Revisitando Técnicas de Bancos de Dados no contexto da Web

Fernanda Lima
e-mail: ferlima@inf.puc-rio.br

Marco Antonio Casanova
e-mail: casanova@inf.puc-rio.br

Rubens Nascimento Melo
e-mail: rubens@inf.puc-rio.br

PUC-RioInf.MCC18/99 August, 1999

Abstract

Database integration has been a research topic for over a decade. Although there are several proposals in the literature, the subject reemerges due to its frequent necessity. Companies have systems (legacy or not) that need to be integrated, and users often need to query disperse data sources. These examples are two common uses of database integration, still up do date. Many of the techniques already researched are still valid, however, nowadays it is necessary to consider a new environment that is becoming widespread in companies and users' realities, the World-Wide-Web (WWW). In this new context, it is necessary to reevaluate the existing techniques. This work is a partial contribution to this challenge, while revisiting database techniques in the Web context.

Keywords: Databases, WWW, Web Mediators, Web Data Models, Web Query Languages.

Resumo

A integração de bancos de dados é tema de pesquisa há mais de uma década. Apesar de existirem diversas propostas, o tópico é frequentemente retomado devido a sua necessidade sempre atual. Empresas possuem sistemas (legados ou não) que precisam ser integrados, e usuários necessitam realizar consultas em bases dispersas a todo momento. Estes são apenas dois exemplos de integração de banco de dados, ainda muito atuais. Muitas das técnicas propostas continuam válidas, entretanto, atualmente é necessário considerar um novo ambiente cada vez mais presente na realidade de empresas e usuários, a Web. Neste novo contexto, é preciso reavaliar as técnicas existentes. Este trabalho visa preencher atender parcialmente este desafio, ao revisitar técnicas de bancos de dados no contexto da Web.

Palavras-chave: Bancos de Dados, WWW, Mediadores para a Web, Modelos de Dados para a Web, Linguagens de Consulta para a Web.

1. Introdução

A utilização da Web tornou possível o acesso a uma vasta quantidade de informação disponível em formato digital. Atualmente, esta informação encontra-se armazenada em fontes de dados diversificadas, tais como Sistemas de Bancos de Dados Legados, páginas WWW e arquivos convencionais.

Apesar da existência de ferramentas de busca na Web, o acesso a esta informação ainda é inadequado, pois o suporte oferecido para pesquisas a documentos ainda é primitivo. Um exemplo da deficiência das ferramentas é a ausência de uso da estrutura dos dados.

Tradicionalmente, pesquisas na área de Bancos de Dados (BD) tratam de acesso otimizado a um grande volume de dados. E é natural imaginar que a comunidade de BD pode contribuir para a solução do problema de acesso a dados da Web. É importante ressaltar, no entanto, que as tecnologias de BD enfrentam novos desafios no contexto da Web, tornando necessária uma reavaliação das soluções existentes no contexto de BDs.

Em [FLORESCU+98], Florescu, Levy e Mendelzon realizam um importante levantamento das contribuições atuais da comunidade de Banco de Dados para o contexto de gerenciamento de informações na Web. Pesquisas são agrupadas em três grandes temas: Extração e Integração de Informação; Modelos de Dados e Linguagens de Consultas para a Web; e Construção e Reestruturação de Web Sites.

Este trabalho complementa os dois primeiros temas apresentados no artigo mencionado, apresentando exemplos de pesquisas e projetos, e também uma análise do impacto de XML. O principal objetivo deste trabalho é fornecer uma visão atualizada do cenário das pesquisas de Bancos de Dados aplicadas no contexto da Web.

O restante deste trabalho está organizado da seguinte forma. Na seção 2 são enumeradas pesquisas tradicionais de Bancos de Dados, seus novos desafios no contexto da Web, bem como definições conceituais de XML, utilizadas no restante do trabalho. A seção 3 apresenta o tema de Extração e Integração de Informações, com exemplos de pesquisas recentes. A seção 4 trata do tema Modelos de Dados e Linguagens de Consultas para a Web. E finalmente, na seção 5 são apresentadas as conclusões deste estudo.

2. Pesquisas Relacionadas

Não é incomum encontrar na literatura autores comentando que a Web pode ser vista como um grande Banco de Dados (BD). Realmente, a essência de diversos trabalhos da área de BD pode ser resumida ao problema do gerenciamento de grandes volumes de dados. E justamente na Web, os dados disponíveis alcançam grandes proporções, tornando as dificuldades aparentemente semelhantes.

No entanto, não se pode desprezar o fato de que a Web traz novos fatores que precisam ser considerados. Nesta seção, temas tradicionais de Bancos de Dados são comentados e associados a temas atuais no contexto da Web. A recente linguagem extensível XML é também apresentada pois será referenciada ao longo deste trabalho.

2.1 Temas de Pesquisa Tradicionais no contexto de Banco de Dados

Dois grandes temas de pesquisas tradicionais de Bancos de Dados podem ser destacados como temas relacionados a este trabalho: (i) Integração de Sistemas Gerenciadores de Bases de Dados (SGBDs) e (ii) Modelos de Dados e Processamento de Consultas.

O primeiro tema diz respeito à integração de dados através do uso de SGBDs Heterogêneos. Diferentes propostas foram elaboradas e não existe um consenso quanto a solução ideal, e nem mesmo existe consenso quanto ao significado de termos como Multi-SGBDs, SGBDs Federados com Acoplamento Fraco ou Forte.

Para esta monografia é importante apenas comentar que as propostas existentes consideram (ou não) a existência de um esquema global, que pode ser único ou parcial. As soluções que consideram o uso de esquema global, subdividem suas tarefas em: (a) tradução do esquema local para um modelo canônico e (b) integração de esquemas. Esta tarefa de integração é subdividida em homogeneização para eliminar heterogeneidades (semântica e estrutural) e integração com métodos binários e n-ários. Vale comentar que o modelo orientado a objetos (OO) foi muito utilizado como modelo canônico.

O segundo tema tradicional de Bancos de Dados está relacionado a Modelos de Dados e Processamento de Consultas para Sistemas Gerenciadores de Bases de Dados Distribuídos. O modelo relacional foi amplamente utilizado e o modelo OO ainda é alvo de pesquisas atuais. Diversas técnicas de projeto de Bancos de Dados Distribuídos foram estudadas e também técnicas de processamento e otimização de consultas.

Mais especificamente em relação a modelo de dados, vale comentar que os SGBDs atuais gerenciam dados estruturados, onde um esquema fixo (a estrutura) é previamente definido. Todos os dados gerenciados por este SGBDs devem obedecer aos esquemas existentes. Um esquema é útil para um SGBD pois permite armazenar dados e indexá-los, bem como processar consultas e atualizações. Usuários também utilizam esquemas para formular consultas e atualizações.

2.2 Os Desafios no contexto da Web

No contexto da Web surgem novas questões não abordadas tradicionalmente por pesquisas de Bancos de Dados. Pesquisas de BD costumam envolver situações estáveis onde assume-se uma quantidade fixa de fontes de dados a serem integradas. Quando esta quantidade não é fixa, é considerada de alteração pouco freqüente. As fontes de dados são geralmente SGBDs com capacidade total para processar consultar e assume-se que os mesmos estarão disponíveis para acesso integrado.

O aspecto dinâmico da Web não permite revalidar estas premissas básicas. Para enfrentar estes novos desafios relacionados à integração de dados, muitos projetos de pesquisa vêm utilizando a proposta de arquitetura de mediadores, a ser apresentada na seção 3.

Quanto aos desafios da Web para modelos de dados e linguagens de consulta, pode-se destacar o fato de que, atualmente, muitas informações não estão disponíveis em formato totalmente estruturado. Como exemplos pode-se citar dados extraídos da Web ou dados integrados/intercambiados em ambientes heterogêneos. Esta categoria de dados é classificada como dados semi-estruturados, que podem possuir as seguintes características básicas: ausência de

estrutura regular, ou estrutura que evolui de forma imprevisível; dados que podem ser incompletos; usuários e SGBDs que desconhecem sua estrutura completa.

