados gerenciados são compartilhados sem a presença de um esquema global e acessados por meio da linguagem SQL.

2.5.3.1 Arquitetura

A Figura 8 ilustra a arquitetura do *Peer*DB que basicamente é composta por três camadas:

Camada P2P: camada responsável por ofertar serviços de descoberta de recursos na rede e por compartilhar dados.

Camada de agentes inteligentes: sistema multi-agentes que viabiliza uma infra-estrutura para que agentes móveis possam operar nos diversos *peers* da rede. E, em cada *peer* existe um agente responsável por gerenciar as consultas dos usuários, esse agente é chamado

[2] SPARQL Query Language for RDF, W3C. http://www.w3.org/TR/rdf-sparql-query/

de DBAgent. Além disso, o DBAgent monitora estatísticas sobre *peers* vizinhos e gerencia políticas de configuração da rede.

Camada de gerenciamento de dados: camada responsável pela manipulação dos dados disponíveis no *peer*. Nesta camada existe mecanismos de *cache* para proporcionar o armazenamento temporário dos resultados provenientes de diversos *peers*, com o intuito de minimizar o tempo de resposta às consultas subsequentes. Para o gerenciamento de cache, o *Peer*DB faz uso do algoritmo LRU [O'Neil et al., 1993].

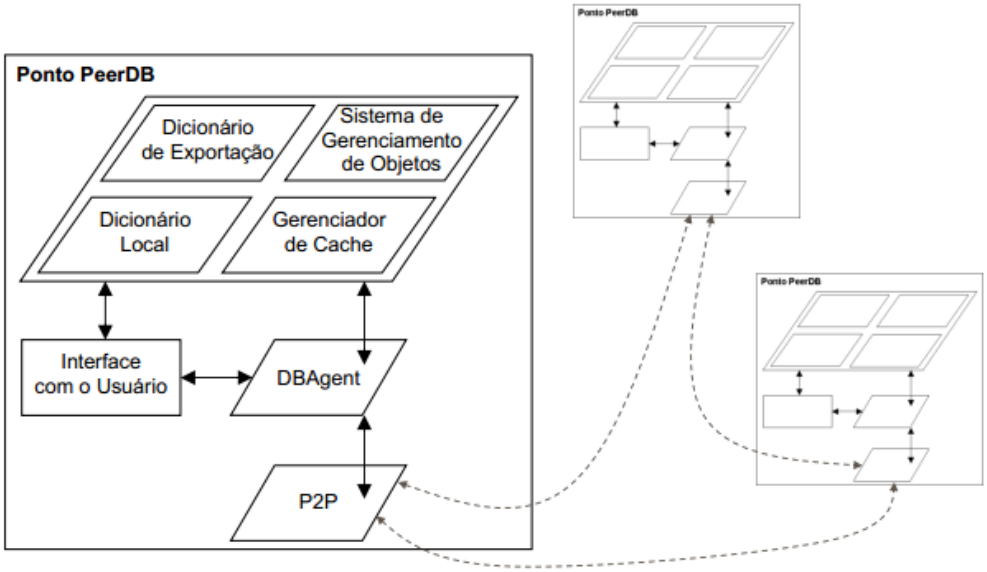


Figura 8 Arquitetura do *Peer*DB [OOI et. al., 2003].

2.5.3.2 Processamento da Consulta

Uma vez submetida uma consulta em um *peer*, um agente DBAgent do *peer* cria agentes auxiliares e os envia ao conjunto de *peers* vizinhos. No entanto, apenas os agentes auxiliares aptos a responderem a consulta retornam metadados referentes às tabelas e às colunas envolvidas dos *peers* de destino. O DBAgent, situado no *peer* onde a consulta foi submetida, compara sintaticamente os metadados recebidos com os metadados do seu *peer*. Se surgirem conflitos semânticos, o DBAgent interage com o usuário a fim de resolvê-los.

Por meio de comparações sintáticas entre os nomes de tabelas e de colunas da consulta e as palavras-chave contidas no dicionário de exportação (metadados referentes às tabelas e às colunas compartilhadas) dos *peers* envolvidos, é que um *peer* é escolhido. Durante o processamento de consultas o usuário necessita confirmar os pontos que devem ser consultados. Com isso, o *Peer*DB fica dependente da interseção do usuário. Além disso, a qualidade do resultado das consultas depende do processo de definição de

palavras-chave e, no caso de ocorrência de homônimos, o resultado da consulta pode ficar comprometido, devido à semelhança entre a grafia das palavras mas com significados diferentes, talvez levando o usuário a ficar confuso [OOI et al., 2003]. No final, os dados (dos *peers* sem conflito) são combinados e apresentados ao usuário.

3. Qualidade da Informação (QI)

Informação está presente em cada dia do nosso mundo em constante desenvolvimento. As pessoas usam informações para tomarem diversas decisões, desde as mais simples para as mais complexas. Independente do domínio de interesse existe uma premissa geral: quanto mais valor a informação agrega, melhor para o seu propósito. A Qualidade da Informação (QI) é um fator crucial na escolha da informação, particularmente no contexto do processo de tomada de decisão.

A avaliação da QI de uma fonte de dados *Web* recebeu muita atenção nas pesquisas realizadas na última década. Duas razões principais para esta atenção são (a) o grande crescimento em números de diversas fontes de dados na *Web* e (b) a natureza de alta acessibilidade destas informações por diversos grupos de consumidores [Azary & Kopak, 2010]. Por tal crescimento, muito do conteúdo disponível pode ser proveniente de fontes errôneas e/ou duvidosas levando a questionamentos sobre a QI fornecida por tal fonte.

Devido à popularidade de sistemas de *data warehouse* e de acesso direto a informações publicadas por várias fontes de dados, incrementaram-se ainda mais as necessidades de tratamento da QI. Entretanto alguns problemas ainda existem e nem todas as informações são adequadas: ferramentas de verificação podem auxiliar na validação de um conjunto de dados e, ainda assim, uma parte do conjunto permanecer inválida. Em geral, os dados podem apresentar baixa qualidade tanto por não refletirem a realidade, como por serem mal utilizados e mal entendidos pelos usuários. Normalmente, o custo de dados com baixa qualidade pode ser medido em termos dos requisitos de usuários. Mesmo dados precisos, se não estiverem disponíveis e dispostos de maneira que sua interpretação seja simples, serão de pouco valor [Wang et. al., 1993].

A qualidade tem se tornado um aspecto crítico em muitas organizações e sistemas de informações. QI como um conceito, pode ser comumente definida como um conjunto de critérios para indicar o grau de qualidade geral de uma informação obtida por um sistema [Batista, 2008]. Estes critérios formam um aspecto multidimensional e o papel de cada um é avaliar e medir a qualidade da informação [Angeles & MacKinnon, 2005]. Outra definição dada por [Nurse et. al, 2011] é que QI pode ser vista como a avaliação ou medida de quanto uma determinada informação está apta para uso por parte do usuário (*fitness for use*). Esta noção de *fitness for use* foi apresentada por [Wang & Strong, 1996] e até hoje é

usada em vários trabalhos [Knight & Burn, 2005] [Shanks & Corbitt, 1999] [Bovee et. al., 2003].

É importante distinguir entre os conceitos de qualidade do dado e qualidade da informação. Embora, às vezes, usados como sinônimos, QI é um termo usado para descrever a qualidade de qualquer elemento ou conteúdo de um sistema de informação [Wang & Strong, 1999], não apenas o dado. A garantia da QI é a certeza que uma parte da informação está em conformidade com alguns critérios de qualidade. O uso do termo *informação* ao invés de *dado* implica que o *uso* e a *entrega* de dados têm que ser considerados em qualquer avaliação da qualidade, i.e., a qualidade do dado entregado representa seu valor para consumidores de informações [Price & Shanksa, 2004]. Em alguns trabalhos tem-se o termo qualidade do dado como similar ao critério de QI *precisão*, ou seja, apenas uma característica ou aspecto do mais amplo conceito de QI [English, 1999].

Naumann, em [Naumann & Rolker, 2000] mostra que a qualidade da informação depende de três fatores maiores: a percepção do usuário, a informação por si só e o acesso às informações. Os três fatores são rotulados como o sujeito, objeto e predicado de uma consulta e cada um deles é uma fonte para os metadados de QI ou escores dos critérios de QI. Um escore é um valor associado a um determinado critério de QI. As três fontes de metadados correspondem a três classes de critérios de QI como mostra a Figura 9.

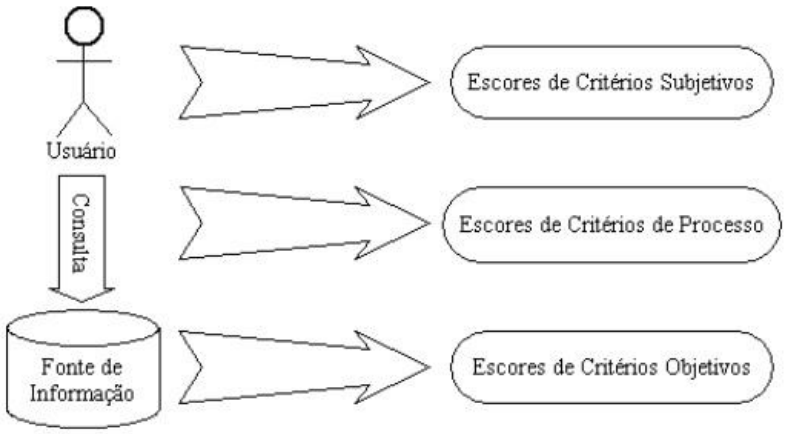


Figura 9 Fontes de escores de critérios de QI adaptada de [Naumann & Rolker, 2000].

