Capítulo 2

Interoperabilidade de Banco de Dados e Sistemas Heterogêneos

A criação de ambientes que permitam um compartilhamento controlado e troca de informação entre banco de dados autônomos e heterogêneos foi proposto em [SSU90] como uma das áreas chave para as futuras pesquisas em banco de dados. Para se entender a complexidade da área, vários pontos devem ser apreciados. Dentre eles, este capítulo aborda aspectos e problemas encontrados na geração de ambientes de interoperação e integração. Este capítulo inicia-se com as definições relacionadas à interoperabilidade de sistemas de banco de dados heterogêneos. Em seguida são discutidos os vários problemas a serem resolvidos na integração de sistemas heterogêneos. Aspectos referentes à interoperabilidade são tratados na seção 2.3 assim como as arquiteturas e metodologias propostas para a integração. Em seguida, é feita uma classificação dos sistemas de integração e são mostrados exemplos de sistemas e suas características. Finalmente são abordados alguns dos trabalhos relacionados com o tema e tendências na área de pesquisa.

2.1 Conceitos

Os Sistemas de Gerenciamento de Banco de Dados (SGBD) foram propostos na intenção de compartilhar recursos sobre uma rede de informações com múltiplas aplicações autônomas. Os sistemas de arquivos, anteriores aos SGBD, funcionavam de maneira autônoma e produziam efeitos colaterais dessas autonomias, como: duplicação de dados, diferenças de tipos de dados e estruturas, difícil administração e muitos outros.

A abordagem centralizada surgiu como proposta para sanar as dificuldades da autonomia das aplicações fazendo com que fosse criado um ambiente centralizado onde residiam todas a informações sobre o sistema. Dessa forma, a administração do sistema foi reduzida a um único local. Apesar das facilidades advindas com a centralização, surgiram pontos críticos como o gargalo administrativo e o risco de uma falha geral no sistema, visto que todos os recursos do banco de dados residem em um único local. A noção de SGBD trouxe a idéia de que as aplicações têm autonomia sobre o banco de dados, causando a ilusão de uma única aplicação acessando o sistema.

Atualmente, muitas empresas compartilham informações com seus afiliados e provavelmente cada um possui um Sistema Gerenciador de Banco de Dados. Obviamente busca-se uma interoperação e distribuição da informação, onde se possa compartilhar dados de estoque, planilhas de custos, gráficos de desempenhos e outros. No nível físico, as redes de computadores oferecem uma boa infraestrutura de distribuição e a cada dia vê-se o avanço das telecomunicações no sentido de interligar os sistemas mais distantes e heterogêneos. O problema agora é intermediar a distribuição no nível lógico, ou seja, das aplicações e dos seus SGBD. O desafio da integração é interligar os diferentes SGBD.

No processo de integração na *web* podem ser utilizadas abordagens virtuais ou materializadas. As abordagens variam na forma como as informações são acessadas no esquema integrado. Na virtual, as informações são recuperadas das origens de dados à medida que o sistema recebe requisições sobre o seu esquema integrado. Já na abordagem materializada, o sistema consulta um repositório resultante da integração das fontes de dados.

Dentre as arquiteturas para sistemas de integração, tem-se esquemas globais e locais [Lev00], sistemas federados [LMR90], sistemas de mediadores [Wie92] e de *data warehouse* [Inm92]. Os esquemas globais são visões que refletem os objetos dos bancos de dados utilizados, portanto geram um esquema lógico integrado onde os usuários acessam sem ter a noção da localização física do SGBD. A construção de esquemas globais pode não ser uma tarefa fácil, pois requer conhecimento sobre os esquemas dos SGBD e por dificilmente tratar o crescimento do próprio esquema.

Nos sistemas federados, ocorre uma cooperação entre os participantes da federação no sentido de compartilhar seus objetos de banco de dados sem perder o controle sobre os mesmos. Um conjunto de múltiplos bancos de dados autônomos gerenciados sem esquema global é chamado de Multibanco de dados (*multidatabase*) ou *banco de dados interoperavél* [LMR90]. Os sistemas que gerenciam estes Multibanco de dados são comumente chamados de sistemas federados ou sistemas de Multibanco de dados.

Os sistemas de mediadores fazem parte de uma arquitetura em camadas onde no mais baixo nível estão as fontes de dados, na mais alta os sistemas que oferecem acesso ao sistema de integração, e na camada intermediária a entidade que resolve o processo de integração das fontes de dados da camada inferior. Na abordagem materializada do *data warehouse*, ocorre a concentração de informação integrada de várias fontes em um repositório

Parte dos problemas de integração dirige-se à heterogeneidade entre os sistemas participantes. Relacionada a essa redundância está a heterogeneidade de nomes, tipos de valores e estrutura de dados. Comumente, sistemas de interoperabilidade possuem uma linguagem comum de acesso ao esquema. A proposta desta linguagem de manipulação ultrapassa os métodos procedurais em razão de torná-la mais próxima da linguagem humana. Neste caso, poder-se-ia, com uma única consulta, acessar diferentes bases de dados dentro de uma interface comum aos sistemas [JR97].

2.2 Heterogeneidade dos Sistemas

Existem vários aspectos que fazem de um conjunto de sistemas um corpo heterogêneo. Quando há uma comparação dos esquemas de vários bancos de dados, notadamente percebem-se os conflitos de esquemas, os quais representam a não congruência de tipos de dados, nomes e representações do mundo real. Os mais comuns são listados abaixo [KS91]:

a) *Conflitos de generalização*: são aqueles onde um atributo ou classe assume o valor de muitos atributos ou representa várias classes em outro banco de dados. Por exemplo, o campo `endereço` em uma tabela de um banco de dados pode ser visto em outro banco de dados como um conjunto dos atributos `rua`, `bairro` e `número`;

b) *Conflitos de tipos de dados*: são aqueles onde um tipo de dados de um atributo difere do tipo de dados do atributo com o mesmo significado em outro banco de dados. Por exemplo, o tipo de dados de um atributo `CPF` pode ser numérico para um sistema enquanto noutro seu tipo é alfanumérico;

