UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Integração Materializada na Web: um Estudo de Caso

por

EIDY LEANDRO TANAKA GUANDELINE

Dissertação submetida à avaliação, como requisito parcial, para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. José Valdeni de Lima
Orientador

Porto Alegre, janeiro de 2002.

CIP – CATÁLOGAÇÃO DE PUBLICAÇÃO

Guandeline, Eidy Leandro Tanaka

Integração Materializada na Web: um Estudo de Caso/ por Eidy Leandro Tanaka Guandeline. Porto Alegre: PPGC da UFRGS.

94f.:il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientador: Lima, José Valdeni de.

1. Integração de dados 2. Dados semi-estrutrados. Wrappers. 4. Extração de Informação. I. Lima, José Valdeni de. II. Título.

Agradecimentos

Agradeço em primeiro lugar a Deus, pela oportunidade que tem me dado.

Ao meu orientador, José Valdeni de Lima, a quem agradeço por todas as coisas que realizou por mim dentro desta instituição e, por sua grande amizade a qual estimo.

Agradeço aos professores pelo profissionalismo mostrado em sala e fora dela, aos conselhos e oportunidades fornecidas para aumentar e desenvolver o conhecimento.

A todos os funcionários do Instituto que sempre se mostraram prestativos nas mais diversas áreas, secretaria, portaria, almoxarifado, rede e etc.

A todos os meus novos amigos que consegui durante estes dois anos nesta nova cidade. Amizades estas que espero poder manter por muitos anos.

Ao CNPq pela bolsa a mim fornecido, cuja ajuda foi imprescindível para a realização deste trabalho.

Enfim, agradeço a todas as pessoas que conheci neste instituto e que me fizeram companhia nestes dois anos.

A todos, muito obrigado por tudo...

Sumário

Lista de Abreviaturas

ADM	Araneus Data Model
ASP	Active Server Pages
CNPq	Conselho Nacional de Pesquisa e Desenvolvimento
DDL	Data Definition Language
DTD	Data Type Definition
HTML	Hypertext Markup Language
LORE	Lightweight Object Repository
ODMG	Object Data Management Group
OEM	Object Exchange Model
OSM	Object Oriented Systems Model
PROPESQ	Pró-Reitoria de Pesquisa da UFRGS
SQL	Structured Query Language
TSIMMIS	*The Stanford-IBM Manager of Multiple Information Sources*
UFRGS	Universidade Federal do Rio Grande do Sul
URL	Unique Resource Location
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
XSL	Extensible Style Sheet Language

Lista de Figuras

Lista de Tabelas

Resumo

A World Wide Web em poucos anos de existência se tornou uma importante e essencial fonte de informação e a localização e recuperação de informações na Internet passou a ser um grande problema a ser resolvido. Isto porque a falta de padronização e estrutura adequada para representação dos dados, que é resultado da liberdade de criação e manipulação dos documentos, compromete a eficácia dos modelos de recuperação de informação tradicionais.

Muitos modelos foram então desenvolvidos para melhorar o desempenho dos sistemas de recuperação de informação. Com o passar dos anos surge assim uma nova área de pesquisa a extração de dados da web que, ao contrário dos sistemas de recuperação, extrai informações dos documentos relevantes e não documentos relevantes de conjunto de documentos.

Tais estudos viabilizaram a integração de informações de documentos distribuídos e heterogêneos, que foram baseados nos mesmos modelos aplicados a banco de dados distribuídos.

Neste trabalho é apresentado um estudo que tem como objetivo materializar informações contidas em documentos HTML de modo que se possa melhorar o desempenho das consultas em relação ao tempo de execução quanto à qualidade dos resultados obtidos.

Para isso são estudados o ambiente web e as características dos dados contidos neste ambiente, como por exemplo, a distribuição e a heterogeneidade, aspectos relacionados à maneira pela qual as informações estão disponibilizadas e como estas podem ser recuperadas e extraídas através de regras sintáticas.

Finalizando o estudo são apresentados vários tipos de classificação para os modelos de integração de dados e é monstrado em detalhes um estudo de caso, que tem como objetivo demonstrar a aplicação das técnicas apresentadas ao longo desta pesquisa.

Palavras-chave: Integração de dados, Dados Semi-Estrutrados, *Wrappers*, Extração de Informação.

TITLE: “INTEGRATION OF SEMI-STRUCTURED DATA OBTAINED FROM THE WEB: A CASE STUDY”

Abstract

In the last few years, the World Wide Web has become an important source of information. However, information searching and retrieving is a hard task due to the lack of standardization and structuring necessary to data representation. The manipulation of these data as complex documents affects the efficiency of traditional models of information searching.

Many models have been developed to improve the performance of information retrieval systems, and a new research area has emerged: Web data extraction, which differs from conventional retrieval systems since aims at obtaining relevant information from documents instead of relevant documents from a set of documents.

These studies enabled the integration of information from distributed and heterogeneous documents, based on the same models applied to distributed databases.

This work aims at materializing information from HTML documents. Such materialization could increase query or search performance with respect to both execution time and quality of results.

To achieve this goal the web environment and its data aspects were studied concerning, for example, data distribution, data heterogeneity, information structuring, information extraction through the use of syntactic rules, etc.

Integration models are classified and described in details. Finally, a case study is presented to demonstrate the suitability of the techniques proposed in this work.

Keywords: Data Integration, Semistructured Data, Wrappers, and Information Extraction.

1 Introdução

A World Wide Web em poucos anos de existência se tornou uma importante e essencial fonte de informação e a localização e recuperação de informações na Internet passou a ser um grande problema a ser resolvido. Isto se deve à falta de padronização e estrutura adequada para representação dos dados, que é resultado da liberdade de criação e manipulação dos documentos, e que compromete a eficácia dos modelos de recuperação de informação tradicionais.

No início do desenvolvimento de sistemas de recuperação de informação dois modelos se destacaram, o *browsing*, utilizado pelo Yahoo, e o *search engine*, utilizado pelo AltaVista. No primeiro modelo, os documentos estavam divididos em grupos de acordo com o assunto e domínio de interesse; já no segundo a pesquisa é realizada em um banco de dados que possui informações sobre um grande conjunto de documentos, sendo que algumas bases de dados possuem um índice invertido para melhorar a eficiência da consulta.

Estes modelos foram sendo aperfeiçoados para melhorar o resultado de suas consultas, mas ainda hoje, um problema comum que atinge estes tipos de sistemas é a quantidade de documentos irrelevantes em seus resultados.

Outro problema que surgiu nos últimos anos é a necessidade de não apenas encontrar documentos relevantes e sim localizar e extrair informações específicas do conteúdo destes documentos. Surge neste contexto o processo de extração de informação na Web que, diferentemente do processo de recuperação de informação, recupera informações contidas nos documentos e não documentos como um todo.

Neste trabalho serão abordados tanto os sistemas de recuperação de informação quanto os sistemas de extração de informação, de forma que a união destas tecnologias possa promover o aperfeiçoamento das tecnologias de sistemas de integração de informação a partir de documentos distribuídos.

1.1 Motivação

Com o crescimento da quantidade de informações disponibilizadas na Internet, surgiu a necessidade de se criar sistemas de recuperação de informação. O objetivo é selecionar um conjunto de documentos relevantes que fazem parte de uma grande coleção de acordo com alguns parâmetros especificados pelo usuário [EIK 99]. Estes sistemas trabalham, na sua maioria, com a busca por palavras chaves, meta informações e índices para localização dos documentos.

Porém, na medida em que a quantidade de informação disponibilizada na Web foi aumentando, estes sistemas foram perdendo sua eficácia. Isto se deve ao fato de que a maioria das técnicas aplicadas foi desenvolvida para dados com características diferentes das encontradas no ambiente Web, que na sua maioria se apresentam em forma de documento semi-estruturados no formato HTML - *Hypertext Markup Language*.

Os primeiros sistemas de recuperação de informação foram desenvolvidos utilizando tecnologias tradicionais de banco de dados, ou seja, os dados possuem uma estrutura fixa, tipo determinado e um esquema rígido.

Já o ambiente Web possui características totalmente diferentes dos dados contidos em banco de dados. Os dados estão contidos numa estrutura que pode variar de documento para documento. Os dados não possuem um tipo e esquema fixo, sendo alterados freqüentemente pelo autor do documento, que tem total liberdade de modificação da estrutura e conteúdo.

Com o objetivo de solucionar alguns destes problemas, a pesquisa realizada visa estudar a interação de várias tecnologias para um melhor controle sobre os dados encontrados na Web, e com isso obter resultados de consultas mais satisfatórios, melhorando o desempenho na recuperação de informações no ambiente web.

1.2 Objetivos

O problema tratado neste trabalho aprofunda-se em um nível de granularidade da informação requerida. A maioria dos sistemas desenvolvidos tem como objetivo encontrar documentos relevantes a partir de parâmetros fornecidos pelo usuário, mas poucos são capazes de extrair informações específicas do conteúdo do documento.

Para melhorar a eficácia e a eficiência dos métodos de consulta de informações, o presente trabalho propõe a integração entre as tecnologias de recuperação de informação, extração de informação e banco de dados, utilizando as melhores técnicas de trabalhos já existentes.

Os sistemas de recuperação de informação possibilitam a recuperação de um subconjunto de documentos de acordo com alguns parâmetros indicados pelo usuário. Alguns sistemas podem possuir além da simples passagem de parâmetros, outras funcionalidades na tentativa de melhorar os resultados ou simplesmente auxiliar o usuário.

As tecnologias de extração de informação têm como objetivo transformar texto em uma forma estruturada e com isso reduzir a informação do documento em uma estrutura tabular [EIK 99] que possa ser consultada através de alguma linguagem de consulta. Ao contrário dos sistemas de recuperação de informação que recuperam documentos de uma coleção, os sistemas de extração de informação extraem informações relevantes dos documentos.

Depois de extraídos os dados de acordo com regras de extração, estes podem ser armazenados em um banco de dados que possui um esquema conceitual global definido para representar o domínio da informação a ser processada. Este esquema é uma visão canônica que tenta representar de forma única os diversos esquemas das fontes dos dados extraídos. Com os dados armazenados em banco de dados, os processos de consulta podem ser realizados em um ambiente com maior controle sobre os dados e desta forma melhorar seus resultados como tempo de consulta, relevância dos dados encontrados e granularidade das informações fornecidas.

Como objetivo, será realizado um estudo de caso que possa utilizar as melhores tecnologias de recuperação e extração de informação, integrando fontes de dados heterogêneas, materializando os dados de acordo com as regras de extração definidas para o domínio da informação escolhido. Assim poderemos também identificar as melhores práticas e os principais problemas neste tipo de aplicação.

1.3 Estrutura do Trabalho

Este trabalho possui a seguinte estrutura. No capítulo 2 é apresentada uma visão geral sobre o ambiente Web, que se tornou uma das maiores fontes de informação, os problemas de recuperação causados pelo crescimento desorganizado e a heterogeneidade de dados. Neste capítulo também é realizada uma classificação dos documentos segundo a estrutura do conteúdo dos documentos. Por último é apresentado um breve estudo referente aos aspectos mais importantes das informações na web: dados heterogêneos e distribuídos.

A partir destas características são apresentados modelos de representação para dados semi-estruturados e a classificação de acordo com características dos dados distribuídos, tais como: autonomia, heterogeneidade e distribuição.

No capítulo 3 são discutidos os modelos de recuperação de informação, suas vantagens e desvantagens. Também são apresentados o processo de extração de informação e os componentes que são utilizados para este processamento. São discutidos dois modelos de extração de informação: o sintático com base na estrutura do documento e sintático com base na estrutura do elemento.

No capítulo 4 são discutidas a interoperabilidade entre sistemas, as técnicas mais utilizadas, a importância dos metadados no processo de interoperabilidade e dois modelos são apresentados: integração virtual e integração materializada. Após esta introdução são expostas as vantagens e desvantagens dos sistemas de integração materializada.

Por fim, é apresentado um estudo de caso, onde os modelos de classificação apresentados durante o trabalho são aplicados e as técnicas são utilizadas, finalizando com as conclusões e os trabalhos futuros nesta área de pesquisa.

1.4 Trabalhos Relacionados

A necessidade de consultar os dados na web através de linguagens de consulta gerou inúmeras pesquisas. Tais pesquisas podem ser divididas em duas correntes distintas. A primeira tinha como objetivo criar linguagens de consulta apropriadas para dados semi-estruturados na web. Logo surgiram as primeiras linguagens para consulta como a W3QS [KON 95], WebOQL [ARO 97], [ARO 98], outros estudos são exibidos em [MEN 96] e [LAK 96]. Nestes estudos a web é

considerado um grafo onde os nós são as fontes de informação e os arcos são os relacionamentos entre os dados.

A proposta da segunda corrente é criar estruturas de dados capazes de representar os dados semi estruturados que são a maioria dos documentos encontrados na web e a partir da extração, integrá-los nesta estrutura mais complexa. O estudo pioneiro nesta área é o trabalho apresentado em [CHA 94] que descreve o projeto TSIMMIS que tem como objetivo integrar bases de informação heterogêneas.

Estes sistemas são baseados em *wrappers* que tem a função de extrair e combinar os dados extraídos em uma estrutura de dados mais complexa. Wrappers possuem regras de extração que podem ser criadas manualmente [BUN 96], [ATZ 97], [ABI 97], ou através de software de auxilio [ASH 97], [KUS 97], [ADE 98]. Com isto os dados são exportados para modelos de dados específicos para dados semi estruturados como o OEM [ABI 97], Araneus [ATZ 98], YAT [CHR 98], LORE [ABI 97]. Para consultar estas informações foram criadas linguagens de consulta apropriadas como a UnQL [BUN 96], Lorel [ABI 97], StruQL [FER 97]. As diversas técnicas apresentadas foram discutidas em [FLO 98] de modo a dar uma visão geral do problema e as possíveis soluções. [EIK 99] faz um estudo sobre os diversos métodos para a criação de wrappers.

A partir do amadurecimento destas técnicas de extração de dados, ocorrida entre os anos de 1997 a 1998, iniciaram-se as primeiras tentativas de exportação para modelos de banco de dados que pudessem ser consultados através de linguagens de consulta como o SQL. Alguns estudos de materialização de dados na web em banco de dados relacionais são apresentados em [LAB 2000] e [ROS 2000]. Estes modelos possuem vários tipos de problemas tais como manutenção dos dados extraídos [ABI 99], [GUP 2001], identificação de inconsistências [COM 98], [TEJ 98], [LIU 99], esquemas de integração [GAR 2000], [JOH 2001].

Os estudos mais recentes focam a utilização do padrão XML - *Extensible Markup Language* - para a criação das fontes de informação que serão processados, descrição de regras de extração e como modelo de dados para exportação de dados já processados.

2 Ambiente Web

A Web providencia a seus usuários um acesso rápido e flexível a vários tipos de aplicação [MAN 96], sendo composta basicamente de vários documentos conectados por "*links*", onde o usuário navega entre os documentos. Muitos destes documentos fazem parte de um grupo de informação, ou seja, dizem respeito a um mesmo contexto de informação [LAW 2000]. Vários autores diferentes publicam seus documentos e conectam seus documentos a outros do mesmo domínio, criando o que podem ser chamadas de ilha de informação, ou seja, vários documentos interconectados referentes a um mesmo assunto.

A Internet pode ser considerada a maior e mais heterogênea fonte de informação já construída pelo homem, e em poucos anos de existência se tornou uma importante e essencial fonte de informação. Devido às características encontradas neste ambiente a recuperação de informação foi comprometida e, ,como resultado, as consultas realizadas por mecanismos especializados geram uma grande quantidade de documentos sem relevância alguma.

Tanto o processo de recuperação quanto o de extração de informação são prejudicados pelas características dos dados encontrados na web. Estas características são apresentadas a seguir, sendo que as mais importantes são expostas com maiores detalhes nos capítulos posteriores. Estas informações são importantes, pois definirão as características dos dados que devem ser processados. São elas:

- Distribuição das informações. A Internet é um grande ambiente distribuído. Nela conseguimos recuperar informações em todas as partes do mundo, que estão armazenadas em vários tipos de plataformas, em várias linguagens e etc. Documentos neste ambiente possuem total autonomia para realizar todo o tipo de operação, inclusão, alteração ou deleção, sem a necessidade de informar a nenhum outro documento a ele relacionado.

- Alto volume de dados é uma das características mais visíveis da web. De acordo com estudos realizados, havia 350 milhões de documentos em 1998 e a estimativa é de um bilhão de documentos no ano 2000, sendo que cerca de 30% dos documentos na Internet são duplicados, ou seja, possuem o mesmo conteúdo com mesmas ou diferentes estruturas de apresentação, alterando somente sua localização na web. Fonte: www.notess.com

- A atualização dos dados também é freqüente na Internet. Os documentos possuem na sua maioria um alto índice de alterações. Seus conteúdos são alterados, podendo mudar de endereço ou simplesmente deixar de existir, com isto sistemas de recuperação de informação podem possuir dados inválidos em suas bases de dados o que representa um quebra de integridade das informações.

- Os dados na web são heterogêneos e dificultam a classificação das informações. A enorme quantidade de tipos de documentos gerados pelos mais diversos autores e ferramentas causou uma heterogeneidade muito grande entre os documentos. Logo uma metodologia de classificação pode ser muito eficaz em uma classe de documentos e ser totalmente ineficaz em outra. Outro fator ligado à heterogeneidade é a composição de documentos com objetos multimídia tais como imagens, sons, vídeos e áudio que aumentam ainda mais a complexidade dos documentos e dificulta a exata classificação do conteúdo do documento.

- Os dados na web são semi estruturados e não possuem um esquema fixo e bem definido como nos modelos de banco de dados. A estrutura do documento varia de documento para documento e é definida pelo usuário que cria o documento de acordo com suas necessidades, expectativas e conhecimento, assim objetos comuns no mundo real podem possuir campos multivalorados e de tipos variados.

Todas estas características geram problemas para os sistemas de recuperação de informação, pois, para se criar um mecanismo eficiente de busca de informações deve-se ter conhecimento do tipo de documentos que serão utilizados. Mas no ambiente Web não existe um tipo padrão, cada autor cria seu próprio formato. Logo, para realizar a integração de informações é preciso classificar os documentos de maneira que se possam criar mecanismos mais eficientes e direcionados a solucionar uma parte do problema.

Uma característica que pode ser utilizada para a classificação dos documentos na web é a estrutura da informação dentro do conteúdo do documento. Com isto podemos classificar os documentos de acordo com as características da tabela 2-1.

TABELA 2.1 - Classificação de Documentos segundo a sua Estrutura

Tipo	**Característica**
Texto Livre	Originalmente, os estudos relacionados à extração de informação foram baseados em sistemas para processamento de pequenos trechos de linguagem natural, limitados a um domínio específico de informação. Estes sistemas utilizavam técnicas de linguagem natural e regras de extração para resolver o problema. As regras de extração eram baseadas em modelos que envolviam relações sintáticas e semânticas das palavras. Estes estudos estão longe de conseguir realizar o processamento de linguagem realizado pelo ser humano.
Texto Semi-Estruturado	São dados intermediários entre o texto livre e o estruturado. Não possuem a liberdade completa da linguagem natural, nem estão totalmente vinculados a uma estrutura pré-definida, ou possuem um formato rígido.
Texto Estruturado	Esta informação textual está normalmente armazenada em banco de dados ou em um formato pré-definido. Informações deste tipo podem ser facilmente manipuladas e trabalhadas, pois possuem um esquema conceitual bem definido e são fortemente tipadas, impedindo a incompatibilidade entre atributos que representam a mesma informação do mundo real.