Critérios subjetivos devem ser fixados pelo usuário por meio de métodos de experiência, amostragem e avaliação contínua.

Critérios objetivos podem ser avaliados automaticamente e determinados por uma análise cuidadosa da fonte de informação, por exemplo, *completude;*

Critérios de processos podem ser determinados através do processamento de consultas, e assim variando de consulta para consulta. São representativos, porém temporários. Um exemplo de critério de processo seria *tempo de resposta*.

Na Figura 10 é ilustrada uma série de critérios de QI compilados de [Naumann & Rolker, 2000] dentro de cada classe é especificado o método de avaliação que deve ser aplicado para a obtenção dos escores de cada um deles.

Classe	Critério de QI	Método de Avaliação
Critérios de Sujeito	Possibilidade	Experiência do Usuário
	Representação Concisa	Amostragem do Usuário
	Interpretabilidade	Amostragem do Usuário
	Relevância	Avaliação Contínua do Usuário
	Reputação	Experiência do Usuário
	Compreensibilidade	Amostragem do Usuário
	Valor Agregado	Avaliação Contínua do Usuário
Critérios de Objeto	Completude	Parsing e Amostragem
	Suporte ao Usuário	Parsing e Contratação
	Documentação	Parsing
	Objetividade	Entradas de Usuário
	Preço	Contratação
	Confiabilidade	Avaliação Contínua
	Segurança	Parsing
	Atualidade	Parsing
	Rastreabilidade	Entradas de Usuário Especialista
Critérios de Processo	Precisão	Amostragem, técnicas de limpeza de dados
	Volume de Dados	Avaliação Contínua
	Disponibilidade	Avaliação Contínua
	Representação Consistente	Parsing
	Latência	Avaliação Contínua
	Tempo de Resposta	Avaliação Contínua

Figura 10 Classificação de critérios de QI adaptada de [Naumann & Rolker, 2000].

Mais especificamente, os métodos de avaliação de critérios subjetivos, de objeto e de processo são:

- Experiência do usuário: diz respeito ao conhecimento do usuário adquirido ao longo do tempo com relação às fontes de dados, por exemplo: uso de relatórios e sistemas de informações;
- Amostragem do Usuário: o uso do método de amostragem deve utilizar resultados obtidos previamente das fontes de informações. Estes resultados, geralmente são adquiridos uma vez e somente quando a fonte de dados sofre alterações de importância;
- Avaliação Contínua do Usuário: trata-se de fazer uma verificação contínua das informações que o usuário recebe da fonte de informações ao longo do tempo, analisa cada informação recebida, não só amostragens. Esse método é o mais trabalhoso e menos compensador dos três, e deve ser aplicado quando é muito difícil conseguir extrair amostras representativas da fonte de informação;

- Contratação: Para alguns os escores podem ser avaliados considerando os termos de contratação entre a fonte de informações e o usuário. Exmplos de critérios que podem ser estimados em um acordo são: *suporte ao usuário* e *segurança*;
- Parsing: efetuar operações de *parsing* significa processar os metadados da fonte de dados. Naumann em [Naumann & Rolker, 2000] faz uma distinção entre *parsing estrutural* e *parsing de conteúdo. Parsing estrutural* considera estruturas da informação, tais como, posicionamento de campos em tabelas, presença de gráficos e tipos de tabelas. Por outro lado, *parsing de conteúdo* considera a informação atual e o conteúdo proveniente da fonte, tais como, *documentação e suporte;*
- *Amostragem:* A fim de evitar a tarefa de avaliar todo o conteúdo da fonte de informação na intenção de estimar o escore correto de um critério, técnicas de amostragem são aplicadas para selecionar um subconjunto representativo das informações e considerar apenas esse subconjunto na determinação do escore do critério de qualidade;
- Entrada de Usuário Especialista: o conhecimento de usuários especialistas é necessário para estimar alguns critérios, por exemplo, o critério de *rastreabilidade* (bem documentado, verificável) de uma fonte de dados. O especialista deve seguir algumas diretrizes para garantir a precisão e comparação dos escores;
- Avaliação Contínua: Trata-se da avaliação contínua do quão bem estão as fontes de informações em relação a cada critério. Um exemplo de critério que se aplica à avaliação contínua é a *confiabilidade*.

3.1 Perda de Informação

Perda de informação significa a diminuição dos escores dos critérios de QI [Mena et. al., 2000]. Como iremos ver nas seções mais adiante, alguns PDMS usam critérios de QI para mensurar os diferentes elementos de um PDMS: *peers*, mapeamentos, ontologias, resposta da consulta, dados, entre outros. Critérios de QI são medidas dinâmicas dentro do sistema, ou seja, suas medidas mudam a todo momento, por esta razão, o planejamento da consulta em um PDMS deve ser feito em tempo de execução, selecionando os melhores mapeamentos que levam a fontes de dados relevantes para o resultado da consulta. Visto que a perda de informações é algo presente nos caminhos dos mapeamentos, além de outros elementos de um PDMS, levar em consideração critérios de QI é especialmente valioso no cenário de sistemas distribuídos e dinâmicos.

3.2 QI em Sistemas de Integração de Dados

Um sistema de integração de dados provê uma visão unificada para usuários enviar consultas por várias fontes de dados autônomas e heterogêneas [Halevy et. al., 2006]. As

consultas são processadas em um esquema global que oferece uma visão integrada de várias fontes de dados. Então, o resultado da consulta de um usuário é na verdade uma integração de dados de várias fontes. Existem vários trabalhos que consideram o uso de QI para sistemas de integração de dados, particularmente em formulação de consultas, processamento (mediação) e otimização [Aggarwal & Yu, 2009] [Duchateau & Bellahsene, 2010] [Naumann & Leser, 1999].

Em um ambiente de integração de dados existem alguns elementos do sistema onde é possível inserir critérios de QI para análise, são eles: esquema da fonte de dados, processamento da consulta, consultas de mediação, esquemas integrados, seleção das fontes, integração dos dados e materialização dos dados [Batista, 2008]. Com base nesses elementos é possível perceber que podem ser associados aos mesmos vários critérios de QI, como detalhado a seguir.

Esquema da fonte de dados: as fontes de dados são autônomas, heterogêneas. Quando uma fonte participa do sistema, ela exporta um esquema que descreve todas as informações que serão compartilhadas. Estas fontes podem entrar e sair do sistema a qualquer momento, além de poderem estar completamente inacessíveis. Consequentemente, algumas questões sobre a manutenção do esquema das fontes surgem: primeiro, se o esquema da fonte mudar, o esquema de mediação precisa ser atualizado para refletir a mudança feita na fonte. Segundo, é desejado que a fonte de dados publique os dados mais atuais possíveis no intuito de manter a consistência entre o seu conteúdo e da informação oferecida pelo sistema de integração. Por esta razão, é interessante a aplicação de critérios de QI em fontes de dados. Podemos considerar os seguintes critérios: *completude do esquema, consistência, reputação, disponibilidade e atualidade.*

Consultas de mediação: em sistemas de integração de dados mapeamentos entre fontes de dados e o esquema global são pré-definidos [Halevy, 2000]. Duas abordagens são bastante usadas (i) *global-as-view* (GAV) e (ii) *local-as-view* (LAV). Na primeira, cada objeto do esquema global é expresso como uma visão (i.e. uma consulta) sobre a fonte de dados, enquanto que no LAV acontece o oposto, i.e., cada objeto em uma dada fonte é definida como uma visão sobre um esquema global. Dado um sistema com mapeamentos do tipo GAV, uma consulta do usuário pode ser decomposta em entidades de mediação, por esta razão é preciso, em tempo de execução, escolher qual a consulta mais adequada para calcular as entidades envolvidas na consulta do usuário. A análise da qualidade das consultas de mediação pode ser vinculada com os seguintes critérios: *disponibilidade*, *atualidade* e *tempo de resposta.*

Esquema de integração: o *esquema de integração* é uma visão integrada das fontes de dados subjacentes. O esquema de integração é composto por várias entidades de mediação e cada uma representa um conceito do mundo real obtidas das fontes de dados. Na avaliação da

qualidade do esquema de integração os seguintes critérios podem ser usados: *completude do esquema*, *consistência* e *minimalidade*.

Seleção da fonte: o sistema deve ser capaz de selecionar a melhor fonte de dados para responder uma consulta do usuário. Neste cenário é onde se tem o maior uso de critérios de QI [Naumann & Leser, 1999]. É possível reduzir significantemente o custo computacional na execução de uma consulta deixando de lado fontes de pouca qualidade. A seleção da fonte envolve os seguintes critérios: *reputação*, *completude dos dados*, *disponibilidade* e *atualidade*.

Resposta da consulta: ambientes de integração de dados oferecem execução de consultas diretamente às fontes de dados autônomas, dinâmicas, distribuídas e possivelmente heterogêneas. Por esta razão, na análise da QI em resposta da consulta os seguintes critérios podem ser levados em consideração: *disponibilidade, relevância, precisão, completude* e *tempo de resposta*.