c) *Conflitos estruturais*: são aqueles onde uma entidade em um sistema é modelada como atributo em outro. O atributo `vendedor` pode ser representado como um atributo em uma tabela enquanto que em outro sistema pode ser uma entidade;

d) *Perda de atributos*: é o caso onde não existe o atributo em outro esquema. Pode ser necessário um atributo `idade` para uma tabela `estudante` de um sistema, enquanto que em outro não seja necessário;

e) *Conflitos de nomes*: é o caso onde as entidades de diferentes sistemas têm o mesmo significado, porém são referenciadas com nomes diferentes;

f) *Conflitos de Escala*: dizem respeito à unidade apropriada para o valor do atributo. Existem sistemas que armazenam valores de moeda em dólares e outros em iene. Apesar de representarem valores para um mesmo atributo (por exemplo, `salário`), possuem valor monetários distintos;

Além das diferenças de esquemas apresentadas anteriormente, pode-se classificar a heterogeneidade de sistemas de forma que as diferenças que denotam a heterogeneidade são

agrupadas em níveis que residem no nível físico (*hardware*) até o lógico [Wie92], como ilustra a figura 2.1. Atingir um nível de integração que resolva a maioria dos problemas encontrados na área ainda é um desafio. Porém o uso de várias técnicas de integração, como o uso de mediadores e suas variações, tem atingido objetivos satisfatórios.

A *heterogeneidade de hardware e de sistemas operacionais* está relacionada com a criação de novos dispositivos e Sistemas Operacionais. Embora a maioria das grandes corporações adote o uso de padrões na criação de novos dispositivos, existem esforços no sentido de permitir que sistemas operacionais e dispositivos de massa tenham uma interface de comunicação padronizada. O uso de APIs (*Application Program Interface*) e as organizações regulamentadoras têm mostrado bons resultados neste nível de heterogeneidade.

A *heterogeneidade de modelos organizacionais* refere-se aos diversos modelos de sistemas de bancos de dados existentes. Os progressos neste nível enfocam mapeamentos dos modelos relacionais, orientado a objetos, hierárquicos, rede e outros modelos derivados.

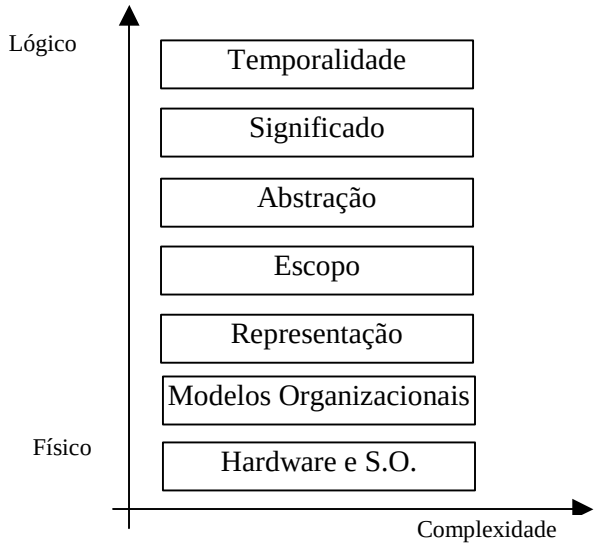

Figura 2.1: Sistema de Integração e Interoperação e suas características.

A *heterogeneidade de representação* refere-se a um dos problemas de natureza semântica existentes. Geralmente ocorre quando tamanhos de campos são diferentes ou possuem tipos semelhantes, mas com representações internas diferentes. Por exemplo, um atributo `cep` com cinco dígitos e em outro sistema com representação de oito dígitos.

A *heterogeneidade de escopo* baseia-se nos diferentes domínios dos objetos de um sistema. Geralmente pode-se ter entidades que representem uma porção do mundo real. Cada sistema, de acordo com as regras do negócio implementado, possui representações de um subconjunto de uma entidade do mundo real. Portanto, a integração de tais sistemas deve gerar uma visão resultante que corresponde a uma entidade mais abrangente. A geração da visão resultante normalmente se dá com operações de junção (JOIN) da álgebra relacional (quando se deseja a associação das entidades) ou ainda uma variação da junção denominada OUTER JOIN. A operação JOIN denota o produto cartesiano entre as relações envolvidas. Porém, alguns atributos podem não ser associados no processo de JOIN por não serem equivalentes nas relações participantes da operação. O processo de OUTER JOIN é semelhante ao JOIN com a característica de exibir atributos que não têm relacionamento com as entidades participantes da operação. Assim os atributos que não pertencem a uma relação são criados com valores nulos e a aplicação deverá tratar tais casos.

Relacionado ao nível de detalhamento do domínio das relações está a *heterogeneidade de abstração*. Esta consiste no nível de detalhamento dos atributos que cada entidade possui no sistema de integração.

A *heterogeneidade de significado* é bem mais complexa e denota a dificuldade em determinar o significado dos atributos com equivalência de nomes. Por exemplo, o atributo `nome` em uma relação `pessoa` denota seu verdadeiro nome na vida real, em contrapartida, em uma entidade `departamento` o mesmo atributo tem significado diferenciado.

Por causa da autonomia e independência dos sistemas, quando ocorrem atualizações assíncronas, podem ocorrer resgates de informações desatualizadas. Seja um sistema que possua atualização feita uma vez por semana e os demais que fazem parte do sistema integrado, com atualizações online. Pode ser gerada uma visão integrada de relações onde atributos semelhantes aos sistemas possam estar atualizados associados com atributos que ainda não sofreram atualizações. Desse modo, a visão apresentaria informações com temporalidades diferentes. Estes efeitos indesejáveis são dirigidos à *heterogeneidade de tempo*.