Propostas de solução para este tema ainda são recentes e existem muitos tópicos ainda não resolvidos. A seção 4 aborda o assunto com maiores detalhes.

2.3 A importância de XML

Desde que o Consórcio da World-Wide-Web, conhecido como W3C [W3C], aprovou a versão 1.0 de XML [W3C-XML98] com o status máximo de recomendação, uma vasta quantidade de pesquisadores desviou seu rumo de trabalho para dar atenção ao novo padrão. Pesquisas ainda estão em fases iniciais e produtos preliminares já começam a ser lançados.

Segundo Bosak e Bray [BOSAK+99], a intensa reação causada pela padronização de XML é conseqüência da esperança de que XML irá resolver um dos maiores problemas da Web atual: a dificuldade de encontrar o item de informação procurado.

Esta dificuldade surge, em parte, devido à natureza da linguagem principal da Web, a linguagem HTML (HyperText Markup Language). Apesar de seu incontestável sucesso como linguagem de publicação eletrônica, HTML trata a informação de forma superficial, descrevendo basicamente como um browser deve apresentar textos ou imagens. O conteúdo de informação presente nas páginas não recebe nenhum tratamento que permita extrair alguma semântica.

A solução proposta por XML é simples: separar o conteúdo e não tratar apresentação, utilizando tags que representam significado e não aparência. Conceitos de metalinguagem de marcação definidos para SGML (Standard Generalized Markup Language) foram reutilizados de forma simplificada, evitando a complexidade que inviabilizou o uso extensivo desta outra excelente proposta.

2.3.1 Definição de XML

A linguagem de marcação extensível XML (eXtensible Markup Language) [W3C-XML98] é um formato de dados para intercâmbio de documentos estruturados na Web. Segundo os relatórios de atividades do consórcio W3C [W3C-XML-Activity], a linguagem XML não é apenas um formato de texto simples, flexível, baseado em SGML e projetado para enfrentar os desafios da publicação eletrônica em larga escala, XML também representará um papel cada vez mais importante na troca de dados na Web.

Basicamente, XML descreve uma classe de objetos de dados chamada documentos XML, e permite descrever parcialmente o comportamento de programas que processam estes objetos. A meta de XML é permitir que SGML genéricos possam ser enviados, recebidos, e processados na Web do mesmo modo atualmente possível com HTML. A linguagem XML foi projetada para interoperar com SGML e HTML.

Ao definir regras para permitir que novas tags sejam utilizadas na linguagem de marcação, XML permite que o conteúdo seja utilizado tanto por seres humanos, quanto por ferramentas capazes de extrair semântica dos dados. Um conjunto de regras define, entre outras características, que: tags são utilizados em pares (na maioria dos casos), aninhamento de tags é permitido e caracteres Unicode são considerados padrão de uso.

XML é adequada para dados semi-estruturados, pois o padrão não impõe restrições nas tags ou nos relacionamentos de aninhamento. Como o uso de "esquemas" não é obrigatório, um dado XML é auto-descritivo, pois a estrutura está misturada com o dado propriamente dito. Para dados que evoluem rapidamente (como dados da Web), XML permite mudanças freqüentes sem atualização de esquema associado. Já para dados mais estáveis, XML oferece o suporte opcional a DTDs (Document Type Definition), restringindo tags e aninhamentos.

Para o contexto de Bancos de Dados, é importante destacar a característica auto-descritiva de XML. O uso de DTDs permite que a estrutura de um documento seja verificada contra a definição da DTD. Entretanto, como o uso de DTDs é opcional em XML, pode ser necessário extrair a estrutura de um documento a posteriori. Em Bancos de Dados, a estrutura é o próprio esquema.

Pode-se observar dois importantes usos de XML, neste contexto. O primeiro refere-se ao fato de que esquemas de Bancos de Dados podem ser expressos em XML e compartilhados na Web. Dados relevantes a estes esquemas podem, por exemplo, povoar bases de dados, após verificação de compatibilidade estrutural. A utilização de um esquema previamente definido através de uma DTD é importante pois permite otimizações em consultas.

Outra possibilidade de uso de XML é a extração de um esquema a partir de documentos existentes. Desta forma, é possível explorar a flexibilidade oferecida pela não obrigatoriedade do uso de DTDs.

2.3.2 Comparando HTML e XML

As Figuras 1 e 3 apresentam um mesmo exemplo em formato HTML e XML, respectivamente. O exemplo em questão contém dados de duas referências bibliográficas.

```
<UL>
<LI> R. Goldman, J. McHugh, and J. Widom.
<A href="ftp://db.stanford.edu/pub/papers/xml.ps"> From Semistructured Data to XML: Migrating the Lore Data
    Model and Query Language </A> Proceedings of the 2nd International Workshop on the Web and
    Databases (WebDB '99), pages 25-30, Philadelphia, Pennsylvania, June 1999.
<LI> T. Lahiri, S. Abiteboul, and J. Widom.
<A href="ftp://db.stanford.edu/pub/papers/ozone.ps"> Ozone: Integrating Structured and Semistructured Data
    </A>Technical Report, Stanford Database Group, October 1998.
</UL>
```

Figura 1 - Exemplo de Referência Bibliográfica em HTML [WIDOM99]

O exemplo apresentado na Figura 1 corresponde ao trecho de página HTML que pode ser “renderizado” em um browser e apresentado conforme a Figura 2. Os mesmos dados constam no exemplo em XML da Figura 3. Entretanto como XML é independente da forma de apresentação dos dados, não incluiremos figuras associadas. Algumas observações importantes podem ser obtidas a partir deste exemplo muito simples: o código XML é mais extenso que o código HTML; entretanto, de um ponto de vista de gerenciamento de dados, XML fornece informação em formato mais conveniente e utilizável [WIDOM99].

