2
Preliminares

2.1
Motivação

Por definição, interoperabilidade é a habilidade de dois ou mais sistemas ou componentes compartilhar informações e utilizar estas informações compartilhadas (IEEE, 1990). Atualmente, o compartilhamento de informações é um fator crítico para o sucesso das organizações, devido à emergente necessidade de comunicação e compartilhamento de informações com seus parceiros de negócios. A habilidade da organização em adaptar-se rapidamente a novas fontes de informação e modelos de negócios é fator determinante para seu crescimento e sobrevivência no mercado atual.

Recentemente, a Internet foi identificada como a principal plataforma para desenvolvimento de aplicações, justamente por viabilizar o desenvolvimento de aplicações entre organizações (Abiteboul et al., 2003).

De fato, historicamente, os sistemas de informação eram internos às empresas e podiam ser especificados e otimizados totalmente para um único domínio administrativo. O uso de um único *sistema de bancos de dados* era suficiente para organizar, manter e gerenciar os dados destas aplicações.

Entretanto, a maioria das empresas hoje estão interessadas em interagir com seus fornecedores, para compartilhar informações, e com seus clientes, para fornecer melhor atendimento. Estes sistemas de informação compartilhados requerem atenção especial em fatores como segurança e integração de informações, gerando novas questões para serem resolvidas pela comunidade de Banco de Dados.

Neste contexto, o Sistema Gerenciador de Bancos de Dados (SGBD) ideal deve oferecer suporte a consultas através de múltiplas fontes e encaminhar as consultas e atualizações diretamente às fontes de onde os dados a serem atualizados são provenientes. Além disso, deve respeitar as restrições de integridade dos múltiplos bancos de dados.

Neste capítulo é feito um levantamento das barreiras para interoperabilidade de sistemas de informação e das abordagens propostas para

integração, incluindo, entre outras: as federações de bancos de dados, a arquitetura de mediadores, o uso de ontologias e repositórios de meta-informação[1].

2.2
Dimensões para classificação de Sistemas de Informação

Fundamentalmente, um sistema de informação fornece acesso a informações extraídas de dados que são mantidos e gerenciados em algum lugar e de alguma forma. As arquiteturas tradicionais utilizam um único sistema gerenciador de banco de dados com múltiplos bancos de dados. Devido à emergente necessidade de integração de informações distribuídas, algumas barreiras precisam ser atravessadas de maneira a promover a interoperabilidade dos sistemas consumidores destas informações.

Sheth e Larson (1990) caracterizam os sistemas de informação segundo três dimensões ortogonais - distribuição, heterogeneidade e autonomia - explicadas nas seções subseqüentes.

A Figura 1 organiza as arquiteturas de sistemas de bancos de dados conforme sua classificação em relação a estas três dimensões.

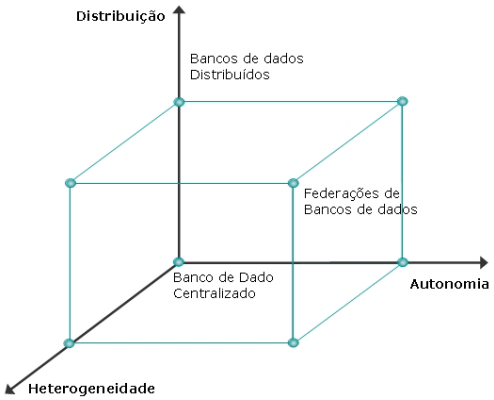

Figura 1 – Caracterização das arquiteturas de bancos de dados.

Em (Busse et al., 1999) é identificada a quarta dimensão relativa à flexibilidade do sistema para evolução, explicada na seção 2.2.4.

[1] O termo “meta-informações” cobre os conceitos de ontologias e metadados.

2.2.1
Distribuição

Uma das barreiras para a interoperabilidade entre sistemas é a característica distribuída das fontes de informação. As fontes podem estar fisicamente localizadas em um único ou múltiplos computadores, que por sua vez podem estar co-localizados ou geograficamente distribuídos, mas interconectados por um sistema de comunicação.

Devido à evolução das redes de computadores, a maioria dos computadores encontra-se conectado a algum tipo de rede. Atualmente, o exemplo mais trivial é a Internet. Portanto, é natural pensarmos em interoperar estes sistemas fisicamente dispersos, realizando sua comunicação via rede.

2.2.2
Heterogeneidade

Naturalmente, a heterogeneidade dos sistemas de informação ocorre devido às diferentes soluções adotadas por equipes de desenvolvimento autônomas. Cada equipe possui uma forma particular de entender o problema, seu próprio ambiente tecnológico e requisitos específicos das aplicações a serem desenvolvidas. Devido a estes e outros fatores, até mesmo os mesmos conceitos do mundo real são modelados de diferentes formas.

Dentre as diversas classificações para heterogeneidade apresentadas na literatura, neste trabalho foi adotada a classificação apresentada em (Busse et al., 1999) que divide o problema da heterogeneidade dos sistemas de informação em três categorias principais: a heterogeneidade sintática, de modelo de dados e a lógica.

- **Heterogeneidade sintática:** engloba problemas sintáticos relacionados aos aspectos técnicos para comunicação dos sistemas de informação envolvidos. A heterogeneidade técnica aborda diferenças nos aspectos técnicos, tais como: hardware, plataforma, sistema operacional e métodos de acesso (protocolos, como por exemplo HTTP, ODBC, CORBA, etc). Já a heterogeneidade de interface de acesso existe quando o acesso aos componentes é feito de forma diferente, devido a diferenças nas linguagens de consulta (heterogeneidade de linguagem) e a diferenças referentes às restrições

das consultas, isto é, quando apenas algumas operações são permitidas.

- **Heterogeneidade de modelo de dados:** este tipo de heterogeneidade reflete as diferenças entre os modelos de dados. Modelos de dados diferentes possuem primitivas estruturais diferentes. Como exemplo, temos o suporte a herança e generalização fornecido pelo modelo orientado a objetos, que não é oferecido pelo modelo relacional.

- **Heterogeneidade lógica:** envolve as diferenças lógicas relacionadas a comunicação dos sistemas de informação envolvidos. É classificada em três tipos: semântica, esquemática e estrutural. A heterogeneidade semântica preocupa-se com a semântica dos dados e metadados. Ela ocorre quando existe discordância a respeito do significado, interpretação ou uso pretendido, por exemplo de um determinado objeto, atributo ou tabela de um banco de dados.

