Integrando Sistemas Legados a Bancos de Dados Heterogêneos

Helio Rubens Soares Claudia Bauzer Medeiros
hrubens@facintr.com.br cmbm@dcc.unicamp.br
IC UNICAMP CP 6176
13081-970 Campinas SP Brasil

Abstract

This paper presents a methodology to construct a federated database infrastructure to help the integration of heterogeneous data sources and which takes legacy data into account. This methodology considers different kinds of data sources and systems to be combined, and gives guidelines to integrate the data for each situation. The last step of the methodology consists in an algorithm that produces mappings from queries on the federated system to the set of queries on the database that participate in the federation. The methodology was validated by a case study on databases and legacy systems of the municipal administration of Paulínia, SP.

Keywords: Bancos de Dados Heterogêneos, Sistema de Bancos de Dados Federados, Sistemas Legados, Sistemas de Informações Geográficas.

1 Introdução

Este artigo aborda o problema de integração de sistemas legados e bancos de dados heterogêneos, motivado pelo processamento de aplicações urbanas. A solução proposta é baseada na criação de um sistema federado e consiste em uma metodologia que parte da padronização dos sistemas envolvidos na integração indo até a especificação do sistema federado e mapeamento de consultas ao sistema federado em consultas aos seus componentes. A integração é vista sob o enfoque de bancos de dados e se restringe aos dados, não considerando problemas relativos ao código dos sistemas legados.

O ponto de partida para este trabalho foi um estudo de caso junto à prefeitura da cidade de Paulínia-SP em 1997, que levantou os problemas relativos à integração de sistemas já existentes a um Sistema de Informações Geográficas (SIG). O objetivo final seria permitir o processamento integrado de todas as aplicações de planejamento urbano usando um sistema geográfico, integrando os departamentos e secretarias da prefeitura através de uma rede, aproveitando aplicações já implantadas (por exemplo, gerenciamento de cadastro urbano, cadastro de desempregados, controle de atividades esportivas e outros.

O levantamento dos dados e aplicações disponíveis nas diferentes Secretarias da cidade e os requisitos encontrados levaram à conclusão que a melhor solução seria a construção de um sistema federado em que um dos componentes fosse um banco de dados geográfico. Esta solução optou por sistemas de bancos de dados federados por serem estes uma alternativa mais equilibrada para o problema de bancos de dados heterogêneos, pois apresentam um nível de acoplamento moderado, possibilitando funções globais eficientes sem comprometer a autonomia dos SGBDs locais. As necessidades deste projeto específico levaram à elaboração da metodologia geral de construção de sistemas federados, que engloba a especificação destes sistemas e algoritmos de mapeamento de consultas.

As principais contribuições deste artigo são: (a) especificação da metodologia de construção do sistema federado, que leva em conta não apenas bancos de dados tradicionais, mas a inclusão de SIG e dados geográficos ao sistema federado; (b) descrição de sua aplicação ao caso de Paulínia (estudo de caso real); (c) implementação de um protótipo que permite mapear consultas ao sistema federado em consultas aos bancos de dados da federação. O uso de bancos de dados geográficos em federações é ainda limitado na literatura correlata, que em geral trata de dados não espaciais ([Int97]).

As próximas seções estão organizadas da seguinte forma. A seção 2 apresenta uma revisão bibliográficasobre bancos de dados heterogêneos e sistemas legados. A seção 3 apresenta a metodologia proposta. A seção 4 discute aspectos específicos relativos a sistemas de informações geográficas. A seção 5 apresenta parte de um estudo de caso implementado a partir do problema de Paulinia. A seção 6 apresenta conclusões e extensões.

2 Bancos de Dados Federados e Sistemas Legados

Há várias propostas para integração de bancos de dados heterogêneos – por exemplo, [Bre90, Agu95], usando ou não o conceito de federação. Várias abordagens são factíveis, dependendo do nível de acoplamento e, consequentemente, da autonomia que se deseja manter nos bancos de dados locais.

Um *sistema de bancos de dados federado* (SBDF) pode ser definido como uma coleção de sistemas de bancos de dados independentes, cooperativos, possivelmente heterogêneos, que são autônomos e que permitem o compartilhamento de todos ou alguns de seus dados, sem afetar as suas aplicações locais [SL90].

2.1 Sistemas de Bancos de Dados Federados

Os SBDFs caracterizam-se pela heterogeneidade, presença de dados distribuídos e autonomia de cada componente, ou seja, de cada banco de dados que compõe a federação. A *heterogeneidade* pode ser identificada em diversos níveis e uma das principais causas refere-se às diferenças entre os bancos de dados componentes, como estrutura de dados, nomes, interpretações semânticas dos atributos e restrições. Quando sistemas legados aparecem como um dos componentes da federação, o grau de heterogeneidade do SBDF aumenta porque alguns sistemas legados não utilizam SGBD, ou então os utilizam de forma rudimentar. Existem igualmente sistemas legados que utilizam SGBD de arquiteturas antigas (como hierárquicos).

Um ponto crítico em SBDF é a *autonomia* inerente aos bancos de dados componentes. Sistemas fortemente acoplados garantem ao usuário uma visão integrada dos componentes e exigem um administrador central. Já sistemas fracamente acoplados permitem a autonomia dos componentes, mas exigem dos usuários maior conhecimento do sistema global [Agu95].

Há vários outros fatores que devem ser revistos para uma eficiente manipulação de um SBDF, como por exemplo questões relativas à *distribuição* e *localização* dos dados, às técnicas de *recuperação de informação* e aos mecanismos de *segurança*.