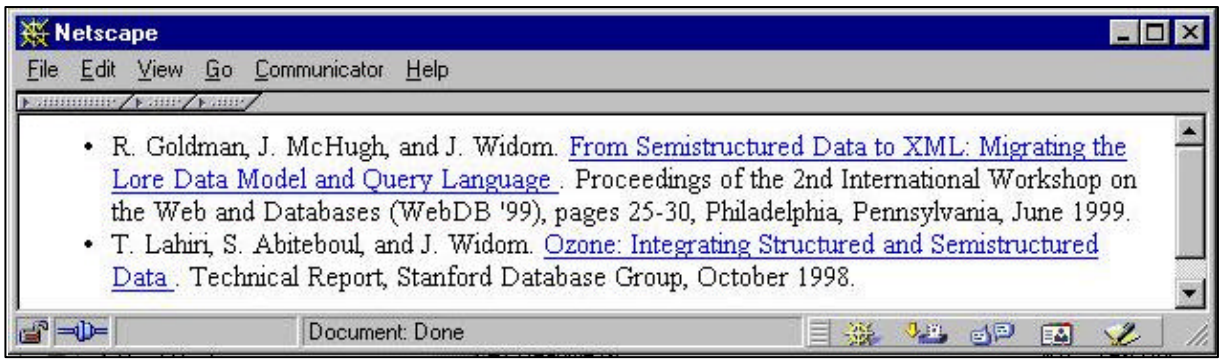

Figura 2 - Apresentação do Exemplo anterior em um browser

Continuando a comparação dos dois documentos apresentados nas Figuras 1 e 2, pode-se confirmar que o documento HTML descreve a apresentação junto com o conteúdo, enquanto que o documento XML descreve apenas o conteúdo. Com o uso de XML, o usuário pode definir novas tags e aninhamentos arbitrários. Além disto, XML permite definir validação, conforme comentado na seção anterior. As tags XML escolhidas neste exemplo possuem semântica relevante, e podem ser utilizadas no processamento de consultas.

```
<Publication URL="ftp://db.stanford.edu/pub/papers/xml.ps" Authors="RG JM JW">
    <Title>From Semistructured Data to XML: Migrating the Lore Data Model and Query Language</Title>
    <Published>Proceedings of the 2nd Int. Workshop on the Web and Databases (WebDB '99)</Published>
    <Pages>25-30</Pages>
    <Location>
            <City>Philadelphia</City>
            <State>Pennsylvania</State>
    </Location>
    <Date>
            <Month>June</Month>
            <Year>1999</Year>
    </Date>
</Publication>
<Publication URL="ftp://db.stanford.edu/pub/papers/ozone.ps" Authors="TL SA JW">
    <Title>Ozone: Integrating Structured and Semistructured Data</Title>
    <Published>Technical Report</Published>
    <Institution>Stanford University Database Group</Institution>
    <Date>
            <Month>October</Month>
            <Year>1998</Year>
    </Date>
</Publication>
<Author ID="SA">S. Abiteboul</Author>
<Author ID="RG">R. Goldman</Author>
<Author ID="TL">T. Lahiri</Author>
<Author ID="JM">J. McHugh</Author>
<Author ID="JW">J. Widom</Author>
```

Figura 3 - O mesmo Exemplo de Referência Bibliográfica em XML [WIDOM99]

2.3.3 Indefinições sobre XML

Em [WIDOM99], Widom comenta que, apesar de ser previsível o fato de que XML terá grande impacto no gerenciamento de informações da Web, ainda não está claro precisamente como XML será usado. Pode-se destacar diversas possibilidades de uso como: formato de troca de dados, formato de armazenamento de dados, e formato de definição de DTDs.

Entretanto, não se pode ignorar o fato de que XML não resolverá todos os problemas. Existirão aplicações que utilizarão XML, mas por questões internas não tornarão seus dados disponíveis para serem acessados por ferramentas de integração.

Pesquisas sobre ontologias podem auxiliar o aspecto de integração de dados. Em [WIEDERHOLD94], pode-se encontrar uma interessante proposta de ontologia para diminuir as heterogeneidades, que tanto dificultam a integração de dados. Porém, não se pode afirmar que esta proposta conseguiu alcançar um amplo consenso.

Com o surgimento de XML, o esforço de criação de ontologias foi parcialmente simplificado, pois não é mais necessário discordar sobre formatos de representação de dados. Resta ainda a importante (e grande) tarefa de definir vocabulários específicos para domínios. Este é justamente o papel das DTDs de domínio específico, tópico de pauta de grupos do consórcio W3C [W3C-XML-Activity].

3. Extração e Integração de Informações

Diversas pesquisas foram realizadas com o intuito de permitir a geração de um esquema global único para acessar SGBDs heterogêneos. No entanto, novos fatores surgiram nos tempos atuais. Com o uso da Web, o escopo dos SGBDs heterogêneos cresceu imensamente. Agora, os SGBDs heterogêneos não pertencem apenas a empresas residindo localmente, existem SGBDs remotos e é necessário realizar consultas em sistemas legados, páginas HTML residindo em servidores remotos, arquivos convencionais. Ou seja, os dados podem ser estruturados, semi-estruturados ou até mesmo não estrurados.

Neste contexto, a proposta apresentada por Wiederhold em [WIEDERHOLD92, WIEDERHOLD97] apresenta uma possível solução para estas dificuldades. A idéia é criar um camada intermediária entre as aplicações e os bancos de dados. Esta camada composta por mediadores tem como objetivo simplificar, abstrair, reduzir, reunir dados e torná-los compreensíveis. Os mediadores são módulos de *software* que ocupam uma camada explícita e ativa em uma arquitetura de compartilhamento e permitem que as aplicações dos usuários sejam independentes dos recursos de dados. Estes módulos exploram o conhecimento codificado sobre subconjuntos de dados criando informação para uma níveis mais altos de aplicação

Na arquitetura de sistemas com mediadores, usuários finais interagem com aplicações escritas por programadores. Aplicações acessam uma representação uniforme das fontes de dados através de linguagens de consulta declarativa. Os mediadores encapsulam a representação das múltiplas fontes de dados para esta linguagem de consulta, fornecendo acesso uniforme. Cabe a estes mediadores resolver conflitos envolvendo representação de conhecimento diferentes como modelos de dados e esquemas, e conflitos devido a diferenças no poder de processamento de cada fontes de dados.

3.1 Novos requisitos para Integração

A Figura 4 descreve a arquitetura inicialmente proposta por Wiederhold [WIEDERHOLD92], onde usuários finais interagem com aplicações escritas por programadores e aplicações acessam uma representação uniforme das fontes de dados através de linguagens de consulta declarativa. Os mediadores encapsulam a representação das múltiplas fontes de dados para esta linguagem de consulta, fornecendo acesso uniforme. Cabe a estes mediadores resolver conflitos envolvendo representação de conhecimento diferentes como modelos de dados e esquemas, e conflitos devido a diferenças no poder de processamento de cada fontes de dados.

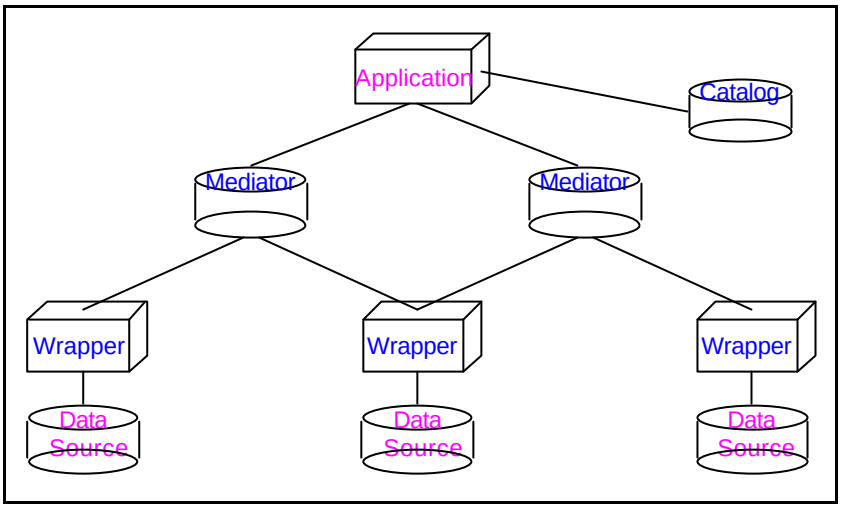

Figura 4 - Arquitetura de Mediadores [WIEDERHOLD92]

Para que múltiplas fontes de dados possam ser acessadas de maneira uniforme, o mediador aceita uma consulta, transforma-as em sub-consultas que são distribuídas pelas fontes de dados. Quando as sub-respostas retornam, o mediador as combina gerando resposta final para a aplicação. Esta arquitetura permite que os mediadores sejam desenvolvidos de forma independente e possam ser combinados, fornecendo um mecanismo para lidar com a complexidade introduzida pelo crescente número de fontes de dados.

Para lidar com a heterogeneidade das fontes de dados, *wrappers* [1] fornecem uma visão estruturada da fonte de dados e transforma sub-consultas do mediador na linguagem particular da fonte de dados. O *wrapper* possui as funcionalidades de transformar consultas para uma fonte de dados em particular e reformatar as respostas apropriadas para cada mediador. Eles contêm informações sobre a estrutura do objeto desejado para integração e seu mapeamento para a fonte de informação. É possível ter vários *wrapper*s para um fonte de dados, se múltiplos objetivos devem ser atendidos. É necessário que o implementador de bancos de dados escreva *wrappers* para cada tipo de fonte de dados.