  Sem dúvida, este é um dos principais problemas a serem solucionados para interoperar diversas fontes de dados. Até mesmo esquemas formulados no mesmo modelo podem ter interpretações diferentes. Os nomes das entidades utilizadas em um esquema carregam uma semântica implícita, representando um conceito num determinado contexto. A interpretação desses nomes não necessariamente irá coincidir quando realizada por diferentes pessoas, caracterizando um conflito semântico. Além desse, podem ocorrer conflitos devido a nomes iguais que representam diferentes conceitos (homônimos) ou quando nomes diferentes representam o mesmo conceito (sinônimos).

  Além disso, um mesmo atributo pode ter a mesma semântica e ainda assim apresentar heterogeneidade na sua representação. Como exemplo, temos “preço” como nome de um atributo de uma fonte de dados A, e o mesmo nome para um atributo de uma fonte de dados B. Na fonte A, o preço esta sendo representado em “dólares americanos”, enquanto na fonte B, em “reais”.

  A heterogeneidade esquemática ocorre quando os conceitos são modelados usando diferentes elementos de um modelo de dados. No modelo relacional, por exemplo, pode existir o conflito entre nomes de atributos e valores de atributos (Busse et al., 1999). Por exemplo, em

sistemas de reservas de passagens aéreas, na Tabela 1 (a) do banco de dados do sistema A, as cidades atendidas pelos vôos são modeladas como atributos, e no sistema B a Tabela 1 (b) modela as cidades como valores do atributo “Cidade”.

Tabela 1 – Exemplo de heterogeneidade esquemática.

Código	Manaus	Joinville
AC1441	X	X
GO6790	X	
VR0990		X

(a)

Vôo	Cidade
AC1441	Manaus
AC1441	Joinville
VR0990	Joinville

(b)

A heterogeneidade estrutural existe quando os elementos com um mesmo significado e modelados seguindo o mesmo modelo de dados, e esquematicamente homogêneos, encontram-se estruturados de forma diferente, por exemplo, os atributos agrupados em tabelas diferentes.

2.2.3 Autonomia

A autonomia das fontes de informação é outro fator que pode influenciar na interoperabilidade de sistemas de informação. A autonomia é uma função de diversos fatores, para determinar o grau de independência das fontes que desejam interoperar, como por exemplo: o fato das fontes trocarem ou não informações, de poderem executar transações de forma independente e de ter ou não permissão de modificar informações. Em (Özsu & Valduriez, 1999) a seguinte classificação é usada para determinar o grau de autonomia em sistemas de bancos de dados distribuídos:

- **Sistemas estreitamente integrados:** é disponibilizada uma única imagem do banco de dados. Da perspectiva do usuário, os dados estão logicamente centralizados em um único banco de dados. O controle do processamento de cada solicitação de usuário fica sob responsabilidade de um sistema gerenciador de dados, mesmo que essa solicitação possa ser atendida por mais de um gerenciador de dados. Nesta classificação os sistemas gerenciadores de dados não operam como SGBDs independentes, embora tenham a funcionalidade para fazê-lo.

- **Sistemas semi-autônomos:** são os SGBDs que podem operar independentemente (e normalmente o fazem), mas que decidiram participar de uma federação para tornar seus dados locais compartilháveis. Cada um desses SGBDs determina que partes de seu próprio banco de dados eles tornarão acessíveis para usuários de outros SGBDs. Eles não são sistemas totalmente autônomos pois precisam ser modificados para que possam trocar informações uns com os outros.

- **Sistemas totalmente isolados:** os sistemas são SGBDs independentes, que não sabem da existência de outros SGBDs nem se comunicam com eles. Nesses sistemas, o processamento de transações do usuário que acessam vários bancos de dados é difícil pois não existe nenhum controle global sobre a execução de SGBDs individuais.

2.2.4 Flexibilidade

Além das características citadas acima de autonomia, heterogeneidade e distribuição, atualmente a maioria das soluções que usam sistemas de informação possuem em comum a necessidade de ser flexíveis como requisito essencial para seu desenvolvimento.

Devido à quantidade de informação disponível e a emergente necessidade de comunicação e compartilhamento de informações com diversas entidades, o sistema deve ter a capacidade de adaptar-se rapidamente a novas fontes de informação.

2.3 Abordagens tradicionais para integração

Tradicionalmente, para interoperar sistemas de informação é necessário integrá-los, ou seja, eles precisam compartilhar as informações de forma homogênea.

De maneira geral, a forma tradicional de endereçar o processo de integração de sistemas de bancos de dados baseia-se no projeto de um esquema conceitual global. Ele pode ocorrer em duas etapas: *conversão de*

esquema (ou apenas *conversão* ou *tradução*) e *integração de esquema* (Özsu & Valduriez, 1999).

Na etapa de conversão de esquema, os esquemas dos bancos de dados são convertidos em uma representação canônica intermediária. O uso de uma representação canônica facilita o processo de conversão do modelo de dados, reduzindo o número de conversores que precisam ser escritos. A etapa de conversão só é necessária se os bancos de dados são heterogêneos e se cada esquema local pode ser definido com o uso de um modelo de dados diferente. Para a conversão devem ser estabelecidas equivalências entre os conceitos do modelo de origem e os do modelo de destino e o esquema conceitual global deve ser especificado seguindo o modelo de dados do modelo canônico.

Na etapa de integração de esquemas, é gerado um esquema conceitual global para integrar os esquemas intermediários produzidos na etapa anterior. A integração de esquemas é o processo de *relacionar* os componentes de um banco de dados uns aos outros, *selecionar* a melhor representação para o esquema conceitual global e, finalmente, *integrar* os componentes de cada esquema intermediário.

Nesta etapa, podem ser usados esquemas externos locais, no lugar dos esquemas conceituais locais, pois talvez não seja desejável incorporar o esquema conceitual local inteiro na arquitetura integrada.

Neste contexto, a principal barreira é o problema da heterogeneidade. Para resolver este problema, diversas soluções têm sido propostas e serão brevemente discutidas nas seções a seguir.

2.3.1 Arquitetura para múltiplos bancos de dados

Originalmente o principal problema da comunidade de banco de dados era integrar dados de diferentes bancos de dados com um mesmo modelo de dados, por exemplo, o relacional. Para isso, é criada uma camada intermediária para acesso integrado aos múltiplos bancos de dados. O problema dessa abordagem está na existência de dados armazenados em sistemas que utilizam outros modelos de dados, como por exemplo os bancos de dados legados (bancos de dados hierárquicos e de rede), bancos de dados orientados a objetos,

documentos estruturados (XML[2]) e documentos não estruturados (arquivos texto e HTML).