Materialização dos dados: um dos principais problemas em um sistema de integração de dados é o processo da materialização seletiva [Batista, 2003]. Algumas partes dos dados usualmente inacessíveis ou estáticos devem ser materializados em um *data warehouse* e os dados mais dinâmicos irão ser acessados por consultas virtuais. Os critérios relacionados com materialização dos dados são: *atualidade*, *tempo de resposta*, *disponibilidade*, *reputação* e *verificabilidade*.

Integração dos dados: problemas na integração dos dados incluem gerenciar um objeto de mediação (diferentes objetos com diferentes valores) e selecionar a fonte quando informações contraditórias são encontradas em diferentes fontes. Outro problema é em relação à conversão de valores vindos de outras fontes e expressos em diferentes representações. QI pode ajudar na melhora de como lidar com informações similares ou contraditórias. Os seguintes critérios podem ser considerados: *completude dos dados*, *atualidade* e *verificabilidade*. O último passo na integração de dados é a entrega dos resultados ao usuário. Aqui, podemos adicionar critérios de QI para enriquecer e facilitar a avaliação geral do escore de qualidade dos resultados da consulta integrada: *precisão.*

A Tabela 1 sumariza para cada elemento de um sistema de integração, quais critérios de QI devem ser atribuídos, avaliados e analisados.

Elemento do Sistema	Critério de QI
Esquema da fonte de dados	Completude do esquema, consistência, reputação, disponibilidade, atualidade
Esquema integrado	Completude do esquema, consistência, minimalidade
Consultas de mediação	Disponibilidade, atualidade, tempo de resposta
Seleção de fontes	Reputação, completude dos dados, disponibilidade, atualidade, verificabilidade
Resposta das consultas	Disponibilidade, tempo de resposta
Materialização dos dados	Atualidade, tempo de resposta, disponibilidade, reputação, verificabilidade
Integração dos dados	Completude dos dados, atualidade, verificabilidade, precisão

Tabela 1 Elementos de um sistema de integração de dados e critérios de QI [Batista, 2008].

3.3 QI em PDMS

PDMS são considerados a evolução dos sistemas de integração de dados (SID) [Pires, 2009] [Souza et. at., 2011], SID oferecem o compartilhamento de dados por meio de um esquema global (esquema mediador) onde consultas são executadas e os dados retornados para o usuário. Contudo, os dados propriamente ditos permanecem nas fontes de dados locais. A principal limitação dos sistemas de integração de dados é que os mesmos necessitam de um esquema de mediação [Tantarinov & Halevy, 2004]. Em alguns sistemas, há a necessidade de compartilhar dados sem depender de um ponto central, por esta questão, que se dá a evolução dos sistemas de integração de dados chegando até os PDMS. PDMSs são diferentes dos sistemas de integração de dados, uma vez que não possuem um esquema de mediação, responsável pela formulação da consulta do usuário. Em um PDMS um *peer* contém toda ou parte de uma informação que serve de resposta para uma dada consulta. Além disso, outros *peers* podem oferecer a mesma resposta com diferentes níveis de qualidade e custo.

Devido ao motivo da dinamicidade em um ambiente PDMS, a relação de confiança, identificação e classificação das fontes de dados tem se tornado de grande relevância. Como resultado, em PDMSs a QI da resposta à consulta (*query answer*) depende não apenas da qualidade dos dados, mas também, da qualidade dos mapeamentos existentes entre os *peers* vizinhos [Yatskevich et. at., 2006]. Particularmente, em relação à qualidade de dados, *peers* podem armazenar dados de baixa qualidade e possivelmente desatualizados, errôneos, duvidosos ou até mesmo incompletos. Com relação à qualidade dos mapeamentos,

habitualmente são estabelecidos entre pares de *peers* que estão semanticamente agrupados [Pires, 2009]. Devido à heterogeneidade, às vezes, um conceito em um *peer* não corresponde exatamente ao conceito de outro *peer*, tornando o mapeamento incorreto ou incompleto [Heese et. al., 2005].

Em um PDMS existem pelo menos três tipos de fatores de tempo de execução relacionados com o sistema que podem influenciar na resposta de uma consulta do usuário, bem como na qualidade das respostas da consulta [Giunchiglia & Zaihrayeu, 2002] [Zaihrayeu, 2006]. São eles:

Variação da Rede: *peers* podem mudar seus próprios dados, mudar seus esquemas, redefinir mapeamentos e novos *peers* podem entrar e sair da rede a qualquer momento. Por esta razão, a mesma consulta enviada para um *peer*, em um momento diferente, pode resultar em diferentes respostas com níveis de qualidade também diferentes.

Variação do *Peer*: *peers* podem sair e entrar na rede a qualquer momento, bem como mudar seus esquemas. Por conseguinte, a mesma consulta submetida ao mesmo tempo pelos mesmos *peers* irá resultar em diferentes grafos de propagação da consulta. Devido a isto, tanto os resultados da consulta quanto a qualidade podem ser diferentes.

Variação da Consulta: as mesmas consultas enviadas para o mesmo *peer* ao mesmo tempo pode levar a diferentes grafos de propagação da consulta, e desse modo, pode produzir diferentes resultados com diferentes níveis de qualidade.

A maioria das abordagens de roteamento e reformulação de consultas em PDMSs não levam em consideração aspectos de qualidade do mapeamento. Como dito anteriormente, mapeamentos são determinados entre pares de *peers* semanticamente agrupados. Por se tratar de um ambiente bastante heterogêneo, algumas vezes, conceitos de um dado *peer* podem não ter uma correspondência exata em outro *peer*. Contudo, os esquemas destes *peers* (normalmente ontologias) comumente se sobrepõem em certo grau de semântica, oferecendo assim significados mais ricos e, consequentemente, melhores correspondências entre os conceitos dos *peers*.

Por estas razões, a consideração de QI em resposta à consulta para PDMSs, consistindo em um grande número de *peers* e mapeamentos entre eles, tem sido considerado um problema importante. Na verdade, alto nível de QI na resposta à consulta tem sido considerado como o fator principal para um bom fluxo de dados entre os *peers*, preservando (com o melhor nível de aproximação) sua *solidez* e *completude* [Giunchiglia & Zaihrayeu, 2002]. Neste contexto, solidez significa que os dados fornecidos pelos *peers* refletem seu esquema local. Enquanto que completude, diz respeito a quão completos os dados são.

3.4Exemplos de PDMS com QI

3.4.1 Humboldt Discoverer

O Humboldt Discoverer é um PDMS que estende os PDMSs clássicos incluindo algumas dimensões a mais em sua arquitetura, para assim oferecer um melhor serviço de descoberta de fontes de informações, bem como fonecer uma melhora significativa no roteamento das consultas [Herschel & Heese 2005].

3.4.1.1 Arquitetura

O Humboldt Discoverer trabalha com quatro dimensões (Figura 11):

(i) a dimensão de PDMS, composta por *peers* e mapeamentos entre eles.

(ii) a dimensão semântica, a qual consiste de ontologias e mapeamentos entre elas.

(iii) a dimensão *Web,* onde cada *peer* mantém um conceito armazenado que indexa o *peer* para seus respectivos vizinhos.

(iv) a dimensão de qualidade, à qual influencia a resposta da consulta em todas as demais dimensões.

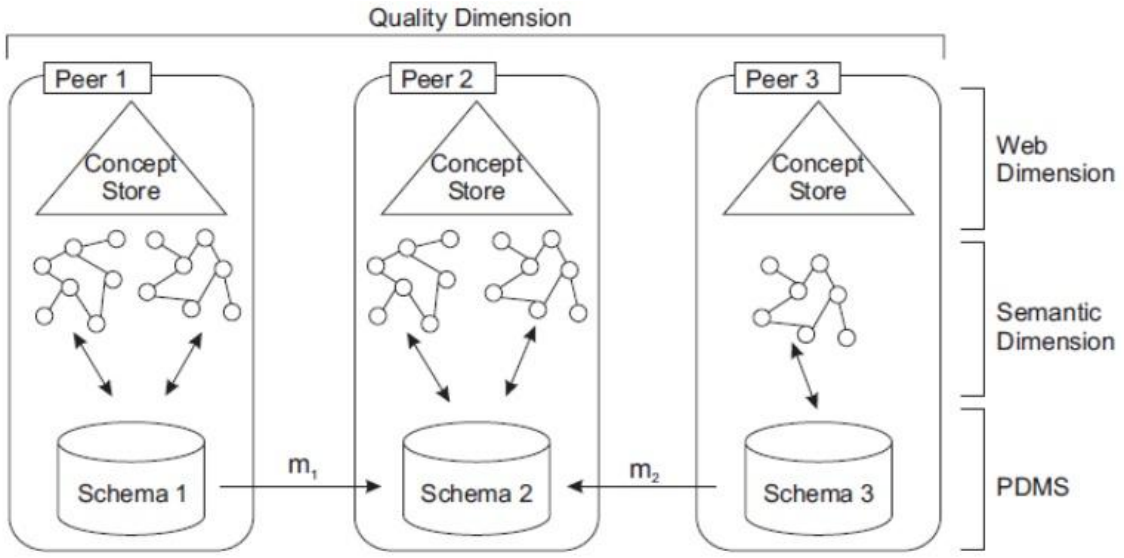


Figura 7. Arquitetura estendida do Humboldt Discoverer [Herschel & Heese 2005].