Vários trabalhos buscam soluções para estes problemas como, por exemplo, [Ber96, VL97, SSR94, Agu95, Jr95, Mot98]. As principais soluções estão relacionadas ao uso das seguintes ferramentas: mediadores, tradutores/adaptadores e visões.

Um *mediador* é um software usado para permitir a interoperabilidade entre dois ou mais SGBDs. Com a utilização de mediadores, o acesso aos dados heterogêneos é efetuado através de consultas que são submetidas ao mediador, que por sua vez as transforma em subconsultas a serem enviadas aos SGBDs componentes. As subconsultas geradas pelo mediador devem ser traduzidas para as linguagens de consulta de cada SGBD componente. Ao final, os resultados das consultas são traduzidos por *softwares* tradutores para a linguagem de consulta do SGBD componente que gerou a consulta (origem) e a resposta final é devolvida ao usuário [VL97].

Já os *Tradutores/adaptadores* convertem os dados fonte para um modelo de dados comum e convertem consultas de aplicações em consultas específicas das fontes de informação envolvidas na consulta [VL97]. Alguns mediadores podem usar tradutores/adaptadores como ferramentas para resolver uma parte específica da conversão de dados.

Finalmente, *visões* são usadas como um mecanismo que auxilia a integração dos componentes de um SBDF. Em SGBDs relacionais, visões são encaradas como relações virtuais não-normalizadas, definidas em função de relações preexistentes e obtidas com o resultado do processamento de uma consulta [Agu95]. Os dados de uma visão podem ficar armazenados fisicamente, constituindo uma *visão materializada*, ou serem gerados a cada utilização da visão e, portanto, ficarem armazenados na memória ou em alguma estrutura temporária, neste caso chamada de *visão virtual* [Hul97]. A integração de bancos de dados via federação pode ser real ou virtual. No primeiro caso, há apenas um banco de dados resultante enquanto que no segundo caso o usuário tem visões do banco de dados integrado.

2.2 Sistemas Legados

Como já mencionado, os componentes de um SBDF podem ser tanto SGBDs como sistemas legados. De forma mais detalhada, *sistema legado* é um sistema de grande porte, com milhões de linhas de código, antigo, autônomo, organizado em algum sistema de arquivos, não sendo gerenciado por um SGBD e que resiste a evoluções e modificações [BS95].

Para determinar a maneira mais eficaz de resolver o problema de sistemas legados é necessário identificar, primeiramente, o grau de organização existente, ou seja, o tipo de arquitetura presente nestes sistemas. De acordo com Brodie et al. [BS95], a arquitetura de um sistema legado tem três camadas básicas: (1) interface, (2) aplicações e (3) serviços de banco de dados. O grau de organização de um sistema legado é determinado, dentre outras, pela capacidade de decomposição desta arquitetura que pode ser: (1) *totalmente decomposta*, (2) *parcialmente decomposta* ou (3) *não decomposta*. O grau de organização de um sistema legado diminui à medida que sua arquitetura perde a capacidade de decomposição.

Existem basicamente três enfoques no tratamento de sistemas legados, dependendo do seu grau de organização: (a) Construção de um novo sistema que substitua o sistema legado; (b) Conversão de esquema; e (c) Migração dos dados legados para um outro sistema que suporte modificações de forma mais organizada;

Se a arquitetura do sistema legado permitir a documentação de um esquema básico, através de técnicas de engenharia reversa, este esquema pode ser convertido para um outro esquema mais robusto e flexível. A conversão de esquema e a migração dos dados legados, muitas vezes, são usadas concomitantemente.

A opção de migrar os dados legados para uma outra forma de armazenamento consiste em partir de um sistema legado e chegar a um sistema destino que possua as mesmas

funcionalidades do anterior. Devido à falta de documentação dos sistemas legados, o que acontece em muitos casos é a criação de um sistema completamente novo, sem aproveitar o que já existe e com uma alta probabilidade de fracassar, além da necessidade de altos investimentos.

Não há muitas propostas para SBDF envolvendo sistemas legados. Alguns trabalhos relacionados a conversão de esquemas (por exemplo, [DK97, BDK92, Qia96]), podem auxiliar no tratamento de aplicações e dados legados. Outros trabalhos diretamente envolvidos com sistemas legados visam propostas metodológicas de migração de dados e aplicações ([BS95, NEK94]) ou técnicas de engenharia de software para mapear ou migrar os dados ([PH95, MNB+94, AMR94]).

Brodie et al. [BS95] apresentam várias estratégias de migração de sistemas legados relevantes para este trabalho, uma das quais, *Chicken Little*, é adotada na metodologia proposta. Ela efetua a migração de sistemas legados através de pequenos passos incrementais, permitindo estimar o risco de falhas em cada passo, facilitando a correção de erros e aumentando as chances de sucesso. Cada passo da migração requer uma alocação pequena de recursos e pouco tempo para ser concluída. Desta forma, *Chicken Little* é classificada como uma *metodologia incremental de migração* [BS95].

Propostas como [PH95, MNB+94, AMR94] podem ser combinadas ao *Chicken Little* para oferecerem soluções a alguns de seus passos. Por exemplo, a análise incremental do sistema legado seguida da decomposição incremental de sua estrutura podem ser realizadas através da engenharia reversa. Outros passos como o projeto incremental das aplicações e do banco de dados destino podem ser realizados através de metodologias clássicas.