A World Wide Web é uma grande coleção de documentos inter-relacionados e, além da estrutura que pode variar, os documentos são dinâmicos e possuem conexões com outros documentos que podem ser utilizados para compartilhar ou estender as informações.

Levando em consideração a classificação dos documentos da Web segundo o seu grau de estruturação, o trabalho descrito em [HSU 98] descreve a seguinte classificação para uma página na Web:

- um documento estruturado é aquele em que cada atributo em uma tupla pode ser extraído baseado em regras sintáticas;
- um documento semi-estruturado é aquele em que cada atributo em uma tupla pode ser extraído baseado em regras sintáticas, porém, podem

faltar atributos, ou possuírem atributos multivalorados ou outro tipo de exceções;

- documento sem estrutura é aquele que necessita de conhecimentos de linguagem natural para a extração de dados. Com isto podemos afirmar que a classificação depende da maneira pela qual os dados estão organizados dentro do documento.

A maior parte dos dados encontrados na web possui um certo grau de organização [EIK 99], com isto, pode-se utilizar esta regularidade de aparência para se definir regras de extração sintática sem utilizar técnicas de processamento de linguagem natural. Isto traz um melhor desempenho para estes sistemas pois, as técnicas de processamento de linguagem natural, geralmente, são mais lentas e isto poderia causar uma grande lentidão no processamento de consultas devido ao alto volume de documentos.

Com isto a web não pode ser vista como um banco de dados, pois não possui uma estrutura regular, regras de integridade, transações, modelo de dados ou uma linguagem padrão para consulta [FLO 98]. Logo, os documentos devem passar por um tipo de pré processamento no qual as informações obtenham um nível superior de abstração e com isto possam ser armazenados em uma estrutura de dados mais complexa. Nos próximos capítulos são discutidas as principais características dos dados na web.

2.1 Dados Semi-Estruturados

Segundo [BUN 97], [FLM 98] dados semi-estruturados são heterogêneos, ou seja, não são rigorosamente tipados e não possuem esquemas bem definidos, que em muitos casos está implícito no contexto da informação. O esquema, na web, descreve o estado atual dos dados, mas não impõe regras de integridade como no modelo relacional. Com isto o esquema é alterado freqüentemente. Podemos ainda citar mais características dos dados semi-estruturados como as apresentadas em [ABI 96], [FLO 98] que são expostas na tabela 2-2.

TABELA 2.2 - Característica dos Dados Semi-Estruturados

Característica	**Descrição**
Estrutura Irregular	Os dados semanticamente similares podem possuir diversos formatos.
Estrutura Implícita	Para se obter a estrutura dos dados semi-estruturados deve-se fazer uma análise do documento onde se torna possível isolar e estabelecer relacionamentos entre os dados.
Estrutura Parcial	Alguns dados encontrados não apresentam uma estrutura implícita, como no caso de imagens e áudio, sendo que é ilusório criar uma estrutura completa para os dados semi-estruturados.
Estrutura Indicativa	A estrutura gerada a partir da análise do documento é indicativa, ou seja, ela indica os tipos de dados atuais que estão sendo utilizados no documento. No modelo de banco de dados relacionais a estrutura é restritiva, os tipos de dados devem pertencer a um domínio específico.
Esquema Posterior	Esquemas de dados semi-estruturados são criados depois da análise do documento; em modelos de banco de dados o esquema é definido em primeiro lugar. Logo o esquema de dados semi-estruturados é adaptativo enquanto o outro é restritivo.
Esquema Grande	Devido à heterogeneidade dos dados, o esquema em muitos casos pode ser maior que os dados em si, diferentemente dos esquemas definidos para os bancos de dados tradicionais.

Devido aos vários problemas encontrados para se criar um esquema, alguns estudos ignoram estes esquemas e as consultas são realizadas de forma navegacional sobre os dados. Porém deve-se tentar manter uma descrição estrutural da base de dados semi-estruturada para que se possa facilitar a utilização de linguagens de consulta de alto nível e a troca de informações entre bases de dados, fornecendo uma visão global do contexto da aplicação ao usuário.

Documentos semi-estruturados não podem ser consultados por linguagens padrão como o SQL. Deve-se usar uma linguagem mais flexível devido ao fato que não se conhece, por completo, a estrutura dos dados como em um banco de dados relacional. Uma linguagem de consulta para dados semi-estruturados deve possuir algumas características que possibilitem uma melhor iteração com estes tipos de dados. Algumas características presentes são a navegação entre os dados de um documento, e a possibilidade de se realizar consultas em um esquema representativo dos dados ou nos dados diretamente.

As primitivas das linguagens de consulta devem ser mantidas. Portanto, o problema de se realizar consultas mais poderosas em documentos semi-estruturados não é resolvido apenas com a utilização de técnicas de banco de dados. Surgem então duas correntes de estudo para a solução deste problema: a criação de linguagens de consulta [ARO 98] e a geração de *wrappers* como componentes da consultas [ATZ 97].

Para se utilizar às linguagens de consulta para dados semi-estrutrados deve-se primeiramente exportar estes dados para um modelo de representação. Na próxima seção são apresentados brevemente alguns destes modelos.

2.1.1 Modelos de Representação

[ABI 97] OEM - *Object Exchange Model* - é um modelo para representação de dados semi-estruturados onde as informações são representadas em forma de um grafo. Neste grafo, cada nó representa um objeto, uma informação, e cada objeto possui seu identificador único.

Porém, neste modelo de representação os dados não possuem uma estrutura rígida, ou seja, os dados podem possuir atributos multivalorados e de tipos diferentes. A linguagem utilizada para consultar estes dados é a LOREL.

A figura 2-1 apresenta um exemplo da utilização do modelo OEM para representar informações referentes a restaurantes.

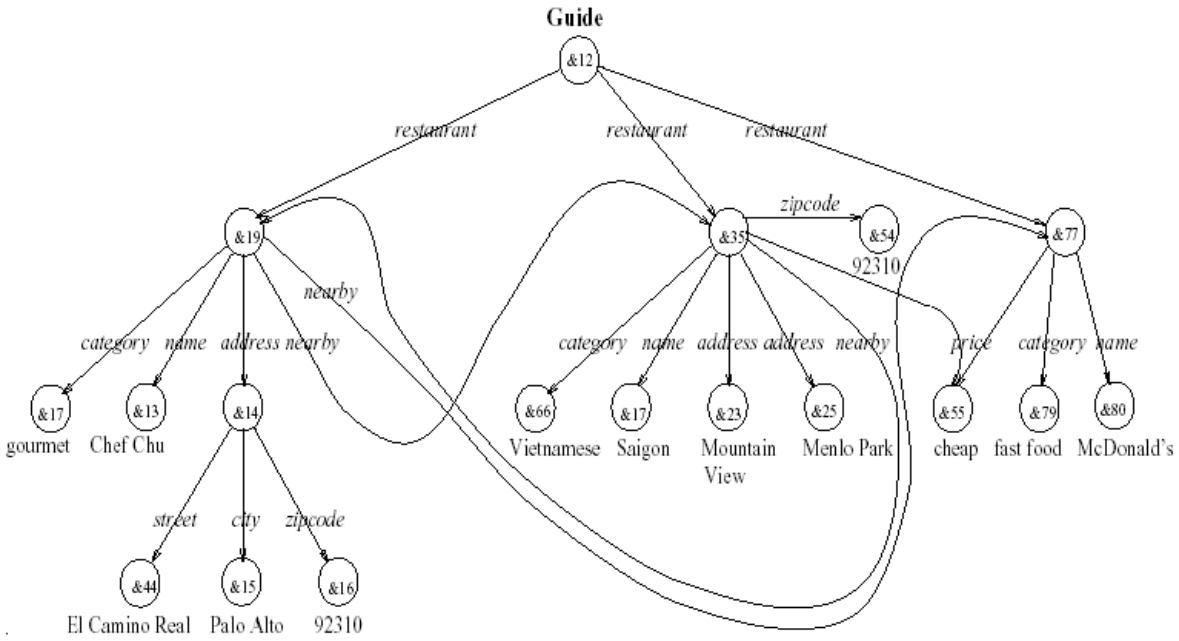

FIGURA 2.1 - Exemplo de Modelo OEM

[ATZ 98] *Araneus Data Model* é um modelo orientado à objetos e pode ser considerado um subconjunto da ODMG, em que o esquema é utilizado para descrever a estrutura do conjunto de dados em um *site*. Neste modelo cada página tem um identificador, no caso a URL do documento, e seus atributos. Os atributos podem ser simples como textos e imagens ou complexos como listas de itens.

A figura 2-2 representa o processo de definição de uma visão no modelo Araneus. Primeiramente os dados são processados por uma linguagem chamada EDITOR [ATZ 97] que é capaz de pesquisar e reestruturar o texto. As informações extraídas são adicionadas ao ADM – *Araneus Data Model*. A partir deste esquema, Ulixes e Penélope geram visões relacionais dos dados da web que podem então ser consultadas.

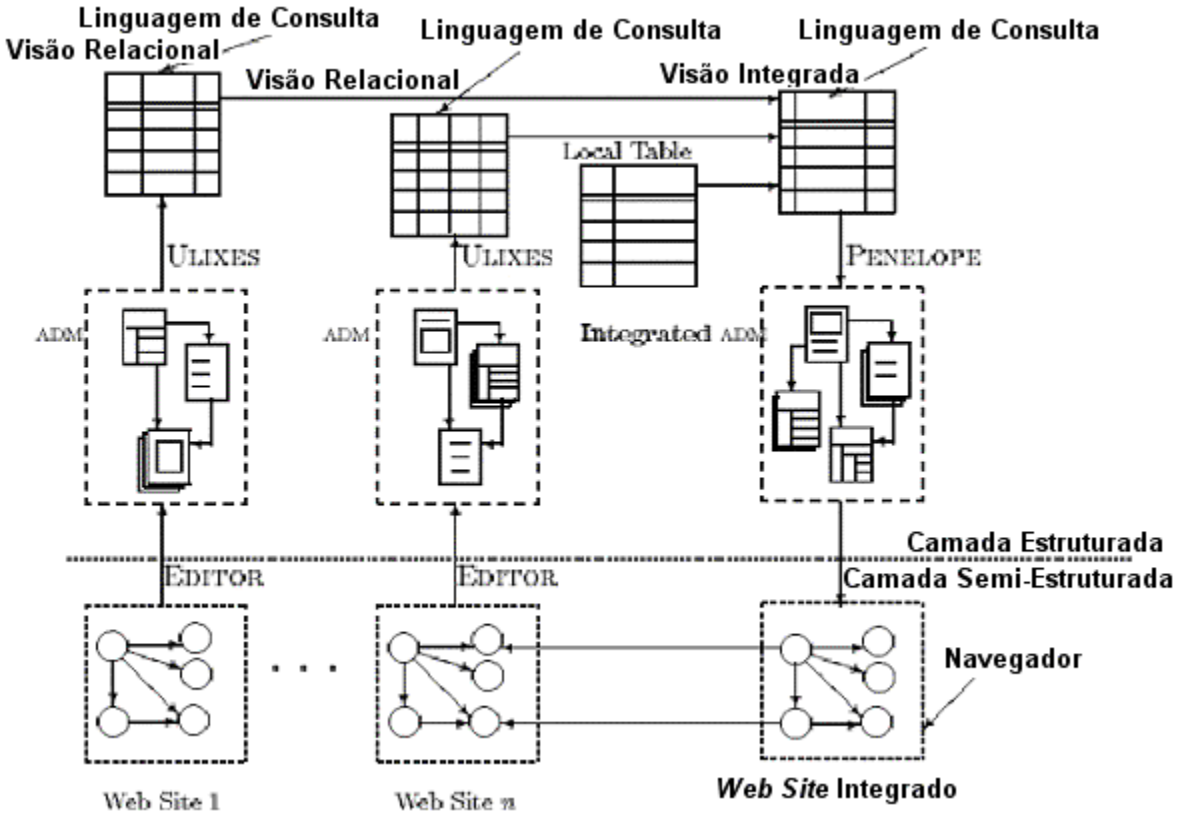

FIGURA 2.2 - Modelo de Execução no Sistema Araneus

A figura 2-3 representa um esquema em ADM – *Araneus Data Model* para dados bibliográficos. Como se pode notar, a notação é muito parecida com a utilizada nos modelos baseados na ODMG.

Outros modelos para dados semi-estruturados podem ser vistos em [LAH 99] *Ozone Data Model* [LAH 98] e [CHR 98] *Yat Data Model*.

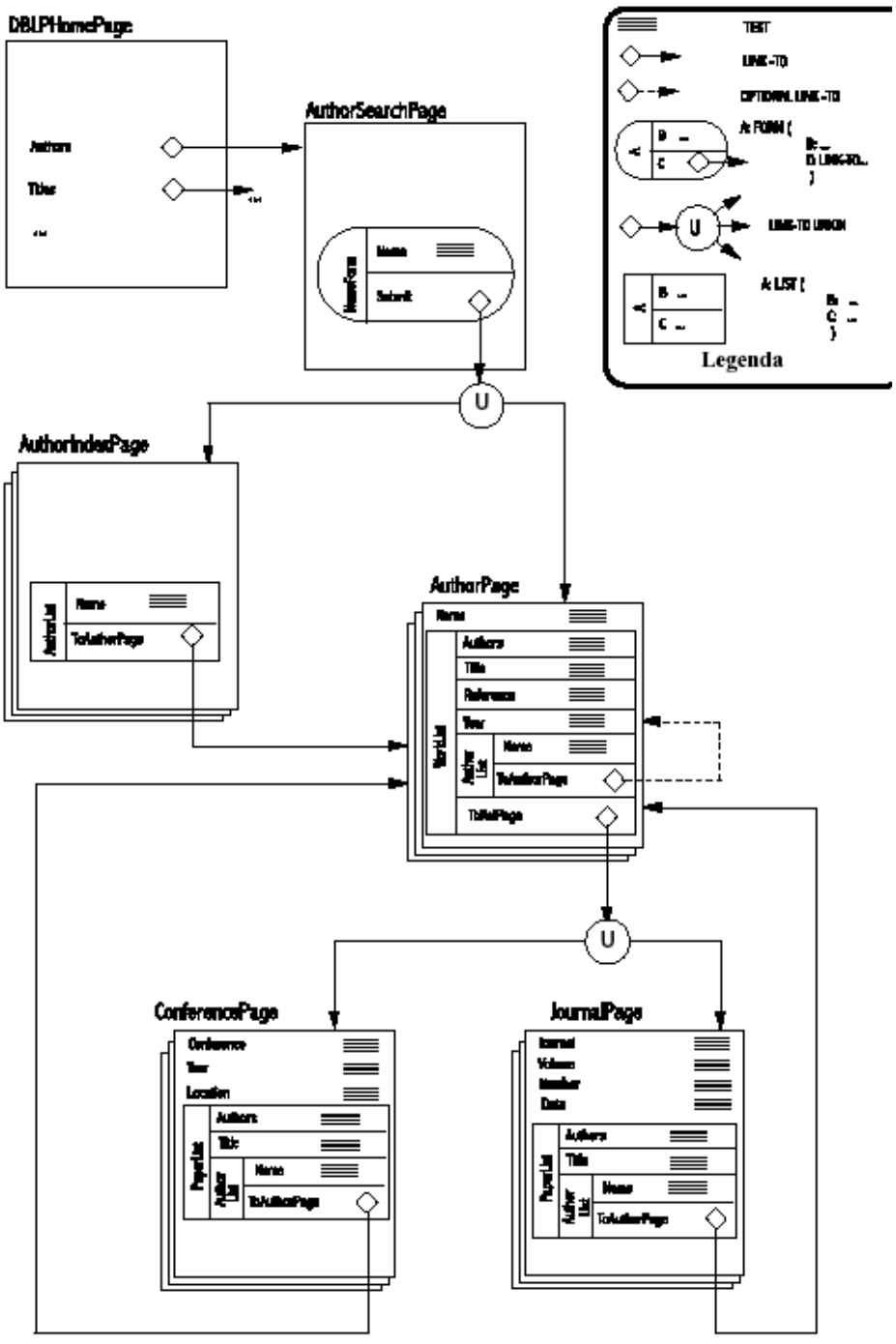

FIGURA 2.3 - Modelo de Representação ADM

2.2 Dados Distribuídos

Os dados na web são distribuídos por natureza. Estes dados podem ser classificados de acordo com o seu grau de autonomia, heterogeneidade e distribuição. Esta classificação tem como objetivo estabelecer relacionamentos e esquemas individuais para que se possa modelar um sistema de integração de dados.

A década de 80 pode ser vista como o momento de transição, onde aplicações proprietárias e/ou incompatíveis foram transferidas para sistemas de gerenciamento de banco de dados centralizados. Após esta fase, veio a necessidade de se combinar informações armazenados em sistemas de banco de dados diferentes ou distribuídos e, foi em 1985 que [HEI 85] definiu o termo “multidatabase systems” e mais tarde em 1990 surgiu o termo “federated database” [LIT 90].

As informações disponibilizadas na web podem ser vistas como um repositório de dados, mas não como um banco de dados, visto que estas não possuem um esquema ou linguagem de consulta padrão. Mesmo assim existem vários aspectos semelhantes aos sistemas de banco de dados distribuídos que serão abordados neste capítulo.

De acordo com [BUS 90], um sistema de informação providencia acesso à informação, baseado em dados que são gerenciados, ou seja, a web pode ser considerada um sistema de informação, pois provê acesso às informações contidas diretamente no documento ou em um banco de dados.

A classificação dos sistemas de informação no ambiente web de acordo com alguns conceitos de classificação de sistemas de informação é apresentada na tabela 2-3.

TABELA 2.3 - Características dos Sistemas de Informação

Característica	**Descrição**
Autonomia de projeto	Estes podem ser projetados independentemente um dos outros, criando modelos, conceitos e estruturas diferentes para um mesmo domínio de informação.
Autonomia de Comunicação	Na web podemos interligar um documento a vários outros documentos sem informar a estes outros sistemas.
Autonomia de Execução	Sistemas de recuperação de informação

	podem ser executados independentemente. Com isto, se torna difícil, por exemplo, definir transações globais, já que existe uma autonomia de execução.
Heterogeneidade Técnica	Na internet são encontrados os mais diversos tipos de hardware e software, porém, estas barreiras foram quebradas com a utilização de padrões de comunicação utilizados na web.
Heterogeneidade de Interface	Em sistema de informação este tópico está relacionado aos diferentes tipos de linguagem de consulta. Na web este tópico estaria mais bem relacionado com as diferentes formas de apresentar um conteúdo, já que uma mesma informação pode ser apresentada com várias tecnologias diferentes.
Heterogeneidade de Modelo de Dados	A liberdade causada pela autonomia de projeto fez da Internet a maior fonte de informação heterogênea conhecida. Cada documento pode apresentar a mesma informação com modelos semânticos totalmente diferentes.
Distribuição	A internet pode ser vista como um grande grafo interconexo, onde os nodos são os documentos e os arcos são os *links* que interligam os documentos.