2.3 Interoperabilidade de Banco de Dados

Há uma necessidade grande em interligar sistemas e como conseqüência disto, ocorrem os problemas discutidos na seção anterior. Com o aumento da quantidade de informação na *Web*, a interligação de sistemas através das redes de computadores e o uso de potentes SGBD nos diversos pontos da rede, torna-se claro que um processo de automação de recuperação das informações seja disponibilizado. No processo de automação da integração dos sistemas, muitas das técnicas para resolver a heterogeneidade semântica ou lógica de um sistema de integração baseiam-se em arquiteturas ou *frameworks* derivados dos conceitos básicos de SGBD. Além disso, para diminuir a heterogeneidade, tecnologias de modelos globais com representações dos diversos sistemas que compõem o ambiente; ambientes de integração que fazem do processo uma rica interface onde usuários criam esquemas para a interoperação de sistemas como se fossem programadores; padrões da indústria do software como: JDBC, ODBC e outros, os quais ajudam no mapeamento do modelo global à fonte de dados; e finalmente a disposição do sistema integrado sobre uma interface de uso comum e largamente utilizada na Internet são de fundamental importância para a redução da heterogeneidade e o crescimento da interoperação dos sistemas [BFM97].

A interoperabilidade tem agregado valor à informação resgatada de múltiplas fontes de dados. Neste domínio, múltiplas fontes de dados tornam-se um campo bastante amplo em razão da autonomia e da diversidade dos sistemas disponibilizados, podendo ser desde banco de dados até documentos XML. O fato de criar sistemas de interoperabilidade e integração de sistemas simplesmente unindo as partes desejadas, não reduz um dos principais problemas que atingem a *Web*: a sobrecarga de informação. É necessário que a *Web* tenha um mínimo de estrutura, expresse relacionamentos e ultrapasse o conceito de grande repositório de dados. Sem estruturas não se torna fácil o processo de indexação e catalogação, e em conseqüência as consultas tornam-se cada vez mais ineficientes.

Adicionando inteligência no processo de integração e interoperação, as informações poderiam tornar-se mais valorosas para os usuários e finalmente as consultas na

Web tornar-se-iam semelhantes às de banco de dados. É provável que a *Web* semântica resolva o problema, porém ainda é trabalho de pesquisa [WWW1].

2.3.1 Metodologias de Interoperabilidade

A maioria das metodologias de interoperabilidade são baseadas em arquiteturas ou *frameworks* que oferecem uma forma genérica para atingir a interoperação. Apesar da possibilidade de construção de sistemas personalizados que dêem suporte à interoperação, as arquiteturas de interoperabilidade são variações de técnicas básicas dos SGBD. As arquiteturas utilizam visões, modelos de dados e interfaces, além de novas implementações de técnicas de atualização de dados e recuperação de informações. Nas subseções seguintes são discutidas algumas arquiteturas que suportam a interoperabilidade.

2.3.1.1 Federação de Banco de Dados

Uma federação de Banco de Dados é uma coleção integrada de banco de dados autônomos, a qual os componentes de administração mantém controle total sobre os seus sistemas individuais, mas cooperam com a federação através do suporte a operações globais [SL90]. Cada participante da federação congrega dados que estão presentes no restante dos membros da federação. Esses dados são utilizados para suportar uma visão virtual das informações de cada banco de dados federado.

Os dados disponibilizados para outros participantes da federação são também descritos por esquemas ou modelos de exportação. Tal modelo agrupa dados do completo modelo conceitual da federação ou é um subconjunto dele. A integração ocorre com o agrupamento destes esquemas de exportação dos diversos participantes da federação e a união de dois ou mais esquemas chama-se esquema ou modelo de importação. Esses modelos de importação são acessíveis pelos usuários do sistema federado. Os esquemas de importação são gerados através de operadores de derivação [LMR90].

As informações relacionadas com a federação em si são armazenadas em um dicionário global. As consultas sobre a federação são submetidas aos operadores de derivação que criam os esquemas de importação e através de um protocolo definido faz a intermeadiação entre os diversos bancos de dados autônomos integrantes [HD87]. A figura 2.2 ilustra o modelo de arquitetura empregado em ambientes federados.

Há uma classificação para os Sistemas de Banco de Dados Federados que se baseia no modo como os usuários interagem com a federação: os fortemente acoplados e os fracamente acoplados. Os Sistemas de Banco de Dados Federados Fortemente Acoplados possuem uma visão integrada estática formulada pelo administrador da federação. O termo fortemente acoplado refere-se exatamente à visão estática e à impossibilidade dos usuários modificarem os esquemas de importação. Ademais, as diferenças semânticas e de esquemas dos federados são resolvidas pelo administrador. Como exemplos destes sistemas temos o Multidatabase [DH84] e Pegasus [Ahm+91].

Já os Sistemas de Banco de Dados Federados Fracamente Acoplados possuem um sistema de integração dinâmica. Neste tipo de sistema os usuários interagem com os objetos do banco de dados. O processo de interação é efetuado por meio de linguagens de manipulação que têm o poder de criar os esquemas característicos da federação. O termo fracamente acoplado reside na propriedade que o sistema tem em se adaptar aos anseios dos usuários. Por causa disto, os participantes da federação não possuem ligações rígidas uns com os outros. Embora os usuários tenham que obter conhecimento da linguagem de manipulação para a criação dos esquemas, os sistemas fracamente acoplados possuem um alto grau de escalabilidade e grande adaptabilidade. O MRDSM [L+89] é um exemplo de sistema fracamente acoplado.

Embora em muitas referências o termo multibanco de dados [L+89] seja tratado como um sistema federado, nota-se que o segundo é uma evolução do primeiro. As diferenças entre os dois acentuam-se mais na nomenclatura de seus componentes do que na funcionalidade das partes da arquitetura. Em relação ao multibanco de dados, a Federação de Banco de Dados adiciona o compartilhamento da cooperação entre os bancos de dados participantes da federação permitindo que uma definição de interoperação seja visível por

todos os usuários. Na prática, os termos Sistema Federado de Banco de Dados, Sistemas multibanco de dados, Sistemas de Banco de Dados Interoperáveis e SGBD Distribuído Heterogêneos são equivalentes e referem-se ao acesso de múltiplos banco de dados autônomos sem o uso de um esquema global físico.