3 Metodologia Proposta

3.1 Visão Geral

A metodologia de integração recebe como entrada o conjunto de esquemas locais, a especificação dos dados dos sistemas legados, os requisitos e restrições dos usuários em relação à exportação de dados e as consultas e transações processadas por estes usuários. Como saída, a metodologia define um SBDF fracamente acoplado. Além disto, o usuário tem à disposição visões integradas de dados e mapeamentos entre visões e sistemas locais, permitindo o processamento de consultas envolvendo dados dos vários bancos de dados.

A metodologia divide-se em três fases principais, iteradas repetidas vezes até que se atinja a estabilização do esquema federado:

- Fase 1: Padronização das entradas através do mapeamento dos modelos de dados de todos os sistemas de bancos de dados (SBD) e sistemas legados para um *modelo de dados canônico* (MDC).
- Fase 2: Definição dos esquemas exportados, que vão determinar o subconjunto de dados de cada esquema local a ser integrado.
- Fase 3: Criação de um sistema integrado (SBDF) que possibilite o acesso a todos os esquemas exportados.

Características	Soluções
Grupo 1: Arquitetura não decomposta e/ou sistemas fechados.	Reconstruir o sistema legado e, se possível, migrar os dados.
Grupo 2: Arquitetura parcial/totalmente decomposta e/ou sistemas de arquivos e/ou organização moderada dos dados e/ou SGBDs rudimentares e/ou documentação desatualizada ou insatisfatória.	1a. opção a) Manter o sistema legado que gerencia os dados; b) Utilizar um mediador. 2a. opção a) Substituir o sistema legado por um SGBD mais robusto; b) Realizar a migração dos dados.

Table 1: Características gerais dos sistemas legados e soluções aplicáveis a cada caso.

As fases de padronização e exportação dos bancos de dados componentes da federação são os passos mais importantes do processo de integração. É nelas que se define quais dados serão compartilhados e para qual finalidade.

A fase de padronização é responsável por traduzir os esquemas locais a serem integrados para um modelo canônico (MDC). O MDC tem por base o modelo relacional, acrescido de informações semânticas adicionais. A opção pelo modelo relacional foi feita baseada no fato de que a grande maioria das aplicações atuais já utiliza este modelo, além da sua ampla utilização por SGBDs comerciais. As informações adicionais são usadas para facilitar o tratamento da heterogeneidade, já que o modelo relacional não possui muitos recursos para armazenar informações semânticas. Os sistemas legados são tratados à parte nesta fase de padronização porque possuem graus de organização diferentes entre si.

Padronização dos Sistemas de Bancos de Dados: A padronização neste caso considera que os bancos de dados a serem padronizados já têm documentação associada segundo algum modelo de dados conhecido e corresponde à aplicação de algoritmos de tradução dos modelos para o MDC. Estes algoritmos incluem adição de informações semânticas através de relações suplementares, considerando também as informações obtidas com os usuários e com os administradores dos bancos de dados. Há vários algoritmos que permitem a tradução entre modelos (vide [Oli93, Agu95]).

Padronização dos Sistemas Legados: O segundo caso a ser tratado na padronização está relacionado aos sistemas legados, e depende do nível de organização do sistema legado em questão – *totalmente decompostos*, *parcialmente decompostos* ou *não decompostos*. Para cada arquitetura, deve ser considerada uma forma de padronização diferente. A tabela 1 mostra um resumo das principais características dos sistemas legados mais comuns nas organizações e as soluções que podem ser a eles aplicadas para chegar ao MDC. As etapas para chegar a este modelo canônico utilizam a metodologia *Chicken Little*.

O *Grupo 1* de sistemas legados corresponde àqueles que apresentam uma arquitetura não decomposta ou a sistemas fechados, com ausência total de documentação. Estas características não permitem que um modelo de dados seja construído e tampouco que um mediador seja criado para fazer o acesso aos dados. Assim sendo, estes sistemas precisam ser refeitos para que sua desorganização não comprometa a eficiência do SBDF. Pode-se

dizer que estes tipos de sistemas legados são os mais críticos na fase de padronização. Todas as aplicações e funcionalidades devem continuar sendo suportadas pelo novo sistema e todos os dados devem ser migrados, corrigindo-se os problemas de integridade que possam existir.

Já o *Grupo 2* considera aqueles sistemas legados que apresentam uma arquitetura parcialmente ou totalmente decomposta, englobando os sistemas de arquivos, sistemas de bancos de dados rudimentares ou programas com pouca documentação ou documentação desatualizada, mas que possuem um nível de organização suficiente para o acesso aos dados através de mediadores, sem a necessidade de reestruturação total dos dados. Duas soluções são factíveis para o *Grupo 2*, dependendo da decisão de se manter ou não o sistema legado. Se a decisão do projetista do SBDF for manter o gerenciamento dos dados através do sistema legado, o acesso a estes dados pode ser feito através de um mediador. Por outro lado, se houver a necessidade, por exemplo, de gerenciar os dados do sistema legado através de um SGBD mais robusto, deve-se optar por migrar os dados do sistema legado para o novo SGBD.

3.2 Exportação

A fase de exportação é subsequente à padronização, embora sejam necessários vários ciclos entre exportação, testes, padronização até estabilizar o que deve ser exportado. De cada esquema local são extraídos os sub-esquemas que efetivamente irão participar da integração – os *esquemas exportados*. A especificação de quais esquemas devem ser exportados depende do conjunto de aplicações que se deseja executar sobre o SBDF.

3.3 Criação do Sistema de Bancos de Dados Federados

Uma vez padronizados e exportados os esquemas a serem integrados, o passo seguinte corresponde à construção do esquema federado. Nesta fase, como os sistemas legados já estão representados através do MDC, tanto os bancos de dados quanto os sistemas legados que compõem a federação serão chamados de *bancos de dados componentes*, formados apenas pelos esquemas exportados. Estes são usados para construir uma federação fracamente acoplada.