Esta arquitetura é indicada para bancos de dados que seguem um mesmo modelo de dados. Além disso, as diferenças entre os esquemas não são tratadas, pois é feita apenas a união dos esquemas das múltiplas fontes.

2.3.2 Bancos de dados federados

Para endereçar o problema da heterogeneidade entre os modelos de dados de cada fonte, surgiu a idéia de um sistema de bancos de dados federados (*Federated DataBase System* - FDBS). Um FDBS é formado por uma coleção de sistemas de bancos de dados cooperativos, autônomos e possivelmente heterogêneos (Sheth & Larson, 1990). Estes sistemas formam uma coleção de bancos de dados autônomos que cooperam com a federação oferecendo suporte às operações globais.

Os conceitos chave de um FDBS são a autonomia dos bancos de dados componentes, e o compartilhamento parcial e controlado dos dados.

Para permitir o compartilhamento controlado preservando a autonomia dos bancos de dados federados e a execução contínua das aplicações existentes, um FDBS suporta dois tipos de operações: locais e globais. As operações globais incluem o acesso aos dados usando o sistema gerenciador da federação e pode incluir o gerenciamento dos dados pelos múltiplos bancos de dados federados. Os SGBDs das fontes de dados componentes devem permitir o acesso aos dados que gerenciam. As operações locais são submetidas diretamente aos SGBDs das fontes federadas.

As federações de bancos de dados podem ser classificadas com base em quem as gerencia e como os bancos de dados são integrados. De acordo com sua classificação elas serão: fracamente acopladas ou fortemente acopladas.

Uma federação é considerada fracamente acoplada se a responsabilidade de criar e manter a federação é dos usuários da federação e não existe controle centralizado por parte do sistema federado e seu administrador. Uma federação fracamente acoplada sempre oferece suporte a utilização de múltiplos esquemas federados.

Já uma federação é fortemente acoplada quando o sistema federado ou seu administrador controla o acesso aos bancos de dados componentes. Este

[2] XML - http://www.w3.org/TR/REC-xml/

tipo de federação pode ter um ou mais esquemas federados. Ela é chamada de uma *federação única* se permite a criação e gerenciamento de apenas um esquema federado.

A Figura 2 mostra a arquitetura clássica em cinco camadas de um FDBS (Sheth & Larson, 1990) composta pelos seguintes itens:

- **Fontes de dados:** representa as diversas fontes de dados que desejam compartilhar informações entre si. Estas fontes componentes da federação podem ser de naturezas diversas: bancos de dados, páginas HTML, arquivos XML, entre outros.

- **Esquema local:** representa o esquema conceitual de uma fonte de dados componente. Este esquema representa o modelo de dados nativo do componente, que pode ser um banco de dados relacional, banco de dados orientado a objetos, arquivo XML, página HTML, entre outros. Portanto, os esquemas locais componentes da federação podem ser representados seguindo diferentes modelos de dados.

- **Esquema componente:** esquema gerado a partir da tradução do esquema local para o modelo de dados canônico adotado pela federação.

- **Esquema de exportação:** representa o subconjunto do esquema componente que será compartilhado na federação. Um banco de dados componente não precisa disponibilizar para a federação e seus usuários a sua coleção completa de dados. Além disso, este esquema inclui informações para controle de acesso relativo a sua acessibilidade para usuários específicos da federação.

- **Esquema federado:** representa a integração de múltiplos esquemas de exportação. Chamado também de esquema conceitual global.

- **Esquema externo:** define a visão que o usuário (aplicação) ou um grupo de usuários (conjunto de aplicações) terão dos dados da federação. O uso de esquemas externos pode viabilizar a personalização das informações disponibilizadas aos usuários. Como o esquema global de uma federação poder ser grande, complexo e

difícil de sofrer mudanças o uso de um esquema externo pode ser personalizado. O esquema externo pode conter apenas um subconjunto das informações da federação contendo apenas aquelas informações relevantes para um usuário ou para um grupo de usuários da federação. Além disso, o esquema externo pode ser representado no modelo de dados de preferência do usuário.

A limitação dessa abordagem está na consulta à federação, pois as aplicações devem especificar explicitamente as fontes de dados federadas utilizadas na consulta. Isto exige mudanças nas aplicações no momento em que uma nova fonte de dados passa a fazer parte da federação.

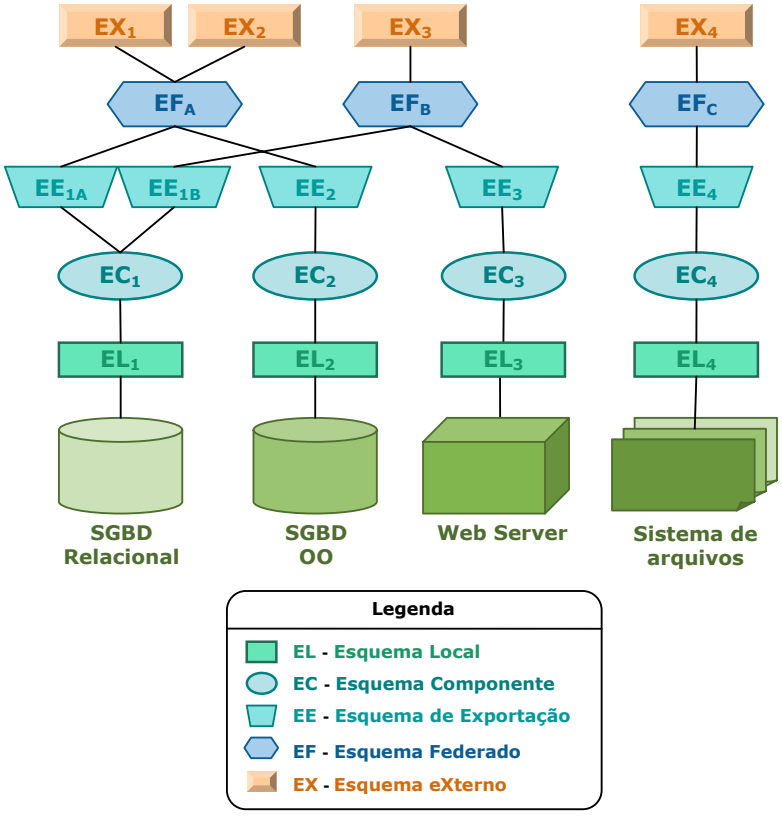

Figura 2 – Arquitetura de cinco níveis de esquemas para FDBS.