Como dito anteriormente, estas dimensões do Humboldt estendem um PDMS tradicional fornecendo busca eficiente de informações e um melhor roteamento das consultas:

- Dimensão semântica: provê um caminho alternativo aos usados em PDMSs clássicos para os mapeamentos semânticos entre os esquemas dos *peers*. Essa camada consiste de ontologias e mapeamentos entre elas. Para se ter vantagem da dimensão semântica para o processamento de consultas, um *peer* pode ter mapeamentos do seu esquema local para uma ou mais ontologias. Desde que as ontologias forneçam informações semânticas sobre diferentes grupos;
- Dimensão web: um índice P2P é introduzido, que auxilia a resolver o problema de localização de fontes relevantes para uma consulta que não foi mapeada anteriormente, ou seja, que está sendo executada pela primeira vez. Este índice P2P é constituído de um *concept store* o qual é responsável por armazenar um resumo dos conteúdos de cada *peer* conhecido. Este resumo é formado por (i) uma representação do conteúdo dos *peers*, baseados nos seus conceitos e propriedades, (ii) metadados sobre QI. Desta maneira, se torna mais fácil localizar mais *peers* que podem contribuir com dados relevantes à consulta, em contraste com o mecanismo adotado por sistemas P2P baseados em índices e palavras-chave.
- Dimensão de qualidade: influencia o processamento da consulta em todas as demais dimensões. No nível do PDMS e na dimensão semântica, a qualidade das respostas da consulta depende do mapeamento entre os *peers*. À medida que uma consulta é encaminhada para outros *peers*, a qualidade dos *mapeamentos* é registrada por meio do escore do critério *completude* de cada *peer* envolvido no mapeamento.

3.4.1.2 Processamento da Consulta

O processamento de consultas no Humboldt ocorre em duas fases. Na primeira fase (passos 1 - 3), a consulta é processada na *camada PDMS*, utilizando os mapeamentos relacionais entre os *peers* (ver Figura 12). Se o resultado dessa primeira fase for insuficiente em relação à quantidade de informações oferecidas pelas fontes ou mesmo resultados vazios, a segunda fase (passos 4 - 7) é iniciada e a consulta é processada ao longo dos mapeamentos das ontologias.

Nessa segunda fase, a consulta é traduzida para uma representação gráfica e a base de conceitos é consultada para obtenção de fontes de informação que possam responder à consulta, ao achar estas fontes de informações, os resultados serão integrados e mostrado ao usuário.

A Figura 12 representa os passos tomados pelo processamento da consulta que são explicados abaixo.

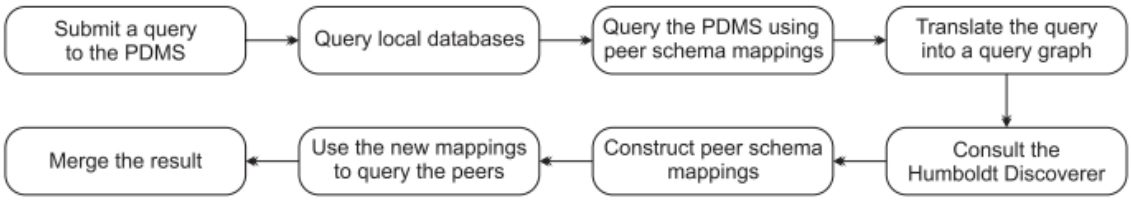


Figura 8 visão geral do processamento de consultas [Herschel & Heese, 2005].

1. Um *peer* recebe uma consulta de um usuário escrita em SQL.
2. Este *peer* usa o seu banco de dados local para responder à consulta.
3. O *peer* identifica mapeamentos entre esquemas e roteia a consulta por estes caminhos. Os *peers* destinos assim que recebem a consulta, executam a mesma em seus bancos de dados e retornam o resultado.
4. Se o resultado proveniente dos *peers* consultados forem insuficientes, ou seja, vazios, continuamos com a segunda fase. Na segunda fase, o *peer* traduz a consulta em um grafo de consulta por meio do mapeamento esquema – para – ontologia e consulta a sua *concept store* no intuito de encontrar *peers* para encaminhar à consulta.
5. A *concept store* devolve uma lista de *peers* os quais podem contribuir para a resposta da consulta.
6. Para cada *peer* na lista, gerada no passo anterior, o *peer* cria um mapeamento por meio da composição dos mapeamentos da ontologia que são definidos na camada semântica.
7. Os novos mapeamentos formam um novo caminho no PDMS chamado de atalho (Figura 13). Por esta razão, no futuro o *peer* pode usar este atalho como uma forma direta de consultar o outro *peer*, melhorando assim o roteamento da consulta.
8. Todos os resultados são agregados em um resultado geral da consulta.

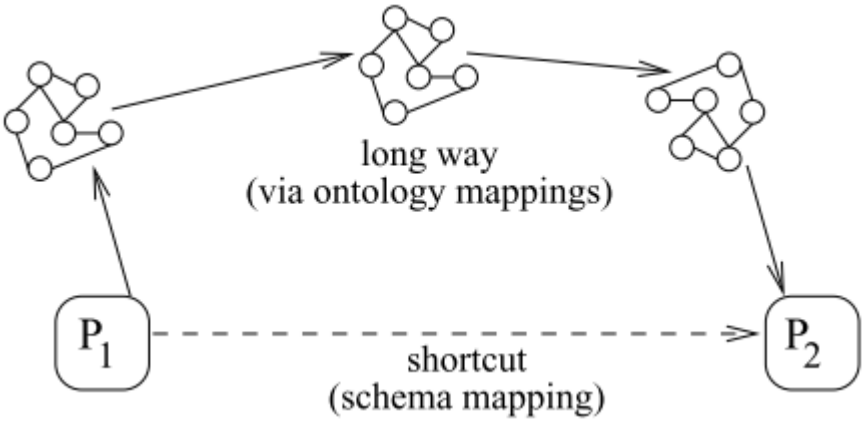


Figura 9 Usando mapeamentos de ontologias para gerar um mapeamento direto entre os esquemas [Heese et. al., 2005].

3.4.1.3 Qualidade da Informação

O Humboldt faz uso de QI para classificar os *peers* relevantes para responderem determinadas consultas, bem como faz uso da mesma para avaliar a qualidade do mapeamento entre os *peers* e suas respostas, isto para as camadas semânticas e de PDMS.

A camada *Web* é a responsável por localizar os *peers* relevantes para a consulta, ou seja, aqueles que possuem dados de interesse da consulta. Nesta camada são aplicadas medidas para localizar e ordenar estes *peers*. Na intenção de ordenar estes *peers*, quando uma consulta chega à camada *Web* a *concept store* é consultada para verificar os *peers* que oferecem conceitos relevantes para a consulta. A ordenação dos *peers* é influenciada por dois fatores: a QI do *peer* e a qualidade do mapeamento entre os grafos dos *peers*. Para obter o primeiro fator são fixados em cada *peer* escores para três critérios de QI, a saber, *concept coverage* (CC), *timeliness* (TLN) e *peer count* (PC). CC é o número de conceitos permitidos por este *peer* em relação ao número de conceitos contidos na ontologia. TLN provê a atualidade dos dados armazenados. Este critério pode influenciar o planejamento da consulta quando informações confiáveis e atualizadas estão disponíveis. PC é o número de *peers* conhecidos pelo respectivo *peer* usando a mesma ontologia que os outros *peers*. Para obter o segundo fator (qualidade do mapeamento) são usados dois critérios de QI, *Extensinal completeness* e *Intensional completeness* [Heese et. al., 2005]. *Extensional Completeness* diz respeito à proporção do tamanho de um conjunto de objetos em relação ao número de *todos* os objetos acessíveis. Enquanto que, *Intensional completeness* pode ser definida de maneira ortogonal à *Extensional* como a proporção dos esquemas dos elementos de um determinado conjunto e o esquema (*intension*) de *todos* os *peers*.

Outro critério levado em consideração é a *relevância*, usada para avaliar o grau de coerência de um resultado da consulta com as expectativas do usuário que efetuou a consulta. Este é um tipo de critério subjetivo que deve ser avaliado pelo usuário.

3.4.2 Chatty Web

O Chatty Web é um PDMS criado na École Polytechnique Fédérale de Lausanne (EPFL), Suíça. Sua estrutura tem como base a Web Semântica e provê o processamento semântico de informações para as máquinas, por meio de meta-modelos como RDF[3], OWL[4].

3.4.2.1 Arquitetura

O modelo de rede do sistema Chatty Web [Aberer et. al., 2003] é baseado no protocolo Gnutella. Por esta razão, *peers* podem enviar mensagens *ping* para outros *peers* e receberem mensagem *pong* no intuito de conhecer a estrutura da rede. Como uma extensão ao protocolo Gnutella, os *peers* podem também enviar seus próprios identificadores dos esquemas como parte da mensagem *pong*.

Cada *peer* mantém uma vizinhança $N(p)$ de *peers* previamente selecionados por meio da mensagem *pong* recebida. Os *peers* nesta vizinhança são diferenciados entre aqueles que compartilham o mesmo esquema, $N_e(p)$, e aqueles que possuem um esquema diferente, $N_d(p)$ com mostrado na Figura 14.