Considerando então a World Wide Web como um grande sistema de informação, e levando em consideração os aspectos relacionados à autonomia, heterogeneidade e distribuição, a Web pode ser considerada um sistema distribuído, pois os dados estão armazenados em diferentes locais e é um sistema heterogêneo visto que existem diferenças de modelo de semântico e de estrutura.

3 A Informação no Ambiente Web

Até o momento discutimos as características dos dados encontrados no ambiente Web e podemos perceber que são inúmeros os problemas a serem resolvidos. Neste capítulo estaremos apresentando de maneira sucinta como estas informações podem ser recuperadas de acordo com as características discutidas anteriormente, e qual o nível de granularidade da informação que se pode obter com a junção de várias tecnologias.

3.1 Recuperação de Informação

É fato que os mecanismos de busca não satisfazem os usuários. Estes sistemas devolvem uma grande quantidade de informações que na sua maioria são parcialmente ou totalmente irrelevantes [GLO 2000].

Grande parte dos usuários não utiliza corretamente os mecanismos de busca, que na sua maioria trabalham com modelos de recuperação de informação, baseado na teoria dos conjuntos e na álgebra boolena, onde as consultas são representadas por expressões booleanas. O modelo booleano considera que um documento é verdadeiro ou falso para uma determinada expressão, expressão esta que representa os parâmetros informados pelo usuário.

As empresas de sistemas de recuperação de informação, tais como o Altavista e Yahoo, possuem dados estatísticos indicando que 70% dos usuários utilizam apenas um termo na consulta, que 80% dos usuários não acessam a segunda página de respostas e 78% não alteram o formato de sua consulta - fonte http://www.notess.com.

Entre os usuários, a principal reclamação é a velocidade de processamento dos mecanismos de busca, seguida de resultados irrelevantes e/ou com “*links mortos*”.

Somente em terceiro lugar aparece a falta de habilidade de se achar informações relevantes por parte dos usuários, onde muitos deles se dizem incapazes de utilizar os recursos adicionais disponibilizados pelos sistemas de recuperação de informação e desconhecem completamente a utilização de álgebra booleana em suas consultas.

Em http://www.gvu.gatech.edu/user_surveys são identificados outros problemas de menor relevância dos sistemas de recuperação de informação.

Outro problema dos mecanismos de busca é a interface utilizada para se fazer consultas, que muitas vezes não permite expressar as necessidades de um usuário em específico.

Para produzir um melhor conjunto de respostas aos usuários, os mecanismos de busca na Internet sofreram vários processos até chegarem aos modelos que nós conhecemos atualmente. A maioria dos sistemas de busca de informações na Web utiliza-se de métodos sintáticos para a consulta de documentos. Porém, este método gera muitas imprecisões, pois não se leva em consideração à semântica das informações do documento.

Sistemas de recuperação de informação ideais deveriam suportar algum tipo de processamento semântico ou processamento de linguagem natural combinados com conhecimentos de documentos semi-estruturados e banco de dados. Com isso se poderia conseguir uma melhor recuperação de documentos relevantes, mas aumentaria muito o tempo de processamento.

Segundo [LAW 98], outro problema que reduz a eficácia destes sistemas é a não consideração do contexto no momento em que se ordenam as informações processadas.

No momento, sistemas de processamento de linguagem natural têm sido usados para melhorar a recuperação de informação com o estudo da semântica dos documentos. Estes sistemas utilizam-se de técnicas de recuperação como *full text retrieval*, metadados, unidos a uma determinada ontologia.

Uma ontologia é um modelo de referência semântica com estruturas que contêm e descrevem relações entre os dados. Os sistemas atuais utilizam técnicas de indexação e classificação por agentes inteligentes que utilizam um processador semântico para melhorar sua classificação dos documentos.

Como dito anteriormente, sistemas de recuperação de informação têm como objetivo recuperar documentos que estejam relacionados à consulta realizada pelo usuário. As técnicas tradicionais foram desenvolvidas para trabalharem com dados armazenados em banco de dados ou em uma estrutura de dados capaz de representar os dados de forma mais estruturada.

Porém, a web possui uma grande quantidade de documentos semi-estruturados e heterogêneos, com isto as tecnologias tiveram que se adaptar para conseguir produzir resultados mais satisfatórios.

Este processo de recuperação de informação tem grande peso na eficácia no método de integração, pois é a partir dos resultados deste processo que as

informações serão extraídas. Por isso, é evidente que este processo deve possuir respostas de qualidade, e ainda que este processo deve conseguir navegar entre os documentos para localizar informações em níveis inferiores.

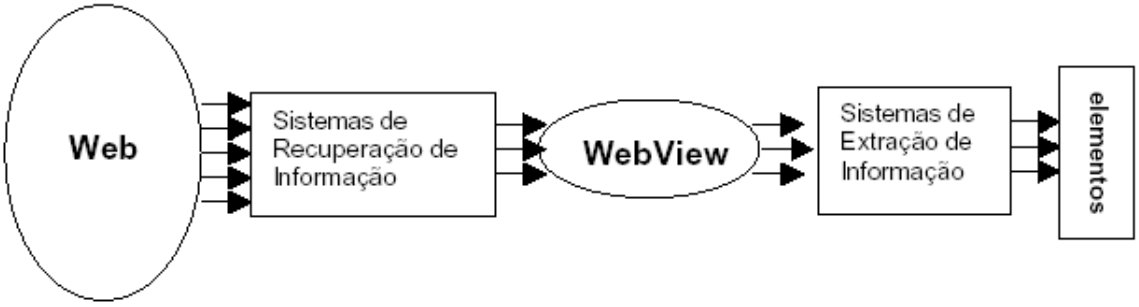

FIGURA 3.1 - Fluxo de Informação

A figura 3-1 representa o fluxo de informação de a web ao elemento extraído. Assim, os sistemas de recuperação de informação devem trabalhar como filtros, impedindo os documentos que não pertençam ao contexto da consulta cheguem ao processador de extração.

Os mecanismos de recuperação de informação podem ser divididos em três grupos. No grupo 1 os documentos são classificados quanto ao seu conteúdo e estes são identificados com um ou mais diretórios que representam aquele domínio de informação. Como exemplo deste grupo temos o *Yahoo*, e o *Open Directory*.

No grupo 2, os documentos são analisados e classificados por palavras chaves encontradas no conteúdo do documento. Esta classificação pode ser feita através de agentes móveis que navegam na web em busca de novos documentos indexando os documentos encontrados, ou manualmente, onde o usuário define quais as palavras chaves as quais ele quer relacionar seu documento. O índice é armazenado em banco de dados que é consultado através de interfaces específicas. Como exemplo dos dois modelos temos o Altavista e o Cadê. Entre os principais problemas deste grupo podemos citar o limite de cobertura e a base de dados com possíveis informações desatualizadas.

No grupo 3, a consulta realizada pelo usuário é enviada, com modificações ou não, aos sistemas do grupo 1 e/ou 2 e as informações extraídas dos diversos sistemas são combinadas de forma que apresentem os melhores resultados entre os sistemas consultados.

Com isto estes sistemas conseguem melhorar sua área de cobertura, visto que os sistemas do grupo 1 ou 2 trabalham com um subconjunto dos documentos existentes. Como exemplo deste modelo podemos citar o *Google, MetaCrawler* e o *Miner*.

Um dos problemas que vem ocorrendo com maior freqüência entre os diversos tipos de mecanismos de recuperação de informação é a presença de documentos irrelevantes para a consulta realizada. Um dos motivos da queda de desempenho é o alto volume de informações com um sistema de *ranking* que não considera o contexto da consulta [LAW 98], [LAW 2000] e nem disponibiliza meios do usuário definir o modo pelo qual o resultado será disponibilizado.

O contexto é um dos fatores mais importantes para o processo de integração entre bases de sistemas distribuídos. Os documentos a serem processados devem possuir relações bem definidas com os outros documentos. Assim, antes de iniciar o processo de integração precisamos organizar o modo pelo qual as fontes de informação serão recuperadas, de forma a agrupar um domínio de informação específico.

Existe espalhado na web, uma infinidade de mecanismos de busca especializados em produtos, medicamentos, pessoas, casamentos, e outras diversas áreas. Para melhor contextualizar as fontes processadas, os sistemas de recuperação de informação utilizados devem ser os especializados em algum domínio.

Outra vantagem é que os resultados destes sistemas podem ser facilmente mapeados para a extração de dados, visto que estes sistemas trabalham com páginas dinâmicas e com isto mantém uma estrutura fixa por algum tempo, estrutura esta que na maioria das vezes é tabular.

Logo, para criar sistemas de integração a partir de documentos espalhados pela web, é necessário localizá-los. Como estes sistemas trabalham, em sua maioria, em um domínio específico de informação, podem ser utilizados sistemas de recuperação de informação específicos para aquela área de conhecimento escolhida. Assim, estaremos garantindo melhores resultados para o processo de extração de informação, além do que, identificando mais claramente os relacionamentos entre os documentos.

3.2 Extração de Informação

A Internet apresenta uma grande quantidade de informação que continua a se expandir devido a vários fatores como a facilidade de publicação e acesso da informação. Estas informações estão isoladas em um documento e este pode estar conectado a outro documento criando uma grande rede de informação – um hiperdocumento. Para encontrar estas informações existem vários sistemas de recuperação de informação, discutidos no capítulo anterior.

O objetivo da extração de informação é transformar dados sem estrutura em um formato estruturado que possa ser consultado [EIK 99]. Através deste processo, informações específicas podem ser extraídas de diferentes fontes e podem ser representadas de forma uniforme a partir da definição das relações entre os dados [SMI 97].

Segundo [EIK 99], a extração de informação é originalmente a tarefa de localizar informações específicas de um documento em linguagem natural, e em particular usando uma determinada área do processamento de linguagem natural.

O processo de extração analisa pequenas porções de texto dentro de vários documentos em busca de informações relevantes. Tais informações são definidas pelo domínio a ser escolhido como objetivo, ou definido implicitamente pelo sistema de extração [ABI 97a].

Este processo é realizado a partir do conjunto de regras de extração que são previamente definidas para um determinado domínio de informação. Para se iniciar um processo de extração de informação, o primeiro passo é a definição de um esquema de banco de dados que irá reorganizar as informações extraídas [EMB 98], e é a partir deste esquema, baseado na estrutura das fontes de informação, que as rotinas de extração serão criadas.

Estas rotinas têm como objetivo, acessar os documentos que serão processados, fragmentar as informações, remover informações indesejadas como as *tags html*, e apresentar o dado em sua forma mais objetiva possível.

Pesquisas relacionadas ao processamento de linguagem natural já vêm sendo realizadas há algum tempo pelos pesquisadores da área de inteligência artificial, que já construíram sistemas capazes de extrair informações relevantes de pequenos domínios de informação. Com o crescimento acelerado da quantidade de informação disponibilizada eletronicamente na Web, esta área recebeu nova atenção da comunidade científica.

A Web é uma grande coleção de documentos que possui a informação distribuída entre os diversos servidores. Esta informação pode possuir diferentes formatos, entre os quais destacamos o formato HTML - *Hypertext Markup Language* - que pode ser processado para a extração de informação de interesse, convertendo os dados de vários documentos em tuplas de banco de dados.

É importante lembrar que os dados a serem extraídos muitas vezes não estão presentes em um único documento e a navegação entre os *links* destes documentos é realizada para a obtenção completa dos atributos de uma tupla.

Em [CHI 97], é apresentado um modelo que classifica os documentos semi-estruturados na web segundo o nível de distribuição da informação. Documentos do nível um-para-um apresentam todas as informações em uma só página, documentos do tipo um-para-muitos apresentam vários *links* para serem seguidos para encontrar o resto da informação.

Como dito anteriormente, os processos de recuperação e extração podem ser combinados para integrar bases de informação distribuídas. A extração de informação pode ser aplicada em textos livres, estruturados e semi-estruturados [GAO 99].

No processo de extração é planejada a decomposição de complexas consultas que serão realizadas sobre o domínio escolhido, de forma a derivar pequenas consultas que podem ser realizadas sobre fontes de informação distribuídas pré-definidas.

As técnicas de extração podem ser divididas em duas categorias: linguagens de consulta devidamente preparadas para a consulta de documentos web e, através de mediadores e *wrappers* [EMB 98] que serão discutidos neste capítulo.

Entre as linguagens de consulta para documentos na web podemos citar a WebOQL que é discutida em [ARO 98], WebSQL [ARO 97] e W3QS [KON 95]. Estas linguagens utilizam a estrutura dos documentos web, para conseguir navegar entre os documentos e encontrar as informações requeridas. Neste modelo os documentos são vistos como grafos onde a consulta pode navegar entre os nós. O escopo deste trabalho não visa utilizar estes tipos de categorias para a extração de dados e sim a utilização de *wrappers*.

Segundo [EIK 99], um *wrapper* pode ser visto como um procedimento que é projetado para extrair o conteúdo de um domínio específico e disponibilizá-lo em outra forma de representação. Outra definição diz que este é um componente de software que converte dados de um modelo para outro, assim dados de modelos diferentes podem ser consultados através de uma linguagem de consulta comum após a conversão para um modelo comum. [EMB 98] define um *wrapper* como um processo de extração de dados a partir de informações contidas em um texto sem estrutura, onde os valores dos atributos extraídos são compostos em uma estrutura de dados complexa.

Para nosso estudo de caso, apresentado posteriormente, a segunda definição nos parece mais apropriada, pois os dados contidos em documentos semi-estruturados na Web devem ser processados e armazenados em banco de dados. Note também que os *wrappers* podem ser executados como um processo da consulta do usuário ou podem ser executados como um processo à parte.

As figuras abaixo representam os dois tipos de processamento que podem ser realizados pelos *wrappers*. A figura 3.2 apresenta o *wrapper* sendo disparado pelo usuário através da consulta realizada, ou seja, o *wrapper* é um dos componentes da consulta.

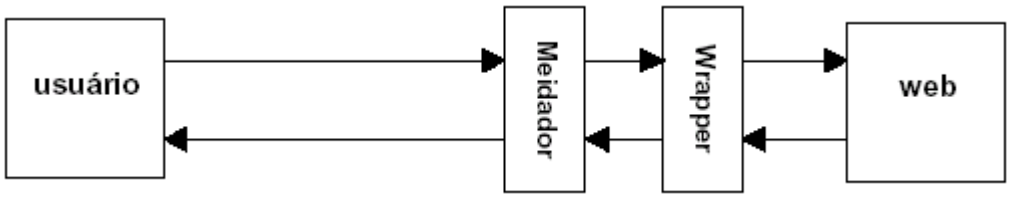

FIGURA 3.2 - Iniciando Wrapper (a)

Já na figura 3.3 o *wrapper* é disparado pelo mediador, que através de alguma rotina de atualização ou de execução, inicia o processo de extração, onde os dados coletados são enviados para um banco de dados.

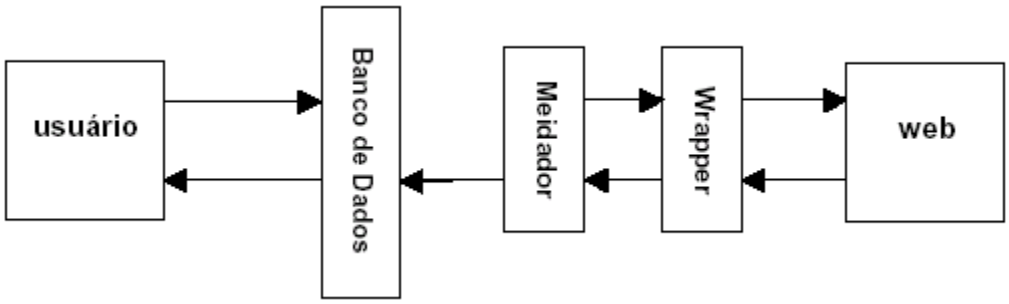

FIGURA 3.3 - Iniciando Wrapper (b)

As vantagens de utilizar o primeiro modelo é que a consulta é sempre realizada sobre os dados reais do domínio escolhido, porém para cada consulta o wrapper é disparado e executado novamente. Segundo dados de análise referente aos tipos de consultas realizadas, os usuários tendem a repetir várias vezes uma mesma consulta. Usuários diferentes também executam consultas semelhantes quando pesquisam um mesmo contexto. Assim o desperdício de processamento pode ser alto se existe um número significativo de consultas e fontes de informação a serem processadas.

Já no segundo exemplo, os usuários consultam uma base local que contém cópias das informações das fontes de informação, ou seja, o processamento por parte do wrapper é bem menor, porém existe o problema da ocorrência de dados inválidos ou desatualizados. Neste modelo é importante criar uma política de atualização de modo que dados modificados nas fontes de informação sejam também alterados na base local.

Wrappers possuem um conjunto de regras de extração que são definidas para um certo domínio. Em alguns casos o wrapper é aplicado a fontes independentes, podendo haver conjuntos de regras específicas para cada fonte de informação, o que aumenta a eficácia. Estas regras, na maioria das vezes, são baseadas em regras sintáticas, que identificam os delimitadores das informações requeridas no documento a ser processado.

Para a construção de um wrapper é necessário definir um modelo de extração da fonte de dados que define os campos a serem extraídos a partir da formatação da fonte de dados e da estrutura sintática que definem ou delimitam a informação a ser extraída.

Os wrappers podem ser construídos manualmente ou utilizando softwares que auxiliam a construção das regras. O problema principal é que na maioria das vezes, estas regras são baseadas na estrutura do documento, e devido ao alto grau de dinamicidade dos dados e estrutura dos documentos na web, estas regras podem não estar corretas.

Os documentos mudam de estrutura constantemente e não existe controle sobre isto. Para garantir wrappers mais genéricos alguns utilizam a técnica de indução, ou seja, métodos de conhecimento indutivo são aplicados para a construção de regras genéricas, garantindo assim que os dados possam ser encontrados no conteúdo do documento de uma forma mais geral.

O nível de complexibilidade para a construção de um wrapper está ligado ao nível de estruturação da fonte de informação. Nos capítulos anteriores havíamos classificados os documentos quanto ao seu grau de estruturação, ou seja, como os dados estão disponibilizados dentro da estrutura interna de apresentação do documento.

A figura 3.4 representa o nível de complexidade ao descrever regras sintáticas de acordo com a classificação de estrutura referente aos documentos. Isto é, quanto melhor representado a estrutura da informação mais facilmente será implementada as regras de extração. Isto pode ser observado em regras de extração para documentos *XML – Extensible Markup Language* – que possui o foco nos dados e, regras para documentos *HTML – Hypertext Markup Language* – que possui o foco na apresentação, no primeiro caso as regras estão muitos mais claras que no segundo modelo de documento.