2.3.3
Arquitetura de mediadores

O termo “mediador”, introduzido em (Wiederhold, 1992), vem sendo utilizado em diversas publicações a respeito de técnicas e projetos de integração de dados. Um mediador é um componente de software que faz a mediação entre

o usuário e as fontes de dados físicas. Em particular, os mediadores são desenvolvidos para usar outros mediadores como componentes.

A Figura 3 mostra a arquitetura clássica de mediador-adaptador, adaptada de (Busse et al., 1999). Ela é composta pelas seguintes camadas:

- **Camada de Mediação:** contém diversos mediadores fornecendo serviço de mediação para fontes de dados ou para outros mediadores, formando redes de mediadores. Cada mediador possui seu próprio esquema federado. O mediador centraliza as informações fornecidas pelos adaptadores numa visão unificada dos dados disponíveis na fonte. Além disso, o mediador decompõe as consultas do usuário em consultas menores posteriormente executadas pelos adaptadores, e reúne os resultados parciais e calcula a resposta à consulta do usuário.
- **Camada de Adaptação:** contém os adaptadores responsáveis pelo acesso às fontes de dados. Cada adaptador esconde a heterogeneidade técnica e de modelo de dados da fonte de dados, tornando o acesso à fonte de dados transparente para o mediador. Para cada fonte de dados existe um adaptador que exporta algumas informações sobre a fonte, tais como: seu esquema, informações sobre seus dados e sobre seus recursos para processamento das consultas.

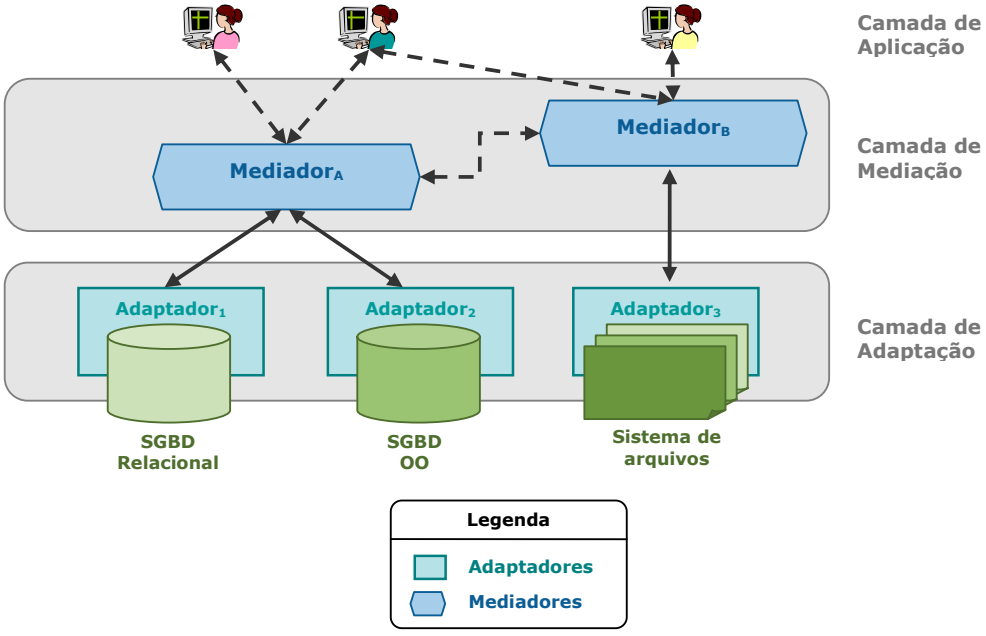

Figura 3 – Arquitetura mediador-adaptador.

Por utilizar um esquema federado em cada mediador para fornecer um acesso integrado aos dados de diferentes fontes de dados, a arquitetura de mediadores é vista como uma evolução da arquitetura de federações. Assim, a arquitetura de mediadores viabiliza a interoperabilidade entre diferentes tipos de fontes de dados, incluindo as fontes disponíveis na Web.

Para integrar fontes na Web, deve-se observar algumas peculiaridades (Özsu & Valduriez,1999):

- o número de fontes de dados pode ser muito alto, tornando a tarefa de integração e da resolução de conflitos um grande problema;
- a Web é um ambiente dinâmico, portanto a inclusão e eliminação das fontes de dados deve ter impacto mínimo sobre o esquema integrado;
- as fontes de dados podem ser recursos diferentes, variando desde SGBDs completos até arquivos simples, incluindo fontes não-estruturadas ou semi-estruturadas e, portanto, não oferecendo praticamente nenhuma informação para criação do esquema integrado.

A arquitetura de mediador-adaptador oferece vantagens. Um exemplo disso é através dos componentes especializados da arquitetura, que permitem que as peculiaridades de diferentes tipos de usuários sejam tratadas separadamente. Em geral, os mediadores se especializam em um conjunto inter-relacionado de fontes de dados com dados “semelhantes”, e assim exportam esquemas e semânticas relacionadas para um domínio específico. A especialização dos componentes leva a um sistema distribuído flexível e extensível.

A Figura 4 ilustra uma hierarquia de mediadores especializados, com um mediador para Recuperação de Informações ($Mediador_{RI}$), responsável por mediar o acesso a fontes de dados na Web, um mediador para bancos de dados ($Mediador_{BD}$) heterogêneos e um mediador de BD e RI ($Mediador_{BD/RI}$) que oferece recursos tanto para recuperação de informações na Web quanto de consultas a bancos de dados.

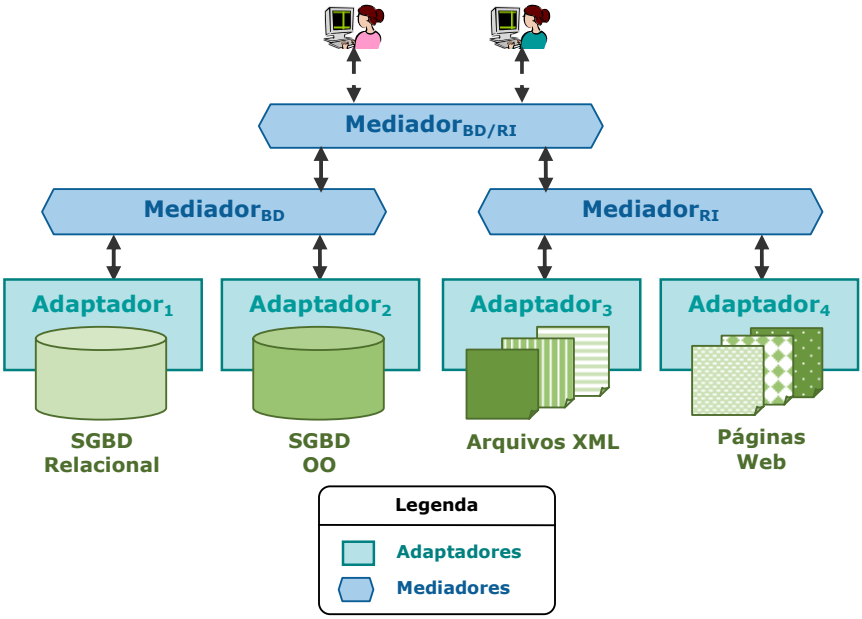

Figura 4 – Hierarquia de mediadores.