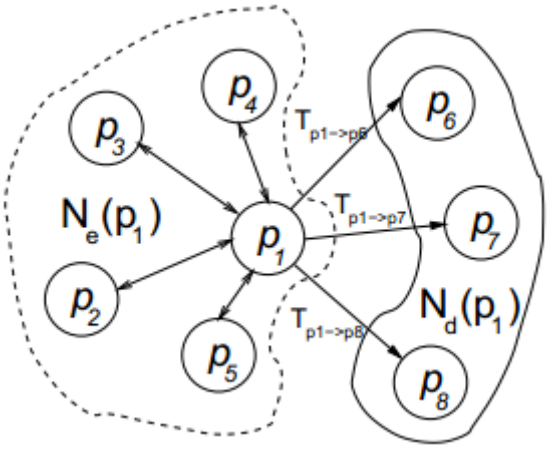


Figura 10 O modelo da rede [Aberer et. al., 2003].

[3] Resource Description Framework, W3C. http://www.w3.org/RDF/
[4] Web Ontology Language, W3C. http://www.w3.org/TR/owl-features/

3.4.2.2 Processamento da Consulta

Em um conjunto de *peers* P e cada *peer* $p \in P$, e todos possuem um mecanismo básico para comunicação que permite estabelecer conexões com outros *peers*. *Peers* podem mandar consultas de um ponto para outro bem como seus identificadores de esquemas.

Consultas podem ser emitidas para qualquer *peer* por meio de *mensagens de consultas*. Uma mensagem de consulta contém um identificador de consulta *id*, a consulta potencialmente transformada () *q*, a mensagem da consulta do *peer* que originou a consulta e o rastreador da tradução *TT*, que é uma lista de pares de *peers* no formato *peer* origem e *peer* destino $\{(P_{origem}, P_{destino})\}$ [Aberer et. al., 2003] o qual tem como propósito acompanhar as transformações já sofridas pela consulta. Ao final, os resultados das consultas em cada *peer* são agregados e retornados ao *peer* de origem.

3.4.2.3 Qualidade da Informação

No sistema Chatty Web, as medições da qualidade do mapeamento entre esquemas são aplicadas às consultas. Elas são atualizadas durante a execução da consulta na rede. A qualidade do mapeamento do esquema é medida por *similaridade sintática* e *semântica*.

Similaridade sintática é um tipo de critério intrínseco, ou seja, que se refere apenas à consulta que está sendo processada e à tradução que está sendo feita. A mesma visa oferecer um resultado quantitativo das informações perdidas, resultante de uma consulta quando um atributo de um esquema não existe em outro esquema.

Similaridade semântica é um tipo de critério extrínseco, ou seja, que se refere ao grau de concordância que pode ser obtida entre vários *peers*.

Uma melhor maneira de entender semântica é considerar que é um acordo entre os *peers*. Se dois *peers* entrarem em um acordo no grau de similaridade de seus esquemas, então eles irão gerar traduções compatíveis. Estas medidas irão permitir que a qualidade dos atributos preservada durante a reformulação seja avaliada. Para que isto seja possível, são apresentados dois mecanismos para *Cycle Analysis* e *Result Analysis*. O primeiro mecanismo é baseado na análise da fidelidade das traduções no nível de esquema. Quando uma consulta entra novamente em um domínio semântico que já passou anteriormente, é iniciado um ciclo que computa a análise da similaridade entre os esquemas. O segundo mecanismo é baseado na análise dos

resultados retornados pelo primeiro mecanismo. Então, se neste resultado os *peers* concordam em seus níveis de semântica entre seus esquemas, consequentemente, eles possuem as mesmas dependências de dados. Para cada dependência funcional, os *peers* analisam se os valores dos atributos dependentes realmente combinam. A porcentagem de combinações provê então o grau de *fidelidade* que pode ser colocado em um *peer*.

3.4.3 ESTEEM

O Esteem (Emergent Semantics and cooperation in multi-knowledgE EnvironMents) é um sistema baseado em comunidade para o apoio à colaboração semântica entre um conjunto de *peers* independentes, sem conhecimento recíproco antecedente e nem relacionamentos pré-definidos [Montanelli et. al., 2010]. O objetivo do Esteem é oferecer uma plataforma integrada para descoberta/compartilhamento de dados e serviços em um ambiente baseado em comunidade e arquitetura P2P. Suas motivações por trás disso são: superar a falta de técnicas para apoio à interpretabilidade semântica entre os sistemas; superar a falta de técnicas para garantir agregação/reconhecimento eficiente e não-supervisionado de *peers* com conteúdos similares; superar a falta de técnicas para avaliar o nível de confiança dos *peers*.

3.4.3.1 Arquitetura

O Esteem é definido como um sistema P2P baseado em comunidade para a colaboração semântica. Este cenário de colaboração possui *peers* que fornecem vários tipos de informações de diversos domínios, por esta razão este cenário é conhecido como multi-conhecimento. Por este motivo, nenhum ponto central é definido no sistema para gerenciar os *peers* que compõem a rede tornando o ambiente autônomo.

Uma comunidade semântica emerge no Esteem a partir de um grupo de *peers* que possuem interesses em comum, explícitos na forma de uma ontologia. Esta ontologia é um meio conceitual que permite que os *peers* no sistema se tornem cientes dos seus interesses em comum, ou seja, reconheçam a existência de conhecimento coletivo o qual pode ser organizado em uma estrutura concreta (por exemplo, comunidade).

A Figura 15 define que o conhecimento que um *peer* pode fornecer para o sistema é obtido por meio de elementos: ontologia do *peer*, ontologia de serviço, contexto atual e perfil de confiança e qualidade dos dados.

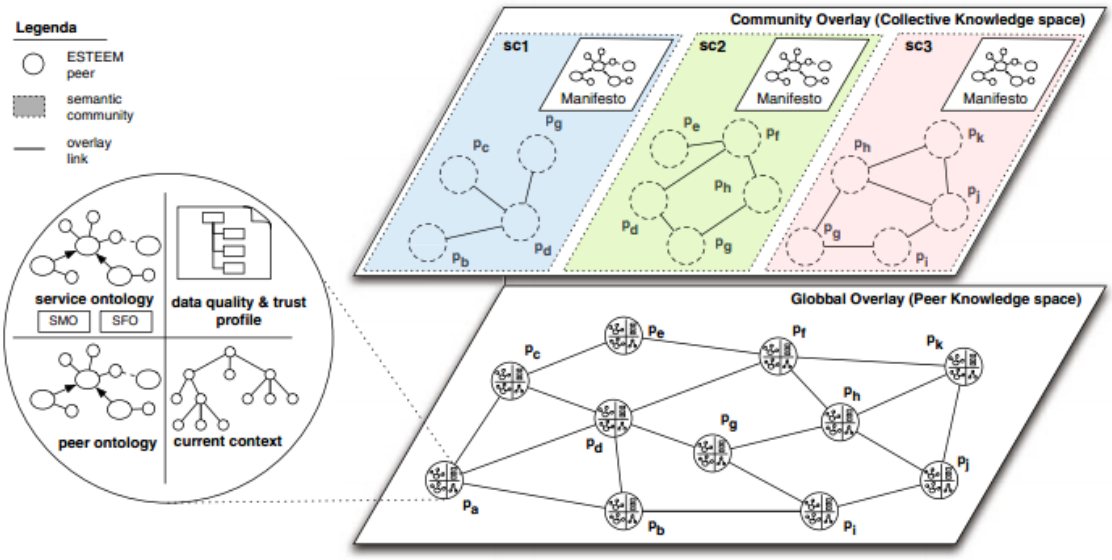


Figura 15 A ferramenta de conhecimento do Esteem [Montanelli et. al., 2010].

Ontologia do *peer*: é o ponto central de conhecimento do *peer* e oferece uma descrição semântica dos dados compartilhados pelo *peer*. A ontologia do *peer* é consultada no intuito de avaliar se o conhecimento fornecido por um *peer* que realiza uma consulta pode ser realmente enviado como resposta à consulta. Consequentemente, a ontologia do *peer* é também consultada para derivar os interesses atuais do *peer*, bem como para determinar as comunidades semânticas a se conectar.

Ontologia do serviço: oferece uma descrição semanticamente rica dos serviços dos *peers* que estão disponíveis para compartilhamento. Por serviço, devemos entender como funcionalidades concedidas pelos *peers* como, por exemplo, parâmetros de entrada e saída (I/O). Na verdade, são aspectos funcionais do serviço.

Contexto atual: é uma estrutura em forma de árvore responsável por descrever o perfil do *peer*, seus interesses, sua situação e coordenadas temporal/espacial no momento de interação com a rede do ESTEEM.

Perfil de confiança e qualidade dos dados: envolve a mensuração das métricas de qualidade sobre os dados do *peer* que estão disponíveis para compartilhamento com outros *peers*. Cada *peer* pode associar metadados de qualidade a seus esquemas exportados. Estes metadados representam medidas de qualidade provenientes de dimensões de qualidade. No ESTEEM, atualmente, as métricas relacionadas à

qualidade usadas para análise são: *accuracy*, *completeness*, *internal consistency* e *currency* (ver seção 3.4.3.3).

