2
Conceitos básicos

2.1
Arquiteturas tradicionais para integração de dados

2.1.1
Arquitetura de mediadores

Um mediador é um componente de software que intermedia o acesso de clientes (usuários ou componentes de software) a fontes de dados (Wiederhold, 1992). Em particular, os mediadores podem ser desenvolvidos para usar outros mediadores como componentes.

A Figura 4 mostra a arquitetura clássica de mediador-adaptador (Busse et al., 1999), composta pelas seguintes camadas:

- *Camada de Mediação:* contém um ou mais mediadores fornecendo acesso integrado a fontes de dados ou outros mediadores, podendo formar redes de mediadores. De forma geral, o mediador centraliza as informações fornecidas pelos adaptadores das fontes mediadas numa visão unificada dos dados disponíveis nas fontes. Para isso, o mediador possui um *esquema mediado*, ou *esquema global*, e alinhamentos entre os elementos do esquema mediado e os esquemas das fontes. O esquema global permite que uma consulta submetida ao mediador possa ser traduzida em consultas às fontes. Dada uma consulta submetida ao mediador, ele cria e submete consultas específicas para as fontes, reúne os resultados parciais das fontes e entrega uma resposta unificada ao usuário.
- *Camada de Adaptação:* contém os adaptadores responsáveis pelo acesso às fontes de dados. Cada adaptador esconde a heterogeneidade técnica e do modelo de dados da fonte de dados a qual adapta, tornando o acesso à fonte de dados transparente para o mediador. Para cada fonte de dados existe um adaptador que exporta as informações relevantes sobre a fonte, tais como: seu esquema, informações sobre seus dados e sobre seus recursos para processamento de consultas. Em geral é utilizado um *esquema de*

exportação para descreve a forma como os dados são disponibilizados por cada fonte. O *esquema de exportação* representa um esquema externo do esquema conceitual da fonte em questão, contendo apenas os atributos relevantes no contexto do mediador.

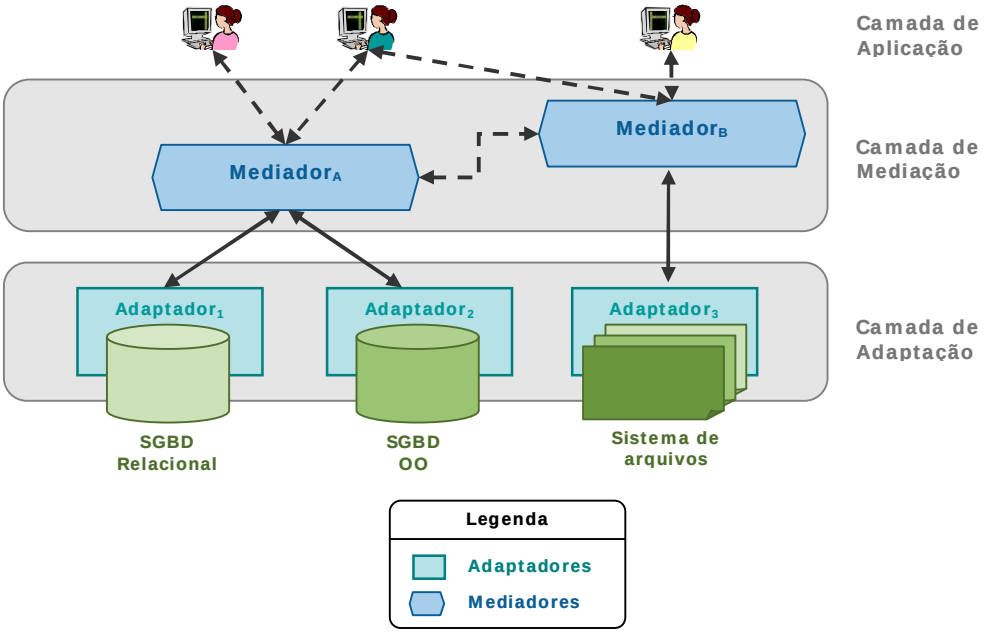


Figura 4 – Arquitetura mediador-adaptador.

Em geral, os mediadores são especializados para um conjunto inter-relacionado de fontes com dados "semelhantes" e, assim, fornecem acesso a dados relacionados a um domínio específico.