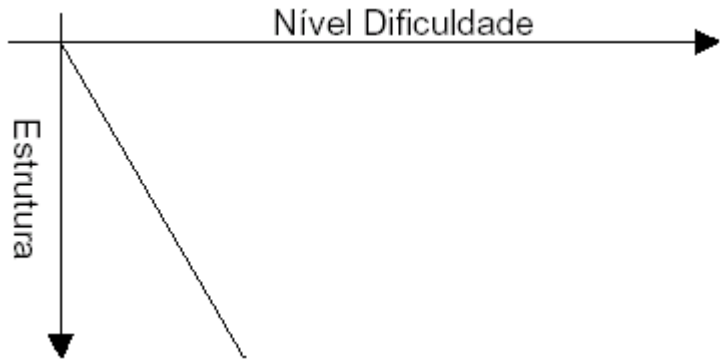

FIGURA 3.4 - Nível de Complexidade de Regras Sintáticas

Um dos primeiros estudos realizados na área de exportação de dados semi-estruturados para modelos mais complexos, é o projeto TSIMMIS - *The Stanford-IBM Manager of Multiple Information Sources* - 1995. Este projeto é descrito com maiores detalhes no próximo capítulo. Em [EIK 99] são discutidos outros exemplos de wrappers.

Outro sistema para a extração de informação é o ARIADNE apresentado em [CRA 97], que é um software que auxilia a construção de agentes de extração de informação. Primeiramente existe um processo de modelagem dos dados, onde um conjunto de fontes de informação são mapeados. Após finalizar esta fase, o usuário pode realizar consultas nestas bases distribuídas. A consulta é então fragmentada em consultas menores e executada em documentos simples, sendo as respostas de cada consulta então combinadas em uma única resposta.

3.3 Extração Sintática com Base no Documento

A extração sintática pode ser realizada sobre qualquer linguagem de marcação tais como o SGML, HTML e XML entre outras. Neste tipo de análise as *tags*, ou marcas, são utilizadas como delimitadores para um *wrapper*, ou seja, determinam o início e o fim de uma informação, não existindo, portanto, um esquema sobre os dados que desejamos extrair.

Neste caso, os *wrappers* são programas que percorre documentos inteiros a procura de conjunto de tags que delimitam uma informação requerida. Embora a sintaxe HTML não exija a utilização de "end-tags" em seus documentos, as aplicações que utilizam a extração sintática podem exigir que seus documentos possuam *start-tags* e *end-tags* em todos os seus elementos.

Após a coleta de informações, as informações são exportadas para um modelo mais representativo, para que possa ser consultado através de alguma linguagem.

No próximo capítulo entraremos em detalhes sobre o projeto TSIMMIS. Este projeto é um dos pioneiros na utilização de wrappers para a extração de dados com o intuito de alocá-los em uma estrutura de dados mais complexa.

3.3.1 Projeto TSIMMIS

Para um melhor entendimento deste tipo de processo será analisada uma ferramenta para extração de dados semi-estruturados de documentos HTML descrito em [HAM 97]. Este protótipo faz parte do projeto TSIMMIS do Departamento de Ciência da Computação da Universidade de Stanford e tem como objetivo a extração de dados e a conversão destes em objetos de banco de dados.

Esta ferramenta tem como entrada uma especificação que declara onde serão encontrados os dados no documento HTML e como eles serão "empacotados" e armazenados no modelo OEM – *Object Exchange Model*. Este extrator não utiliza técnicas de Inteligência Artificial, apenas é baseado em modelos sintáticos que identificam o ponto onde começam e terminam as informações.

Inicialmente, deve-se especificar um arquivo de extração cujo conteúdo são várias seqüências de comandos do tipo:

[variável, fonte, modelo]

onde ***variável*** é o nome da variável que receberá a informação extraída, ***fonte*** é o texto em que deve ser encontrado o modelo, e ***modelo*** determina como deve ser encontrada a informação. Note que uma variável pode ser a fonte de um comando subseqüente.

A seguir, um exemplo detalhado demonstrado em [HAM 97] será analisado a fim de esclarecer o método de extração em questão. Suponha que uma aplicação na web que tenha como resultado um documento HTML com dados meteorológicos como o exemplo da figura 3.5.

		Tue, Jan 28, 1997		Wed, Jan 29, 1997	
Contry	**City**	**Forecast**	**hi/lo**	**Forecast**	**hi/lo**
Áustria	Vienna	snow	-2/-7	snow	-2/-7
Belgium	Brussels	ptcldy	3/-4	ptcldy	3/-4
...					

FIGURA 3.5 - Fonte de Dados HTML

O código HTML gerado por esta aplicação pode ser visto abaixo

```
1 <HTML>
2 <HEAD>
3 <TITLE>INTELLICAST: europe weather</TITLE>
4 <A NAME="europe"></A>
5 <TABLE BORDER=0 CELLPADDING=0 CELLSPACING=0 WIDTH=509>
6 <TR>
7 <TD colspan=11><I>Click on a city for local forecasts</I><BR></TD>
8 </TR>
9 <TR>
10 <TD colspan=11><I> temperatures listed in degrees celsius </I><BR></TD>
11 </TR>
12 <TR>
13 <TD colspan=11><HR NOSHADE SIZE=6 WIDTH=509></TD>
14 </TR>
15 </TABLE>
16 <TABLE CELLSPACING=0 CELLPADDING=0 WIDTH=514>
17 <TR ALIGN=left>
18 <TH COLSPAN=2><BR></TH>
19 <TH COLSPAN=2><I>Tue, Jan 28, 1997</I></TH>
20 <TH COLSPAN=2><I>Wed, Jan 29, 1997</I></TH>
21 </TR>
22 <TR ALIGN=left>
23 <TH><I>country</I></TH>
24 <TH><I>city</I></TH>
25 <TH><I>forecast</I></TH>
26 <TH><I>hi/lo</I></TH>
```

```
27 <TH><I>forecast</I></TH>
28 <TH><I>hi/lo</I></TH>
29 </TR>
30 <TR ALIGN=left>
31 <TD>Austria</TD>
32 <TD><A HREF=http://www.intellicast.com/weather/vie/>Vienna</A></TD>
33 <TD>snow</TD>
34 <TD>-2/-7</TD>
35 <TD>snow</TD>
36 <TD>-2/-7</TD>
37 </TR>
38 <TR ALIGN=left>
39 <TD>Belgium</TD>
40 <TD><A HREF=http://www.intellicast.com/weather/bru/>Brussels</A></TD>
41 <TD>fog</TD>
42 <TD>2/-2</TD>
43 <TD>sleet</TD>
44 <TD>3/-1</TD>
45 </TR>
.
.
</TABLE>
</HTML>
```

Após uma análise do código HTML, o usuário deve definir um arquivo de especificação, que é constituído por vários comandos do tipo *[variável, fonte, modelo]*, mostrado anteriormente. Este arquivo irá especificar para o extrator o arquivo do qual se deseja extrair os dados, quais os dados importantes que devem ser extraídos e como estes devem ser “desmontados” do código HTML.

O código exibido é um dos possíveis códigos de extração para o exemplo citado anteriormente. Este código pode ser visto como um pseudocódigo para a abstração das funções específicas das linguagens de programação.

```
1 [["root",
2 "get('http://www.intellicast.com/weather/europe/')",
3 "#"
4 ],
5 ["temperatures",
6 "root",
7 "*<TABLE*<TABLE*</TR>#</TABLE>*"
8 ],
9 ["_citytemp",
10 "split(temperatures,'<TR ALIGN=left>')",
11 "#"
12 ],
13 ["city_temp",
14 "_citytemp[1:0]",
15 "#"
16 ],
```

```
17
["country,c_url,city,weath_today,hgh_today,low_today,weath_tomorrow,hgh_tomor
row,low_tomorrow",
18 "city_temp",
19
"*<TD>#</TD>*HREF=#>#</A>*<TD>#</TD>*<TD>#/#</TD>*<TD>#</TD>*<TD>
#/#*"
20 ]]
```

Analisando-se o arquivo de especificação anterior, temos que as linhas 1, 2, 3 e 4 definem um comando na forma *[variável, fonte, modelo]*. Estas linhas indicam que a variável se chama *root*, a fonte é *http://www.intellicast.com/weather/europe/* e a palavra reservada # indica que deve se extrair todo o conteúdo para a variável *root*.

As linhas de 5 a 8 indicam que a variável *temperatures* receberá valores da variável *root* de acordo com o modelo apresentado na linha 7. Este modelo indica a seguinte ação: desconsidere tudo (*) a'te encontrar a primeira tabela (<TABLE), desconsidere tudo até encontrar a segunda tabela, desconsidere tudo até encontrar a primeira tag </TR>, extrair todo o conteúdo até encontrar a tag </TABLE>.

Os comandos das linhas 9 a 12 indicam que a variável *_citytemp* receberá todo o conteúdo da variável *split(temperatures,'<TR ALIGN=left>')*. A função split, de modo conceitual, cria um vetor com as partes da variável *temperatures* separadas pela tag <*TR ALIGN=left>.*

O comando incluso nas linhas 13 a 15 indica que a variável *city_temp* receberá os elementos do vetor *city_temp*.

Note que o primeiro inteiro do comando da linha 14, indica que deve ser lido o segundo elemento do vetor, assumindo que o vetor comece com o elemento "0". O segundo inteiro apresentado na linha 14, indica que o vetor deve ser lido por inteiro, este número consiste em um índice partindo do ultimo elemento.

Nas linhas de 17 a 19 são armazenados os valores nas variáveis definidos pelo usuário no arquivo de especificação. Após serem executados os comandos, as variáveis são empacotadas em objetos OEM que são modelos sem esquema que são particularmente bem adaptados para representar dados semi-estruturados como os encontrados na Web. Os dados extraídos do exemplo podem ser vistos no formato OEM exibidos abaixo:

root *complex* {
    temperature *complex* {
        city_temp *complex* {
            country *string* “Austria” **city_url** *url* http://www…
            city *string* “Vienna”
            weather_today *string* “snow”
            high_today *string* “-2”
            low_today *string* “-7”
            weather_tom *string* “snow”
            high_tomorrow *string* “-2” **low_tomorrow** *string* “-7”
        }
        city_temp *complex* {
            country *string* “Belgium”
            city_url *url* http://www…
            city *string* “Brussles”
            …
            }
        …
        }
}

Alguns aspectos devem ser levados em consideração, como o fato de que podem existir modelos OEM diferentes para a mesma base de dados, e que, mudanças na estrutura do documento HTML forçam mudanças no arquivo de especificação. Existem ainda alguns comandos que não foram utilizados no exemplo descrito, o **extract_table** e o **case**. O primeiro comando extrai automaticamente o conteúdo de uma tabela, enquanto o segundo permite ao usuário definir vários arquivos de especificação, com isto o *parser* poder verificar qual modelo ele deverá usar sobre o documento HTML.

Extraídos os dados, devemos prover alguma interface de consulta para estes dados. No trabalho desenvolvido no projeto TSMMIS não foi desenvolvida nenhuma nova ferramenta para este fim. Em vez disso, foi usado um conjunto de *wrappers* que suportam uma grande variedade de tipos de consulta. Estes mecanismos podem utilizar as linguagens de consulta de dados semi-estruturados como o Lorel [ABI 97] ou o MSL [PAP 96], retornando os resultados no formato OEM. O objetivo agora é armazenar os objetos OEM no LORE (*Lightweight Object Repository*) que podem trabalhar com processadores de consulta mais otimizados.

3.4 Extração Sintática com Base no Dado

Um dos problemas apresentados pela extração sintática com base no documento, é que o processo de extração deve ser alterado sempre que alguma mudança ocorrer no documento modificando a sua estrutura. Além disso, os modelos gerados pelos processos sintáticos não levam em consideração a informação semântica dos dados. Em [BER 99] é descrito um modelo específico de integração semântica de dados semi-estruturados.

Muitos documentos possuem relacionamentos entre seus dados e juntos descrevem o contexto da informação. Nestes documentos podemos definir um modelo conceitual que tem como finalidade extrair a semântica dos dados [NES 98]. Este modelo é baseado em uma ontologia que descreve os dados de interesse, incluindo relações, aparência léxica e chaves de contexto.

A partir deste modelo é criado um esquema de banco de dados. Um banco de dados é então criado e populado com dados extraídos do documento a partir de regras pré-estabelecidas pela ontologia. Este processo de extração possibilita a identificação dos relacionamentos existentes no domínio da aplicação possibilitando consultas mais precisas. Para um melhor entendimento deste método de extração será realizada uma análise do estudo proposto em [EMB 98].

É apresentado abaixo um modelo de extração sintática baseada nas características dos elementos a serem extraídos. Este modelo é descrito em [EMB 98] e foi definido para processar documentos que possuem grande quantidade de informações inter-relacionadas e que podem ser representados por uma pequena ontologia.

Neste modelo são apresentados cinco passos para a extração:

1. Modelagem de uma ontologia sobre uma área de interesse: nesta fase são definidas as relações existentes no domínio da aplicação, ou seja, a semântica dos dados;

2. Realiza-se uma análise desta ontologia para gerar um esquema de banco de dados e regras para adaptar constantes as palavras-chaves. Nesta fase é também criado um banco de dados que será populado posteriormente. São criadas as expressões regulares para os tipos de dados da aplicação, isto é, a forma em que os dados estarão disponíveis no documento;

3. Os dados do documento da Web são quebrados em pequenos fragmentos, eliminadas as *tags* HTML e estes pedaços são apresentados como pequenos registros desestruturados para posterior processamento. Nota-se

que este modelo de extração sintática considera que o documento apresenta uma estrutura dividida em pequenos registros;

4. Um reconhecedor aplica as regras nos pequenos fragmentos de forma que seja possível extrair os dados e as relações esperadas: nesta fase os pequenos registros são analisados de acordo com as expressões regulares definidas anteriormente;

5. Nesta fase é populado o banco de dados utilizando-se de heurísticas que levam em consideração a cardinalidade descritas na ontologia e também determinando com serão montados os registros no banco de dados.

A estrutura do processo pode ser vista na figura 3.6. Neste modelo é mostrado que a única entrada do sistema é o documento web. A ontologia é escolhida de acordo com o domínio do documento e o resultado do processo é um banco de dados populado.

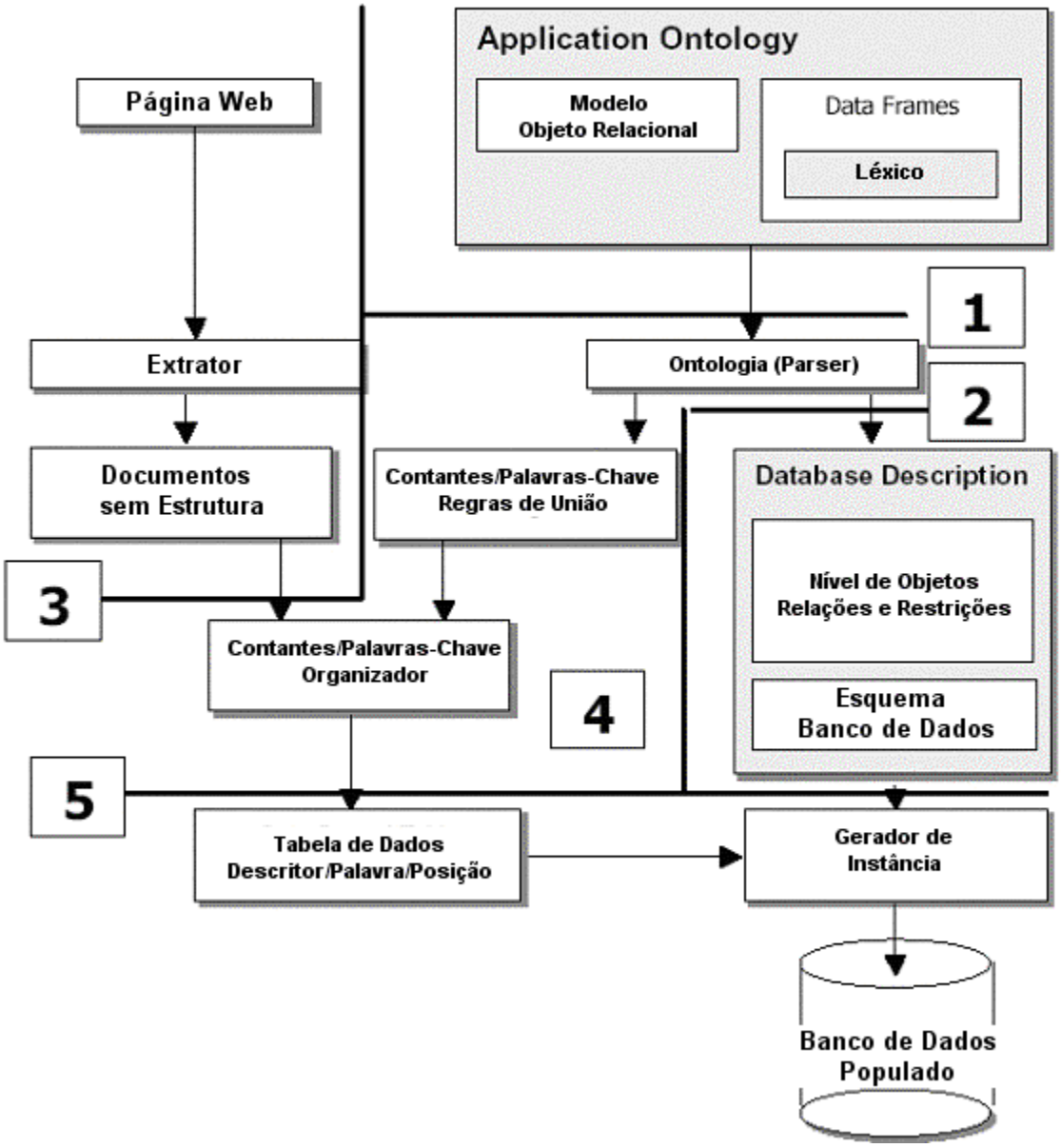

FIGURA 3.6 - Modelo de Extração

A seguir iremos analisar uma fonte de dados referente à venda de automóveis e será iniciado o processo descrito na figura 3.6. Im possível código fonte para venda de automóveis é o seguinte:

'97 CHEV Cavalier, Red, 5 spd, only 7,000 miles on her.
Previous owner heart broken! Asking only $11,995. #1415.
JERRY SEINER MIDVALE, 566-3800
####
'94 CHEV Corsica, 88,281 miles. Ask for #16. $4,900.
Government Surplus533-5885
####

A especificação da ontologia consiste em informações referentes à estrutura de dados, a informações léxicas que são entradas para o quadro *Application Ontology.* São criadas regras léxicas, *DataFrame*, compatíveis com as entradas dos dados e as constantes do domínio. Um modelo objeto-relacional pode ser visto na figura 3.7 que descreve um modelo sobre venda de carros onde foi utilizado o *Object Oriented Systems Model* – OSM [EMB 92].