3.2 Pesquisas sobre Extração e Integração de Informações

Uma grande quantidade de pesquisas e projetos foram desenvolvidos utilizando a arquitetura de mediadores. De uma forma geral, pode-se destacar que alguns dos principais problemas

[1] O termo *wrapper* pode ser utilizado em português como adaptador ou tradutor.

abordados são: quantidade de fontes de dados, autonomia de cada uma das fontes envolvidas e metadados.

O próprio artigo {FLORESCU+98] destaca alguns dos principais aspectos estudados: Especificação do esquema no mediador usando conceitos como: GAV (Global as View) x LAV (Local as View); completitude das Fonte de dados (resposta negativa); diferença no poder de procesamento de consultas(operações); otimização de consultas (custo); processadores de consulta (atraso), construção de Wrappers ("templates"); identidade de objetos em diferentes fontes.

3.3 Um Exemplo de Extração e Integração de Informações

O Projeto DISCO utiliza a arquitetura apresentada na seção 3.1 e define a interação entre mediadores e wrappers em duas fases: o registro e o processamento de consultas. Na primeira fase, a chamada fase de registro, o mediador registra diversos *wrapper*s através de chamadas (*register_call*). Cada *wrapper* informa seu esquema local, as capacidades de processamento de consultas e qualquer informação de custo existente. O administrador de mediadores define o esquema global e visões para conectar o esquema global com os locais, pois as aplicações são escritas com relação a esquemas globais.

Na segunda fase ocorre o processamento de consultas propriamente dito. Aplicação envia consulta ao mediador, que a transforma em um plano de sub-consultas e uma consulta "composição". Este plano é otimizado de acordo com os custos importados pelos *wrappers* e respeita a capacidade destes. O mediador executa o plano enviando as sub-consultas aos *wrappers*, que as processam e retornam sub-respostas. O mediador combina as sub-respostas usando a consulta composição e retorna o resultado para a aplicação. Se algum *wrapper* não está disponível, o mediador retorna a resposta parcial para a aplicação.

3.3.1 O Modelo de Dados e A Linguagem de Consultas do Projeto DISCO

O modelo de dados DISCO é baseado na especificação ODMG-93 [CATTELL+96], composta de linguagem de definição de objetos (ODL), linguagem de consulta (OQL) e uma linguagem para *binding*. O modelo DISCO estende a ODL em dois aspectos:

- *extents* associam múltiplos extents com cada tipo de interface definida para o mediador;
- *type mapping* associa informações de mapeamento de tipos e o tipo associado a uma determinada fonte de dados.

No modelo de dados, interface define a assinatura do tipo para acessar um objeto. Uma extensão associada a uma interface armazena objetos de forma persistente.

Para exemplificar o funcionamento da arquitetura DISCO, pode-se considerar um sistema contendo duas fontes de dados r0 e r1. Suponha que r0 contém uma relação de pessoa chamada Mary com salário = 200, e r1 contém uma relação de pessoa chamada Sam com salário = 50. O mediador modela r0 e r1 em extensões person0 e person1, do tipo Person, conforme Figura 5.

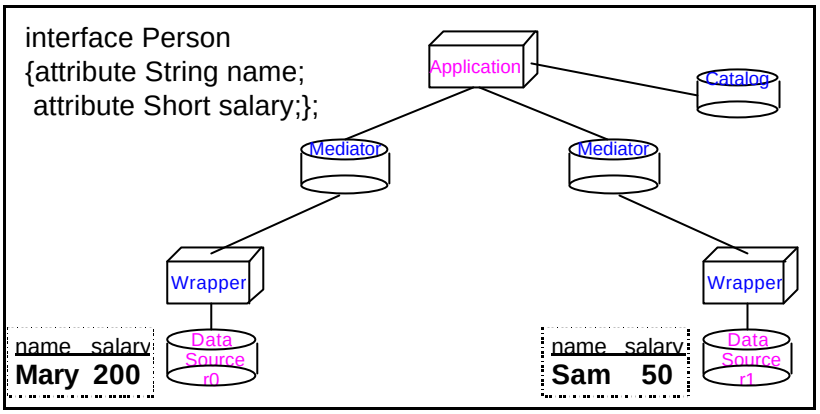

Figura 5- Exemplo de consulta do Projeto DISCO

Para acessar as fontes de dados, o usuário executa uma consulta na linguagem de consulta DISCO:

```
select x.name from x in person
where x.salary > 10;
```

A consulta acima constrói um conjunto de nomes de pessoas de r0 que possuem salário maior que alguém em r1. A resposta é um conjunto de *strings* Bag((“Mary”, “Sam”)).

3.3.1.1 Passos para definição de acesso a fonte de dados no DISCO

Para que as fontes de dados possam ser acessadas de forma transparente, é necessário realizar diversos passos, descritos em detalhes por TOMASIC, RASCHID e VALDURIEZ em [TOMASIC+98]. A seguir, um exemplo é reproduzido sem as instruções associadas, apenas com o intuito de fornecer uma visão geral. O leitor interessado em detalhes deve procurar [TOMASIC+98, TOMASIC+95].

Inicialmente, é preciso modelar cada fonte de dados associando endereços a repositórios. Em seguida, é necessário que o Administrados de Banco de Dados (ABD) localize o *wrapper* correspondente para a cada tipo de fonte de dados, defina o tipos de dados no mediador que correspondem aos tipos nas fontes de dados e, finalmente, especifique a extensão associada ao mediador que acessa o repositório através do *wrapper*.

É importante comentar que cada extensão DISCO representa uma coleção de dados em uma fonte de dados. Desta forma, para incluir a outra fonte de dados é necessário repetir o último passo, permitindo que as duas bases sejam acessadas. A nova consulta poderia ser feita explicitamente através do comando:

```
select x.name from x in union (person0, person1) where x.salary > 10;
```

retornando Bag (Mary, Sam)

Entretanto, pode-se elaborar consultas sem explicitar as extensões, o que é mais conveniente. Para isto, deve ser utilizado um tipo de meta-dado chamado MetaExtent. Com isto, extensões para tipos do mediador podem ser adicionadas ou removidas ao adicionar ou remover objetos do tipo MetaExtent.

Usando o meta-dado, DISCO pode fornecer uma referência implícita para todos as extensões associadas ao tipo do mediador, ao declarar uma extensão na definição da interface.Esta definição de interface assume implicitamente uma expressão de definição de consulta para a extensão person.

Após estes passos mencionados, as consultas podem acessar dinamicamente todas as extensões definidas para o tipo Person.

Vale ressaltar que estas fontes de dados possuem estruturas similares. Entretanto, pode ser necessário integrar múltiplas fontes de dados com estruturas diferentes, assunto mencionado a seguir.

3.3.1.2 Integração dos dados no DISCO

O Projeto DISCO não busca integração das fontes de dados através da obtenção de um tipo único [TOMASIC+98]. São fornecidos mapeamentos de estruturas diferenciados para cada caso, a saber:

- sub-tipos são usados para sub-estruturas similares,
- mapeamentos são realizados quando tipos de dados DISCO são diferentes dos tipos das fontes de dados,
- visões são utilizadas para estruturas diferentes.

Além disto, transformações arbitrárias na representação das fontes de dados são permitidas.

3.3.1.3 Processamento de Consultas no Mediador

O mediador contém: otimizador de consultas, sistema *run-time* e Banco de Dados interno com informação de fonte de dados, tipos e visões. O objetivo do otimizador é buscar a melhor forma de executar a consulta no sistema run-time. Para isto, o otimizador transforma a consulta em expressões lógicas alternativas que podem ser executadas pelo mediador ou pelo *wrapper*. Cada expressão possui um custo estimado associado e a expressão de menor custo será executada pelo sistema *run-time.*

A Figura 6 apresenta o processamento de consultas no Mediador DISCO.