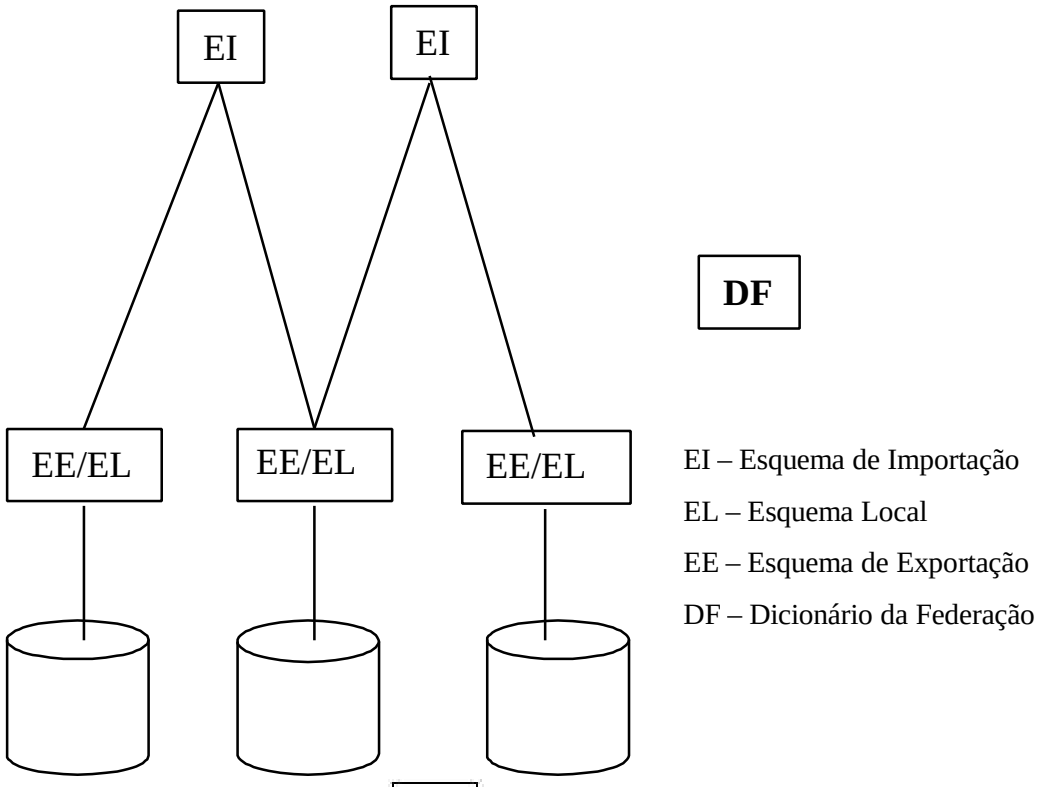

Figura 2.2: Arquitetura de uma Federação de Banco de Dados

2.3.1.2 Sistema de Mediadores

Um Sistema de Mediadores é uma coleção de fontes de informação que são integradas para oferecer uma interface uniforme somente de leitura para usuários finais, e um conjunto de ferramentas para executar as tarefas de integração [JR97]. Apesar do conceito referir-se a uma interface somente de leitura, extensões da arquitetura foram criadas para suportar atualizações de visões globais como nos Mediadores de Integração do SQUIRREL [HZ96].

Esses sistemas possuem módulos que fazem a mediação entre usuários finais e as fontes de dados. Os módulos são conhecidos como Mediadores. Um mediador é um módulo do programa que explora o conhecimento representado em um conjunto ou subconjunto de dados para gerar informações para aplicações residentes em uma camada superior [Wie92].

A arquitetura dos Sistemas de Mediadores contempla o particionamento de recursos e serviços em duas dimensões, como mostra a figura 2.3 [Wie99]. As camadas na horizontal contribuem para a integração das fontes de dados através de chamadas a funções bem definidas. A camada superior, chamada de camada de aplicação, congrega a logística das operações de integração. Nesta camada o usuário desenvolve uma decisão final sobre o negócio desejado.

Na camada intermediária, chamada de mediação, ocorre o processo de derivação das consultas submetidas na camada de aplicação. O processo de derivação consiste de módulos que procuram as fontes de dados e processadores de consultas que reformulam a consulta inicial para facilitar o acesso aos dados. Após a etapa de processamento das consultas é necessária a integração dos resultados para que seja enviada uma resposta adequada para os usuários. O mediador pode ainda recorrer a regras que possam reduzir a redundância de dados. Tais regras são do tipo: “A fonte de dados A é preferível à fonte B” ou ainda “dados mais recentes são preferíveis” [Wie99].

Na camada inferior têm-se as fontes de dados que podem ser um conjunto homogêneo ou heterogêneo de bancos de dados e fontes de dados semi-estruturados.

Para que se obtenha os resultados desejados, nos Sistemas de Mediadores existem usuários especializados nas tarefas sobre as camadas ilustradas na figura 2.3. Cada qual atua num segmento específico e isto não leva a crer que há a necessidade de três indivíduos, mas de conhecimentos pertinentes a cada camada. Nos Sistemas de Mediadores estão envolvidas três categorias de usuários [PMU96]:

a) Usuário final: É aquele que submete a consulta para o sistema e conclui com uma decisão final. É o usuário que atua na camada de aplicações;

b) Integrador de domínios: é aquele que procura resolver as heterogeneidades do sistema global através de um modelo comum de dados. É o usuário que atua na camada de fonte de dados e geralmente é o administrador de banco de dados;

c) Criador de Mediadores: É o responsável pela integração das fontes de dados traduzindo as fontes de dados em uma visão global comum.