A construção de um mediador impõe alguns desafios importantes, entre eles a heterogeneidade semântica das fontes de dados. Estes desafios tornam-se ainda mais complexos no contexto da Web, onde a quantidade de fontes de dados distintas pode ser enorme e é desejável a inclusão e remoção de fontes dinamicamente, sem controle do mediador. Isto torna mais difícil a tarefa de mapear as informações entre os esquemas de exportação e global.

2.1.2 Armazém de dados

Um armazém de dados ou *data warehouse* (DW) é uma plataforma para integração de dados para facilitar o desenvolvimento de sistemas de apoio a decisão e de processos empresariais (Kimball, 2002). O objetivo principal é a

integração efetiva de fontes de dados operacionais em um único repositório que facilite o uso estratégico dos dados.

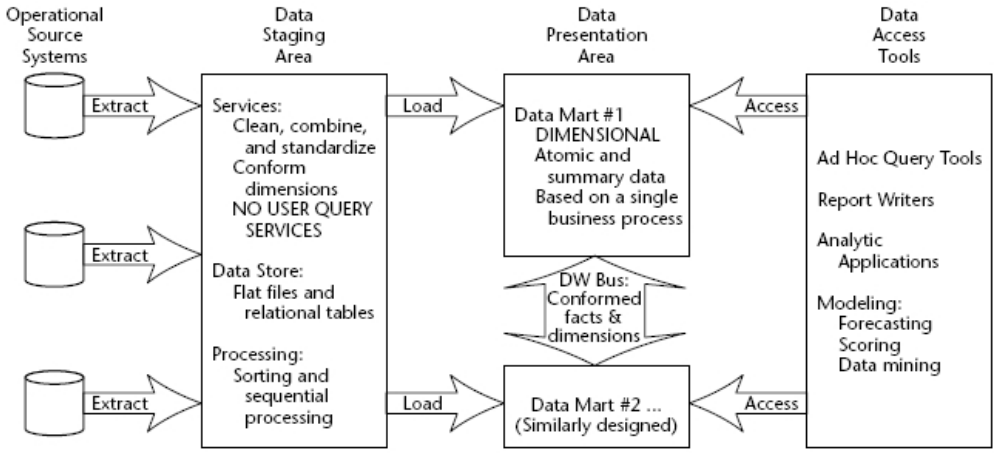


Figura 5 – Elementos básicos de uma data warehouse e o processo ETL.

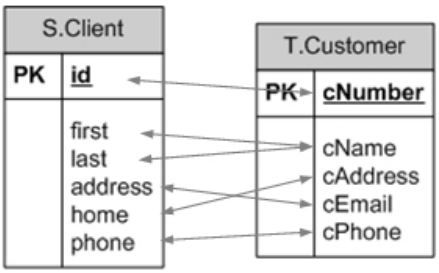
A criação de uma DW segue um processo de extração, transformação e carga dos dados (*ETL - extract-transformation-load*). A Figura 5 ilustra o processo ETL e posteriormente o acesso aos dados da DW. A extração é a primeira etapa do processo de ETL. Nesta etapa, os dados necessários para a DW são lidos, entendidos e copiados das fontes de dados de origem para a área de preparação (*Data Staging Area*) da DW. Na área de preparação é realizada a etapa de transformação de dados, onde diversas operações podem ser aplicadas, tais como: a limpeza (que inclui resolução de conflitos, tratamento de elementos nulos, ou a transformação para um formato padrão), a combinação de dados de múltiplas fontes, a eliminação de duplicatas, etc. A etapa de transformação precede a última etapa do processo de ETL conhecida como carga dos dados. Finalmente nesta etapa, os dados são carregados para a área de apresentação (*Data Presentation Area*) da DW, de onde os dados estarão prontos para serem acessados pelas ferramentas de acesso.

Para exemplificar a etapa de transformação de dados utilizamos um exemplo simples onde se deseja carregar os dados de uma base de dados com esquema *S* para uma DW com esquema *T*. Primeiramente, *S* precisa ser comparado contra *T*, de forma a identificar os elementos similares e distintos. Os elementos distintos e suas instâncias são descartados enquanto os elementos similares serão utilizados para a transformação de dados.

A Figura 6 ilustra os esquemas *S* e *T*. Para exemplificar foram utilizados apenas os elementos similares *S.Client* e *T.Customer*, que armazenam

informações de consumidores de lojas *online*. A Figura 6 (a) ilustra as correspondências entre os atributos dos esquemas de ambas as tabelas, tais como: *id* ↔ *cNumber*, *first* ↔ *cName*, *last* ↔ *cName*, *address* ↔ *cEmail*, *home* ↔ *cAddress* e *phone* ↔ *cPhone*. Note que os elementos *first* e *last* do esquema *S* representam o primeiro nome e o sobrenome do consumidor, respectivamente. Ambos mapeiam para o mesmo atributo *cName* do esquema *T*, que representa o nome completo do consumidor.