Em (Gupta et al., 1999) é proposta uma arquitetura baseada em mediadores para integração de informações de Sistemas de Informações Geográficas e de bancos de dados de imagens geo-referenciadas. A abordagem propõe a arquitetura padrão para mediadores utilizando um modelo de dados em XML na camada de mediação. Os adaptadores são responsáveis pela conversão dos dados e pelas consultas requisitadas entre o modelo de dados da fonte e o modelo XML do mediador. As conclusões da pesquisa incluem o desenvolvimento de regras e um modelo de custos para selecionar as fontes de dados no mediador baseado nos metadados de cada fonte; a incorporação de um mediador de alinhamento "inteligente"; o suporte a mecanismos alternativos para associação dos nomes a objetos geográficos para tratar o problema dos n:n mapeamentos.

2.3.4 Armazém de dados

O valor da informação nas organizações impulsionou a origem desta abordagem. Um armazém de dados ou data warehouse (DW) é um ambiente expansível e planejado para a análise de dados não voláteis. Uma DW é uma plataforma com dados integrados para apoiar vários sistemas de apoio de decisão e processos empresariais (Kimball, 1996; Inmon, 2002).

Uma DW combina várias tecnologias, tendo como objetivo a integração efetiva de bases de dados operacionais em um ambiente que viabilize o uso estratégico dos dados. Os dados provenientes das múltiplas bases de dados são materializados e integrados num repositório centralizado. Estas cópias estáticas dos dados não podem ser atualizadas, de forma a manter o histórico das informações.

De fato, para criação de um DW são enfrentados os mesmos problemas identificados para integração.

2.4 Abordagens inovadoras para integração

Devido à atual característica dinâmica das organizações, com as constantes mudanças no modelo de negócios das aplicações, o surgimento de cada vez mais parceiros de negócios e a avalanche de informações disponibilizada, é inviável a utilização das abordagens tradicionais para promover a interoperabilidade.

Atualmente, o sucesso está na utilização de arquiteturas mais extensíveis, flexíveis e com baixo custo de implementação e manutenção, viabilizando a rápida adaptação a novas bases de clientes e a novos parceiros de negócios.

Além da evolução das abordagens tradicionais para integração de sistemas, diversas novas idéias vêm emergindo de pesquisas envolvendo conceitos relacionados à *Web Services* e ontologias.

2.4.1 Arquitetura baseada em serviços

Para atender os requisitos de ambientes em constante evolução, foi proposta uma arquitetura baseada em serviços para integração de sistemas (Layzell et al., 2000).

Um serviço é uma implementação de uma funcionalidade de negócio bem definida que opera independentemente do estado de qualquer outro serviço definido no sistema. Os serviços possuem um conjunto bem definido de interfaces e operam através de um contrato pré-definido entre a aplicação cliente e o serviço em si.

Numa Arquitetura Baseada em Serviços (*Service Oriented Architecture* - SOA), o sistema opera como um conjunto de serviços (Gupta, 2004). Cada serviço pode interagir com vários outros para executar uma determinada tarefa.

Essencialmente, os serviços operam como entidades independentes, podendo evoluir sem comprometer a arquitetura como um todo.

A arquitetura de *Web Services* (Booth et al., 2004), proposta pela W3C, surgiu com o propósito de servir de padrão para interoperar diferentes aplicações, executando em diferentes plataformas. Para descrição dos *Web Services*, foi proposta uma linguagem baseada em XML chamada *Web Services Description Language* (WSDL) (Christensen et al., 2001). Para publicação e localização de serviços numa arquitetura baseada em serviços (*Web Services*), foi criado o *Universal Description Discovery and Integration* (UDDI) (Atkinson et al., 2003).

Outro exemplo usando a abordagem baseada em serviços encontramos na infra-estrutura para *Grid Computing*. Os Grids são ambientes computacionais distribuídos para fornecimento de recursos computacionais de larga escala. O objetivo é fornecer o compartilhamento de aplicações visando o alto desempenho na execução destas aplicações (Foster et al., 2001). Os serviços são fornecidos por instituições ou indivíduos particulares para consumo de outros usuários no Grid.

2.4.2
Abordagens baseadas no uso de ontologias

A chave para interoperabilidade é dar aos sistemas a habilidade para compartilhar informações de maneira automática e com entendimento comum e não ambíguo dos termos e conceitos utilizados nas aplicações envolvidas. Neste contexto, as ontologias mostram-se importantes artefatos que podem ser utilizados para viabilizar o tratamento da heterogeneidade.

Berners-Lee, criador da Web (Berners-Lee, 1989), propôs a Web Semântica (Berners-Lee et al., 2001) como uma evolução da Web atual para viabilizar a manipulação do conteúdo por aplicações (agentes de software) com capacidade de interpretar a semântica das informações. Assim, o conteúdo da Web, que não passava de informação sem significado para os computadores, pode ser interpretado por máquinas através do uso de ontologias, tornando a recuperação de informação na Web menos ambígua e fornecendo respostas mais precisas às consultas dos usuários.

Neste trabalho usamos o termo ontologia referindo-se a uma descrição explícita de conceitos e relações de um domínio de aplicação, incluindo o

vocabulário de termos da área e um conjunto de sentenças lógicas (axiomas) que expressam as restrições para interpretação deste vocabulário.

Ontologias são usadas para diversos fins, que vão desde dar apoio aos processos de desenvolvimento de software ou qualquer outro trabalho executado em equipes geograficamente distribuídas, até mesmo para dar apoio em tempo de execução aos sistemas de informação (Guarino, 1998).