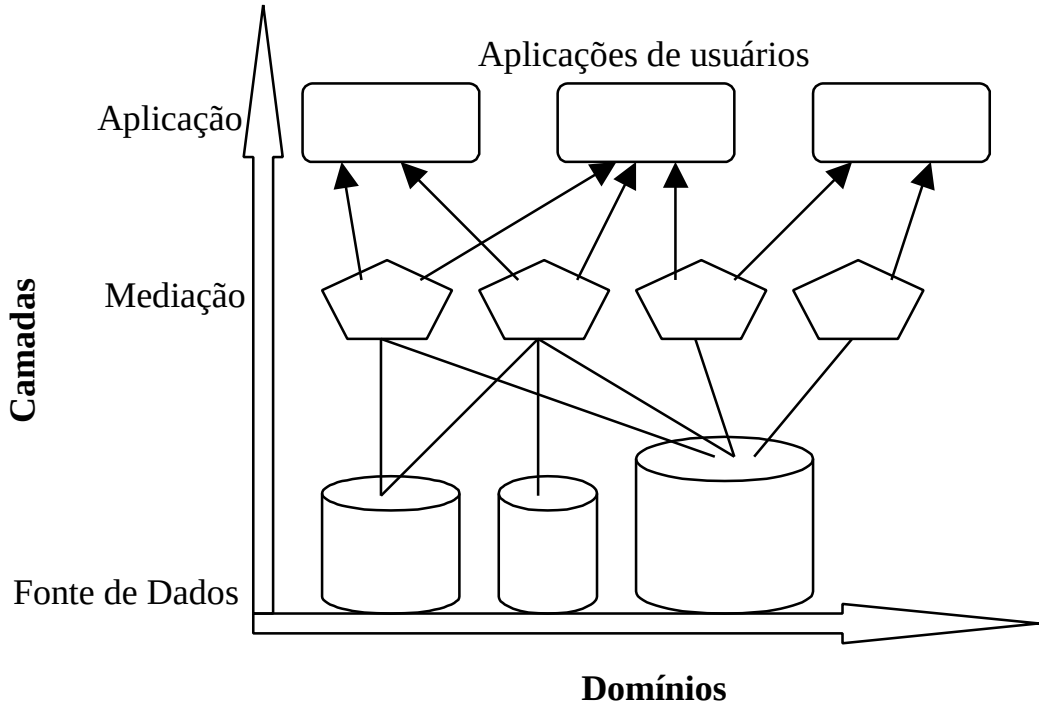

Figura 2.3: Arquitetura de Sistema de Mediadores

O esquema produzido da camada intermediária é feito a partir de métodos virtuais, materializados ou um composto deles. O método virtual, inicialmente proposto em [LMR90], baseia-se em um modelo de decomposição de consultas, onde cada parte decomposta é uma subconsulta para a fonte de dados específica, resultando em uma fusão de todas as subconsultas derivadas para produção de um resultado. O método materializado, comercialmente denominado *data warehouses* [IK93], consiste em criar um modelo global derivado das fontes de dados que reflita a integração das diversas fontes de dados. Desta maneira, as consultas são submetidas ao *data warehouse* sem acessar diretamente as fontes de dados. Com a implantação do método materializado, ocorre o problema da atualização no

repositório do *Data warehouse*. Algumas técnicas de Banco de Dados Ativos têm sido utilizadas na manutenção do repositório, como também técnicas de atualização incremental e de replicação de dados.

2.3.1.3 Sistemas de Workflow

Sistemas de Workflow enfatizam a coordenação e cooperação dos usuários, e claramente dos repositórios de dados que participam da cooperação. A cooperação se dá de maneira transparente, pois a ferramenta de workflow fornece uma interface que omite a localização da fonte de dados. O Sistema de Workflow é definido como sendo uma execução coordenada de um conjunto de tarefas por diferentes entidades de processamento, seja em um processo manual ou automatizado.

Os Sistemas de Workflow podem ser baseados em técnicas de Redes Petri e grafos direcionados. Cada nó da rede atua cooperando com a execução do processo. Muitas vezes segue uma ordem de execução em cada nó da rede ou a distribuição das tarefas são feitas de forma a obter resultados simultâneos em diversos nós da Cooperação. Um bom exemplo é o INTERFYS [SU99], que utiliza técnicas de workflow para integração com interface pela *web* sem o uso de gerenciadores de workflow.

2.3.1.4 Sistemas Declarativos

Os sistemas que utilizam a lógica na descrição do sistema de integração fazem parte de um grupo que utiliza técnicas de Inteligência Artificial, regras de primeira ordem e uma linguagem muito intuitiva para a criação de ferramentas de interoperação e integração de dados. São sistemas que incorporam o uso de *middleware* para mediar o acesso às fontes de dados e linguagens declarativas para especificar as conversões de dados dos sistemas heterogêneos que compõem o sistema.

Information Manifold [LRO96] e o YAT [CDS+98] estão incluídos neste grupo de sistemas. Eles utilizam métodos semelhantes aos sistemas de mediadores, porém admitem uma única visão global dos dados. O YAT[1] utiliza a poderosa linguagem YATL (YAT *Language*) para resolver a heterogeneidade. A YATL é uma linguagem baseada em regras onde são utilizadas declarações avançadas de casamento de padrões, reestruturação de primitivas e funções *Skolem*, que tratam da criação e uso de novos identificadores [CDSS98].

2.3.1.5 Sistema de Repositórios de Metadados

Os Repositórios de Metadados são dicionários que agregam as informações disponibilizadas no sistema de integração e interoperação. Neste caso, quando um usuário solicita uma informação, ocorre uma consulta no repositório. Deste modo, os usuários não acessam diretamente as fontes de dados do Sistema. Os repositórios armazenam metadados sobre as diversas fontes de dados e são úteis na verificação semântica das informações envolvidas na integração.

Os repositórios são somente leitura e não permitem atualizações. As heterogeneidades são resolvidas pelo próprio usuário com o auxílio de uma interface gráfica e de assistentes de integração inteligentes. Diferentemente dos modelos anteriores, o repositório é criado parcialmente pelo próprio sistema. Assim, os mapeamentos entre as fontes de dados e a visão do sistema de integração são feitos parcialmente. O Metadatabase [CC96] é um sistema que se enquadra nas características dos Sistemas de Repositórios de Metadados. Ele possui uma linguagem de consulta ao repositório chamada MQL e caracteriza-se por uma interface visual de formação de consultas com o apoio de assistentes.