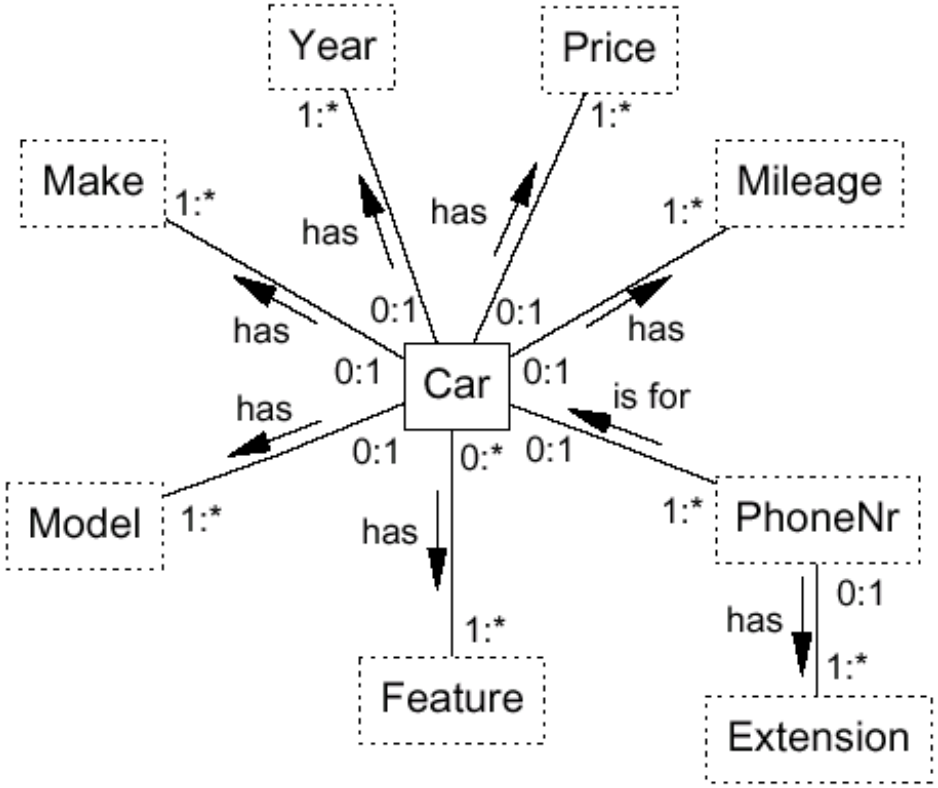

FIGURA 3.7- Modelo de Ontologia no formato OSM

O texto abaixo é a representação textual da ontologia descrita acima:

```
Car [0:1] has Year [1:*];
Year {regexp[2]: "\d{2} : ([^\$\d]|^)\d{2}[^,\dkK]",
"\d{2} : ([^\$\d]|^)\d{2},[^\d]",
"\d{2} : \b'\d{2}\b" };
Car [0:1] has Make [1:*];
Make {regexp[10]: "\bchev\b", "\bchevy\b", ... };
Car [0:1] has Model [1:*];
Model {regexp[16]: "88 : \bolds\S*\s*88\b",
"80 : \baudi\S*\s*80\b", "\bacclaim\b", ... };
Car [0:1] has Mileage [1:*];
Mileage {regexp[8]: "\b[1-9]\d{1,2}k",
"[1-9]\d?,\d{3} : [^\$\d][1-9]\d?,\d{3}[^\d]" }
{context: "\bmiles\b", "\bmi\.", "\bmi\b"};
Car [0:*] has Feature [1:*];
Feature {regexp[20]:
-- Colors
"\baqua\s+metallic\b", "\bbeige\b", ...
-- Transmission
"(5|6)\s*spd\b", "auto : \bauto(\.|,)",
-- Accessories
"\broof\s+rack\b", "\bspoiler\b", ...
-- Engine characteristics
"\bv-?(6|8)", "\b6\s*cyl\b", ...
-- Body/Style
"\b4\s*d(oo)?r\b", "\b2\s*d(oo)?r\b", ...
-- Low mileage
"\blow\s+miles\b", "\blow\s+mi\.", ... };
Car [0:1] has Price [1:*];

...
```

A partir desta ontologia, que define um esquema de banco de dados, é criado um banco de dados utilizando comandos DDL do SQL. A DDL da ontologia citada acima pode ser visualizada no código abaixo:

```
create table Car (
Car integer,
Year varchar(2),
Make varchar(10),
Model varchar(16),
Mileage varchar(8),
Price varchar(8),
PhoneNr varchar(8));
```

```
create table PhoneNr (
PhoneNr varchar(8),
Extension varchar(4));

create table CarFeature (
Car integer,
Feature varchar(10));.
```

O *DataFrame* tem a finalidade de descrever palavras chaves e as expressões regulares que indicam a presença de um objeto em um registro. Estas regras podem definir rotinas de conversões para uma representação comum, como exemplo, representar a constante *ano* com dois ou quatro dígitos (98 ou 1998). Um exemplo de *dataframe* pode ser visto abaixo:

```
Year : \d{2} : ([^\$\d]|^)\d{2}[^,\dkK]
Year : \d{2} : ([^\$\d]|^)\d{2},[^\d]
Year : \d{2} : \b'\d{2}\b
Make : \bchev\b
Make : \bchevy\b ...
Model : 88 : \bolds\S*\s*88\b
Model : 80 : \baudi\S*\s*80\b
Model : \bacclaim\b ...
Mileage : \b[1-9]\d{1,2}k
Mileage : [1-9]\d?,\d{3} : [^\$\d][1-9]\d?,\d{3}[^\d]
KEYWORD(Mileage) : \bmiles\b
KEYWORD(Mileage) : \bmi\.
KEYWORD(Mileage) : \bmi\b
Feature : \baqua\s+metallic\b
Feature : \bbeige\b ...
Feature : (5|6)\s*spd\b
Feature : auto : \bauto(\.|,)
Feature : \broof\s+rack\b
Feature : \bspoiler\b ...
Feature : \bv-?(6|8)
Feature : \b6\s*cyl\b ...
Feature : \b4\s*d(oo)?r\b
Feature : \b2\s*d(oo)?r\b ...
Feature : \blow\s+miles\b
Feature : \blow\s+mi\. ...
Price : [1-9]\d?,\d{3} : \$[1-9]\d?,\d{3}
Price : [1-9]\d{2,3} : \$[1-9]\d{2,3}
PhoneNr : [1-9]\d{2}-\d{4} : (\b|[^\d])[1-9]\d{2}-\d{4}([^\d]|$)
KEYWORD(Extension) : \bext\b
Extension : \d{1,4} : \d{1,4} : (x|ext\.\s+)\d{1,4}\b
```

O segundo passo é extrair os dados do arquivo HTML, que consiste em dois passos:

- localizar as informações referentes à ontologia. Neste modelo não se leva em consideração que as informações podem estar presentes em documentos separados;

- dividir o documento em pequenos registros e deletar as *tags* do HTML.

Cada registro é analisado pelo quadro *Constant/Keyword/Recognizer* que gera uma tabela com o seguinte formato: descritor, string, posição. Veja o exemplo abaixo:

```
Year|97|2|3
Make|CHEV|5|8
Model|Cavalier|10|17
Feature|Red|20|22
Feature|5 spd|25|29
Mileage|7,000|37|41
KEYWORD(Mileage)|miles|43|47
Price|11,995|101|106
PhoneNr|566-3800|140|147
```

Note a existência da palavra reservada *"KEYWORD(X) \ variável \ posição inicial \ posição final"*. Este comando denota que a variável é uma expressão regular e que o *dataframe* contém regras que possibilitam extraí-lo e formatá-lo de acordo com as regras pré-estabelecidas. Por exemplo, formatar a variável *ano* com quatro números (1998 em vez de 98).

A partir destes dados o quadro *Database-Instance Generator* insere os elementos em um banco de dados como no exemplo abaixo, podendo ser realizadas consultadas através de comandos SQL:

```
insert into Car values(1001, "97", "CHEV", "Cavalier", "7,000","11,995", "566-3800")
insert into CarFeature values(1001, "Red")
insert into CarFeature values(1001, "5 spd").
```

Após a inserção dos elementos no Banco de Dados podemos realizar consultas do tipo:

```
select Year, Make, Model, Price
from Car C, CarFeature CF
where C.Car = CF.Car and Year >= "87"
and (Feature = "red" or Feature = "white")
```

A resposta para a consulta anterior é dada na forma de tabelas, da mesma maneira que os banco de dados relacionais trabalham. Note que na extração sintática a resposta era fornecida no formato OEM. A consulta acima implicaria em uma resposta do tipo:

Year	Make	Model	Price
94	DODGE		4,995
94	DODGE	Intrepid	10,000
91	FORD	Taurus	3,500

4 Integração de Dados

O acesso integrado a bases de dados distribuídas e heterogêneas é um dos grandes problemas encontrados pelas organizações. Em [BRA 94] já eram discutidas técnicas recuperar informações em hiperdocumentos e em [CHA 96] a web já é vista como um sistema distribuído que pode ser integrado. Para providenciar interoperabilidade entre sistemas heterogêneos, pode-se estabelecer uma visão global e uniforme para dados e serviços [KAL 99], [CHR 99].

A interoperabilidade entre sistemas pode ser vista ainda como a extração e combinação de múltiplas e heterogêneas fontes de informação buscando derivar a informação em um novo nível de qualidade ou abstração [SAT 99].

Entre as técnicas mais utilizadas para a solução deste problema estão a integração de modelos conceituais entre os participantes através da definição de um modelo conceitual global, ou a adição de uma camada de software para a integração lógica dos dados, que integra fontes de dados específicas sendo que alguns softwares utilizam um tipo de linguagem de consulta particular.

No primeiro caso, a definição do modelo conceitual global é realizada através da comparação entre modelos conceituais locais, identificação de equivalência, identificação e resolução de conflitos [KAN 2000].

No segundo caso, uma camada de software providencia a integração a partir da definição de regras entre os participantes. Esta camada é muitas vezes citada como mediador, que também tem a finalidade de fundir as informações de fontes de dados heterogêneas removendo redundâncias e resolvendo inconsistências [PAP 96], sendo a peça fundamental no processo de integração.

Os mediadores possuem um conjunto de regras em que informações extraídas de uma fonte de informação real são inseridas em uma fonte de informação virtual, possibilitando aos usuários trabalharem diretamente sobre esta visão única criada. As informações são extraídas utilizando *wrappers* que já foram discutidos neste trabalho.

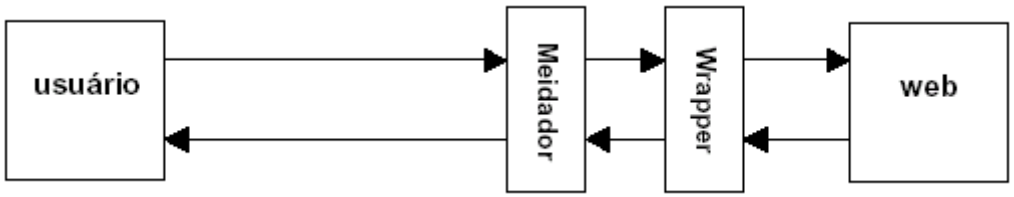

FIGURA 4.1 - Interface entre Usuários e Wrappers

Nos últimos anos, a World Wide Web providenciou inúmeras tecnologias para a disseminação de informação. Porém, esta grande quantidade de possibilidades de dispor as informações, aumentou a heterogeneidade dos dados encontrados na Web. Este fato trouxe problemas referentes a como mapear esta enorme variedade de tipos de dados, de forma que seja possível providenciar um acesso a informações heterogêneas de forma integrada.

Sistemas de banco de dados exploram o conteúdo semântico definido pelo esquema para melhorar as funções de indexação, otimização de consultas, gerenciamento de restrições e outras funcionalidades [KAL 99]. Já no ambiente Web não existe um modelo formal ou um esquema determinado para os dados. Isto prejudica ainda mais as funcionalidades relacionadas à consulta e integração na Web, por isso, existe uma necessidade de se mapear fontes de informação semi-estruturadas e heterogêneas de forma a representá-las em um formato estruturado a partir de um modelo conceitual único. Este modelo pode ser baseado na regularidade de estrutura dos documentos HTML.

Esta regularidade de estrutura é a base principal para a construção de wrappers, que tem como objetivo extrair os dados dos documentos HTML como se fossem pequenos bancos de dados autônomos.

Outro aspecto importante para tratar a interoperabilidade entre sistemas é o conceito de metadados. Metadados é uma descrição explícita dos dados do sistema e que auxiliam a documentação, reusabilidade e interoperabilidade [BUS 90]. Em [ATZ 98a] é descrito um modelo lógico para representação de metadados no ambiente Web.

Na tabela 4-1 são definidos alguns tipos de metadados que são importantes para o processo de integração.

TABELA 4.1 - Metadados para Sistemas de Integração

Tipo	Descrição
Metadados Técnicos	Informações referentes ao acesso às bases distribuídas, tais como: velocidade de processamento, protocolos de comunicação.
Metadados Lógicos	Informações sobre os relacionamentos entre os diversos esquemas participantes.
Metadados Semânticos	Descrição da semântica da aplicação, ou seja, uma ontologia que descreve o domínio específico da aplicação.
Metadados de Sistema	Dados sobre a freqüência de atualização, segurança e outro que auxiliam a otimização do processo de integração.

Metadados de Infra-estrutura	Dados que auxiliam a aplicação a localizar as informações, como por exemplo, a navegação entre os sistemas.
Metadados de Usuários	Dados referentes às preferências dos usuários, como por exemplo, o que o usuário consulta, como ele recebe as informações.

Quanto maior o nível de qualidade das informações dos metadados, melhor será o desempenho do sistema de integração. É claro que, no ambiente Web, muitos destes dados não existem ou simplesmente estão implícitos na aplicação. Outras informações por sua vez, somente serão conhecidas após o processo de integração, tais como a velocidade de processamento real dos sistemas distribuídos.

Com as informações sobre os sistemas autônomos podemos construir uma integração entre sistemas de duas formas: Com Esquema Global e Sem Esquema Global.

Nos sistemas com Esquema Global, existe um esquema único que é utilizado para manipular os dados. Este esquema é gerado a partir dos diversos esquemas semânticos dos sistemas participantes, e tem como principal vantagem a necessidade do usuário conhecer apenas o esquema global e não os diversos esquemas. A figura 4.2 demonstra este tipo de sistema.

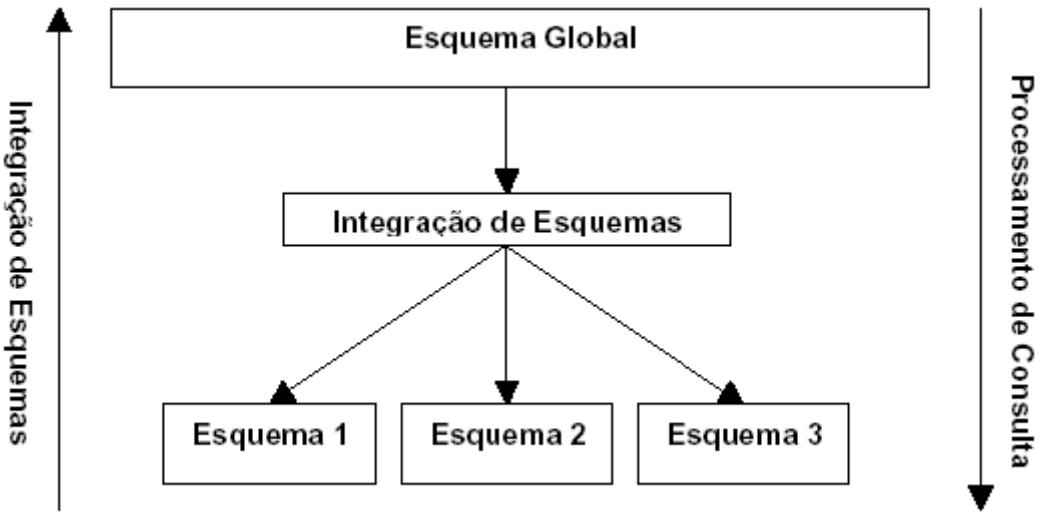

FIGURA 4.2 - Esquema Conceitual Global

O processamento de consulta neste caso possui um plano para executar as consultas que são realizadas no esquema global, assim a consulta é fragmentada e traduzida para os diversos esquemas participantes.

Nos sistemas que não oferecem um Esquema Global, existe uma linguagem de consulta uniforme que abstrai a heterogeneidade referente à linguagem de consulta dos sistemas participantes. Em [LIT 90] é apresentado a MDBQL – *Mulitdatabase Query Language.* A figura 4.3 representa o esquema deste tipo de sistema.

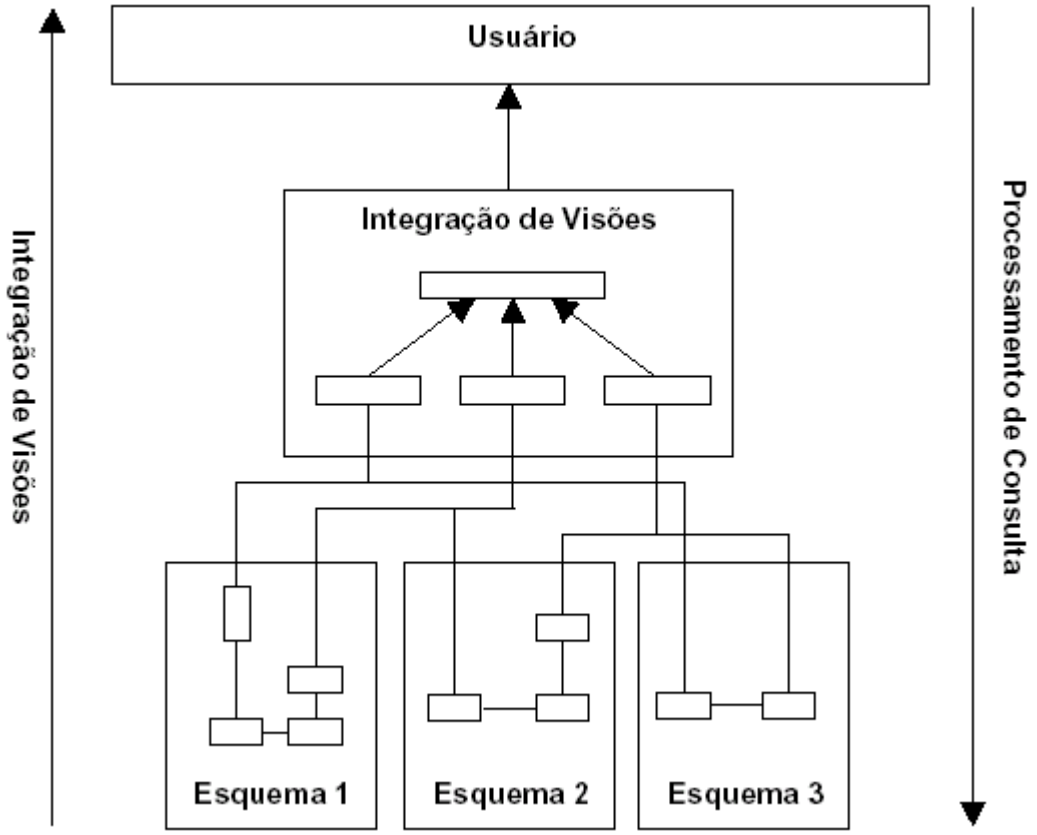

FIGURA 4.3 - Linguagem de Consulta Global

O processamento de consulta é realizado através da decomposição e tradução da consulta realizada pelo usuário.