O elemento chave para construção de ontologias é a definição de uma linguagem de ontologias. A linguagem deve ter semântica bem-definida, ser expressiva o suficiente para descrever inte-relacionamentos complexos e restrições entre os objetos, e ser capaz de manipular e inferir automaticamente com limites aceitáveis de tempo e recursos (McIlraith & Martin, 2003). O *World Wide Web Consortium*[3] (W3C), órgão que regula o desenvolvimento da Internet, orienta no desenvolvimento, organização e padronização de linguagens para promover a interoperabilidade entre aplicações na Web. Entre estas linguagens, temos: XML, RDF[4], DAML+OIL[5] e mais recentemente OWL[6].

Para atacar o problema da interoperabilidade usando ontologias, inúmeras soluções têm sido propostas na comunidade científica (Wache et al., 2001).

O SIMS (Arens et al., 1996), utiliza uma base de conhecimento hierárquica de termos, relacionando cada fonte de dados a ela e atuando como um esquema global para formulação das consultas dos usuários.

No sistema OBSERVER (*Ontology Based System Enhanced with Relationships for Vocabulary hEterogeneity Resolution*) (Mena et al., 2000) a semântica de cada fonte de dados é descrita por uma ou mais ontologias usando *Description Logic* (Baader et al., 2003). A vantagem desta abordagem, usando múltiplas ontologias, está em não precisar se comprometer com uma ontologia global, dando autonomia para que cada fonte possa construir sua própria ontologia. Por outro lado, a falta de um vocabulário comum torna difícil a integração das diferentes ontologias das diversas fontes. Para resolver este problema, foram definidos mapeamentos entre as ontologias para identificar os termos correspondentes. Essa solução pode se tornar ineficiente dependendo do número de fontes utilizadas.

Abordagens híbridas também foram propostas, utilizando múltiplas ontologias para descrever as fontes e um vocabulário global compartilhado.

[3] W3C - www.w3c.org
[4] RDF - http://www.w3.org/TR/rdf-primer/
[5] DAML+OIL - http://www.w3.org/TR/daml+oil-reference
[6] OWL - http://www.w3.org/TR/owl-ref/

Nestas abordagens as ontologias são construídas utilizando termos compartilhados do vocabulário global. Exemplos dessa abordagem são: COIN (Goh, 1997), MECOTA (Wache et al., 1999) e BUSTER (Stuckenschimidt et al., 2000).

Em (Hakimpour & Geppert, 2001) é apresentada uma abordagem para integração de esquemas de comunidades distintas que utilizam ontologias diferentes. Um esquema integrado é derivado da combinação (*merge*) das ontologias através dos relacionamentos de similaridade entre os conceitos. Este esquema integrado pode ser usado como o esquema global em sistemas de bancos de dados federados. A detecção das similaridades é feita por uma máquina de inferência usando definições intencionais dos termos representadas em *Description Logic*.

Em (Broekstra et al., 2004; Stuckenschmidt et al., 2004) é relatado o projeto DOPE (*Drug Ontology Project for Elsevier*) que faz uso de um tesauro[7] para auxiliar o acesso a dados de fontes distribuídas e heterogêneas. O tesauro utilizado no estudo de caso foi convertido para um formato em RDF Schema, e fornece um vocabulário compartilhado para indexação e agrupamento dos dados. Os dados são indexados e armazenados em um servidor de metadados. Estes metadados são transformados em arquivos RDF (mapeados para o modelo RDF) e armazenados no repositório Sesame[8]. O Sesame desempenha o papel central do protótipo do DOPE, atuando como mediador através da API *Storage And Inference Layer* (SAIL), encaminhando as consultas em SeRQL[9] às fontes relevantes.

Em (Tsai et al., 2003) é apresentado o *Smart Office Task Automation* (SOTA) *Framework*. O SOTA é um *framework* usando *Web Services*, ontologias em DAML+OIL, e agentes. O objetivo do *framework* é criar uma plataforma integrada de serviços de informação para fornecer, aos usuários de uma Intranet, o suporte a tarefas automatizadas. Para acessar as aplicações são criados adaptadores (*wrappers*) com interfaces utilizando *Web Services*. A plataforma SOTA usa uma ontologia de mediação para integrar as aplicações da Intranet, fornecendo uma interface integrada única aos usuários para acesso de diversas operações. Esta ontologia contém as definições dos conceitos abstratos das aplicações reais e seus relacionamentos. A arquitetura usa duas

[7] Um tesauro (do latim *thesaurus*) contém uma lista classificada de sinônimos (Fonte: WordNet Online 2.0 - http://www.cogsci.princeton.edu/cgi-bin/webwn).
[8] Sesame - http://www.openrdf.org/
[9] SeRQL Query Language - http://www.openrdf.org/doc/users/ch06.html

ontologias para homogeneizar o entendimento entre os *Web Services*: uma Ontologia de Operação modela os principais tipos de ações, e uma Ontologia de Recursos fornece o significado dos parâmetros de entrada e saída dos serviços. Para registrar os serviços é utilizado o *Web Service Semantic Registry* (WSSR). O WSSR armazena a semântica dos serviços extraída a partir do documento WSDL. Para isso é utilizada a ferramenta WSDL *Semantic Annotation* para anotar capturar os métodos e parâmetros de entrada e saída dos documentos WSDL, baseado nas ontologias de operações e de recursos.

2.5 Servidores de ontologias e catálogos de metadados

Com a emergente utilização de metadados e ontologias por sistemas, agentes de software e serviços como artefatos para apoiar suas funcionalidades, torna-se necessário que eles estejam acessíveis às aplicações e aos usuários. Para viabilizar isto, são utilizados repositórios para armazenar e gerenciar metadados e ontologias. A funcionalidade principal de um repositório é armazenar e disponibilizar estas meta-informações (metadados e ontologias) para viabilizar o entendimento compartilhado.

2.5.1 Servidores e serviços de ontologias

Diversos tipos de servidores de ontologias já foram propostos (Li et al., 2003; Pan et al., 2003; Duke & Patel, 2003; Suguri et al., 2001). Basicamente, um servidor de ontologias provê funcionalidades para armazenamento e consulta de ontologias. Em alguns casos fornece uma máquina de inferência que permite que as sentenças sobre os relacionamentos entre entidades em ontologias diferentes sejam recuperadas. Alguns servidores provêem acesso às ontologias através das URIs (*Unified Resource Identifiers*) de origem das ontologias, outros, por sua vez, materializam as ontologias em arquivos locais, outros, por sua vez, armazenam as ontologias em um repositório local, facilitando na manutenção e controle de versões.