[1] Yet Another Tree-Based System.

2.3.2 Classificação para Integração

O objetivo da extração de informação e técnicas de integração é construir uma descrição sintetizada e uniforme das informações extraídas de múltiplas fontes de dados [BCV+99]. De acordo com aspectos apresentados anteriormente, existem dois grandes grupos de metodologias para alcançar a integração. Essa divisão parte de como a tarefa de resolução dos problemas da integração é resolvida e em que nível é solucionado. Desde modo, se encaixa a classificação: sistemas de integração estáticos e sistemas de integração dinâmicos.

2.3.2.1 Integração Estática

Neste grupo alojam-se sistemas que resolvem os problemas de integração de esquema e semântico na inserção de um novo componente de banco de dados. O termo estático está relacionado à impossibilidade de modificação do sistema na inserção de uma nova fonte de dados. Também em virtude do nível muito baixo de automação nos níveis de mediação e do *wrapper*.

O papel de resolução da heterogeneidade é praticado no nível de usuários especializados na integração de domínios (novas fontes de dados). Quando uma nova fonte de dados deseja participar do sistema, o integrador avalia os objetos necessários e os disponibiliza na visão global do sistema de integração. Entretanto, com a adição da nova fonte de dados, um novo intercâmbio de informações entre os componentes do esquema local, ao qual pertence o recém participante do sistema de integração, e o esquema global do sistema deve ser definido. Logo, como resultado do processo de integração, são produzidos *wrappers* para acesso às novas fontes de dados.

Os *wrappers* disponibilizam componentes do esquema local que são acessíveis pelo mediador e mais adiante pelos usuários finais. Este processo procura esconder dos usuários finais todo o caminho percorrido das diversas fontes de dados participantes do sistema de integração até a interface do usuário.

A partir dos componentes de esquemas gerados pelos *wrappers*, são criados esquemas integrados pelos mediadores, que também têm a tarefa de resolver os problemas das heterogeneidades. A maioria dos esquemas gerados é somente leitura e por não serem definidos pelos usuários finais são ditos como estáticos. Ou seja, o papel do integrador de domínios é definir um esquema que atenda às definições do sistema de integração. Desse modo, é feita uma intersecção entre os objetos dos participantes da integração, formado um esquema integrado acessível pelos usuários finais.

Todo o processo de integração é chamado de *processo de integração de sistemas em dois estágios*, ou seja, um estágio feito pelos *wrappers* e outro pelos mediadores. Os *wrappers* mantêm contato com as fontes de dados e fornecem um esquema visível pelos mediadores. Os mediadores utilizam os esquemas gerados pelos *wrappers* e os agrupa, resultando na integração dos esquemas e na resolução dos conflitos da integração. A figura 2.4 ilustra a arquitetura básica deste tipo de integração.

Os sistemas federados fortemente acoplados se enquadram nos modelos de integração estáticos. Trabalhos têm sido feitos no sentido de prover alguma automatização no processo de duas fases de integração, como em [Dav+99].

2.3.2.2 Integração Dinâmica

Diferentemente dos sistemas estáticos, os usuários finais dos sistemas de integração dinâmica têm acesso aos esquemas das fontes de dados gerados pelos *wrappers*. Através de uma interface, tais usuários formulam suas consultas e as submetem para um acesso direto a esses esquemas. A interface é uma ferramenta visual para facilitar a interação com o sistema e permite que o usuário manipule os objetos do sistema de integração como se fossem objetos locais.

Para a construção de consultas, uma interface e um dicionário de metadados estão disponíveis para os usuários. Na verdade, a interface gráfica é uma facilidade provida pelo sistema para a criação das consultas na linguagem de consultas definida no sistema de

integração, como MQL [CC96] do Metadatabase e OEM-QL [PMU96] do projeto TSIMMIS. Além de facilitar o acesso aos objetos do sistema, a interface gráfica utiliza informações do dicionário de metadados para proceder com a verificação semântica e resolução de heterogeneidades.