A federação é criada construindo-se um *esquema coordenado*, responsável por resolver a heterogeneidade semântica em nível de nomes de atributo. A seguir descrevemos o algoritmo implementado para criar o esquema federado. Este algoritmo, baseado no trabalho de Zhao [Zha97], estende este trabalho de forma a considerar vários tipos de restrições de integridade e permitir maior flexibilidade na tradução de consultas ao sistema federado.

Ele recebe como entrada os esquemas exportados a serem integrados, definidos utilizando o MDC. Como saída, o algoritmo constrói:

1. uma tabela contendo o nome e o significado semântico de cada atributo da federação (atributo federado), chamada de *Tabela Semântica*;

2. uma matriz de correspondência entre atributos federados e atributos dos esquemas exportados, chamada de *Matriz de Correspondência de Atributos*; e

3. uma tabela para cada esquema exportado contendo os nomes dos atributos tal qual eles aparecem nos esquemas exportados, os tipos destes atributos (chave, não chave, chave estrangeira) e um campo que representa as dependências entre as relações criadas pelas

chaves estrangeiras. Este campo armazena nomes de atributos, a partir dos quais é possível realizar as operações de junção entre as relações dos esquemas. Cada tabela é chamada de *Tabela Exportada*.

A matriz de correspondência de atributos e a tabela semântica devem ser construídas simultaneamente da seguinte forma: Seja M a matriz de correspondência de atributos para um conjunto genérico n de esquemas exportados E_i $(i : 1..n)$; e S a tabela semântica associada à matriz M. A matriz M possui os seguintes atributos: *Atributo-Federado*, *Componente-*E_1*, Componente-*E_2*, ..., Componente-*E_n e a tabela S possui os atributos: *Semântica* e *Atributo-Federado*. As construções de S e M seguem os seguintes esquemas:

S.Semântica : Significado semântico para um conjunto de atributos dos esquemas exportados.
S.Atributo-Federado : Nome do atributo federado escolhido para referenciar um significado semântico único.
M.Atributo-Federado : *S.Atributo-Federado*
M.Componente-E_i : Nome do atributo no esquema E_i cujo significado semântico é igual a *S.Semântica*, relativo a *S.Atributo-Federado*.

Alguns campos *i,j* da matriz de correspondência podem ficam vazios, informando que não existe relação entre o atributo federado da linha i com o esquema da coluna j. Construídas a matriz de correspondência de atributos e a tabela semântica, passa-se à construção das tabelas que armazenam as informações dos esquemas exportados (tabelas exportadas). Estas tabelas possuem os atributos *Nome*, *Tipo*, *Ponteiro* e *Domínio*. Seja T uma tabela exportada para um esquema genérico E. *T.Nome* é o do atributo no esquema exportado E; *T.Dominio* descreve o domínio de cada atributo; *T.Tipo* indica o tipo do atributo: [chave primária, chave estrangeira, primária-estrangeira, não chave]; e *T.Ponteiro* é uma lista de atributos que permite estabelecer as ligações entre as relações do esquema exportado.

R_i.

Com a matriz de correspondência de atributos, a tabela semântica e as tabelas exportadas, é possível processar consultas envolvendo todos os componentes da federação, sem se preocupar com a heterogeneidade de nomes entre eles. Devido às próprias características da federação fracamente acoplada, são consideradas apenas consultas aos dados, desconsiderando os procedimentos de atualização. A alteração de dados neste tipo de federação não é aconselhável devido à autonomia dos bancos de dados componentes. Permitir atualizações neste tipo de ambiente deixaria o sistema integrado vulnerável a problemas de integridade de dados [SL90]. Uma consulta à federação tem a seguinte forma:

SELECT *atributos federados*
FROM *nomes dos bancos de dados componentes*
WHERE *condições de seleção e junção entre os esquemas exportados*

A tradução da consulta do usuário para subconsultas a serem processadas por cada banco de dados componente fica a cargo do SBDF. O processo de tradução da consulta ocorre em 4 fases: (1) Divisão da consulta, (2) Conversão de atributos, (3) Especificação de junções e (4) Identificação dos objetos originais. O processamento da consulta exige que se combine os resultados das subconsultas a cada componente. Para maiores detalhes sobre os algoritmos de construção das tabelas, sugere-se a leitura de [Soa98].

4 Extensão a SIGs

Esta seção analisa à parte a questão de dados geográficos, devido às suas peculiaridades. Um problema adicional que surgiu a partir do trabalho em Paulínia foi a necessidade de

combinar os dados de mapas, fornecidos por um SIG. Basicamente SIGs são sistemas automatizados usados para armazenar, analisar e manipular dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la [Agu95]. Para isto, é necessário que o SIG possua como componente um sistema gerenciador de banco de dados (SGBD) não convencional. Este SGBD é responsável por permitir o uso conjunto de dados espaciais e convencionais.

Basicamente, quando se deseja considerar SIGs no processo de integração de dados, deve-se separar dois tipos de gerenciamento: (1) gerenciamento dos dados convencionais e (2) gerenciamento dos dados espaciais. Desta forma é preciso acrescentar ao algoritmo o processo de integrar dados espaciais.

Há vários problemas em aberto relativos à integração de dados espaciais, como por exemplo: (a) representação de dados (matricial/vetorial); (b) escala; (c) projeção e (d) qualidade e heterogeneidade temporal e espacial dos dados.