McBride (2001) descreve o desenvolvimento da API Jena[10]. Jena é um *framework* para construção de aplicações para Web Semântica. Ela provê um ambiente para manipulação de RDF, RDFS e OWL, incluindo uma máquina de

[10] Jena API - http://jena.sourceforge.net/

inferência baseada em regras. A implementação da API Jena é código aberto e foi desenvolvida no projeto *HP Labs Semantic Web Programme* da *Hewlett-Packard Development Company*. Dentre outras funcionalidades, a API fornece meios para leitura e escrita de documentos RDF nos formatos: RDF/XML[11], N3[12] e N-Triples[13]; armazenamento em memória e em banco de dados dos modelos manipulados; e suporte à linguagem RDQL[14] (*RDF Data Query Language*) para consultas a documentos RDF.

Em (Broekstra et al., 2002) é apresentado o Sesame[15], um *framework* em Java para manipulação de documentos RDF e RDFS, oferecendo suporte ao armazenamento, consulta e inferência. O ponto central do *framework* é o repositório. A implementação do repositório do Sesame é independente de um sistema de armazenamento, por isso, pode ser implantado utilizando diversos sistemas (bancos de dados relacionais, armazenamento em memória, sistema de arquivos, etc.). O Sesame oferece uma API de acesso flexível, com suporte tanto para acesso local quanto remoto (via HTTP, SOAP ou RMI). As consultas podem ser feitas utilizando diversas linguagens.

O KAON (*The KArlsruhe ONtology and Semantic Web Tool Suite*) (Volz et al., 2003) é uma infra-estrutura para gerenciamento de ontologies, que inclui um conjunto de ferramentas para criação e gerenciamento de ontologias, e um *framework* para construção de aplicações baseadas em ontologias. O objetivo do KAON é assegurar a escalabilidade de inferência em grandes ontologias e bases de conhecimento.

Em (Suguri et al., 2001) é descrita a implementação de um serviço de ontologias baseado na especificação da FIPA[16] (FIPA, 2000). Para verificar e demonstrar a especificação e a implementação foi desenvolvida uma aplicação de compras on-line usando o serviço de ontologias e integrando múltiplos esquemas de bancos de dados. O Serviço de Ontologias da FIPA é basicamente um agente adaptador (*wrapper*) do OKBC[17] (*Open Knowledge Base Connectivity*). O OKBC é uma API que conecta aplicações a bases de conhecimento. Originalmente, o OKBC não fornecia interface de comunicação com agentes de software utilizando a linguagem de comunicação de agentes

[11] RDF/XML Syntax Specification - http://www.w3.org/TR/rdf-syntax-grammar/
[12] Notation3 (N3) - http://www.w3.org/DesignIssues/Notation3
[13] N-Triples - http://www.w3.org/TR/rdf-testcases/#ntriples
[14] RDQL - http://www.w3.org/Submission/RDQL/
[15] Sesame - http://www.openrdf.org/
[16] FIPA - http://www.fipa.org/
[17] OKBC - http://www.ai.sri.com/~okbc/

(*Agent Communication Language* – ACL) especificada pela FIPA. Por isso, a FIPA especificou um Agente de Ontologia (*Ontology Agent* – OA) para adaptar a API do OKBC, viabilizando a sua utilização por agentes de software. Os agentes clientes do OA comunicam-se com o OA para acessar o serviço de ontologias através de padrões especificados pela FIPA. Os clientes podem armazenar, alterar, remover ou consultar as ontologias através do OA.

Em (Duke & Patel, 2003) é apresentado o Servidor de Ontologias desenvolvido para a plataforma de agentes UKOLN para o projeto Agentcities.NET[18]. O projeto AgentCities foi desenvolvido para auxiliar pesquisas usando aplicações baseadas em agentes. O objetivo do projeto é construir uma rede de plataformas de agentes geograficamente distribuídas baseada nos padrões FIPA. Cada plataforma de agentes é chamada de "City" e hospeda populações de agentes que podem acessar os serviços uns dos outros. O Servidor de Ontologias da plataforma UKOLN é uma implementação do MEG Registry (Heery et al., 2002) com interface para agentes. A comunicação dos agentes com o Registro é mediada por um agente, chamado Agente Servidor. O MEG Registry é um repositório de documentos RDF baseado no Redland (Beckett, 2001), um conjunto de ferramentas para manipulação de RDF. O Agente Servidor comunica-se com o Servidor via HTTP.

Em (Li et al., 2003) é descrito o *AgentCities Ontology Service* (ACOS), um serviço desenvolvido usando a API Jena para armazenamento e consulta de ontologias descritas em DAML+OIL e XML. O ACOS introduz funcionalidades de inferência para verificação de consistência e oferece suporte ao desenvolvimento colaborativo de ontologias. A arquitetura do servidor expõe interfaces para acesso dos clientes (usuários, aplicações e agentes de software), oferecendo serviços de gerenciamento de ontologias e controle de acesso. A comunicação do servidor com os agentes é feita por intermédio do *Ontology Agent*, utilizando a linguagem ACL definida pela FIPA.

Em (Pan et al., 2003) é apresentado um protótipo do Repositório de Ontologias Otago. Para modelar a estrutura das ontologias armazenadas no repositório, foi desenvolvida uma meta-ontologia. Estes metadados são armazenados num servidor para ser consultados por usuários humanos e agentes de software. Um dos objetivos deste trabalho é permitir que agentes se comuniquem diretamente com o repositório usando o protocolo HTTP.

[18] AgentCities.NET - http://www.agentcities.net/

OntoRama (Eklund et al., 2002) é um editor de ontologias para a WebKB, uma base de conhecimento que contém uma hierarquia de tipos e sentenças. A base suporta um conjunto de construtores incluindo subtypeOf, partOf, memberOf, urlOf, entre outros. As funções principais do OntoRama incluem a pesquisa, comparação e modificação das ontologias. O OntoRama foi implementado usando Jena e armazena ontologias em RDF.

SchemaWeb[19] é um repositório para esquemas descritos nas linguagens RDFS, OWL e DAML+OIL que permite o acesso de usuários humanos, agentes de software e aplicações. Para usuários é fornecida uma interface Web onde podem ser realizadas consultas, buscas por palavra chave, submissões de esquemas para inclusão no repositório e visualização das classes e propriedades dos esquemas. Agentes de software e outras aplicações podem acessar o SchemaWeb via *Web Services*, REST[20] e SOAP[21].

2.5.2 Serviço de catálogo de metadados no Grid

O ambiente de Grid Computacional foi projetado para oferecer suporte a aplicações que produzem e manipulam grandes conjuntos de dados. Para dar suporte a estas aplicações, é necessário disponibilizar serviços que suportem o registro e consulta de informações a respeito dos metadados.