Com base nestas correspondências, no momento de carregar os dados da fonte de origem para a DW, uma consulta em SQL pode ser gerada para transformar as instâncias de *Client* de *S* em instâncias de *Customer* de *T*. A Figura 6 (b) ilustra a consulta SQL citada anteriormente. Note que os valores das colunas *id*, *home, address* e *phone* de *S* são utilizados para popular as colunas *cNumber*, *cAddress, cEmail* e *cPhone* de *T*, respectivamente, enquanto os valores de *first* e *last* de *S* são concatenados para popular a coluna *cName* de *T*.



(a) Alinhamento

S.Client

id	first	last	address	home	phone
13	Daniela	Brauner	db@m	Av.Br., 10	9447
...	...	...		...	...

→

T.Customer

cNumber	cName	cAddress	cEmail	cPhone
13	Daniela Brauner	Av.Br., 10	db@m	9447
...	...	...		...

```
INSERT INTO T(cNumber, cName, cAddress, cEmail, cPhone)
SELECT id, Concat(first, last), home, address, phone
FROM S
```

(b) Transformação de dados

Figura 6 – Alinhamento de esquemas conceituais e transformação de dados.

2.2
Alinhamentos sintáticos, semânticos e *a priori*

Casanova et al. (2007) apresenta três classificações para abordagens de alinhamento de esquemas: *sintática, semântica* e *a priori*[3].

A primeira abordagem, chamada de *sintática*, consiste em alinhar dois esquemas utilizando indicações sintáticas, tais como os tipos de dados dos atributos e as similaridades (sintáticas) entre os nomes dos elementos dos esquemas. Esta abordagem depende da suposição implícita de que a proximidade sintática implica em similaridade semântica. Entretanto, tal suposição é freqüentemente injustificável e pode levar a alinhamentos errôneos.

Por exemplo, suponha dois esquemas *S* e *T* descrevendo bancos de dados em domínios de aplicação não muito evidentes. *S* possui um conjunto de objetos chamado *Games*, com atributos *Name* e *ESRB*, e *T* possui um conjunto de objetos denominado *Gaming*, com atributos *Name*, *Price* e *Rating* (veja Figura 7). Utilizando apenas uma abordagem de comparação sintática, *Games* provavelmente seria alinhado com *Gaming*, os atributos *Name* seriam alinhados entre si, mas *ESRB* não seria alinhado com *Rating*. Se o domínio de *S* e *T* fosse em jogos de computador, então este alinhamento estaria razoável, exceto pela falta do alinhamento entre *ESRB* e *Rating*, o qual deveria ter sido capturado, pois ambos se referem a avaliação de jogos (*Entertainment Software Rating Board* - ESRB). Porém, se *S* descreve um banco de dados de uma agência de turismo especializada em safáris, é injustificável o alinhamento de *Games* (significando neste caso caçadas - *game hunting*) com *Gaming*, do esquema *T* de um banco de dados de jogos de computador.

A segunda abordagem, chamada de *semântica*, e abordagem alvo desta tese, utiliza indicações semânticas para gerar hipóteses sobre o alinhamento dos esquemas. Esta tenta detectar como os objetos do mundo real são representados em diferentes fontes de dados e utiliza esta informação para alinhar os esquemas. Esta abordagem é mais robusta do que a abordagem sintática, porém só pode ser aplicada a esquemas simples. Por esquemas simples entende-se aqueles que possuem tabelas com poucos relacionamentos entre si expressos através de chave estrangeiras.

[3] Note que este conceito é diferente da abordagem *a priori* introduzida nesta tese. Nesta tese, o termo *a priori* é usado para uma abordagem de alinhamento semântica que realiza os alinhamento antes de permitir o acesso às fontes utilizadas.