O processo de integração também pode ser dividido de acordo com o modelo de integração semântica, ou seja, como a camada de integração combina os dados extraídos das fontes de informação. Os modelos de integração, levando em consideração a semântica segundo [BUS 90], podem ser classificados de acordo com a tabela 4-2.

TABELA 4.2 - Classificação dos Sistemas de Integração

Tipo	**Descrição**
Coleção	Os dados são extraídos sem se preocupar com objetos equivalentes de outras fontes.
Fusão	Os dados são extraídos levando em consideração os objetos equivalentes de outras fontes.
Abstração	Utilizado para resolver conflitos semânticos entre fontes heterogêneas no processo de extração.
Suplementação	Os dados não só são extraídos das fontes, mas combinados com outras fontes, adicionando conteúdo descritivo ou semântico para o dado.

Um das características principais do processo de integração é a transparência. Isto é, o usuário não tem conhecimento da localização física dos dados, não tem conhecimento dos diferentes esquemas e linguagens de consulta utilizadas pelos sistemas participantes do processo de integração.

É importante salientar que um sistema de integração centrado em dados disponíveis em documentos espalhados na web permitirá apenas a consulta de dados. É importante escolher os métodos de acesso aos dados para o processamento. Uma aplicação de integração pode acessar as informações utilizando uma linguagem de consulta como o SQL, ou utilizando consultas já parametrizadas como as utilizadas nos sistemas de recuperação de informação na Web.

4.1 Integração Materializada

O processo de integração tem como objetivo possibilitar o acesso à informação através de uma visão canônica dos vários modelos envolvidos. O processo de materialização visa o armazenamento real desta visão única dos modelos envolvidos. Já a integração "*virtual*" tem a mesma finalidade, porém, os dados são consultados através de um mediador que interage com as fontes de informação durante a execução da consulta, ou seja, como nos sistemas de banco de dados, podemos criar visões de duas formas: virtual e materializada [LAB 2000]

O problema de integração de dados em documentos na web possui várias características que a tornam mais difícil, entre as principais podemos citar o grande volume de fontes envolvidas, pouca ou nenhuma meta-informação sobre os dados e o alto grau de autonomia das bases de dados.

Devido à liberdade de criação na Web, onde cada autor pode apresentar o mesmo conteúdo de maneiras diferentes segundo a sua percepção, a *World Wide Web* se tornou a maior fonte de informação heterogênea. Esta liberdade de iteração e manipulação gerou uma total falta de padronização ou esquema global para a representação de informação, tendo como resultado a queda de eficiência dos sistemas de recuperação de informação baseadas em técnicas tradicionais.

A idéia básica é partir de um domínio específico de interesse [COL 97], [DEA 99], [GAR 99], [KRU 2000] e definir um modelo conceitual, que irá representar cada instância [ROS 2000] coletada a partir dos métodos de extração de informação, definidos para os documentos do contexto. Neste processo são utilizadas técnicas de representação de conhecimento tanto para a definição do modelo conceitual quanto para o modelo de extração.

O processo de materialização do domínio de pesquisa traz vários benefícios entre os quais o principal é a facilidade e eficiência de acesso à informação [ROS 2000]. Porém, vários são os problemas que devem ser resolvidos tais como a manutenção dos dados armazenados e a própria criação do banco de dados.

Um aspecto importante a ser levado em consideração na escolha de um método de integração é a dinamicidade das fontes de informação. A integração materializada consegue melhores resultados em conteúdos mais estáticos enquanto que a integração "*virtual*" em conteúdos altamente dinâmicos.

Outro fator é o tempo de resposta à consulta. Um sistema que utilize a integração materializada executará a consulta em um tempo muito menor, visto que este deve apenas processar a consulta. A integração virtual necessita realizar o processamento de extração das informações para posteriormente realizar a consulta. O controle sobre os dados, na integração materializada, pode ajudar a criar índices que aumentem ainda mais a eficiência das consultas, pois uma vez que possuímos a base materializada e a maneira pela qual os dados são consultados podemos otimizar o banco de dados.

O desenvolvimento de um sistema para integração materializada ocorre em quatro etapas:

1. escolha do Domínio de Interesse;
2. definição do Modelo Conceitual para representação.
3. definição da Lógica de Extração dos dados;
4. modelagem da Interface de Consulta.

As etapas 2 e 3 são os mais importantes. O modelo conceitual que irá se criar deve poder representar de forma única todos as fontes de informação envolvidas de forma que as consultas possam ser otimizadas. Já o processo de Lógica de Extração deve fornecer um plano de execução de consulta as fontes de

informação de maneira que os wrappers possam extrair as informações e inseri-las no modelo de dados escolhido, respeitando regras de integridade definidas no modelo conceitual.

Com isto, a visão única do modelo conceitual é fragmentada em visões individuais de cada fonte de informação, visão esta que define os dados que serão extraídos.

A classificação das informações contida em um domínio de interesse deve ser realizada com base na representação da estrutura genérica dos documentos bem como a terminologia utilizada, sendo que os termos utilizados na classificação podem ser diferentes dos termos encontrados nos documentos. A representação do modelo conceitual pode ser realizada utilizando o modelo entidade-relacionamento visto que este modelo também representa o banco de dados que armazenará as informações.

Já a lógica de extração de dados fornece a rotina para a extração de forma que sejam criadas tuplas a serem adicionadas no banco de dados que obedeçam à semântica definida no modelo conceitual.

As rotinas que processam a extração de dados seguem na maioria das vezes a mesma seqüência de passos. Primeiramente o extrator realiza uma análise da estrutura do documento, localizando os elementos a serem extraídos. Estes elementos podem ainda ser reprocessados para a extração de atributos, que finalmente são inseridos no banco de dados. Após a extração e inserção o extrator segue para o próximo documento a ser processado.

Com os dados devidamente armazenados no banco de dados, estes podem ser consultados e disponibilizados na Web. Pode-se ainda, a partir de determinadas consultas, reconstruir o documento de forma integral ou parcial. O termo "*WebView*" foi utilizado por [LAB 2000] para indicar o documento HTML que é criado com base em dados armazenados em um sistema de banco de dados, como isto os documentos são atualizados imediatamente após a atualização no banco.

A principal vantagem em se possuir uma visão materializada dos dados extraídos de vários documentos da web é sem dúvida o tempo de execução das consultas. Isto se deve ao fato de que uma consulta de usuário é refletida diretamente no banco de dados e não em um processo que irá primeiramente extrair os dados, inseri-los em uma estrutura de dados mais complexos para posteriormente executar a consulta.

Um dos problemas principais deste modelo é que os dados podem estar desatualizados e a consulta pode não responder às expectativas reais do usuário. Assim o processo de atualização deve ser feito com base em informações

referentes às diversas bases envolvidas no processo, que em alguns casos possuem dados importantes referentes às atualizações.

5 Intercâmbio de Informações para Integração Materializada

Após o processo de integração materializada, os dados estão prontos para serem processados pelas aplicações. Neste momento existe um controle total sobre os dados armazenados no banco de dados local. Mas, devemos lembrar que estes dados são apenas uma cópia dos dados originais; sendo assim, o processo de integração deve possuir alguma política de atualização baseada nas meta informações de cada sistema participante.

Com isto o processamento dentro de um sistema de integração materializada não tem um fim. Após a materialização completa dos dados distribuídos, um processo de atualização deve ser disparado de tempos em tempos.

Outro fator importante para os sistemas de integração de dados na web é não prejudicar os sistemas remotos. Se vários sistemas de integração extraem dados de uma mesma base, esta base pode ter a sua velocidade de processamento de requisições prejudicada. Veja o exemplo onde vários sistemas de integração fazem milhares de requisições para um mesmo ponto da rede.

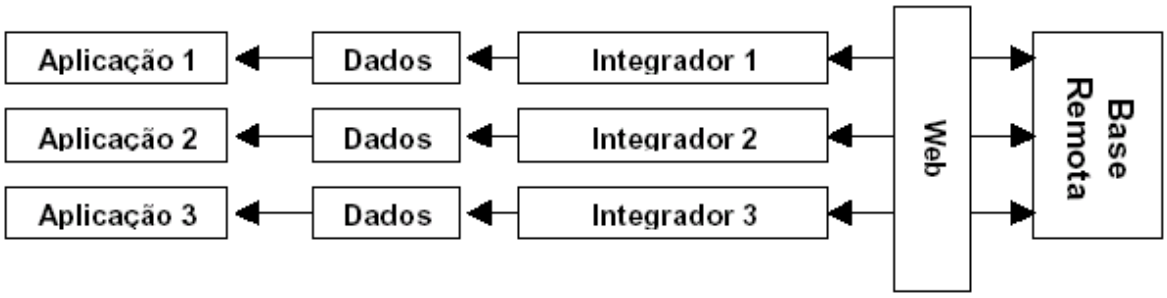

FIGURA 5.1 - Múltiplos Acessos de Sistemas Remotos

A figura 5.1 apresenta vários sistemas de integração processando uma mesma base remota e tendo como resultado um mesmo conjunto de dados que são processados por aplicações diferentes. Podemos observar que a base remota pode ter seu desempenho prejudicado, visto que os sistemas de integração podem fazer milhares de requisições por minuto. Além disso, algumas políticas de segurança poderiam negar acesso aos sistemas de integração, por estes estarem prejudicando o sistema remoto.

Para resolver este problema, temos que analisar a natureza das aplicações envolvidas. Se as aplicações estão dentro de uma única organização, todos os dados extraídos pelos processos de integração devem estar em um único banco de dados evitando a redundância de informação. Nesta situação devemos também verificar se todas as informações extraídas pelos sistemas de integração são armazenadas no mesmo esquema semântico para que as aplicações não sofram modificações. A figura 5.2 demonstra este modelo.

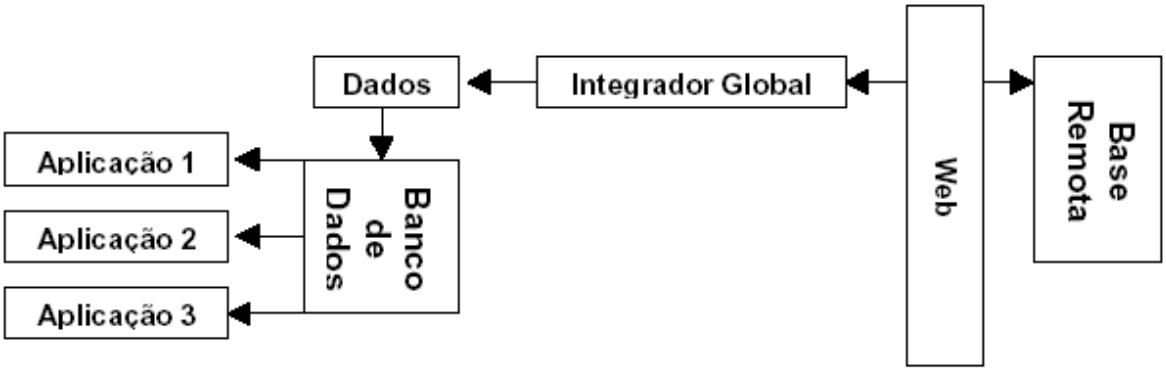

FIGURA 5.2 - Acesso Único para Vários Sistemas Remotos

Neste modelo a base remota é processada por um único sistema de integração global. Possíveis incompatibilidades podem ser tratadas e mapeadas no processamento do integrador global, mantendo as informações necessárias para cada aplicação.

Outro possível cenário seria as aplicações estarem também distribuídas no ambiente web. Assim, seria mais difícil fornecer acesso às aplicações diretamente ao banco de dados. É neste momento que precisamos de uma interface que possibilite a troca de informações de maneira rápida, estruturada e simples. A figura 5.3 demonstra este cenário.

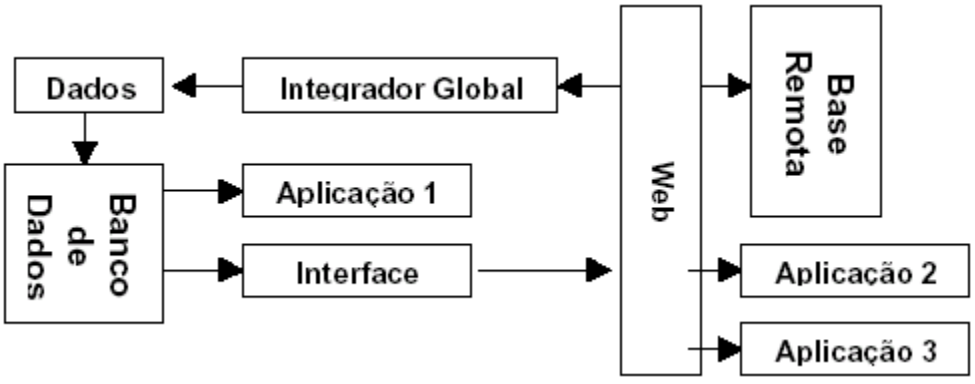

FIGURA 5.3 - Interface Web para acesso a Dados de Extração

Esta interface para o compartilhamento de informação deve permitir a consulta ao banco de dados e o resultado deve ser altamente estruturado para que as aplicações não percam desempenho. Logo, a melhor solução para esta interface é criar um script acessível via Web que possa ser executado através da simulação por linguagem de programação. A resposta do servidor pode ser um documento XML – *Extensible Markup Language* – que é criado especialmente para conter as informações necessárias para cada aplicação. Com isto, trabalha-se com um padrão definido pelo W3C para troca de informações, a maioria das linguagens de programação já possuem algum tipo de recurso para documentos XML.

6 Estudo de Caso

O protótipo desenvolvido como estudo de caso é um sistema para consulta sobre a produção bibliográfica disponibilizada no Currículum Lattes CNPq – conselho Nacional de Pesquisa e Desenvolvimento. A linguagem de programação que foi utilizada foi o Visual Basic 6, por possuir a maioria das rotinas necessárias já implementadas em componentes ActiveX. O estudo de caso tem como objetivo validar alguns dos estudos realizados até aqui e verificar a validade de alguns métodos de extração e integração.

6.1 Definição de Domínio

O currículo Lattes é um formulário eletrônico para cadastros curriculares de pesquisadores. Seus dados são utilizados para avaliação de competências e obtenção de bolsas de auxílio à pesquisa. Estas informações permitem uma melhor seleção de consultores e formação de comitês assessores. Com os dados armazenados em um único repositório espera-se poder melhor avaliar a pesquisa e programas de pós-graduação brasileiras.

Atualmente, estas informações não podem ser recuperadas através de uma conexão com banco de dados tradicional. A única interface de visualização destas informações é um documento HTML dinâmico que a partir do código do pesquisador monta o currículo completo com todos os dados relacionados a esta pessoa. Logo, não existe maneira de recuperar as informações por palavras-chaves ou saber quem publicou em um determinado evento ou ano, ou outro tipo de consulta que relacione atributos de currículos diferentes.

O processo de integração tem como objetivo criar uma base de dados capaz de melhorar o desempenho das consultas, e fornecer acesso completo às informações nele contidas.

O primeiro passo é identificar as possíveis fontes de informação e o relacionamento entre elas. Em nosso caso as fontes escolhidas foram o sistema de consulta de currículo Lattes do CNPq e o sistema de pesquisa da PROPESQ - Pró-Reitoria de Pesquisa.

Esta escolha foi baseada nos seguintes critérios: o sistema do CNPq possui as informações que devem ser copiadas e armazenadas em banco de dados; o sistema da PROPESQ possui os códigos dos pesquisadores junto ao CNPq. É

importante frisar que o sistema que será processado inicialmente deve possuir algum tipo de facilidade na consulta que possa ser implementado em linguagem de programação, ou seja, no gerenciador de extração de informação.

O sistema da PROPESQ foi escolhido para ser a fonte de informação primária, pois neste sistema podemos simular a requisição de informações referentes aos pesquisadores fornecendo um identificador numérico e incremental para cada registro. Assim, podemos facilmente implementar uma função em linguagem de programação, que requisite os documentos de forma incremental. Já no sistema do CNPq o código de cada pesquisador é uma string com letras e números gerado aleatoriamente, o que seria impossível de implementar.

6.2 Modelo Entidade Relacionamento

A figura 6.1 demonstra o relacionamento entre os sistemas remotos que serão processados pelo sistema de integração.

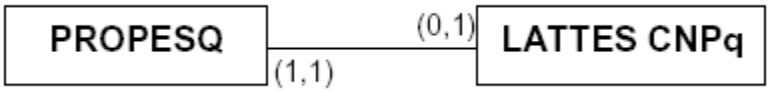

FIGURA 6.1 - Relacionamento PROPESQ - CNPq

Dentro do sistema da PROPESQ, cada pesquisador pode ou não possuir um código para seu currículo na Plataforma Lattes.

6.3 Modelo Relacional

Para armazenar os dados extraídos dos dois sistemas, um esquema foi criado baseado na DTD do currículo lattes do CNPq. Esta DTD pode ser obtida no endereço http://200.215.9.189/lmpl/LMPLCurriculo.dtd. No anexo 1 apresentaremos apenas um trecho desta DTD referente a publicações. O Instituto de Informática da UFRGS também participou efetivamente da definição deste documento.

A figura abaixo demonstra o modelo relacional criado para armazenar os dados extraídos.