A plataforma Esteem é constituída sobre uma rede P2P não-estruturada onde uma comunidade semântica é definida por meio de uma rede de sobreposição (*overlay network*) [Valduriez & Pacitti, 2004]. No Esteem uma comunidade semântica é dada por uma 4-tupla na forma *sc* = (*UCI, N, L, M*) onde *UCI* é o identificador universal da comunidade (*Universal Community Identifier*) que caracteriza uma comunidade unicamente, *N* e *L* são um nome simbólico e uma linguagem natural para descrição dos interesses da comunidade, respectivamente, e *M* é a ontologia da comunidade. De acordo com seus interesses um *peer* no Esteem pode ser incluído em nenhuma comunidade ou em várias por meio da entrada em uma comunidade correspondente. Por exemplo, na Figura 14, o *peer* P_g, se junta às comunidades sc_1, sc_2 e sc_3, enquanto que o *peer* P_a não participa em nenhuma comunidade semântica.

Um mecanismo para busca é definido no Esteem para diferenciar:

A **fase de descoberta**, que é baseada no *matching* de ontologia, onde consultas são definidas para identificar as comunidades/*peers* que são capazes de oferecer conhecimento relevante respeitando o domínio de interesse dado previamente.

A **fase de compartilhamento**, que é baseada na definição de mapeamentos P2P, onde consultas padrões são definidas ponto-a-ponto para aquisição de dados atuais bem como execução de serviços.

Para este fim, um *peer* no Esteem é organizado em uma arquitetura baseada em componentes como descrito na Figura 16. O usuário explora o Esteem satisfazendo suas necessidades de colaboração (***Data & service discovery***). Técnicas para contexto e gerenciamento de qualidade/confiança podem ser executadas durante as interações dos *peers* para melhorar a eficiência da fase de compartilhamento de acordo com os requisitos da colaboração considerado no dado momento. No nível de sistema, comunidades semânticas são usadas durante a fase de descoberta, oferecendo uma arquitetura para que a requisição de um único usuário seja propagada para grupos de potenciais *peers* que podem contribuir para a melhora da resposta à consulta (***Semantic community & routing***). O gerenciamento da conectividade do *peer* bem como a manutenção da comunidade *overlay* é feito pela camada ***Network & overlay***.

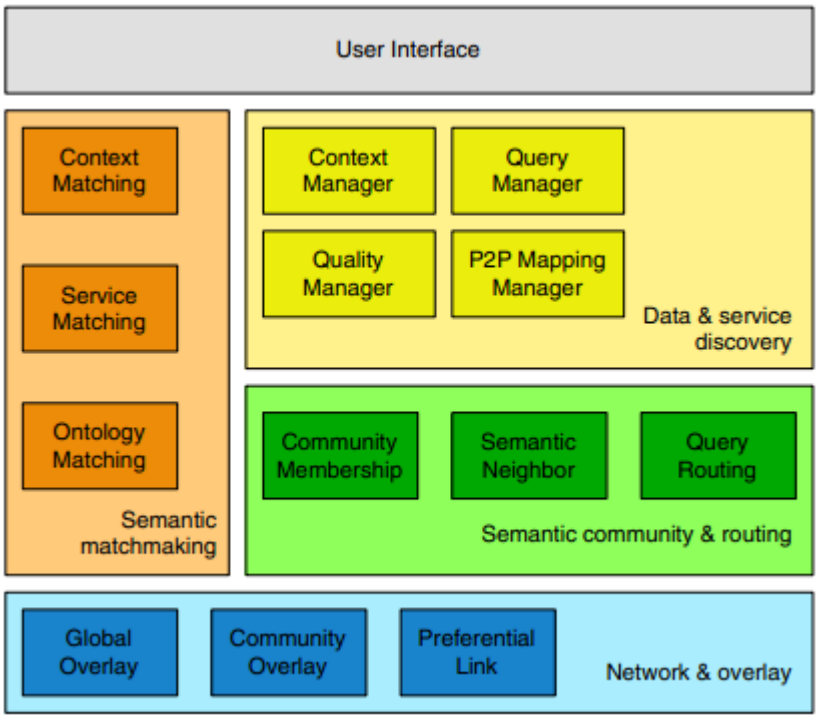


Figura 11 Arquitetura de um *peer* no Esteem [Montanelli et. al., 2010].

3.4.3.2 Qualidade da Informação

No intuito de apoiar a confiança e a qualidade da informação o sistema ESTEEM faz uso do sistema DaQuinCIS [Scannapieco et. al., 2004], uma arquitetura para gerenciar a qualidade dos dados em sistemas de informação cooperativos. O DaQuinCIS permite a difusão de dados com qualidade e explora a replicação de dados no intuito de melhorar a qualidade geral dos dados. Quatro critérios de QI são levados em consideração no sistema DaQuinCIS. Estes critérios estão relacionados apenas aos dados, não se estendendo a outros elementos como o esquema lógico e o formato do dado. Os critérios mencionados anteriormente são os seguintes: (i) *Accuracy*: se refere à proximidade de um valor *v* para um valor *v'* considerado como correto. Mais especificamente, *accuracy* é a distância entre *v* e *v'*, sendo *v'* o valor considerado como correto; (ii) *Completeness*: é o grau de cada valor do elemento de um esquema que está presente na instância do elemento do esquema; (iii) *Currency*: refere-se apenas a dados cujos valores mudam ao passar do tempo, por exemplo, *endereço*. Mais especificamente, *currency* é a distância entre o instante em que o valor muda no mundo real e o instante em que o valor é modificado no sistema de informação; (iv) *Internal Consistency*: é o grau em que cada valor dos atributos de uma instância de um esquema satisfaz o conjunto de regras semânticas específicas definidas no esquema.

O Esteem faz a computação dos escores de QI durante a fase de processamento da consulta por meio do DaQuinCIS.

3.4.4 System P

O System P [Roth et. al., 2006] é um PDMS apoiado no modelo relacional que fornece uma estrutura para comunicação entre *peers* na Internet. Cada *peer* consiste de um conjunto de fontes locais representadas pelo modelo relacional, mapeamentos *Local-as-View* (LaV) e *Global-as-View* (GaV) entre os esquemas dos *peers* e características de planejamento da consulta e execução.

3.4.4.1 Arquitetura

A Figura 16 ilustra sua arquitetura. *Peers* no System P consistem em esquemas relacionais, um conjunto de fontes locais conectada ao esquema do *peer* por meio de mapeamentos locais, os quais são usados para conectar um *peer* a outros *peers* (*Peer Schema*). No System P qualquer Sistema de Gerenciamento de Banco de Dados Relacionais (SGBDR) com um *driver* JDBC[5] pode ser acoplado como uma fonte local, por meio do *JDBC Local Source* mostrado na Figura 17. O planejamento das consultas é completamente descentralizado e implementado localmente nos *peers*, por meio dos mapeamentos GAV e LAV (*Local Mappings*), uma árvore baseada em regras é criada no *peer* que está recebendo a consulta original bem como em cada *peer* que é consultado durante o processamento da consulta. Depois que a consulta passa por otimizações e possíveis podamentos o planejamento da consulta local é executada no módulo *Query Execution.*

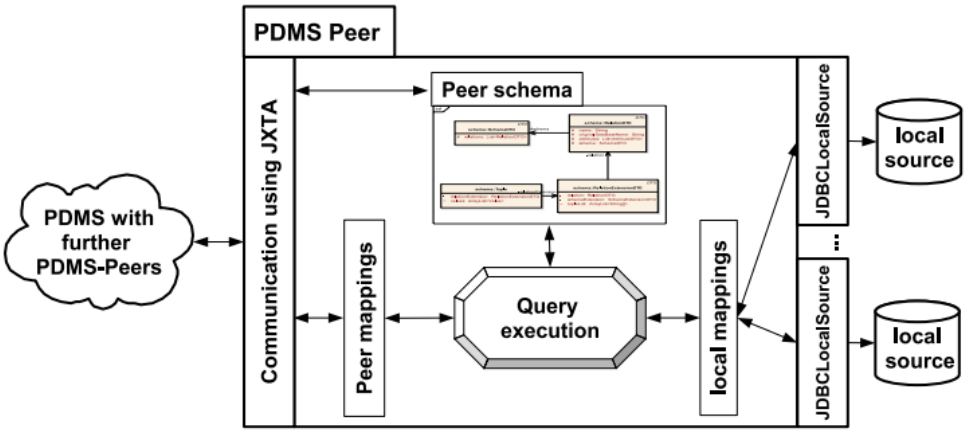


Figura 17 Arquitetura do System P [Roth et. al., 2006].

[5] Java DataBase Connectivity: http://docs.oracle.com/javase/tutorial/jdbc/overview/index.html

3.4.4.2 Processamento da Consulta

Em [Roth & Naumann, 2005], é apresentada uma solução para o processo de reformulação de consultas em um PDMS que explora características de completude nos mapeamentos entre os *peers*. Na abordagem apresentada por eles, faz-se uso de uma estratégia descentralizada que guia os *peers* na decisão de qual mapeamento provavelmente resultará em uma melhor reformulação da consulta. O objetivo é decidir se vale a pena enviar a consulta àquele *peer* vizinho ou se a mesma deve ser planejada e podada (fragmentada) naquele momento.

Como um exemplo ilustrativo do processamento da consulta no System P, levamos em consideração a Figura 18, um conjunto pequeno de *peers* com seus mapeamentos. Neste exemplo, o processamento da consulta considera a *competude* da resposta detalhada na Seção 3.4.4.3.