Em (Singh et al., 2003) é apresentado o projeto de um Serviço de Catálogo de Metadados (*Metadata Catalog Service* – MCS). O MCS fornece um mecanismo para armazenamento e acesso de metadados, permitindo aos usuários consultar os dados a partir de seus atributos descritivos.

Este serviço prevê o uso de diferentes serviços no Grid para tratar dos diferentes tipos de metadados utilizados.

2.5.3 Catálogo de feições do OpenGIS

O OpenGIS Consortium[22] (OGC) é um consórcio internacional que reúne empresas, agências governamentais e universidades com o objetivo de criar especificações para a indústria de geoprocessamento, fornecendo os requisitos

[19] Schema Web - http://www.schemaweb.info
[20] REST - http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
[21] SOAP - http://www.w3.org/TR/soap/
[22] OpenGIS Consortium – http://www.opengis.org

técnicos mínimos para promover a interoperabilidade entre produtos e serviços que implementem estes padrões.

Em (Nebert & Whiteside, 2004) são especificados os modelos abstratos e os modelos de implementação necessários para publicar e acessar catálogos digitais de metadados para dados e serviços geoespaciais.

O modelo de interface de catálogo especificado pelo OGC fornece um conjunto de interfaces abstratas para serviços, oferecendo suporte a descoberta, acesso, manutenção e organização de catálogos de informações geoespaciais e recursos relacionados. As interfaces especificadas devem permitir aos usuários ou aplicações localizar informações existentes em múltiplos ambientes computacionais distribuídos, incluindo a Web.

Um serviço de catálogo oferece suporte à publicação e busca de coleções de informações descritivas (metadados) sobre dados, serviços e objetos de informação. Os metadados atuam como propriedades generalizadas que podem ser consultadas e requisitadas por humanos e aplicações. Eles são consultados através dos serviços de catálogos. Os serviços de catálogo suportam o uso de uma ou mais linguagens de consulta para localizar e retornar os resultados usando os esquemas dos metadados armazenados.

A Figura 5 mostra a arquitetura de referência especificada pela OGC para desenvolvimento dos serviços da interface do catálogo. A arquitetura é uma organização multicamadas de clientes-servidores. A aplicação cliente interage com o serviço de catálogo usando a interface de catálogo definida pela OGC. O serviço de catálogo pode fazer uso de uma das três fontes para responder a requisição: um repositório de metadados local ao serviço, um serviço de recursos, ou outro serviço de catálogo.

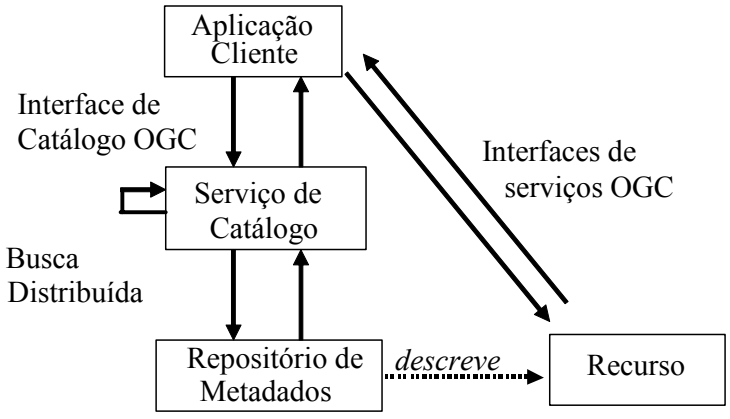

Figura 5 – Arquitetura do modelo de referência.

Para compartilhar informações com a comunidade, um esquema de metadados deve ser definido, fornecendo um vocabulário comum para suportar pesquisa, recuperação, descoberta e associação entre as descrições e os objetos que estão sendo descritos. As coleções de dados que desejam fazer parte de um catálogo devem seguir os padrões de metadados deste catálogo.

A OGC define propriedades de metadados gerais para caracterizar qualquer recurso. O objetivo é definir propriedades consultáveis para viabilizar que as mesmas consultas possam ser executadas contra quaisquer serviços de catálogos sem necessidade de alterações e sem conhecimento do modelo utilizado pelo catálogo.

2.6 *Dataweb*: infra-estrutura para compartilhamento de dados na Web

Independente da abordagem adotada para interoperar sistemas e compartilhar informações, os dados compartilhados devem ser disponibilizados de alguma forma para tornarem-se acessíveis. Neste contexto, foram criados novos padrões para viabilizar a criação de uma Web de dados, chamada Dataweb.

Publicado pela OASIS[23] em (Reed & Strongin, 2004), o XRI[24] (*eXtensible Resource Identifier*) é um novo protocolo para identificação abstrata de recursos. A idéia é que estes identificadores sejam independentes de aplicação, localização e transporte, e possam ser compartilhados entre inúmeros domínios e diretórios. O uso do XRI viabiliza a identificação, descrição e o versionamento dos dados, tornando possível a criação do formato de intercâmbio universal chamado XDI[25] (XRI *Data Interchange*).

O XDI é um protocolo para compartilhamento de dados distribuídos e mediação usando XRIs. O objetivo do XDI é criar um formato universal para compartilhamento de dados. Assim, dados XML provenientes de qualquer fonte de dados, podem ser identificados, compartilhados, conectados e sincronizados numa Web de dados, chamada Dataweb. A Dataweb será entendida por máquinas, de forma análoga à Web atual, feita para humanos, onde páginas HTML provenientes de diversas fontes de conteúdo estão conectadas. A Tabela

[23] OASIS - http://www.oasis-open.org/
[24] OASIS XRI Technical Committee - http://www.oasis-open.org/committees/xri
[25] OASIS XDI Technical Committee - http://www.oasis-open.org/committees/xdi

2 mostra os padrões da Dataweb em comparação aos padrões da Web (Reed et al., 2004).

Tabela 2 – Web versus *Dataweb*.

Padrão	Web	Dataweb
Endereçamento	URIs	XRIs
Representação dos dados	HTML	XML/XDI
Intercâmbio de dados	HTTP	XDI/HTTP, XDI/SOAP

Em virtude do controle necessário para mediar o acesso e o uso de dados compartilhados estarem nos links XDI, o desenvolvimento da Dataweb pode fornecer a infraestrutura para compartilhamento de dados distribuídos ideal para o desenvolvimento do potencial total da Web, da Web Semântica e dos *Web Services* (XDI.ORG, 2004).