O primeiro problema acontece quando alguns dados espaciais utilizam a representação matricial e outros utilizam a representação vetorial. O segundo e terceiro problemas estão relacionados com dados armazenados em diferentes escalas e/ou projeções, inviabilizando seu uso integrado. Por fim, o quarto problema está relacionado à fase de coleta de dados. Dados espaciais coletados em periodos distintos e/por equipamentos diferentes são muitas vezes impossíveis de serem integrados.

Para resolver os três primeiros problemas, é necessário uma padronização dos dados espaciais, escolhendo uma representação, uma escala e uma projeção padrão e, posteriormente, converter todos os dados para este padrão. Vale ressaltar que a padronização neste caso só é possível se os dados estiverem sendo usados por SIGs iguais, pois cada SIG depende totalmente de formatos quase que proprietários dos dados espaciais. Quando os SIGs são diferentes é preciso migrar os dados de cada SIG para um banco de dados independente, o que deixa de ser uma integração de dados. O quarto problema, em inúmeros casos, não permite a realização de algum tipo de padronização e dificilmente os dados podem ser usados de forma integrada.

Desta forma, a metodologia deve ser ampliada de forma a considerar o seguinte:

Fase 1: Padronização: Padronizar os bancos de dados geográficos (SGBD do SIG), determinando a representação básica e escala/projeção padrão e transformar todos os dados espaciais para este padrão.

Fase 2: Exportação: (1) Exportar o esquema referente à parte dos dados não espaciais; e (2) Exportar as definições dos arquivos espaciais.

Fase 3: Construção da federação: Os dados não espaciais devem ser tratados como anteriormente. A única diferença é que, em vários SIGs, existem atributos reservados, com nomes particulares e que são utilizados para ligar dados convencionais a espaciais. Em especial, dificilmente um SIG consegue acessar arquivos definidos por um SIG diferente, sem que haja um pré-processamento (por exemplo, usando moderadores).

5 Aplicação a Dados Reais

Esta seção mostra um estudo de caso da aplicação da metodologia ao projeto de Paulínia, limitado a apenas três componentes heterogêneos. Os três componentes foram especificados

a partir dos bancos de dados das secretarias de Educação, de Esportes e do SAE (Serviço de Apoio ao Empregado). A Secretaria de Educação gerencia seus dados usando *dBASE*, para matrículas de alunos no sistema municipal de ensino. Já a Secretaria de Esportes utiliza planilhas EXCEL para processar dados das pessoas que praticam algum tipo de esporte oferecido pela prefeitura, os locais e os responsáveis pelo esporte praticado. Por fim, o SAE mantém arquivos com extensão *DBF* gerenciados pelo programa *CLIPPER*. No resto do texto, estes componentes são referenciados pelos nomes Educação, Esporte e SAE.

5.1 Padronização e Exportação

O componente Educação foi considerado um banco de dados relacional pelo fato dos dados estarem sendo gerenciados pelo *dBASE* e por existir uma documentação que permita a representação dos dados no MDC. Já o componente Esporte se enquadra no pior caso de sistema legado, pois seus dados são pouco flexíveis, difíceis de gerenciar e estão armazenados em um formato que exige migração para um novo sistema. O componente SAE também é considerado um sistema legado. No entanto, seus dados possuem uma organização interna maior do que Esporte e já utilizam o modelo relacional.

Devido a estas características, a fase de padronização foi diferente em cada caso. Em princípio, não é necessária para os componentes Educação e SAE, devido à sua documentação e a descrição (relacional) dos dados armazenados. Já o componente Esporte não possui documentação e necessitou inicialmente de um modelo de representação para seus dados. Inicialmente, foi construído um diagrama ER a partir de entrevistas com o pessoal da Secretaria e análise dos próprios dados armazenados nas tabelas *EXCEL*. A partir deste diagrama, construiu-se o modelo canônico através dos algoritmos de tradução de modelos citados em [Oli93, Agu95], gerando três esquemas relacionais (Atleta, Controle, Esporte).

Para nenhum dos três componentes foi necessário criar tabelas adicionais para armazenar restrições ou informações semânticas dos dados. O MDC, neste caso, corresponde apenas ao modelo relacional. Além disto, a heterogeneidade de domínio não foi considerada neste estudo de caso e, portanto, o campo *Domínio* foi ocultado das tabelas exportadas de cada componente.

A migração de dados foi realizada seguindo a metodologia *Chicken Little* citada anteriormente, para que esta operação fosse feita de forma mais controlada. Os esquemas dos componente Esporte, Educação e SAE estão representados a seguir.

EDUCACAO: Aluno(Cod, nome, endereco, bairro-al, sexo, naturalidade)
EDUCACAO: Escola(ID, responsavel, endereco, bairro)
EDUCACAO: Matricula(Cod, escola_ID, tipo, turno)
ESPORTE: Atleta(Codigo, nome, endereco, bairro, origem, fone, sexo)
ESPORTE: Controle(esporte, coordenador, investimento, endereco, bairro)
ESPORTE: Matresp(Codigo, nome-esp, inicio, turno)
SAE: Pessoa(codp, nomep, endp, sexop, naturalp, bairrop)
SAE: Ult-emp(codp, descricao, periodo, cidade)

No componente Educação, a relação "Aluno" armazena os dados dos alunos que estão ou já foram matriculados em algum curso oferecido pela prefeitura. A matrícula e os dados dos cursos estão armazenados na relação "Matrícula". Por exemplo, o atributo *tipo* em "Matrícula" indica o curso em que a matrícula foi realizada.