Os mapeamentos podem ser incompletos, por exemplo, o mapeamento entre $P_1 \rightarrow P_5$ não mapeia o atributo *E*. Mapeamentos podem também ter predicados que agem como filtros e são expressos como porcentagens. Por exemplo, o mapeamento $P_2 \rightarrow P_6$ possui o predicado ($D > 10$), o qual restringi 60% das tuplas.

Durante o planejamento da consulta um modelo de custo e o modelo de completude (*completeness model* Seção 3.4.4.3) são levados em consideração no intuito de escolher *peers* que potencialmente contribuam com um grande número de resultados e avaliar a qualidade do resultado da consulta, respectivamente.

No modelo de custo citado, para acessar um *peer* é cobrado um valor de custo 1. Para assegurar que os valores de custo são usados de forma inteligente, i.e., que muitos resultados sejam recuperados, é usado um modelo simples de completude (Seção 3.4.4.3).

Assumindo que o PDMS por completo possui 100 itens nos cinco atributos *A, B, C, D, E*. Por exemplo, o *peer* P_5 armazena 90 tuplas, todas com dados sobre os cinco atributos. Por esta razão P_5 tem uma completude de 90%. Contudo, se estes dados são passados pelo mapeamento até P_1, o número de tuplas é reduzido para 36 tuplas (40% de 90 tuplas) e o número de atributos é reduzido para quatro. Por este motivo, P_5 tem uma completude de aproximadamente 29% (144/500) visto da perspectiva de P_1. Este efeito de decadência da completude é acumulado durante os caminhos percorridos entre os mapeamentos, ou seja, durante o processamento da consulta.

Considere uma consulta *Q* no *peer* P_1 pedindo por todos os objetos da relação R_1. O custo afixado a esta consulta é de 5, i.e., nós podemos acessar cinco *peers* para

responder à consulta. O próprio banco de dados de P_1 pode responder à consulta com uma completude de 20%. P_1 tem mapeamentos para P_5 e P_2 e agora tem que decidir para onde enviar à consulta. Supondo que o *peer* escolhido seja P_2, a completude total é incrementada para incluir os dados de P_2: 20% + 40% - (20% * 40%) = 52%. A subtração realizada é para computar os elementos redundantes em P_1 e P_2. Neste ponto, o custo restante é de 3 ainda afixado ao P_2 que decide usar no encaminhamento da consulta para o P_4. P_4 incrementa a completude para 55.45%, visto que este *peer* oferece apenas quatro dos cinco atributos e apenas 10% das tuplas. Como não há mais mapeamentos a seguir, P_4 devolve o custo restante no valor de 2 para P_2, o qual gasta este custo para encaminhar a consulta para o P_6, previamente ignorado.

Como P_6 não possui uma fonte de dados, consequentemente, não pode contribuir para a completude do resultado da consulta, então, encaminha a mesma para o P_5. O conjunto de dados do P_5 é alcançado, todavia, por um caminho alternativo ao anterior (P_1 → P_5), este caminho alternativo conserva mais dados. O resultado final depois de todos os *peers* responderem a consulta e enviarem seus resultados de volta tem uma completude total de 67.68%.

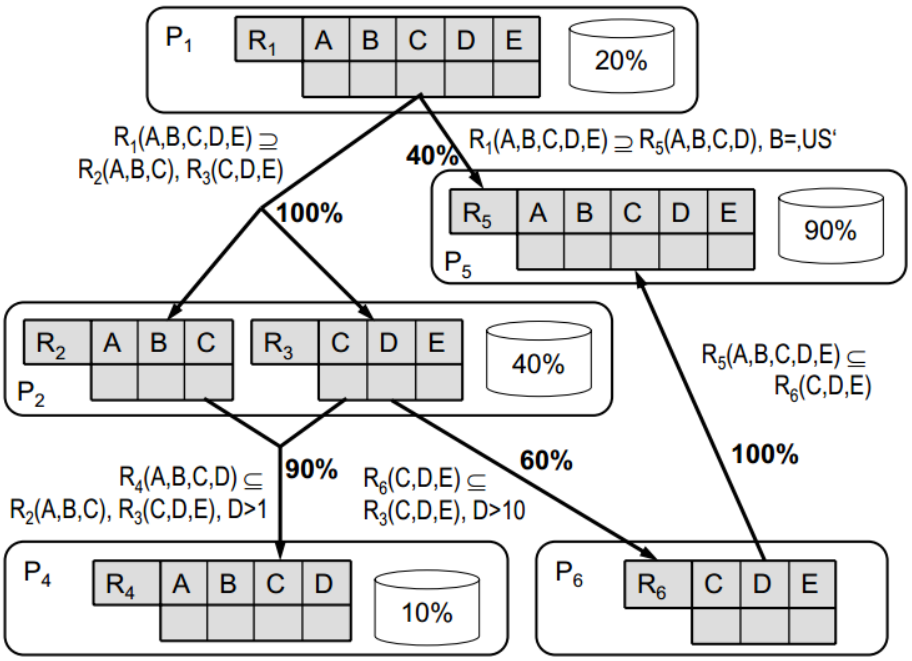


Figura 12 Um pequeno PDMS com mapeamentos e filtros [Roth et. al., 2006].

3.4.4.3 Qualidade da Informação

No System P, os autores fazem uso de duas dimensões *density* e *coverage* para estimar, de forma geral, o critério de *completeness* da resposta da consulta [Roth et. al., 2006].

Coverage descreve a proporção do tamanho de um conjunto de tuplas em relação ao número de todas as tuplas que compõem o PDMS. Esta medida é aplicada nos dois elementos do PDMS: *peers* e resultado da consulta.

Para calcular a *coverage* os autores partem do pressuposto do "mundo fechado" (*closed world*) para todo o sistema. Geralmente, os usuários enxergam um PDMS como um único banco de dados descrito por seu esquema local. Por esta razão, o tamanho do mundo $|W_Q|$, referido por uma consulta Q sobre seu próprio esquema, é o número de tuplas correspondentes à consulta que pode ser alcançada por meio dos mapeamentos existentes na rede.

Definição para Coverage: Seja D_Q seja um conjunto de tuplas respondendo a uma consulta Q. A *coverage* de D_Q com respeito a um mundo W_Q é $c(D_Q) := |D_Q|/|W_Q|$.

Density é o conjunto de atributos A_Q requisitados em uma consulta. Primeiramente, existe um problema inerente a *density*, valores nulos podem ser retornados de um *peer*. Em segundo lugar, atributos requisitados em uma consulta podem não estar disponíveis em determinados *peers* no PDMS.

Definição para density: Seja a_R um atributo da relação R. Uma *projeção* da tupla t desta relação para a_R é representada por $t[a_R]$. Com $\perp$ denotanto *null*, o atributo *density* de um conjunto de tuplas D para R é definida como $d(a_R) := |\{t \in D \mid t[a_R] \neq \perp\}| / |D|$.

Completeness de forma intuitiva, *completeness* pode ser entendida como uma medida agregada da taxa de quantidade de dados em um determinado conjunto de dados em relação à quantidade de dados no mundo W_Q. Na verdade, é uma combinação de *coverage* e *density*, cujo objetivo é a maximização dos mesmos. O escore da *completeness* de um conjunto de dados D pode ser calculado como $C(D) = c(D) * d(D)$ e $0 \leq C \leq 1$.

Usando o modelo de completude acima é possível calcular a contribuição oferecida por todos os *peers* para a resposta da consulta e também o impacto causado pela perda da informação. Além disso, este modelo apóia a distribuição do esquema de custo para selecionar potenciais *peers* que forneçam resultados relevantes à consulta.

Os autores chamam a atenção para a inclusão de outros critérios de QI no contexto de PDMS, além do usado *completeness*.

4. Conclusões

Neste presente documento, foi descrito em detalhes o que é um PDMS e suas principais utilizações. Além disso, conceitos como qualidade da informação foram introduzidos visando dar ênfase a um assunto que pode ser muito proveitoso para a avaliação da qualidade dos diversos elementos de um PDMS. Uma descrição de alguns PDMS que fazem uso de qualidade da informação também foi apresentada.

Referências Bibliográficas

Arazy, O., Kopak, R. (2010). On the Measurability of Information Quality. Journal of The American Society For Information Science and Technology.

M. C. Batista. Otimização de Acesso em um Sistema de Integração de Dados através do uso de Caching e Materialização de Dados, Master Thesis, Federal University of Pernambuco, 2003.

[Batista, M. C. M. 2008] "Schema Quality Analysis in a Data Integration System". Tese de Doutorado, Centro de Informática – UFPE.

[Angeles, P. and MacKinnon, L. 2005]. Quality Measurement and Assessment Models Including Data Provenance to Grade Data Sources. In Conference on Computer Science and Information Systems, pages. 101-118, Greece, June 2005.

[Naumann, F.] and Rolker, C. "Assessment Methods for Information Quality Criteria". InProceedings of the Conference on International Quality (IQ00) Boston, 2000.

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

[Nejdl, W.], SIBERSKI, W., & SINTEK, M. (2003). Design issues and challenges for RDF- and schema-based *peer*-to-*peer* systems. ACM SIGMOD Record, vol. 32, pp. 41 - 46.

[Batista, M. C. M.] "Schema Quality Analysis in a Data Integration System". Tese de Doutorado, Centro de Informática 2008 – UFPE.

[Sung, L. G.] A Survey of Data Management in *Peer*-to-*Peer* Systems. School of Computer Science, University of Waterloo 2005.