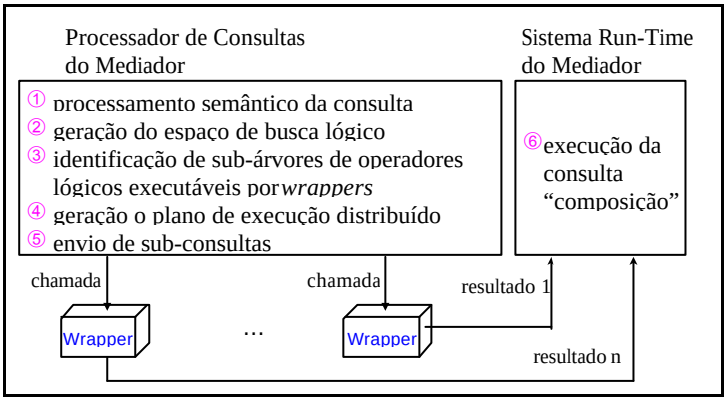

Figura 6 - Processamento de Consultas no DISCO [KAPITSKAIA+97]

O processamento de consultas no mediador DISCO pode ser descrito da seguinte forma:

O passo 1 corresponde ao processamento semântico da consulta realizado em otimizadores tradicionais. Aqui acontece a fase de parse da consulta e através de visões ocorre a reformulação da consulta no esquema local gerando sub-consultas. Além destas, será gerada a consulta composição a ser executada quando retornarem as respostas das sub-consultas. Neste primeiro passo, o processamento de consultas ainda ignora as funcionalidades dos *wrapper*s, de modo que as sub-consultas geradas assumem que todos os operadores lógicos podem ser suportados por todos os *wrapper*s.

No segundo passo, ocorre a geração do espaço de busca lógico. A transformação das consultas gera árvore de operadores lógicos preliminar. Para cada *wrapper* há uma sub-árvore que corresponde a um plano preliminar, o mesmo ocorrendo para o plano preliminar da consulta composição. Estes planos são preliminares pois ainda neste passo, as funcionalidades dos *wrapper*s são desconhecidas.

Somente no terceiro passo que o processamento de consultas começa a considerar as funcionalidades do *wrapper* . Baseado neste conhecimento, são identificadas as sub-árvores executáveis por cada *wrapper*. Este passo é descrito em detalhes em [KAPITSKAIA+97].

O passo quatro realiza a geração do plano de execução distribuído, através da transformação de operadores lógicos em físicos (index-scan, hash-join,). Modelo de custos próprios são utilizados para escolher a árvore de menor custo. Em [KAPITSKAIA+97] o problema de custos não é tratado, porém em [NAACKE+98], os autores descrevem um método para otimização de consultas baseado em custos.

O quinto passo corresponde ao envio das sub-consultas a cada *wrapper*, enquanto que o sexto passo representa a reunião dos resultados das sub-consultas através da execução da consulta composição.

3.3.1.3.1 Definição de Funcionalidade e Custo

Para processar as consultas de forma eficiente, o mediador deve otimizar o plano de consultas. Em Bancos de Dados clássicos, a otimização de consultas baseada em custos é um método eficaz, porém em SBDs distribuídos e heterogêneos, otimização de consultas baseada em custos é difícil de ser alcançada pois as fontes de dados não exportam informação de custo. Em [NAACKE+97], os autores descrevem um novo método que permite ao programador do *wrapper* exportar estimativas de custo (fórmulas de custo estimadas e estatísticas). Ao invés de esperar todas as estimativas de custo, o componente DISCO facilita a responsabilidade do implementador de *wrapper* aprimorando o modelo de custo genérico do mediador com estimativas de custo específicas vindas dos *wrapper*s. Para isto, uma linguagem de especificação de estimativas de custo é definida, e também um algoritmo para mesclar as estimativas de custo durante a otimização da consulta. Testes foram realizados com base na combinação de fórmulas analíticas e medidas reais em uma base de objetos no ObjectStore com 007.

O mediador usa uma chamada *register_call* para obter um esquema local e receber as funcionalidades de cada fonte de dados.As informações de custo são definidas pelo IBD, de modo que para um determinado *wrapper* determinadas expressões lógicas serão possíveis.

O mediador irá incorporar estas informações automaticamente no processo de transformação da consulta. As informações de custo geradas no *wrapper* irão sobrepor as informações genéricas que o mediador tinha, através do mecanismo de orientação a objetos conhecido como anulação.

3.3.1.3.2 Obtenção do Plano de Consultas

Todas as chamadas ao *wrapper* são feitas com o operador lógico submit (source, expression), que significa que a expressão lógica de *expression* está localizada na fonte de dados declarada como *source*.

Quando o otimizador traduz consulta OQL em expressão lógica, as referências a extensões são traduzidas no operador submit. O otimizador gera um operador submit para cada referência a uma extensão. Esta tradução é dividida em tantas partes quanto for a quantidade de extensões. É importante comentar que, desta forma, a projeção de atributos serão realizadaa pelo sistema *run-time* do mediador, o que causará sobrecarga.

Regras de transformação podem re-escrever expressões lógicas contendo o operador lógico submit. Por exemplo, uma regra de transformação válida é levar a projeção para o argumento do submit, fazendo com que a projeção seja realizada na própria fonte de dados. Existem restrições a estas regras de transformações baseadas na álgebra e outras impostas pela funcionalidade do *wrapper*.

O componente de busca DISCO precisa consultar a interface do *wrapper* para determinar as regras de transformação aplicáveis para o operador submit. Esta consulta é feita através de um método que retorna o sub-conjunto da gramática (da Universal Abstract Machine) que cada *wrapper* é capaz de suportar.

3.4 A importância de XML para Extração e Integração de Informações

Pode-se destacar diversos benefícios que o uso de XML pode trazer para o tema de Extração e Integração de Informações. Primeiramente, XML pode ser utilizada como modelo canônico. Com esta escolha, diminui-se a necessidade de construção de wrappers, uma vez que fontes de dados estruturados(SGBDs) e semi-estruturados (páginas HTML) podem exportar seus dados diretamente no formato XML. Isto ja é uma realidade em SGBDs atuais que oferecem suporte a XML e em páginas da Web escritas em XML.

Atualmente, devido as limitações de HTML, a construção de wrappers sobre páginas HTML necessita de intervenção manual. A tendência é que as fontes de dados passem a ser auto-descritivas, pelo menos em relação a seus dados.

Em [PRASAD+99], os autores estimam que a linguagem XML será útil também para descrição das funcionalidades das fontes de dados[2]. Como exemplo, pode-se imaginar um site de uma livraria descrevendo sua capacidade de realizar consultas sobre autores e títulos, mas não permitindo consultas que enumerem todos os livros da loja. Uma segunda geração de ferramentas

[2] Esta estimativa foi posta em prática conforme mencionado no projeto Disco, porém ainda existem muitos aspectos em aberto.

de busca deve ser capaz de explorar além das tradicionais buscas por palavras-chave, também buscas baseadas em atributos específicos de um determinado domínio.

Em [LEVY99], Alon Levy comenta não só aspectos de integração de dados, mas também a necessidade de novas formas de mensurar desempenho através de novas medidas. A quantidade crescente de fontes de dados XML é fator relevante para o desempenho, bem como o grau de irregularidade dos dados fornecidos por estas fontes.

4. Modelos de Dados e Linguagens de Consultas para a Web

O paradigma de Bancos de Dados tradicionais define, inicialmente, a criação de um esquema para descrever a estrutura de uma base de dados, e posteriormente, o povoamento desta base através da interface fornecida pelo esquema. Os Sistemas Gerenciadores de Bases de Dados (SGBDs) se encarregam de realizar o mapeamento da entrada de dados para as estruturas de armazenamento.