O componente Esporte armazena na relação "Atleta" os dados das pessoas que praticam

Educação.Nome	*Educação.Tipo*	*Educação.Ponteiro*
Aluno.cod	primária-estrangeira	Matrícula.cod
Aluno.nome	não chave	Aluno.cod
Aluno.endereço	não chave	Aluno.cod
Aluno.bairro-al	não chave	Aluno.cod
Aluno.sexo	não chave	Aluno.cod
Aluno.naturalidade	não chave	Aluno.cod
Escola.ID	primária-estrangeira	Matrícula.escola
Escola.responsável	não chave	Escola.ID
Escola.endereço	não chave	Escola.ID
Escola.bairro	não chave	Escola.ID
Matrícula.cod	primária	
Matrícula.escola	primária	
Matrícula.tipo	primária	
Matrícula.turno	não chave	Matrícula.cod

Table 2: Tabela exportada do componente Educação.

ou já praticaram algum esporte oferecido pela Secretaria. As matrículas realizadas pelos atletas são armazenadas na relação "Matresp". Os nomes dos esportes disponíveis, os responsáveis, o valor investido e os locais onde se pratica cada esporte ficam armazenados na relação "Controle".

Finalmente, o componente SAE armazena na relação "Pessoa" os dados das pessoas desempregadas. Além disto, armazena na relação "Ult-emp" as informações relativas aos últimos empregos em que a pessoa trabalhou.

Terminada a fase de padronização dos componentes, passou-se para a fase de exportação. Neste caso, o usuário (Prefeitura) deseja exportar todos os dados. Desta forma, os esquemas exportados e os esquemas resultantes da fase de padronização são os mesmos.

5.2 Construção da Federação e Processamento de Consultas

Inicialmente, são criadas as tabelas exportadas de cada componente – por exemplo, a tabela 2 refere-se ao componente Educação. Em seguida, são criadas a tabela semântica (tabela 3) e a matriz de correspondência de atributos (tabela 4) relativas a estes componentes.

Uma vez criada a federação, os usuários podem realizar consultas tanto à federação quanto aos componentes locais. As consultas à federação precisam ser mapeadas em conjuntos de consultas aos componentes, sendo o resultado final passado ao usuário. Exemplos de consultas solicitadas são

- Mostrar bairros onde moram as pessoas desempregadas (espacial)
- Qual o bairro que possui mais pessoas estudando e praticando esportes (espacial)
- Quais as pessoas que praticam esporte e que estão desempregadas (textual)
- Qual o nome e a naturalidade das pessoas que não são naturais de Paulínia, estão matriculadas em algum curso, praticam esporte e estão desempregadas (textual)

A última consulta utiliza dados de todos os componentes federados e será usada para exemplificar o processamento de consulta nesta federação. A notação usada, descrita em

Semântica	*Atributo-Federado*
Identificação da pessoa	IDPessoa
Nome da pessoa	NomePessoa
Endereço da pessoa	EndereçoPessoa
Bairro da pessoa	BairroPessoa
Naturalidade da pessoa	NaturalPessoa
Telefone da pessoa	FonePessoa
Sexo da pessoa	SexoPessoa
Identificação do participante da atividade	PessoaAtividade
Nome da atividade	NomeAtividade
Turno da atividade	TurnoAtividade
Data de início da atividade	InícioAtividade
Classificação da atividade	TipoAtividade
Identificação da atividade	IDAtividade
Recursos destinados à atividade	GastosAtividade
Responsável pela atividade	ResponsávelAtividade
Endereço onde se realiza a atividade	EndereçoAtividade
Bairro onde se realiza a atividade	BairroAtividade
Identificação da pessoa desempregada	PessoaEmprego
Descrição do Último emprego da pessoa	DescriçãoEmprego
Intervalo de tempo de trabalho no emprego	PeríodoEmprego
Cidade do Último emprego da pessoa	CidadeEmprego

Table 3: Tabela semântica.

Atributo-Federado	*Educação*	*Esporte*	*SAE*
IDPessoa	Aluno.cod	Atleta.código	Pessoa.codp
NomePessoa	Aluno.nome	Atleta.nome	Pessoa.nomep
EndereçoPessoa	Aluno.endereço	Atleta.endereço	Pessoa.endp
BairroPessoa	Aluno.bairro-al	Atleta.bairro	Pessoa.bairrop
NaturalPessoa	Aluno.naturalidade	Atleta.origem	Pessoa.naturalp
FonePessoa		Atleta.fone	
SexoPessoa	Aluno.sexo	Atleta.sexo	Pessoa.sexop
PessoaAtividade	Matrícula.cod	Matresp.código	
NomeAtividade	Matrícula.escola	Matresp.nome-esp	
TurnoAtividade	Matrícula.turno	Matresp.turno	
InícioAtividade		Matresp.início	
TipoAtividade	Matrícula.tipo		
IDAtividade	Escola.ID	Controle.esporte	
GastosAtividade		Controle.investimento	
ResponsávelAtividade	Escola.responsável	ControleCoordenador	
EndereçoAtividade	Escola.endereço	ControleEndereço	
BairroAtividade	Escola.bairro	Controle.bairro	
PessoaEmprego			Ult-emp.codp
DescriçãoEmprego			Ult-emp.descrição
PeríodoEmprego			Ult-emp.período
CidadeEmprego			Ult-emp.cidade

Table 4: Matriz de correspondência de atributos.