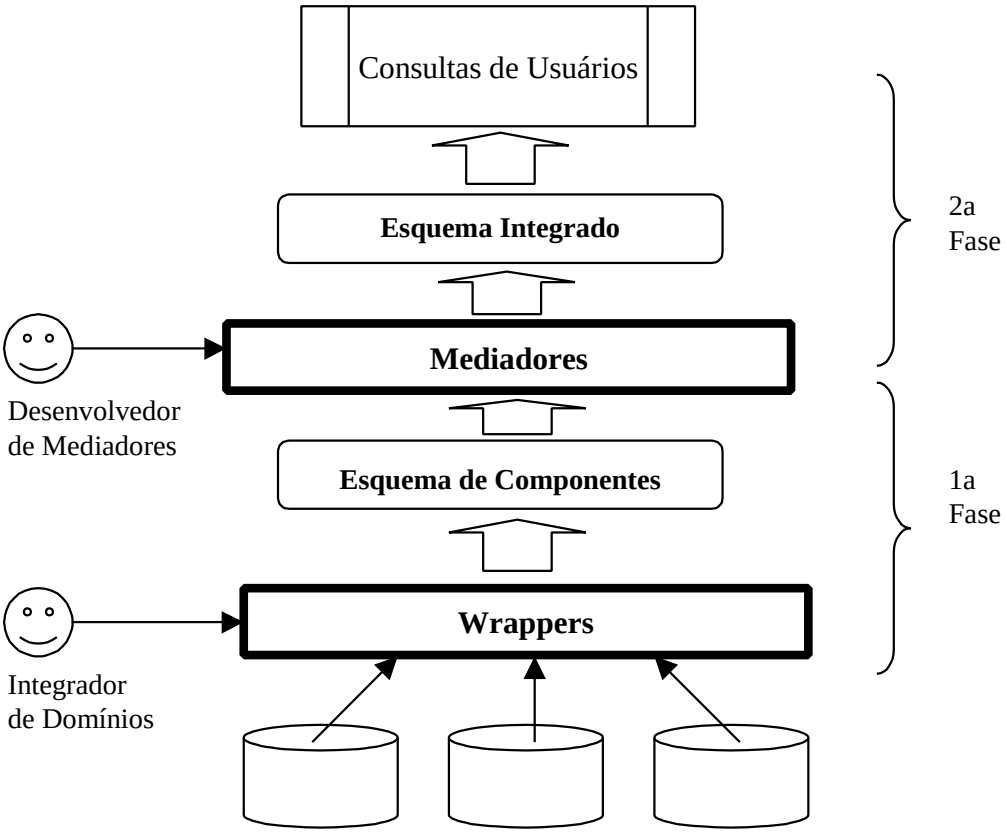

Figura 2.4: Arquitetura de um sistema de Integração Estático

Uma parte das tarefas de resolução de heterogeneidades fica a cargo dos usuários finais e por isso há uma necessidade de maior conhecimento sobre o processo de integração. Nos sistemas dinâmicos o segundo passo do processo de integração é disseminado entre a interface do usuário e o acesso dos *wrappers*. Ainda é necessária a inclusão do integrador de domínios para a geração de definições no dicionário de metadados e esquemas das fontes de dados.

O dicionário de metadados acumula informações a respeito dos esquemas gerados pelos *wrappers* e informações sobre os próprios componentes do sistema de integração. Como definido no COOPWARE [MGK+96], o dicionário de metadados armazena, acessa e gerencia

informações sobre as fontes de informação e/ou processos de informação. O uso de tais dicionários permite uma escalabilidade maior ao sistema de integração.

Os usuários envolvidos no processo devem possuir atribuições de administrador de banco de dados, para a criação dos dicionários de metadados. Ademais, são responsáveis pela geração da camada de *wrappers*. Os *wrappers* fornecem esquemas acessíveis pelos usuários finais e juntamente com o repositório de metadados auxiliam na geração do resultado final do sistema de integração.

O processo de execução de consultas envolve máquinas de execução de consultas. Tais máquinas decompõem a consulta a fim de gerar subconsultas com acesso à fonte de dados relacionada. No final, todas as subconsultas geradas são integradas na interface, como resposta à consulta do usuário. A figura 2.5 denota a arquitetura de sistemas dinâmicos de integração.

2.3.3 Trabalhos relacionados

O processo de integração tem concentrado suas técnicas na unificação de esquemas, seja ela total ou parcial. O que é praticamente impossível é obter um grau de interoperação e integração sem a intervenção dos usuários. Portanto, os diversos sistemas têm-se apoiado na mínima intervenção humana para a resolução da heterogeneidade semântica de um corpo heterogêneo de sistemas.

Além disso, a unificação dos esquemas que formam uma visão integrada dos sistemas agregam somente os modelos estruturais e não resolvem problemas de ambigüidade, por exemplo. Além das dificuldades de interoperabilidade, os sistemas heterogêneos podem possuir grupos de entidades e/ou objetos que não possuem características em comum, tornando difícil a integração. Isto é um fato pelo qual as pesquisas na área têm progresso lento [HM93].

Os sistemas de integração de informação pela *web* possuem a tarefa de responder consultas que podem requerer a extração e combinação de múltiplas fontes de dados na *Web*

[FLM98]. São relacionados os problemas da integração pela *Web* como sendo: a dinâmica evolução das fontes de dados na *Web*; a pouca quantidade de metadado referente às fontes de dados e a grande autonomia que cada fonte de dados possui.