Retornando ao exemplo, para decidir se *Games* deve ser alinhado com *Gaming* (veja Figura 7) utilizando a abordagem semântica, o mediador deve implementar o seguinte procedimento. Primeiramente, ele deve selecionar um conjunto de objetos típicos armazenados em *Gaming*, tais como “Flight Simulator”© e “Super Mario Bros”©, e sondar *S* verificando a ocorrência de tais objetos em *Games*. Esta informação é utilizada então para alinhar *Games* e *Gaming*, alinhar os atributos *Name*, e alinhar os atributos *ESRB* e *Rating*, como é esperado. Note que esta abordagem será executada com sucesso quando *S* e *T* descrevem bancos de dados de jogos de computador. Generalizando esta abordagem, o mediador deve ignorar o fato de que *Games* e *Gaming* são sintaticamente similares e tentar alinhar *S* e *T* utilizando somente o conjunto de objetos típicos.

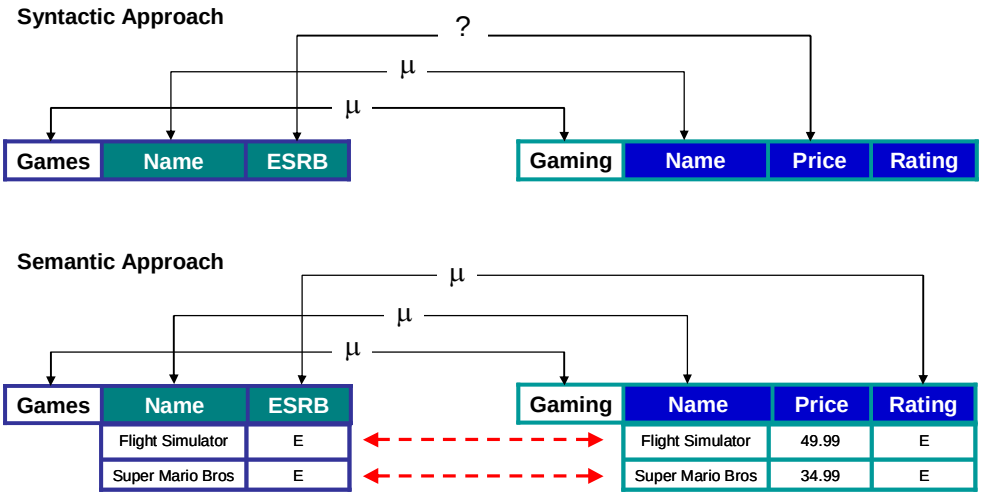


Figura 7 – Exemplos das abordagens sintática e semântica.

Vale ressaltar que, na abordagem semântica, deve ser possível detectar as instâncias equivalentes armazenadas em diferentes fontes de dados. Uma técnica simples consiste em utilizar os próprios valores dos atributos das instâncias em questão. Porém, detectar que “Flight Simulator” denota o mesmo jogo de computador em ambas as fontes de dados é uma tarefa aparentemente simples, pois se trata de um nome comercial, mas não integralmente à prova de erros devido às várias versões existentes do jogo, evidenciando objetos diferentes.

Para resolver isso, devem ser utilizados valores de atributos que identifiquem unicamente as instâncias no domínio das fontes em questão, aqui chamados de *identificadores de domínio*. No exemplo, pode ser utilizado o

código do produto (*part number - PN*) ao invés do nome comercial. Ainda assim, podem existir casos onde os jogos vendidos nos Estados Unidos são identificados usando o código de produto universal (*Universal Product Code* - UPC) com 12 dígitos, enquanto os jogos vendidos na Europa usam o código global do item (*Global Trade Item Number* - GTIN) com 14 dígitos. Portanto, a identificação de duplicatas é um processo nada trivial que requer esforço adicional para amadurecimento das técnicas existentes.

No domínio de dados geográficos, o identificador de domínio pode ser definido com base nas propriedades espaciais dos objetos, que são representadas em atributos como *bounding-boxes* ou *centróides*. Para exemplificar esta abordagem são apresentadas na Figura 8 duas tabelas contendo instâncias nomeadas como “Rio de Janeiro” de duas fontes distintas *A* e *B*. Note que, a partir dos centróides dos objetos, representados pelos atributos LAT e LONG da fonte A e *footprintX* e *footprintY* da fonte B, é possível notar que os objetos “Estado do Rio de Janeiro” da fonte A e “Rio de Janeiro, Estado do - Brazil” são o mesmo objeto, visto que os atributos de A: LAT e LONG possuem os mesmo valores dos atributos de B: footprintY e footprintX, respectivamente. Portanto, os atributos LAT e footprintY e LONG e footprintX, quando analisados em conjunto, atuam como identificadores de domínio para as bases de dados A e B.