Subconsulta criada	SELECT	Educação.[NomePessoa], Educação.[NaturalPessoa]
	FROM	
	WHERE	Educação.[NomePessoa] = Esporte.[NomePessoa]
	AND	Educação.[NaturalPessoa] ≠ "Paulínia"
Conversão de atributos	SELECT	Educação.Aluno.nome, Educação.Aluno.naturalidade
	FROM	
	WHERE	Educação.Aluno.nome = Esporte.Atleta.nome
	AND	Educação.Aluno.naturalidade ≠ "Paulínia"
Especificação de junções	SELECT	Educação.Aluno.nome, Educação.Aluno.naturalidade
	FROM	
	WHERE	Educação.Aluno.nome = Esporte.Atleta.nome
	AND	Educação.Aluno.naturalidade ≠ "Paulínia"
	AND	Educação.Aluno.cod = Educação.Matrícula.cod
	AND	Esporte.Atleta.código = Esporte.Matresp.código
Identificação dos objetos originais	SELECT	Educação.Aluno.nome, Educação.Aluno.naturalidade
	FROM	Educação.Aluno, Educação.Matrícula, Esporte.Atleta, Esporte.Matresp
	WHERE	Educação.Aluno.nome = Esporte.Atleta.nome
	AND	Educação.Aluno.naturalidade ≠ "Paulínia"
	AND	Educação.Aluno.cod = Educação.Matrícula.cod
	AND	Esporte.Atleta.código = Esporte.Matresp.código

Figure 1: Tradução da subconsulta envolvendo os componentes Educação e Esporte.

[Soa98], permite definir na consulta os componentes que darão origem a cada atributo. Esta consulta pode ser expressa como:

SELECT Educação.[NomePessoa], Educação.[NaturalPessoa]
FROM Educação, Esporte, SAE
WHERE Educação.[NomePessoa] = (Esporte, SAE).[NomePessoa]
AND Educação.[NaturalPessoa] ≠ "Paulínia"

O primeiro passo para responder a esta consulta é dividi-la, de acordo com os relacionamentos entre os componentes, produzindo subconsultas que juntam os componentes Educação e Esporte e os componentes Educação e SAE. Cada subconsulta retorna uma relação contendo as informações requeridas pelo usuário para cada componente. Sobre as tabelas resultantes é aplicada uma operação de união. A figura 1 apresenta o processamento da primeira subconsulta.

A partir desta federação, é possível inserir outros componentes de bancos de dados, criando outras tabelas exportadas e adicionando os novos atributos à matriz de correspondência de atributos e, se necessário, à tabela semântica. Exemplos de entidades que deverão constituir futuros componentes: hospitais, creches,transporte dentre outras.

O exemplo não mostrou como dados geográficos poderiam ser considerados. Para este caso, é preciso estabelecer ligações entre os atributos convencionais e os dados espaciais, processados pelo SIG. Em Paulínia, a opção foi a de efetuar esta ligação a partir dos atributos que armazenam endereços (por exemplo, Aluno.endereço, Atleta.endereço, Pessoa.endp). Pressupõe-se, além do mais, que todos os endereços sejam padronizados em um banco de dados único de endereços.

O mecanismo de mapeamento de consultas foi implementado em um protótipo no IC-UNICAMP. A figura 2 mostra um exemplo de tela com formulação de consultas. O usuário tem à disposição listas com os nomes dos atributos federados, os operadores disponíveis para formação de predicados e os nomes dos bancos de dados componentes.

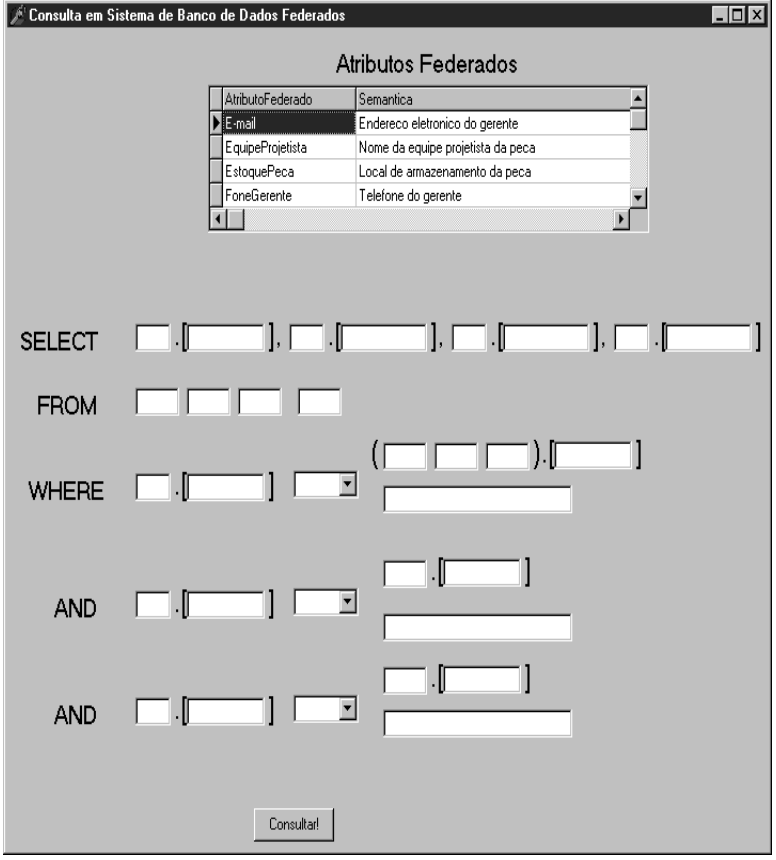

Figure 2: Tela para formulação de consultas no sistema federado.