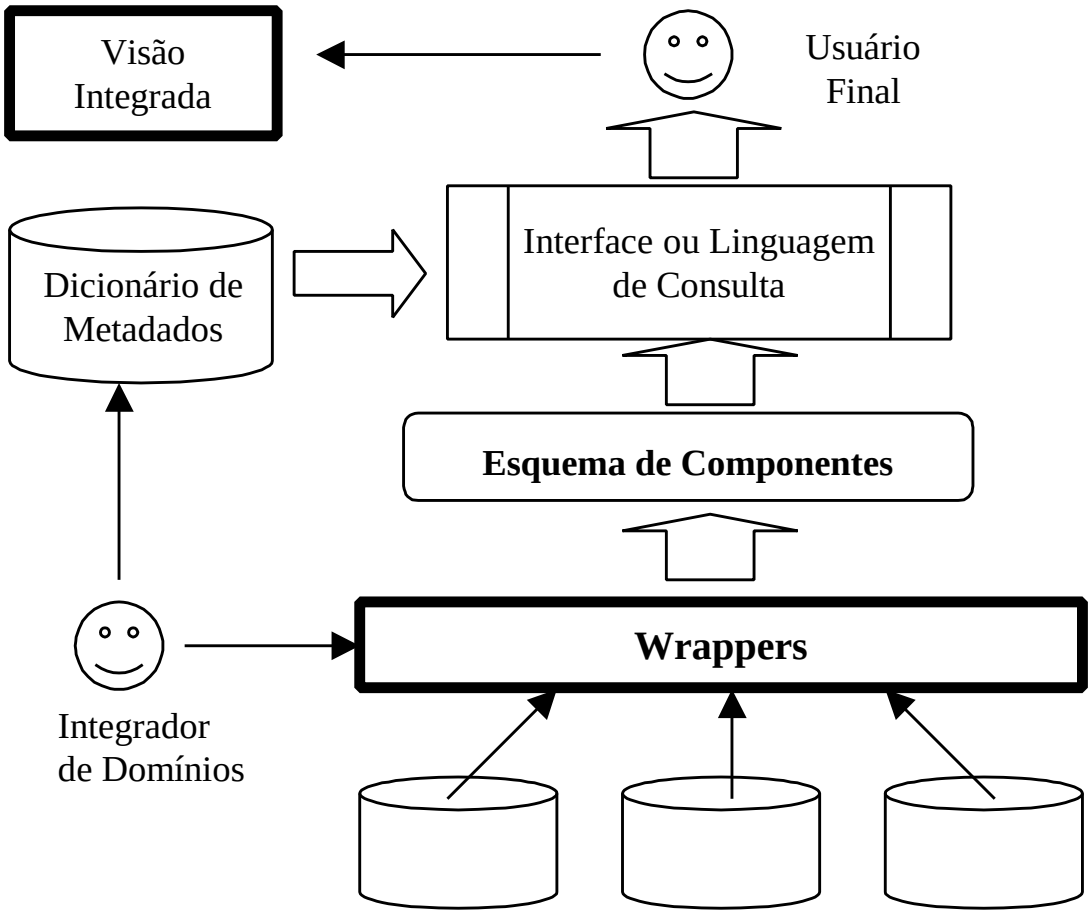

Figura 2.5: Arquitetura de Sistemas de Integração Dinâmicos

Na construção de sistema de integração estão envolvidos vários módulos com características intrínsecas às da integração de informação. São elaboradas especificações de esquemas de mediação e reformulação para que as consultas construídas no esquema de mediação possam ser reformuladas para as origens de dados. Geralmente, no processo de reformulação, são utilizados descritores de fontes de dados para endereçar as origens de dados. São necessárias diferentes capacidades de processamento de consultas, otimização de consultas, máquinas de execução de consultas, construtores de *wrappers* e suporte ao tratamento de casamento de objetos através das fontes de dados [FLM98].

Muitos dos trabalhos relacionados à área têm forte impacto na criação de sistemas comerciais para *Web*. Uns dos que mais se destaca é o uso de informações semi-estruturadas. O padrão XML tem incrementado a *Web* como peça de extrema importância na interoperação de sistemas. Primeiro, seu formato texto o torna legível pela grande maioria dos sistemas. Segundo, pela sua habilidade de expressar informações sobre os próprios dados, e finalmente por ter uma estrutura semelhante a estruturas de dados que há anos os sistemas de informação trabalham: árvores e grafos.

Em contraste às metodologias de integração e interoperabilidade abordadas, existem meios de integração que vários sistemas comerciais utilizam. Os destaques das técnicas paralelas sobressaem-se no uso de JDBC [WFC+99] e ODBC [Gei96] que contrários ao uso de uma única interface para a integração, admitem a criação de interfaces padrões em cada sistema. São formados *gateways* ponto-a-ponto que se comunicam por intermédio de rotinas padronizadas, embora os problemas semânticos sejam tratados de forma manual. O uso de objetos distribuídos tem auxiliado no processo de integração à medida que a comunicação entre os sistemas seja padronizada. A padronização comumente utiliza CORBA (*Common Object Request Broker Architecture*) para criar interface de comunicação padrão entre os sistemas [Vin97].

A tabela 2.1 relaciona alguns dos trabalhos na área de integração e interoperação relacionados com as características básicas de cada sistema.

2.4 Considerações finais

Neste trabalho é utilizada a abordagem virtual para a integração pela *web* visto que as informações são recuperadas diretamente da fonte de dados sem a utilização de repositório central.

XMQL não possui automação no processo de resolução das heterogeneidades dos participantes do processo de integração. Esta tarefa é responsabilidade do usuário desenvolvedor do documento XMQL e é feita no momento da criação das consultas SQL. Neste caso, podem ser utilizadas as funções de conversão de tipos e operadores de cada

gramática SQL das fontes de dados.

A arquitetura de integração para XMQL utiliza uma variação da integração dinâmica face a inexistência de uma camada de mediação e do acesso direto ás fontes de dados. Nesta arquitetura, não é utilizado um dicionário de metadados e também não há uma interface gráfica para a criação do documento XMQL.

Tabela 2.1: Sistema de Integração e Interoperação e suas características.

Produto	***MultiBase***	***MRDMS***	***TSIMMIS***	***HERMES***	***Information Manifold***	***MetaDatabase***
Características dos Sistemas						
Classificação	Federado Fortemente Acoplado	Federado Fracamente Acoplado	Sistema de Mediadores	Sistema de mediadores com fusão de conhecimento	Agentes de agrupamento de informações	Sistema de Repositório de Metadados
Instituição	Computer Corporation of America	INRIA, França	University of Stanford	University of Maryland	AT&T Bell Labs	Rensselaer Polytechnic
Modelo de Dados Comum	Funcional + generalização	Relacional	Object Exchange Model (OEM)	Entidade relacionamento de dois estágios	Relacional com hierarquia de classes	Nenhum (os metados são expressos no modelo Entidade relacionamento)
Atualizável	Sim	Sim	Não	Não	Não	Não
Fonte de Dados destino	Banco de dados estruturado	Banco de dados relacionais	Semi-estruturado ou não estruturado, dinâmico	Bases de conhecimento, espacial e sistema de raciocínio temporal.	Fontes de dados para Web.	Banco de dados Enterprises e CIM
Características de Integração						
Arquitetura	Estático	Dinâmico	Estático	Estático	Estático	Dinâmico
Visão	Esquema Global	Nenhuma	Nenhuma	Nenhuma	Visão do Mundo	Nenhuma
Metadado	Esquema Auxiliar	Nenhum	Nenhum	Yellow pages	Descritores de Fontes	Global Information Resource Dictionary
Linguagem de Especificação	DAPLEX e NQUEL	MSQL	Mediator Specification Language (MSL)	Generalization Annotated Program Framework	CARIN – CLASSIC	MQL
Técnica de Integração Semântica	generalização	Junção implícita, atributos dinâmicos	Especificação de regras, objetos virtuais	Fusão usando Lógica anotada	Funções de Correspon-dência	Processador de regras, modelo transversal
Ferramenta de Integração	Extensões da Linguagem	Extensões da Linguagem	MEDMAKER	Ambiente de Programação Mediatory	Trabalho Futuro	Model Transversal Interface