A

ID	FULL_NAME_ND	DSG	DMS_LAT	DMS_LONG	LAT	LONG
76153	Estado do Rio de Janeiro	ADM1	-220000	-423000	-22.0	-42.5
56032	Serra do Rio de Janeiro	HLLS	-175700	-445700	-17.95	-44.95
67203	Rio de Janeiro	PPLA	-225400	-431400	-22.9	-43.2333333
39870	Rio de Janeiro	STM	-94900	-671600	-9.8166667	-67.2666667

B

identifier	display-name	class	footprintY	footprintX
adlgaz-1-1457057-00	Rio de Janeiro, Estado do - Brazil	administrative areas	-22.0	-42.5
adlgaz-1-1457059-20	Rio de Janeiro, Serra do - Brazil	mountains	-17.95	-44.95
adlgaz-1-1457061-32	Rio de Janeiro - Brazil	populated places	-22.9	-43.2333
adlgaz-1-1437138-6b	Janeiro, Rio de - Brazil	streams	-11.85	-45.15
adlgaz-1-3223719-6f	Rio de Janeiro - Loreto, Departamento de - Peru	populated places	-4.3833	-71.8167

Figura 8 – Exemplos de instâncias equivalentes.

Ambas as abordagens, sintática e semântica, trabalham *a posteriori*, no sentido de que elas tentam alinhar os esquemas de bancos de dados pré-existentes. A terceira abordagem, chamada *a priori*, enfatiza que, ao especificar bancos de dados que irão interagir entre si, o projetista deve começar pela adoção de um padrão apropriado (um esquema global), se existir, para guiar o

projeto dos esquemas. Se não existir, o projetista deve publicar uma proposta para um esquema global cobrindo o domínio de aplicação. Se cuidadosamente aplicada, a abordagem *a priori* contorna o problema de heterogeneidade semântica entre esquemas de bancos de dados. Se ambos os esquemas *S* e *T* seguirem o mesmo esquema global, o alinhamento de *S* e *T* se torna trivial: o alinhamento μ dos conceitos de *S* nos conceitos de *T* é tal que $t = \mu(s)$ sse *s* e *t* denotam o mesmo conceito do esquema global.

Atualmente, existem diversos padrões que podem ser seguidos por projetistas de bancos de dados durante o projeto e publicação das fontes. Por exemplo, o padrão ISO 19115:2003 *Geographic information – metadata* define um esquema de metadados para descrever objetos geográficos e é utilizado para definir esquemas para catálogos geográficos. Um esquema de metadados possui um conjunto de elementos principais e um conjunto de elementos opcionais. Para utilizar um esquema de metadados, cada aplicação deve definir seu perfil (*profile*), isto é, a lista de elementos opcionais que implementa. Assim, as fontes que seguirem este padrão podem interoperar de forma simples, ou seja, apenas compartilhando seus perfis. Desta forma, a semântica dos esquemas é fixada *a priori* pelo padrão utilizado.

As ontologias, no contexto utilizado pelo *World Web Consortium* (W3C), podem ser utilizadas como uma forma de abordagem *a priori*. Uma estratégia razoável para definir um esquema comum em determinado domínio de aplicação pode ser descrita da seguinte forma: (1) selecionar fragmentos de ontologias populares, por exemplo *Wordnet*, que cubra os conceitos pertencentes ao domínio de aplicação; (2) alinhar os conceitos dos fragmentos distintos criando conceitos unificados; (3) publicar os conceitos unificados na forma de uma ontologia, indicando quais os conceitos obrigatórios e opcionais. A ontologia publicada pode ainda guardar a origem dos conceitos unificados de forma a garantir a clareza da semântica e a proveniência dos termos.

Porém, é importante ressaltar que, qualquer que seja a abordagem adotada - sintática, semântica ou *a priori* - o problema de detecção de duplicatas é inerente ao processo de integração. Seja em mediação de consultas, no momento de unificar os resultados obtidos das fontes mediadas, ou em data warehouses, no momento de carregar os dados de uma fonte para outra, é essencial que seja possível detectar as instâncias equivalentes armazenadas nas diferentes fontes de dados.

Nesta tese, são investigadas abordagens semânticas utilizando instâncias para alinhamento de esquemas conceituais e esquemas de classificação. De

modo geral, as instâncias das fontes de dados são utilizadas para gerar evidências sobre os mapeamentos entre os elementos dos esquemas. A seguir, as abordagens serão introduzidas em capítulos separados para o alinhamento de tesauros e para alinhamento de esquemas conceituais.