6 Conclusões e Extensões

Este artigo apresentou uma metodologia para se estabelecer um sistema integrado de dados através da criação de um sistema de bancos de dados federados que possui como componentes bancos de dados heterogêneos e sistemas legados. Evidentemente, pode haver algumas variações nos passos desta metodologia, dependendo das particularidades dos ambientes computacionais de cada organização. No entanto, ela apresenta um avanço quanto à situação atual, estendendo um conjunto básico de passos a serem seguidos. Uma contribuição adicional é considerar sistemas de informações geográficas, apresentando as facilidades e limitações existentes na sua utilização em bancos de dados federados.

Ressalte-se que em muitos casos não é possível realizar a integração de dados e, nestes casos, a reconstrução dos sistemas passa a ser inevitável. O estudo de caso realizado na cidade de Paulínia possibilitou aplicar a metodologia utilizando especificações de bancos de dados reais, o que auxiliou o processo de refinamento da metodologia e permitiu associar problemas de ordem prática ao trabalho.

Há várias extensões possíveis, tanto teóricas quanto práticas. Do ponto de vista prático, é preciso realizar mais testes com dados reais, em especial utilizando sistemas legados de maior complexidade e dados espaciais. É preciso, igualmente, desenvolver ferramentas adicionais de apoio ao usuário, principalmente no que tange solucionar as ambigüidades geradas durante os processamentos de consulta. Do ponto de vista teórico, é preciso estender a metodologia considerando os aspectos de distribuição de dados, acesso concorrente e recuperação de informação. O trabalho não considerou estes aspectos porque a distribuição nem sempre é encontrada em sistemas federados. Finalmente, cabe um estudo mais aprofundado dos

problemas e soluções envolvendo integração de SIGs. Infelizmente, o projeto de Paulinia foi cancelado antes de serem terminados os mapas, o que impediu testes neste sentido.

Agradecimentos. Este trabalho foi parcialmente financiado pelo CNPq e pelo projeto PRONEX MCT Sistemas Avançados de Informação (SAI). A implementação do protótipo foi realizada por Ricardo Torres.

References

[Agu95] C. D. Aguiar. Integração de Sistemas de Banco de Dados Heterogeneos em Aplicações de Planejamento Urbano. Master's thesis, UNICAMP, march 1995.

[AMR94] P. Aiken, A. Muntz, and R. Richards. Dod legacy systems - reverse engineering data requirements. *Comm ACM*, 37(5):26–41, 1994.

[BDK92] P. Buneman, S. Davidson, and A. Kosky. Theoretical aspects of schema merging. *In Proc. Extending Database Technology*, pages 152–167, 1992.

[Ber96] P. A. Bernstein. Middleware. *Comm ACM*, 39(2):86–98, 1996.

[Bre90] Y. Breitbart. Multidatabases interoperabitily. *SIGMOD RECORD*, 19(3):53–60, 1990.

[BS95] M. L. Brodie and M. Stonebraker. *Migrating Legacy Sistems - Gateways, Interfaces and the Incremental Approach.* Morgan Kaufmann Publ., Inc., 1995.

[DK97] S. B. Davidson and A. S. Kosky. Wol: A language for database transformations and constraints. *13th Intl. Conf. on Data Engineering*, pages 55–65, 1997.

[Hul97] R. Hull. Managing semantic heterogeneity in databases: A theoretical perspective. *PODS97*, 1997.

[Int97] Interop. International Conference and Workshop on Interoperating Geographic Information Systems. Web site http://www.ncgia.ucsb.edu/conf/interop97, 12 1997. Site address valid as of 07/99.

[Jr95] G. S. Novak Jr. Creation of views for reuse of software with different data representations. *IEEE Transactions on Software Engineering*, 21(12):993–1005, 1995.

[MNB+94] L. Markosian, P. Newcomb, R. Brand, S. Burson, and T. Kitzmiller. Using an enabling technology to reengineer legacy systems. *Comm ACM*, 37(5):58–70, 1994.

[Mot98] R. Motz. Instanciating integrated schemas. *Anais XIII Simpósio Brasileiro de Banco de Dados*, pages 53–68, 1998.

[NEK94] J. Q. Ning, A. Engberts, and W. Kozaczynski. Automated support for legacy code understanding. *Comm ACM*, 37(5):50–57, 1994.

[Oli93] R. L Oliveira. Transparência de modelos em sistemas de bancos de dados homogêneos. Master's thesis, UNICAMP, 1993.

[PH95] G. Pernul and H. Hasenauer. Combining reverse with forward database engineering - a step forward to solve the legacy system dilemma. *DEXA 95*, pages 177–186, 1995.

[Qia96] X. Qian. Correct schema transformations. *International Conference on Extending Database Technology - EDBT*, 1996.

[SL90] A. P. Sheth and J. A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3):183–235, 1990.

[Soa98] H. R. Soares. Uma Metodologia para Integração de Sistemas Legados e Bancos de Dados Heterogêneos. Master's thesis, UNICAMP, 1998.

[SSR94] E. Sciore, M. Siegel, and A. Rosenthal. Using semantic values to facilitate interoperability among heterogeneous information systems. *ACM Transactions on Database Systems*, 19(2):254–290, 1994.

[VL97] V. M. P. Vidal and B. F. Lóscio. Especificação de mediadores para acesso e atualização de múltiplas bases de dados. *Anais XII Simpósio Brasileiro de Banco de Dados*, 1997.

[Zha97] J. L. Zhao. Schema coordination in federated database management: a comparison with schema integration. *Decision Support Systems*, 20:243–257, 1997.