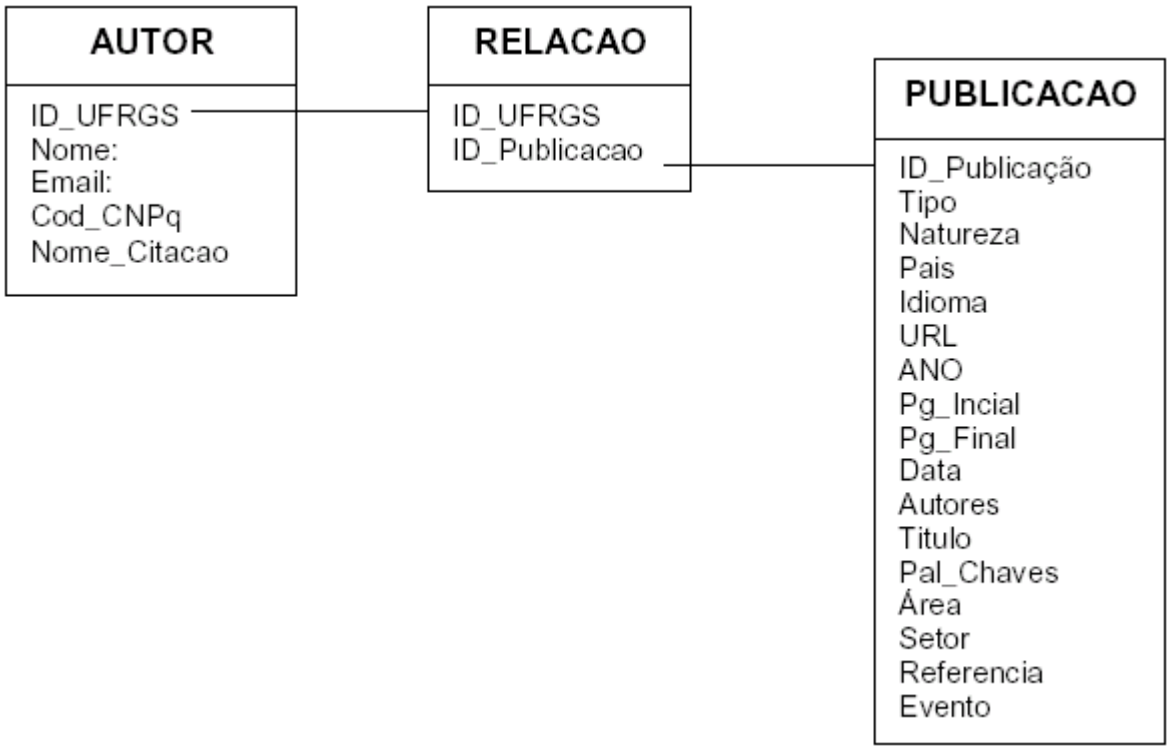

FIGURA 6.2 - Modelo Relacional entre Bases Distintas

6.4 Dicionário de Dados

6.4.1 Tabela Autor

Atributos:

- ID_UFRGS: identificador do autor junto ao sistema da PROPESQ. Tipo: inteiro longo, não nulo;
- Nome: nome do autor junto ao sistema da PROPESQ. Tipo: texto;
- COD_CNPq: Identificador do autor junto ao sistema do currículo Lattes CNPQ. Tipo: texto, não nulo;
- Nome_Citação: Nome para citação bibliográfica junto ao sistema Lattes - CNPq. Tipo: texto;

6.4.2 Tabela Publicação

Atributos:

- ID_Publicacao: identificador da publicação no sistema local. Tipo inteiro não nulo;
- Tipo: tipo da publicação. Os tipos são apresentados na página 64.
- Pais: Pais em que a obra foi publicada. Tipo texto;
- Idioma: idioma da obra publicada. Tipo texto;
- URL: localização na Internet da obra publicada. Tipo texto;
- Ano: ano da publicação. Tipo inteiro;
- Pg_Inicial: número da página inicial anais ou livros. Tipo inteiro;
- Pg_Final: número da página final anais ou livros. Tipo inteiro;
- Data: data completa de publicação. Tipo data;
- Autores: demais autores da publicação. Tipo texto;
- Titulo: título da publicação. Tipo texto;
- Palavras_Chave: palavras chave utilizadas para localização. Tipo texto;
- Área: área de conhecimento. Tipo texto;
- Setor de conhecimento. Tipo texto;
- Referencia: informações adicionais da publicação. Tipo texto;
- Evento: nome do evento da publicação, caso congresso ou seminários. Tipo texto;

6.5 Processo de Extração

Com o modelo relacional bem definido, o próximo passo é definir os passos para o processo de extração.

O sistema da PROPESQ, como já dito antes, retorna um registro para cada indivíduo pertencente à UFRGS. Uma análise deste sistema mostrou como os e quais parâmetros devem ser repassados para o servidor.

A requisição da URL http://astra.ufrgs.br/pesquisa/pesquisador.asp?Localiza=2540 deve ser simulada em software para que o sistema gere um relatório do pesquisador, contendo dados como nome, e-mail, código no CNPq e outros de menor importância para este sistema. O número 2540 pode ser substituído pelos números de 1 a 117098, e representa todos os pesquisadores filiados a UFRGS.

O resultado da requisição é o seguinte código, onde as informações importantes são apresentadas em negrito:

```
<HTML><HEAD>
<TITLE>Sistema Pesquisa - Pesquisador: JOSE VALDENI DE LIMA</TITLE>
</SCRIPT></HEAD>
<link href="pesquisa_css"
    rel="STYLESHEET"
    type="text/css">
<BODY>
<br>
<CENTER>
<div class=title>Sistema Pesquisa</div>
</center>
<br>
<br>
<dd>
<font color=darkred ><b>Informações sobre <i>Pesquisador</i></b></font>
</dd>
<hr>
<blockquote>
<font color=black><b>Nome:</b></font><b> JOSE VALDENI DE LIMA</b><br>
<font color=black><b>Lota&ccedil;&atilde;o:</b></font><b> Departamento de Informática
Aplicada</b> - <b>Instituto de Informática</b><br><font color=black><b>E-Mail:
</b></font><b>valdeni@inf.ufrgs.br</b><br>
<center>
<table width=80% ><tr><td class=tablehead>
<i>Este pesquisador possui v&iacute;nculo com</i>:</td>
<td>
<left>
<font color=darkred>
<a style='color:darkred' href=pesqsel.asp?Localiza=2540&TipoBusca=Projetos> 13 Projetos de
Pesquisa</a><br><a style='color:darkred' href=pesqsel.asp?Localiza=2540&TipoBusca=Linhas> 8
Linhas de Pesquisa</a><br><a style='color:darkred'
href=pesqsel.asp?Localiza=2540&TipoBusca=Grupos> 3 Grupos de
Pesquisa</a><br></left></font></td></tr></table></center><a href=
http://genos.cnpq.br:12010/dwlattes/owa/prc_imp_cv_ext?f_cod=K4781125U1 target=_blank><img
src=linklattes.gif>Ver currículo LATTES do Pesquisador</img></a>
</blockquote>
<HR>
<CENTER>
<div class=credits>
Atualização dos Dados é feita pelo pesquisador através do Sistema Pesquisa
- Módulo Pesquisador
</div>
</CENTER>
<hr>
<table width=100%>
<tr>
<td align=center><a href="javascript:history.go(-1)">Voltar para página anterior</a></td>
<td align=center><a href="pesquisa.asp">Voltar para a página inicial</a></td>
</tr>
</table>
</form>
```

```
</body>
</html>
```

Os dados são então extraídos a partir de regras sintáticas que localizam e extraem a informação. O processamento deste documento pode ser analisado no pseudocódigo abaixo:

Para cod de 1 até 117098 Faça

source = http://astra.ufrgs.br/pesquisa/pesquisador.asp?Localiza=cod

nome = Extrai(source, "Nome:#")
email = Extrai(source, "E-Mail: #")
Cod_CNPq = Extrai(source, " f_cod=# target ")

Insert into Autor (id_ufrgs, nome, email, cod_cnpq)
values (cod, nome, email, cod_cnpq)

Fim Para

A função Extrai() tem como parâmetros um código fonte, *source*, e um texto que identifica o trecho que deve ser localizado no documento. Já o caracter # indica o elemento que deve ser extraído.

Logo, se a função acima fosse aplicada ao registro de número 2540, os dados extraídos seriam os seguintes:

ID_UFRGS = 2540
Nome: José Valdeni de Lima
E-mail: valdeni@inf.ufrgs.br
COD_CNPq = **K4781125U1**

No final deste processamento, todos os pesquisadores ou pessoas ligadas à UFGRS estariam com os dados armazenados em uma base local. Lembrando que apenas alguns dados da PROPESQ foram armazenados.

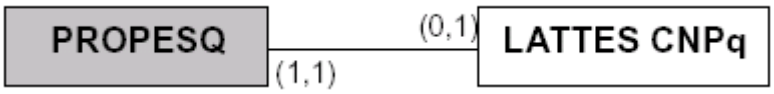

FIGURA 6.3 - Processamento Primário

Após os dados do sistema de PROPESQ preencherem a tabela Autor, o próximo passo é extrair os dados do currículo Lattes referente às publicações. O código é muito parecido com o primeiro, porém, bem mais complexo devido à estrutura dos documentos a serem pesquisados.

Os dados que devem ser extraídos do currículo Lattes são o seguintes:

1. Dados Pessoais
2. Produção Científica, Tecnológica e Artística Cultural;
    - Artigos Publicados em Periódicos;
    - Artigos Completos Publicados em Periódicos;
    - Livros Publicados;
    - Capítulos de Livros Publicados;
    - Trabalhos Completos Publicados em Anais de Eventos
    - Trabalhos Resumidos Publicados em Anais de Eventos
    - Softwares sem Registro um Patente;
    - Dissertação de Mestrado;
    - Dissertação de Doutorado;
    - Trabalhos Técnicos
    - Demais Trabalhos;

O algoritmo para a extração dos dados do currículo Lattes pode ser visto no anexo 2. Este código é semelhante ao primeiro pseudocódigo exibido. A única diferença é que os atributos não são extraídos em um único processamento; o que é extraído são elementos relacionados a uma aplicação. A figura 6.4 demonstra a estrutura básica dos dados relacionados a publicações dentro de um currículo no formato Lattes.

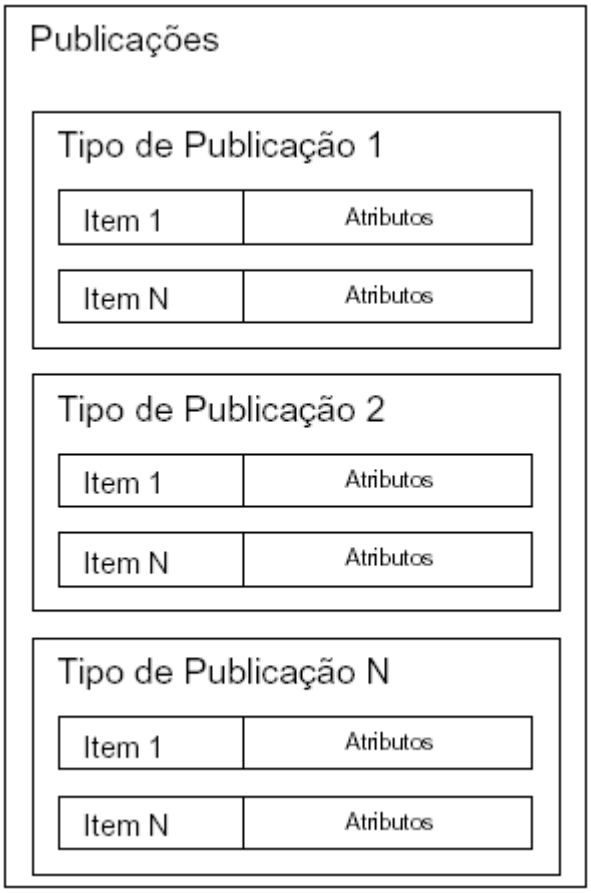

FIGURA 6.4 - Modelo Estrutural do Currículo Lattes CNPq

O código responsável pela extração tem como parâmetro principal o documento no formato Lattes. Este documento então é processado de modo que o conteúdo seja fragmentado em Tipos de Publicação, em seguida vários fragmentos contendo itens de publicação são armazenados em banco de dados.

Os itens de publicação materializados na base de dados são filtrados a partir de regras sintáticas que extraem os atributos de cada elemento de publicação. Este processamento é realizado diretamente no banco de dados local, de maneira a evitar conexões muito longas com servidores remotos na web. O código abaixo representa um elemento contento uma publicação.

<TD VALIGN=TOP WIDTH="90%" BGCOLOR="#CCFFFF">
<FONT FACE="Arial,Helvetica">
<FONT SIZE=-1>
VALIATI, E. R. A., LIMA, J. V., PIMENTA, M. S., LEVACOV, M**.
**
Guia-GESEPE : Um guia de recomendações específico para software educacional **In**: IHC´2000 - II Workshop Brasileiro de fatores Humanos em Sistemas Computacionais, **2000**, Ciudad Real-Espanha**.
**
****IHC´2000 - III Workshop Brasileiro de Fatores Humanos em sistemas Computacionais
Espanha - 2000. **
**
****
Palavras-chave: Interface Homem Máquina, Software Educacional.
****

<**font face="Arial" size="1">**Áreas do conhecimento : Sistemas de Informação, Engenharia de Software
****
</TD>

Os caracteres marcados em negrito servem como indicadores de atributos para os filtros de extração. Os dados extraídos são então armazenados na base local.

Com isto conseguimos materializar os dados referentes a publicações em um banco de dados local, aumentando a velocidade de processamento de consultas e auxiliando sistemas que necessitem acessar os mesmos dados através de dados no formato XML.

6.6 Diagrama de Atividades

O sistema proposto foi dividido em módulos que executam métodos atômicos e posteriormente podem ser alteradas com maior facilidades. Os diversos métodos envolvidos podem ser visualizados na figura 6.5.

O processo é iniciado com uma mensagem entre *Manager* e *Mediador.* Em seguida o *Mediador* começa a executar o *Wrapper.* O *Wrapper* envia e recebe mensagens da *Web* e do *Database* através dos métodos descritos abaixo:

- GetID(): Indentifica unicamente os documentos pertencentes ao domínio de pesquisa.
- PutID(): Insere no banco de dados os identificadores para os documentos a serem pesquisados.
- GetURL(): Recebe o conteúdo dos documentos selecionados.
- PutURL(): Insere em banco de dados, ou arquivo o conteúdo do documento.
- GetElement(): Extrai os elementos extraídos sintaticamente do documento.

- PutElement(): Insere os elementos extraídos sintaticamente do documento no banco de dados.
- GetAttribute(): Fragmenta os elementos extraídos sintaticamente dos documentos em atributos.
- PutAttribute(): Insere os atributos no modelo canônico para posterior consulta e exportação no formato XML.

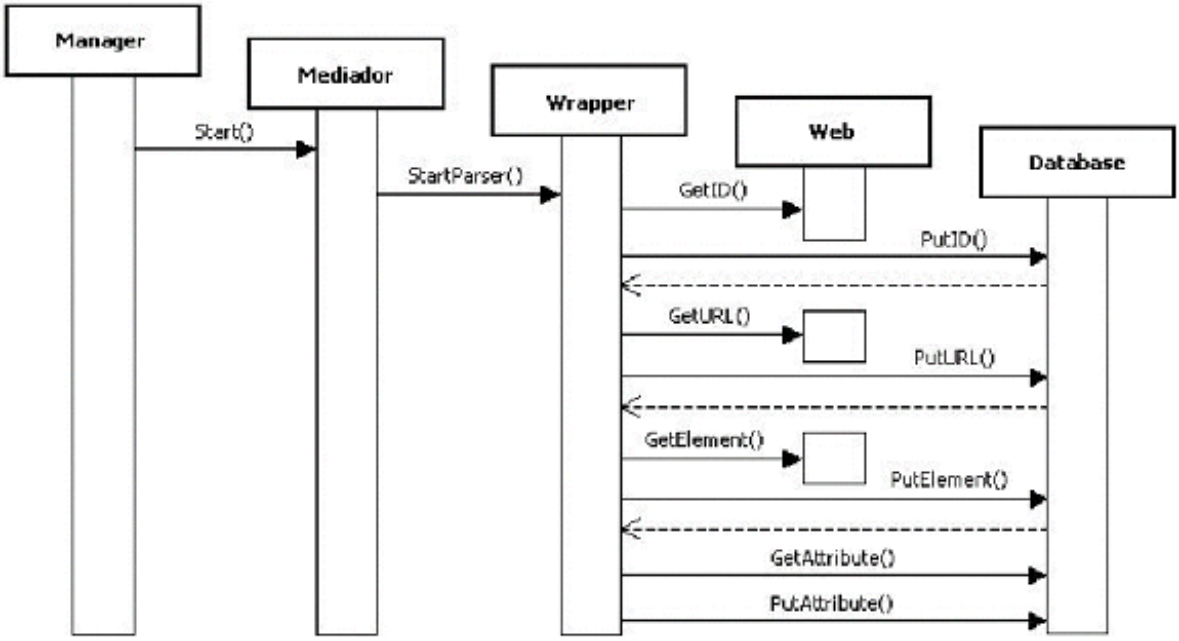

FIGURA 6.5 - Diagrama de Atividades

6.7 Problemas Encontrados

O primeiro problema encontrado foi definir as regras de extração, que são baseadas na estrutura sintática do documento. Estas regras podem ser modeladas manualmente ou através de softwares que automatizam este processo.

Em nosso caso, estamos exportando dados a partir de páginas HTML que apresentam seus dados de forma tabular, ou seja, os dados estão contidos dentro de uma tabela que pode facilmente ser processada, além do que, todos os documentos processados possuem a mesma estrutura sintática. Sendo assim, a construção de extratores de forma manual é melhor empregada visto que algumas otimizações podem ser feitas. Além disso, as regras de extração podem ser

diretamente escritas na linguagem de programação que será utilizada para a construção do sistema de extração de dados.

Softwares que automatizam estas regras são mais bem utilizados quando existem muitas estruturas diferentes entre os documentos que serão processados. Assim estes softwares podem criar regras genéricas que podem ser aplicadas a conjunto de documentos com estruturas diferentes.

A vantagem de se possuir regras genéricas é que mesmo ocorrendo mudanças na estrutura do documento, as mesmas regras podem continuar sendo utilizadas, uma vez que estas podem ser baseadas em algum conhecimento semântico. Já nas regras sintáticas, modeladas manualmente, qualquer alteração de estrutura irá prejudicar o processo de extração. Quando a regra é criada manualmente, busca-se otimizá-la ao máximo.

Outro ponto importante a ser considerado é o fato do documento HTML ser ou não gerado automaticamente. Documentos dinâmicos possuem uma mesma estrutura que é preenchida com dados dinâmicos, assim documentos gerados dinamicamente podem ser analisados, de forma a extrair o documento modelo e com isto facilitar a análise sintática do mesmo.

Neste trabalho foram classificadas várias formas de integração. A integração realizada neste estudo de caso mantém a autonomia das fontes de informação. Este sistema classificado de acordo com a semântica dos dados é do tipo coleção, ou seja, não leva em consideração dados duplicados que representam um mesmo objeto do mundo real [CON 97], [TEJ 98], [LIU 99].

Um exemplo deste caso é quando uma mesma publicação possuir vários autores, cada autor possui uma cópia da mesma publicação, porém no processo de integração realizado no software desenvolvido, todas estas cópias são consideradas como um objeto único.

Este é um problema difícil de se resolver devido a características dos dados que são semi-estrutrados, onde não existe um identificador para cada elemento no documento. Para solucionar este problema o sistema implementa um filtro que é aplicado dobre o resultado da consulta. Este filtro apresenta o nível de similaridade sintática entre as publicações.

Por último podemos citar a necessidade de uma política de atualização que deve ser aplicada sobre os dados de forma a apresentar o mínimo de dados desatualizados possíveis. A solução para o domínio de informação do estudo de caso foi baseada em meta informações das bases remotas. Ou seja, foi analisada a periodicidade das atualizações sobre os dados remotos. Constatou-se que os dados remotos possuem um grau de atualização que pode influenciar qualitativamente as consultas sobre os dados materializados a partir de 30 dias.