[ISMAIL A.], QUAFAFOU M., DURAND N., NACHOUKI G. HAJJAR M. (2010). “Queries Mining for Efficient Routing in P2P Communities”. In Proc. of Journal of Database Management Systems (IJDMS), Vol.2, No.1, February 2010

[Tatarinov, I.], Havely, A.Y. “Efficient Query Reformulation in *Peer* Data Management Systems” In SIGMOD 2004 – Paris – France.

[Heese, R.], Herschel S., Naumann F., and Roth A. (2005) “Self-extending *peer* data management”. In G. Vossen, F. Leymann, P. C. Lockemann, and W. Stucky, editors, Proceedings of the German Conference on Datenbanksysteme in Business, Technologie und Web, volume 65 of LNI. GI, March 2005.

[Roth, A.], Naumann, F., Hübner, T., Schweigert, M. (2006). System P: Query Answering in PDMS under Limited Resources. In G. Vossen, F. Leymann, P. C. Lockemann, and W. Stucky, editors, Proceedings of the German Conference on Datenbanksysteme in Business, Technologie und Web, volume 65 of LNI. GI, March 2006.

[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.

[VALDURIEZ, P.], & PACITTI, E. (2004). Data Management in Large-scale P2P Systems.Proceedings of the International Conference on High Performance Computing for Computational Science. Valencia, Spain.

[Ratnasamy, S.] et al. A scalable content-addressable network. Proc. of SIGCOMM, 2001.

[Stoica, I.] et al. Chord: A scalable *peer*-to-*peer* lookup service for internet applications. Proc. of SIGCOMM, 2001.

[Fioriano]. *Super-peer* Architectures for Distributed Computing. White Paper, Fiorano Software, Inc 2003. Disponível em http://www.fiorano.com/whitepapers/super*peer*.pdf.

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

[Silva, E. R.] (2011). Sistemas P2P e PDMS. Trabalho Individual I. Doutorado. Universidade Federal de Pernambuco. Recife, Brasil.

[HALEVY, A.], IVES, Z., SUCIU, D., & TATARINOV, I Schema mediation in *peer* data management. Proceedings of the 19th IEEE International Conference on Data Engineering, 2003a.

[Halevy, A.], Rajaraman, A. and Ordille, J. Data Integration: the Teenage Years, In Proceedings. of the 25th International Conference on Very Large Data Bases (VLDB), pages 9-16, Seoul, Korea, September 2006.

A. Halevy. Theory of Answering Queries Using Views. SIGMOD Record, vol. 29, no.4, December 2000.

VU, Q. M., LUPU, M., & OOI, B. C. (2010). *Peer*-to-*Peer* Computing - Principles and Applications. Editora Springer.

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

C. Aggarwal and P. Yu. A Survey of Uncertain Data Algorithms and Applications. IEEE Transactions on Knowledge and Data Engineering, Vol. 21, No. 5, May 2009.

F. Duchateau and Z. Bellahsene. Measuring the Quality of an Integrated Schema. In Conceptual Modeling – ER 2010, Lecture Notes in Computer Science, 2010.

F. Naumann and U. Leser. Quality-driven Integration of Heterogeneous Information Systems. In Proceedings of the 25th International Conference on Very Large Databases (VLDB' 99). pages 447-458, Edinburgh, UK, September 1999.

M. Yatskevich, F. Giunchiglia, F. McNeill, and P. Shvaiko. OpenKnowledge Deliverable 3.3: A methodology for ontology matching quality evaluation, 2006.

F. Giunchiglia and I. Zaihrayeu I. Making *peer* databases interact - a vision for an architecture supporting data coordination. In Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA), pages 18–35, Madrid (ES), 2002.

H. Zhuge, J. Liu, L. Feng, X. Sun, and C. He, "Query Routing in a *Peer*-To-*Peer* Semantic Link Network," Computational Intelligence, vol. 21, N. 2, pages 197-216, May. 2005.

K. Aberer, P. Cudre-Mauroux, and M. Hauswirth. The chatty web: emergent semantics through gossiping. In WWW03 Conference Proceedings, pages 197-206, May 2003.

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

S. Castano, A. Ferrara, S. Montanelli, and D. Zucchelli, HELIOS: a general framework for ontology-based knowledge sharing and evolution in P2P system. In Proceedings of the 14th International Workshop on Database and Expert Systems Applications, pages 597-603, 2003.

S. Montanelli and S. Castano, "Semantically routing queries in *peer*-based systems: the H-Link approach," The Knowledge Engineering Review, vol. 23, pages. 51-72, Mar. 2008.

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

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

Wang, R. Y., Kon, H., Madnick, S. Data Quality Requirements Analysis and Modeling. In Proceedings of the 9th International Conference on Data Engineering, pp. 670 – 677, 1993.

R. Wang and D. Strong, " Beyond accuracy: What data quality means to data consumers," Journal of Management Information Systems, vol.12, no. 4, pp. 5–34, 1996.

S.-A. Knight and J. Burn, " Developing a framework for assessing information quality on the world wide web," Informing Science Journal, vol. 8, pp. 160–172, 2005.

G. Shanks and B. Corbitt, " Understanding data quality: Social and cultural aspects," in 10th Australasian Conference on Information Systems, 1999, pp. 785–797.

M. Bovee, R. Srivastava, and B. Mak, " A conceptual framework and belief-function approach to assessing overall information quality," International Journal of Intelligent Systems, vol. 18, p. 5174, 2003

R. Price and G. Shanksa. Semiotic Information Quality Framework. In Proceedings of the 2004 IFIP International Conference on Decision Support Systems (DSS), pages 658-672, Prato, Italy, July 2004

L. English. Seven Deadly Misconceptions about Information Quality. DM Review Magazine, 1999. Disponível em: http://www.dmreview.com/issues/19990701/1239-1.html. Último acesso em 13 de outubro de 2011.

F. Giunchiglia and I. Zaihrayeu I. Making *peer* databases interact - a vision for an architecture supporting data coordination. In Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA), pages 18–35, Madrid (ES), 2002.

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

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, vol 25, no. 1, pp 53-63.

MANDREOLI F., MARTOGLIA R., PENZO W., SASSATELLI S, VILLANI G. (2007b) "SUNRISE : Exploring PDMS Networks with Semantic Routing Indexes" in Proc of eswc07.

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), pp 94-101, Ontario, Canada.

BENEVENTANO D., BERGAMASCHI S., GUERRA F., VINCINI M. (2007). "The SEWASIE Network of Mediator Agents for Semantic Search". Journal of Universal Computer Science, vol. 13, no. 12, pp.1936-1969.

D. Souza, C. E. Pires, Z. Kedad, P. C. A. R. Tedesco, A. C. Salgado. A Semantic-based Approach for Data Management in a P2P System. To be published in LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems, 2011.

Kantere, V., Tsoumakos, D., Sellis, T., Roussopoulos N. Grou*Peer*: Dynamic clustering of P2P databases. Information Systems 34 (2009) 62– 86, 2009.

Ng, W. S., Ooi, B. C., & Tan, K.-L. (2002). Best*Peer*: A Self-Configurable *Peer*-to-*Peer* System. In Proceedings of the 18th International Conference on Data Engineering, p. 272. San Jose, United State of America.

O'NEIL, E., O'NEIL, P., & WEIKUM, G. (1993). The LRU-K page replacement algorithm for database disk. Proceedings of the 1993 ACM Sigmod Internacional Conference on Management of Data, vol. 22, pp. 297-306.

OOI, B. C., SHU, Y., & TAN, K.-L. (2003). Relational Data Sharing in *Peer*-based Data Management Systems. ACM SIGMOD Record, vol. 32, pp. 59-64.

Codd, E. F., A relational model of data for large shared data banks, Communications of the ACM, v.13 n.6, p.377-387, June 1970

S. Russell and P. Norvig, Artificial Intelligence – A modern Approach, Prentice Hall Series in Artificial Intelligence, New Jersey, 1995.

E. Mena, V. Kashyap, A. Illarramendi, and A. P. Sheth. Imprecise answers in dis-tributed environments: Estimation of information loss for multi-ontology based queryprocessing.Intl. Journal of Cooperative Information Systems, 9(4):403–425, 2000.

HALEVY, A., IVES, Z., MONK, P., & TATARINOV, I. (2003b). Piazza: data management infrastructure for semantic web applications. Proceedings of the 12th World Wide Web Conference (WWW), pp. 556-567. Budapest, Hungary.

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

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

Freire, C. A. 2011. Roteamento de Consultas em PDMS. Ph.D. TI2, Federal University of Pernambuco (UFPE/CIn). Recife, PE, Brazil. Disponível em http://www.cin.ufpe.br/~speed/Trabalho%20Individual-Roteamento-crishane.pdf

D. Barkai. An Introduction to *Peer*-to-*Peer* Computing. Intel Developer Update Magazine, pages 1–7, February 2000.

Nurse, J.R.C., Rahman, S.S., Creese, S., Goldsmith, M. and Lamberts, K. (2011) 'Information Quality and Trustworthiness: A Topical State-of-the-Art Review', *The International Conference on Computer Applications and Network Security (ICCANS) 2011,* 492 - 500

[RDF, W3C] http://www.w3.org/RDF/ acessado em 17/09/2011.

[JXTA, Java.net] http://java.net/projects/jxta acessado em 17/09/2011;