Conforme Silberschatz e Zdonik [SILBERSCHATZ+97], cada vez mais é impossível contar com um esquema definido a priori. Atualmente, diversas aplicações criam dados independentes de bancos de dados. E para podermos utilizar estes dados em SGBDs é necessário utilizar uma abordagem bottom-up para povoamento não usualmente suportada por sistemas atuais.

Esta seção comenta a situação atual de propostas recentes para solucionar os problemas mencionados, e também aponta novas direções para modelos de dados e linguagens de consulta para a Web, diante do surgimento de XML.

4.1 Novos requisitos para Modelos e Linguagens

Recentes pesquisas em dados semi-estruturados [ABITEBOUL97, BUNEMAN97] trouxeram importantes contribuições para questões inerentes a Web: como representar os dados armazenados e como consultá-los. No entanto, com o surgimento de XML, até mesmo estes novos projetos estão buscando adaptações para este novo contexto.

De maneira informal, pode-se definir dados semi-estruturados como quaisquer dados que não se encontrem nos dois extremos mencionados a seguir: dados estruturados (como aqueles armazenados em SGBDs) ou dados não estruturados (como arquivos convencionais ou mesmo imagens). Alguns exemplos de dados semi-estruturados são: páginas HTML, Web-sites inteiros e arquivos de referências bibliográficas BibTex.

Dados semi-estruturados diferem de dados estruturados em muitos aspectos, fazendo com que modelos convencionais tornem-se inadequados para resolver as novas questões da Web. Primeiramente, o conhecido conceito de esquema de Bancos de Dados foi alterado. Agora um "esquema" de dados semi-estruturados não é mais prescritivo e sim apenas descritivo, pois não é mais obrigatório que seja fornecido antecipadamente. Além disto, sua estrutura pode ser parcial, permitindo irregularidados como dados ausentes. Um esquema deste tipo pode também mudar com muita frequência e ser muito grande em relação ao tamanho do dado.

Além de todas estas diferenças citadas, também é preciso considerar que os tipos de dados também mudaram, uma vez que objetos e atributos, neste contexto da Web, não são mais

fortemente tipados. É possível que uma mesma coleção de dados possua diferentes representações para o mesmo tipo de objeto.

Portanto, não podemos tratar dados semi-estruturados da forma tradicional [SUCIU97]. Torna-se necessário desenvolver linguagens de consulta específicas, técnicas de avaliação e decomposição de consultas, bem como novos métodos para atualização, extração de estruturas, etc.

Devido a esta constatação, diversos grupos de pesquisa iniciaram projetos com propostas alternativas, comentadas na próxima seção. Vale comentar a existência de dois tutorias [ABITEBOUL97, BUNEMAN97] que contêm uma excelente introdução a dados semi-estruturados, e uma bibliografia relevante.

4.2 Algumas Pesquisas e Projetos sobre Modelos e Linguagens

Dados semi-estruturados podem ser naturalmente modelados como uma coleção de objetos, onde cada objeto pode ter qualquer quantidade de atributos, com possíveis repetições [SUCIU97]. Valores destes atributos podem ser outros objetos ou dados atômicos. Diversos modelos propostos para dados semi-estruturados consistem de alguma forma de grafo rotulado, onde cada nó corresponde a objetos ou valores, e os arcos correspondem a atributos [PAPAKONSTANTINOU+95, QUASS+95, BUNEMAN+96].

Desta forma, a Web é modelada como um grafo com arcos (links) e páginas (nós). Consultas podem ser baseadas no conteúdo (inclusive com predicados complexos) ou na estrutura (utilizando *regular path-expressions*).

As diversas propostas de linguagens de consultas para a Web, podem ser classificadas em duas gerações, conforme [FLORESCU+98]. A primeira geração é composta de linguagens como W3QL, WebSQL e WebLog, capazes de combinar buscas por conteúdo com buscas explorando estrutura de documentos. Estas linguagens tratam páginas da Web como objetos atômicos com propriedades de apontar para outros objetos e conter (ou não) um determinado texto.

Ainda conforme [FLORESCU+98], uma segunda geração de linguagens, exemplificada pelas linguagens WebOQL, StruQL e Florid, trouxe avanços em tópicos como acesso à estrutura do objeto manipulado. Estas linguagens permitem tanto a modelagem interna da estrutura de documentos, quanto a modelagem dos links externos que os conectam. Além disto, também é possível criar novas estruturas complexas como resultado de uma consulta.

Atualmente pode-se destacar o surgimento de uma terceira geração de linguagens de consulta, agora relacionadas a XML, assunto a ser comentado na seção 4.4.

4.3 Exemplos de Modelo de Dados e Linguagens de Consultas

Esta seção descreve um exemplo de modelos de dados semi-estruturados: o modelo OEM, extremamente bem aceito pela comunidade acadêmica, sendo utilizado em diversos projetos. Em seguida, a linguagem de consulta Lorel é mencionada, valendo comentar que esta linguagem é um exemplode uso do modelo OEM.

4.3.1 O Modelo de Dados OEM

O modelo de dados OEM foi proposto inicialmente para o projeto TSIMMIS [HAMMER+95], e devido a sua simplicidade e flexibilidade foi reutilizado em outros projetos. OEM é um modelo de objetos aninhados, onde objetos são auto-descritos através de rótulos. Não há esquema fixo. Conforme apresentado na Figura 7, um objeto OEM consiste de:

- um rótulo cujo nome é o nome da classe do objeto;
- um tipo que pode ser atômico ou um composto (conjunto);
- um valor que pode ser atômico ou um conjunto de objetos;
- um identificador de objetos (IDO), que é opcional.

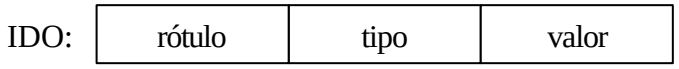

Figura 7 - Um objeto OEM

Apesar de ser considerado um objeto, vários conceitos de orientação a objetos são ampliados neste modelo. Por exemplo, o sistema de tipagem é elementar e bastante flexível.

Um IDO pode ser uma expressão descrevendo a origem do objeto, ou um ponteiro para outro objeto, ou ainda, o identificador pode ser local a uma consulta, sem ser persistente. Um rótulo nomeia a representação semântica de um objeto, sendo por esta razão o modelo chamado de auto-descritivo. Com estas duas primitivas e as duas adicionais: tipo e valor, é possível simular as estruturas encontradas em SGBDs orientados a objetos convencionais, como por exemplo composições de objetos complexos.

A Figura 8 apresenta um exemplo de grafo rotulado composto de uma coleção de objetos OEM. No topo encontra-se o objeto raiz cujo rótulo é DBGroup. Seu valor é um conjunto de objetos, portanto seu tipo é "conjunto". Dentre o conjunto de objetos que compõem o valor do DBGroup, pode-se observar três rotulados como Member e dois como Project. Um objeto Member possui um valor cujo tipo também é conjunto. É importante destacar que os valores deste objeto são conjuntos de sub-objetos com diferentes rótulos, cada um simulando um atributo de uma estrutura. No entanto, não há esquema neste exemplo.

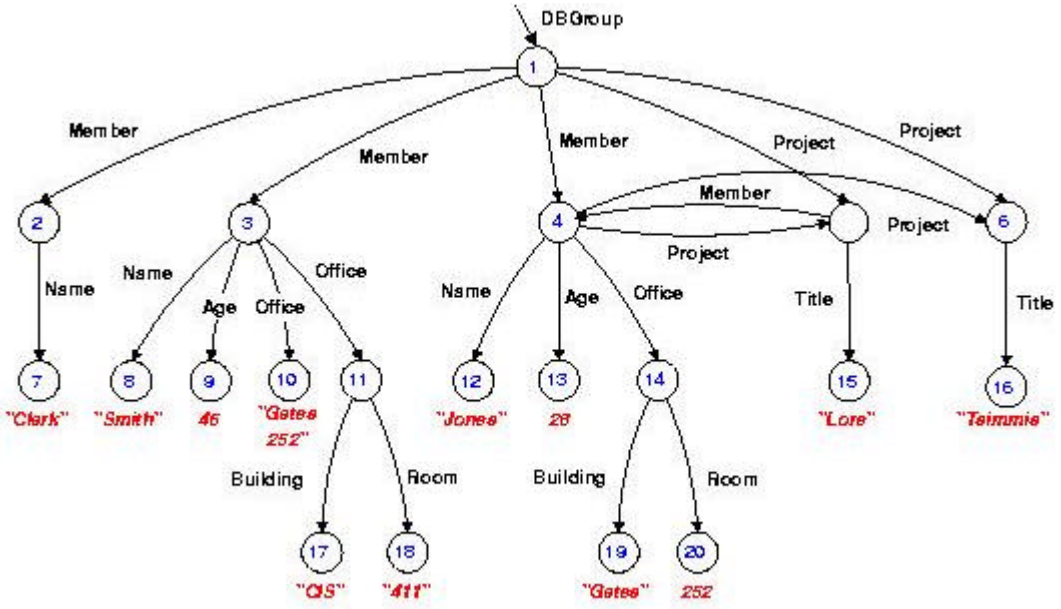

Figura 8 - Exemplo de Grafo Rotulado utilizando o modelo OEM

O modelo OEM pode ser visto como um modelo orientado a objetos, pois fornece algumas das vantagens como a representação natural de estruturas complexas. Entretanto, o modelo OEM também pode ser visto como uma forma de lógica de primeira ordem, onde rótulos são predicados e eles relacionam IDOs a outros IDOs (ou valores atômicos) de forma semelhante a objetos [GM+97].

4.3.2 Um Exemplo de Linguagem de Consulta para a Web

Atualmente, consultas utilizando técnicas de *Information Retrieval* ignoram a noção de existência de esquemas em dados da Web. As ferramentas atuais baseiam-se em busca de palavras-chave, considerando apenas a palavra e no máximo o seu posicionamento no documento. Este tipo de abordagem se mostra inadequado toda vez que um usuário tenta realizar uma busca e recebe como retorno páginas irrelevantes.

Parece vantajoso adicionar a abordagem típica de consulta em Bancos de Dados, onde uma linguagem de consulta declarativa e expressiva leva em consideração a estrutura do dado. É justamente neste contexto que surgiram as propostas de linguagens para consultas na Web.

A linguagem Lorel (Lighweight Object REpository Language) é uma extensão de OQL para o modelo OEM e foi originalmente criada para ser a linguagem de consulta do Projeto TSIMMIS. A linguagem Lorel também foi utilizada como linguagem de consulta para o repositório do Projeto Lore (Lighweight Object REpository), que tinha como objetivo principal a construção de um SGBD para o modelo OEM.

A especificação completa de Lorel, incluindo sua sintaxe e semântica formal é apresentada em [QUASS+95].

Os princípios básicos da linguagem Lorel podem ser resumidos em tratamento de dados irregulares e incompletos, bem como não obrigatoriedade de conhecimento da completa estrutura do objeto. Lorel é uma extensão de OQL [CATTELL+96], provendo conceitos de: conjuntos

heterogêneos; ausência de classes, permitindo que a checagem de tipos não seja restritiva; coerção de tipos extensiva e expressões de caminho genéricas. A linguagem também oferece suporte a quantificadores existencial e universal, variáveis tipo rótulo e caminho, acesso a funções externas, agregações, etc.

Indexação em SGDBs tradicionais envolvem índices em atributos (SGBDs Relacionais) e índices em caminhos (SGBDs Orientados a Objetos). No Projeto Lore, existem diversos tipos de indexação, como: indexação por valores (ex.: atributos opcionais); indexação de links (ex.: rótulos de arcos que alcançam um determinado objeto); indexação de nós (ex.: todas as páginas); indexação de texto e indexação de caminho.

A execução das consultas adota uma abordagem simplificada de navegações exaustivas descendentes. Mas os índices permitem outras abordagens, como: ascendente, híbrida ou por caminho.

As otimizações de consultas são baseadas em custo, estimando quantidade de I/O. Planos de consulta lógicos flexíveis são possíveis, devido aos fatos mencionados acima. Planos físicos são gerados em grande variedade. As decisões do plano ótimo para uma determinada consulta são basicamente locais (também há heurísticas). Um novo tipo de estatística pode ser obtido através do uso de DataGuides, sumários dinâmicos da estrutura de uma determinada fonte.

4.4 A importância de XML para Modelo de Dados e Linguagens de Consultas

Com a utilização de XML como padrão para descrição de conteúdos na Web, tornou-se necessário reavaliar as linguagens de consulta existentes. Atualmente ainda não há consenso na escolha da linguagem adequada. Diversas propostas vem sendo desenvolvidas, conforme pode-se observar na quantidade de publicações do *W3C Query Language Workshop* [W3C-QL'98] e do *Int. Workshop on Web and Databases* [WEBDB'99]. Em [MAIER98], Maier enumera um conjunto de características desejadas em linguagens de consultas XML.

Duas propostas atuais merecem destaque: XML-QL [DEUTSCH+98] e XQL [ROBIE+98]. A primeira foi desenvolvida pelo AT&T Labs, com base no sistema Strudel, enquanto que a segunda proposta foi desenvolvida pela Microsoft, tendo como base principal o uso de XSL (eXtensible Style Sheets). Além do fato da proposta XML-QL possuir tratamento de junções e XQL não, outro fator tem adiado a aceitação da proposta da Microsoft: XSL ainda não está aprovada como recomendação do consórcio W3C.

Na época da publicação deste trabalho, foi anunciado o lançamento de Lore para XML [GOLDMAN+99], um SGBD para XML, utilizando o modelo OEM. Lore-XML foi resultado da migração para XML de um SGBD desenvolvido para dados semi-estruturados genéricos [MCHUGH+97]. O sistema inicialmente não explorava a possível estrutura dos dados, definida através de DTDs, deixando em aberto a questão de otimização de consultas com estruturas pre-definidas. Estudos com DataGuides [GOLDMAN+97] demonstraram progressos nesta direção. Vale comentar que estes pesquisadores da universidade de Stanford iniciaram uma inovadora proposta para busca por proximidade que permite benefícios para buscas em XML.

5. Conclusões

Com o crescente uso da Web, é possível identificar uma grande volume de dados disponíveis em diferentes fontes de dados. Entretanto, a tecnologia atual ainda não é adequada para extrair informação relevante de forma automática. Diversas áreas de pesquisa vêm desenvolvendo propostas para estes desafios. Pesquisadores da área de Bancos de Dados têm procurado revisitar temas tradicionais como integração de informações, modelos de dados e linguagens de consulta para este novo contexto da Web.

Apesar da Web ser considerada um grande Banco de Dados Distribuído, é importante ressaltar que este BD não é trivial. Na Web não há estrutura uniforme, modelo de dados padrão ou linguagem de consulta padrão. É necessário, portanto, adotar novas abordagens.

Este trabalho tem como objetivo, fornecer uma visão atualizada do estado da arte nas pesquisas dos temas mencionados, complementando o trabalho realizado por Florescu, Levy e Mendelson em [FLORESCU+98]. Exemplos de projetos são apresentados e também uma avaliação preliminar do impacto da recente linguagem extensível XML.

Referências Bibliográficas

[ABITEBOUL97] ABITEBOUL, Serge "Querying Semi-Structured Data", Proceedings of the 6th International Conference on Database Theory (ICDT'97), 1997.

[BONNET+98] X BONNET, Philippe TOMASIC, Anthony "Partial Answers for Unavailable Data Sources", Proceedings of the 3rd International Flexible Query Answering Systems Conference (FQAS'98), Lecture Notes in Computer Science, vol. 1495, Springer, 1998.

[BOSAK+99] BOSAK, Jon BRAY, Tim "XML and the Second-Generation Web", *Scientific American*, maio 1999

[BUNEMAN96] BUNEMAN, Peter DAVIDSON, Susan HILLEBRAND, Gerd SUCIU, Dan "A Query Language and Optimization Techniques for Unstructured Data", *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, SIGMOD RECORD, vol. 25, n. 2,* junho 1996.

[BUNEMAN97] BUNEMAN, Peter "Semistructured Data", Proceedings of the 16th Symposium on Principles of Database Systems (PODS'97), 1997.

[CATTELL+96] CATTELL, Rick et al "The Object Database Standard - ODMG 93, Release 1.2", Morgan Kaufmann, 1996.

[DEUTSCH+98] DEUTSCH, Alin FERNANDEZ, Mary FLORESCU, Daniela LEVY, Alon SUCIU, Dan "XML-QL", *http://www.w3.org/TR/1998/NOTE-xml-ql-19980819/*, 1998.

[FLORESCU+98] FLORESCU, Daniela LEVY, Alon MENDELSON, Aberto "Database Techniques for the World-Wide Web: A Survey", *SIGMOD Record, vol. 27, n. 3*, setembro 1998.

[GOLDMAN+97] GOLDMAN, Roy WIDOM, Jennifer "DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases", *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB'97)*, agosto 1997.

[GOLDMAN+98] GOLDMAN, Roy SHIVAKUMAR, Narayanan VENKATASUBRAMANIAN, Suresh GARCIA-MOLINA, Hector "Proximity Search in Databases" *Proceedings of the 24th International Conference on Very Large Databases (VLDB'98)*, 1998.

[GOLDMAN+99] GOLDMAN, Roy MCHUGH, Jason WIDOM, Jennifer "From Semistrutured Data to XML: Migrating the Lore Data Model and Query Language", *Proceedings of the 6th International* 2nd Workshop on Web and Databases (WebDB'99), *http://www-rocq.inria.fr/~cluet/WEBDB/procwebdb99.html*, 1999.

[GARCIA-MOLINA +97] GARCIA-MOLINA, Hector et at "The TSIMMIS Approach to Mediation: Data Models and Languages", *Journal of Intelligent Information Systems (JIIS)* , vol. 8, n. 2, 1997.

[HAMMER+95] HAMMER, Joachim GARCIA-MOLINA, Hector IRELAND, Kelly PAPAKONSTANTINOU, Yannis ULLMAN, Jeffrey WIDOM, Jennifer "Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System", *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, SIGMOD RECORD, vol. 24, n. 2, junho 1995.

[KAPITSKAIA+97] KAPITSKAIA, Olga TOMASIC, Anthony VALDURIEZ, Patrick "Dealing with Discrepancies in Wrapper Functionality", *Technical Report RR-3138 INRIA*, Rocquencourt, França, março 1997.

[LEVY99] LEVY, Alon "More on Data Management for XML", *http://www.cs.washington.edu/homes/alon/widom-response.html*, versão on-line obtida em 27/06/99, 1999.

[MAIER98] MAIER, David "Database Desiderata for an XML Query Language", *Eletronic Proceedings of the W3C Query Language Workshop, http://www.w3.org/TandS/QL/QL98*, 1998, obtido em 21/06/98.

[MCHUGH+97] MCHUGH, Jason ABITEBOUL, Serge GOLDMAN, Roy QUASS, Dallan WIDOM, Jennifer "Lore: A Database Management System for Semistructured Data", *SIGMOD Record, vol. 26, n. 3*, setembro 1997.

[NAACKE+98] NAACKE, Hubert GARDARIN, Georges TOMASIC, Anthony "Leveraging Mediator Cost Models with Heterogeneous Data Sources", *Proceedings of the 14th International Conference on Data Engineering (ICDE'98),* 1998.

[ÖZSU+99] ÖZSU, Tamer VALDURIEZ, Patrick *"Principles of Distributed Database Systems*", Prentice Hall, 1999.

[PAPAKONSTANTINOU+95] PAPAKONSTANTINOU, Yannis GARCIA-MOLINA, Hector WIDOM, Jennifer "Object Exchange Across Heterogeneous Information Sources", *Proceedings of the 11th International Conference on Data Engineering (ICDE'95),* 1995.

[PRASAD+99] PRASAD, S. RAJARAMAN, Anand "Virtual Database technology, XML, and the Evolution of the Web*", IEEE Data Engineering Bulletin, Special Issue on Databases and the World Wide Web, vol. 21, n. 2,* junho 1998.

[QUASS+95] QUASS, Dallan RAJARAMAN, Anand SAGIV, Yehoshua ULLMAN, Jeffrey WIDOM, Jennifer "Querying Semistructured Heterogeneous Information"*Proceedings of the Deductive and Object-Oriented Databases, 4th International Conference, (DOOD'95),* dezembro 1995..

[ROBIE+98] ROBIE, Jonathan LAPP, Joe SCHACH, David "XQL - XML Query Language", *http://www.w3.org/TandS/QL/QL98/pp/xql.html,* 1998.

[SILBERSCHATZ+97] SILBERSCHATZ, Abraham ZDONIK, Stanley "Database Systems - Breaking Out of the Box", *SIGMOD Record, vol. 26, n.3,* setembro 1997.

[SUCIU:97] SUCIU, Dan "Management of Semistructured Data", SIGMOD Record, Foreword of special session dedicated to the Workshop on Management of Semi-Structured Data, vol. 26, n. 4, dezembro 1997.

[TOMASIC+95] TOMASIC, Anthony RASCHID, Louiqa VALDURIEZ, Patrick "Scaling Heterogeneous Databases and the Design of Disco", *Technical Report RR-2704 INRIA,* Rocquencourt, França, novembro 1995.

[TOMASIC+98] TOMASIC, Anthony RASCHID, Louiqa VALDURIEZ, Patrick "Scaling Access to Heterogeneous Data Sources with DISCO",*IEEE Transactions on Knowledge and Data Engineering, vol. 10, n. 5,* setembro/outubro 1998.

[W3C] World Wide Web Consortium Home Page (W3C) , *http://www.w3.org/*

[W3C-QL'98] W3C Eletronic Proceedings of the W3C Query Language Workshop, *http://www.w3.org/TandS/QL/QL98*, 1998.

[W3C-XML-ACTIVITY] W3C XML Activity Web Page , *http://www.w3.org/XML/Activity.html*

[W3C-XML98] W3C "XML 1.0 Recommendation" , *http://www.w3.org/XML*, 1998.

[WEBDB'99] Eletronic Proceedings of the 2nd International Workshop on Web and Databases (WebDB'99), *http://www-rocq.inria.fr/~cluet/WEBDB/procwebdb99.html*, 1999.

[WIDOM99] WIDOM, Jennifer "Data Management for XML", *http://www-db.stanford.edu/~widom/xml-whitepaper.html*, versão on-line obtida em 27/06/99, a ser publicado em IEEE Data Engineering Bulletin, Special Issue on XML, vol. 22, n. 3, setembro 1999.

[WIEDERHOLD92] WIEDERHOLD, Gio "Mediators in the Architecture of Future Information Systems", *IEEE Computer*, vol. 25, n. 3, 1992.

[WIEDERHOLD94] WIEDERHOLD, Gio "Interoperation, Mediation, and Ontologies", *Proceedings of the International Symposium on Fifth Generation Computer Systems (FGCS'94)*, Workshop on Heterogeneous Cooperative Knowledge-Bases, vol.W3, dezembro 1994.

[WIEDERHOLD97] WIEDERHOLD, Gio GENESERETH, Michael "The Conceptual Basis for Mediation Services", *IEEE Expert, vol. 12, n. 5* , 1997.