A atualização da base local materializada pode ocorrer então de duas maneiras: parcial, onde somente dados alterados são processados; total, onde a base de dados é criada novamente. No modelo de atualização parcial cada currículo é analisado para verificar possíveis alterações. A característica que é analisada é o tamanho da fonte de informação. Em todos os processamentos o tamanho do documento é armazenado em uma tabela de registro. Em processamentos posteriores este tamanho, é verificado e validado e, em caso de alterações significativas de tamanho o currículo é extraído da base materializada e em seguida é processado novamente. A decisão de deletar o currículo completamente na base local foi escolhida devido ao fato que a tentativa de se localizar um elemento específico de publicação pode comprometer o desempenho do processo de atualização. Um estudo sobre manutenção de visões materializadas é descrito em [ABI 98] e [SIN 98].

No modelo de atualização total, o processo de integração é realizado novamente criando uma nova fonte de dados materializada. Esta fonte substitui então a base de dados antiga.

Entre os dois modelos, o processamento de atualizações parciais se mostrou mais rápido cerca de 25% do que o método de atualização total nos testes realizados. Outro fator que deve ser levando em consideração é que operações de alterações realizadas na base local são perdidas em ambos os casos, sendo que o processo parcial leva pouca vantagem sobre o outro método. Isto porque alterações locais em dados que não foram extraídos e integrados novamente são mantidas, somente os dados atualizados a partir da base remota perdem as atualizações em base local. Já o processo de atualização total, todas as atualizações locais são perdidas.

6.8 Dados Estatísticos

No estudo de caso foram localizados e consultados 117098 integrantes da base de dados da PROPESQ. Este processamento para a materialização de dados referentes à tabela autores levou aproximadamente 3 horas e meia sendo realizado por um computador Pentium II 266 MHz, com 128 MB de memória executando Windows 2000 Professional. Este tempo foi reduzido consideravelmente para 2 horas e meia de processamento quando executado em um computador Pentium III 750 MHz com 256 MB de memória executando Windows 2000 Professional.

A velocidade da rede também influenciou o tempo de execução no processo de integração. A velocidade da rede utilizada é de 100 Mbps e os dois tempos acima foram conseguidos em processamento realizado no período

vespertino – pior caso. O tempo de execução reduz cerca de 15 % no período matutino e 30% no período noturno.

Com relação aos dados da Plataforma Lattes, o processamento é mais demorado devido ao alto processamento de texto na linguagem Visual Basic. Foram levados em consideração dois modelos para a extração de dados do currículo Lattes. No primeiro modelo os dados eram requisitados do sistema Lattes e em seguida processados pelo sistema Extrator. No segundo modelo todos os currículos Lattes eram requisitados e armazenados em disco; finalizado este processo, o sistema Extrator recuperava os currículos do disco local. No primeiro modelo, foram gastas 8 horas de processamento, no segundo caso quase não houve diferença de tempo de processamento.

Em um dos processamentos foi desconsiderado o tempo de requisição a sistemas remotos, ou seja, todos os currículos já estavam em armazenados em disco. A velocidade de processamento é de aproximadamente de quarenta currículos por minuto.

De um total de 117089 integrantes da Universidade Federal do Rio Grande do Sul foram encontrados 1827 pesquisadores com vínculo junto ao CNPq. Destes currículos foram extraídas 110876 publicações.

O último processamento ocorreu no dia oito de outubro de 2001.

Um fato importante ocorrido no processamento de extração referente à base da PROPESQ é que muitos integrantes localizados pelo sistema de integração não são visíveis a partir da interface de consulta disponibilizada pela PROPESQ.

Os exemplos abaixo estão armazenados na base de dados local, porém não são acessados da interface da PROPESQ:

- http://astra.ufrgs.br/pesquisa/pesquisador.asp?Localiza=111520
- http://astra.ufrgs.br/pesquisa/pesquisador.asp?Localiza=30824
- http://astra.ufrgs.br/pesquisa/pesquisador.asp?Localiza=26808

7 Conclusões

A internet é a maior e mais heterogênea fonte de informação criada pelo homem. Desde a antigüidade o homem busca modelos de organização de informações para facilitar a recuperação dos mesmos. Estes modelos vieram mais tarde a constituir a base dos sistemas de banco de dados, onde as consultas são baseadas na semântica das informações armazenadas e o desempenho é garantido pelas estruturas de dados complexas disponibilizadas pelo sistema de banco de dados.

No entanto, a web surgiu como uma das bases de informação mais utilizadas pelas pessoas. Assim, os métodos de recuperação utilizados nos bancos de dados tradicionais foram aplicados buscando ter a mesma eficácia conseguida em modelos de banco de dados. Porém a web possui características que são completamente diferentes das tradicionais, tais como: esquema mal definido, atributos sem tipos definidos, alto volume de dados, alto grau de alterações da informação.

Para solucionar este problema vários modelos de recuperação de informação foram criados para melhorar o desempenho das consultas e melhorar a relevância dos documentos encontrados. Destes estudos surgiram os sistemas de busca mais utilizados, destacando o Altavista, com base de dados baseado em palavras chaves extraídas dos documentos HTML; Yahoo, baseado em diretórios de domínios de informação; e *Google*, que obtém a consulta combinando o resultado de vários sistemas de busca melhorando assim sua área de cobertura.

A partir de um certo momento, não somente recuperar os documentos relevantes era o objetivo principal, mas sim recuperar as informações contidas nos documentos e combiná-las com outras informações, conseguindo um novo grau de abstração da informação. Neste contexto, a web é vista como um base de dados heterogênea e distribuída que pode ser processada e integrada de forma a possibilitar a consulta a informações contidas em documentos diferentes distribuídos na web.

Com o intuito de resolver o processamento de consultas a dados em documentos heterogêneos foram definidos vários modelos de representação para dados semi-estruturados como o OEM, Araneus e outros. Estes modelos possuem uma linguagem própria para consulta dos dados, e são baseados em extratores que extraem a informação e a inserem em um modelo de dados mais complexo.

Outra forma de solucionar este problema foi desenvolver linguagens de consulta que visualizam a web como um grande grafo onde cada nó é um documento e cada arco é um *link*. Estes nós são explorados de acordo com os termos da consulta.

Por fim a distribuição dos dados levou à realização de estudos referentes não somente à extração de dados de um documento singular, mas de vários documentos onde os dados são combinados e relacionados formando uma pequena base de dados. Alguns estudos desta corrente realizavam o processamento através da modelagem de um esquema de integração global, onde um modelo canônico para representação dos dados distribuídos é definido [KAL 99].

A partir da definição de um esquema global criado a partir dos relacionamentos entre os dados das várias bases de informação, inicia-se o processo de integração. Este processo tem como objetivo disponibilizar uma camada intermediária entre usuário e dados distribuídos, de modo que o usuário possa consultar os dados de forma transparente em relação à distribuição e a estrutura da informação nas diversas bases. Alguns autores comparam o processo de integração como a criação de uma visão em sistemas de banco de dados. Com esta comparação podemos ter um processo de integração virtual ou materializada.

Na integração virtual, todo o processamento de extração dos dados e criação de relacionamentos no modelo canônico é realizado em tempo de execução da consulta realizada pelo usuário. A vantagem principal deste método é que a consulta sempre retorna dados atuais, porém existe um tempo de processamento adicional utilizado para o processo de integração que reflete o tempo de processamento da consulta como um todo.

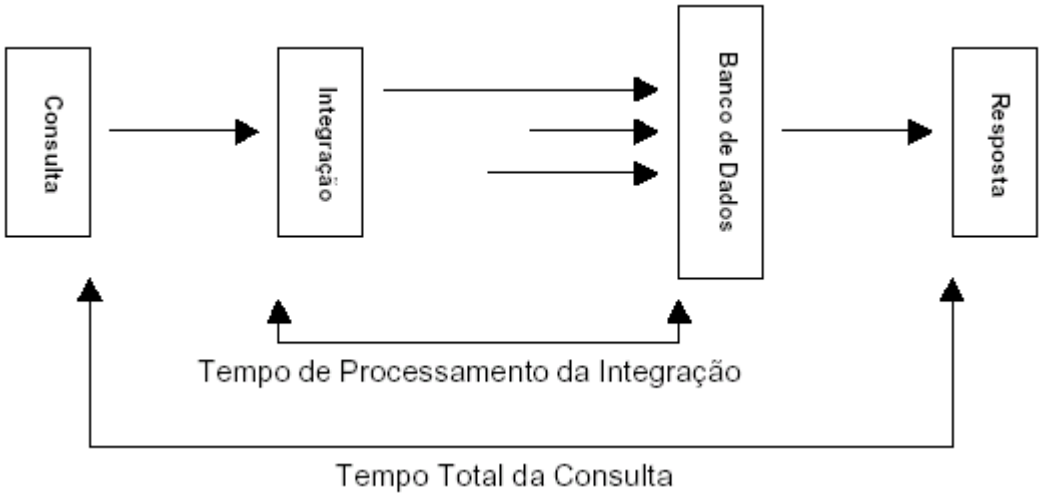

FIGURA 6.1 - Tempo de Execução (a)

Como é mostrada na figura 6.1, a cada consulta à base distribuída é disparado o processo de integração. Segundo análises de *log* de sistemas de consulta, os usuários tendem a repetir uma mesma consulta várias vezes, e uma mesma consulta pode ser realizada por usuários diferentes inúmeras vezes. Com isto o processamento é realizado novamente mesmo quando disparado por uma consulta semelhante.

Já no modelo de integração materializada, o processo de integração é realizado uma única vez para várias consultas. Isto porque a integração materializada armazena de forma persistente os dados extraídos em uma estrutura de dados complexa, como por exemplo, um banco de dados. Assim o tempo total para realizar uma consulta é menor se comparado com a integração virtual. A figura 6.2 mostra este modelo

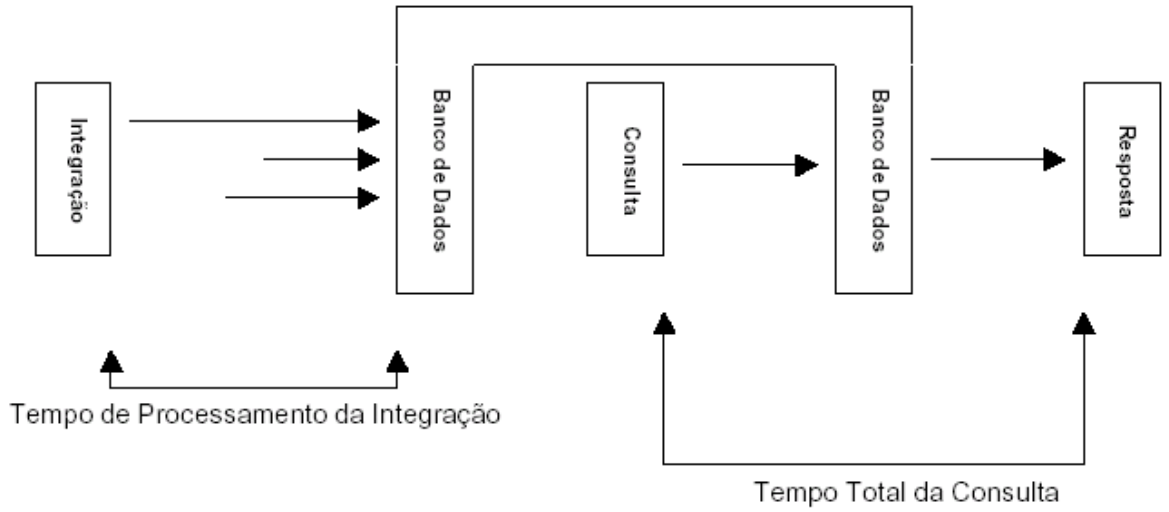

FIGURA 6.2 - Tempo de Execução(b)

O problema principal deste modelo é que os dados consultados são uma cópia dos dados originais. Se as bases distribuídas sofrem alterações constantes, os dados extraídos podem conter informações inválidas. Por isso é necessário um estudo da periodicidade de atualizações nas bases distribuídas de maneira que possa ser criada uma política de atualização dentro do processo de integração. Este modelo de integração é mais utilizado para bases de dados muito grandes onde o processo de extração consome muito tempo de processamento. Já bases menores podem utilizar o processo de integração virtual.

A política de atualização dentro de um sistema de integração pode ser total ou parcial. Na atualização parcial cada documento a ser processado possui uma *flag* que indica as atualizações ocorridas no documento. Por exemplo, tamanho do documento ou data de atualização que em alguns documentos na web são fornecidos através de uma função em *Javascript*. Assim podemos comparar com a

flag armazenada pelo sistema de integração caso as duas sejam diferentes, os dados na base local referentes ao documento são extraídos e processados novamente. Na atualização total, o processo de integração é executado novamente criando uma base mais atual dos dados.

Atualizações realizadas na base local em tuplas que não são alteradas pela atualização parcial são mantidas. Já no outro modelo, todas as alterações na base local são perdidas.

Atualizações parciais removem todos os elementos de um documento que foi alterado e refazem a extração e inserção em banco de dados. O método mais eficiente seria localizar apenas as informações que foram alteradas, porém, na maioria das fontes heterogêneas não existe identificador para um conjunto de dados e a análise destes para a localização de possíveis atualizações poderia aumentar muito o tempo de processamento influenciando no desempenho do processo de atualização.

A parte mais trabalhosa dos sistemas de integração é a definição de regras de extração. Estas podem ser criadas manualmente ou através de algum software de edição. Em alguns casos a definição de regras de extração manual é mais eficiente, pois as regras podem ser otimizadas considerando aspectos da semântica da aplicação e características da linguagem de programação que será utilizada. Na Internet existem várias documentos que são gerados automaticamente a partir de *scripts* que inserem informações que estão armazenadas em banco de dados. Estes documentos podem ser vistos como uma visão de banco de dados com marcas especiais para apresentação na Web, e são facilmente mapeados pelas regras de extração, pois tem seu conteúdo em uma estrutura que é mantida ao longo do documento.

Como contribuições deste estudo de caso, podemos citar em primeiro lugar o sistema de extração de dados curriculares da plataforma Lattes, que cria uma base de dados que pode ser consultada de acordo com as necessidades da aplicação, sendo que estes dados armazenados podem ser importados para outros sistemas em formato XML a partir de interfaces apropriadas. Outras contribuições são o conjunto de classificações de documentos na web referentes à distribuição, heterogeneidade e autonomia; a classificação dos sistemas de integração referentes ao controle semântico dos dados; e por fim, uma análise dos problemas e vantagens em se construir um sistema de integração para dados distribuídos na web.

8 Trabalhos Futuros

Alguns estudos são previstos para a continuidade deste trabalho. Um dos mais importantes é agregar ao sistema de integração funcionalidades que garantam a integridade semântica da aplicação, ou seja, o processo de integração deve possuir meios de identificar objetos semi-estrutruados de bases diferentes que identificam um mesmo objeto do mundo real.

Outro importante aspecto está relacionado ao processo de atualização da base integrada. Neste caso deve se criar modelos de atualização que garantam a validade dos dados e que o processo de atualização consiga identificar com precisão quais itens de informação em um documento HTML foi alterado.

Muitos estudos estão sendo realizados nesta área. Os primeiros trabalhavam como modelos de dados para dados semi estruturados. Os estudos mais recentes iniciaram um processo de materialização diretamente em banco de dados, garantindo com isso que a maioria das aplicações acessem os dados de maneira tradicional utilizando comandos SQL já conhecidos.

O sistema criado para o estudo de caso possui vários componentes separados em programas diferentes. Um ponto a ser estudado é a criação de uma interface única para estes componentes de maneira que o usuário consiga gerenciar o processo de integração sem maiores conhecimentos do processo.

Neste ultimo mês de novembro o CNPq anunciou um formato em XML para os dados do Currículo Lattes. Logo estes dados possuem agora uma estrutura muito mais elaborada para um processo de integração. Este processo ainda é necessário, pois cada documento em XML possui dados referentes a um único pesquisador. A materialização destes dados em uma única base possibilitaria a realização de vários tipos de consulta, inclusive a apresentação do currículo completo do pesquisador. Não é preciso dizer que o sistema de atualização será o mais beneficiado pois agora cada item de publicação pode possuir um atributo identificador, ou um atributo que informe à ultima data de atualização deste item.

Anexo 1 Arquivo DTD do Currículo Lattes

O arquivo DTD - Currículo Lattes define regras para formação da estrutura de transporte das informações relativas ao Currículo Lattes. Assim, será possível para as universidades tanto gerar currículos importáveis pelo sistema de currículos da Plataforma Lattes, quanto receber currículos do CNPq (ou de outras fontes), sem que seja necessário modificar-se a lógica de segurança ou, muito menos, de interação do CNPq com indivíduos. Fonte : http://lattes.cnpq.br

```
<!--                                        -->
<!--                                        -->
<!--        SEGMENTO DA PRODUCAO BIBLIOGRAFICA              -->
<!--                                        -->
<!--                                        -->
<!ELEMENT PRODUCAO-BIBLIOGRAFICA (TRABALHOS-EM-EVENTOS?, ARTIGOS-PUBLICADOS?,
LIVROS-E-CAPITULOS?, TEXTOS-EM-JORNAIS-OU-REVISTAS?, DEMAIS-TIPOS-DE-PRODUCAO-
BIBLIOGRAFICA?)>
<!ELEMENT TRABALHOS-EM-EVENTOS (TRABALHO-EM-EVENTOS+)>
<!ELEMENT TRABALHO-EM-EVENTOS (DADOS-BASICOS-DO-TRABALHO, DETALHAMENTO-DO-
TRABALHO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-
ATIVIDADE?, INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-TRABALHO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-TRABALHO
        NATUREZA (COMPLETO | RESUMO) #REQUIRED
        TITULO-DO-TRABALHO CDATA #IMPLIED
        ANO-DO-TRABALHO CDATA #IMPLIED
        PAIS-DO-EVENTO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DO-TRABALHO EMPTY>
<!ATTLIST DETALHAMENTO-DO-TRABALHO
        CLASSIFICACAO-DO-EVENTO (INTERNACIONAL | NACIONAL | REGIONAL | LOCAL)
#REQUIRED
        NOME-DO-EVENTO CDATA #IMPLIED
        CIDADE-DO-EVENTO CDATA #IMPLIED
        ANO-DE-REALIZACAO CDATA #IMPLIED
        TITULO-DOS-ANAIS-OU-PROCEEDINGS CDATA #IMPLIED
        VOLUME CDATA #IMPLIED
        FASCICULO CDATA #IMPLIED
        SERIE CDATA #IMPLIED
        PAGINA-INICIAL CDATA #IMPLIED
        PAGINA-FINAL CDATA #IMPLIED
        ISBN CDATA #IMPLIED
        NOME-DA-EDITORA CDATA #IMPLIED
        CIDADE-DA-EDITORA CDATA #IMPLIED
>
<!--   ############################ OBSERVACAO ##############################  -->
<!--  PARA MANTER A COMPATIBILIZACAO COM O DATACAPES (COLETA), E HAVENDO A   -->
<!--  INFORMACAO SOBRE O AUTOR, INCLUIR O NUMERO DO CPF PARA BRASILEIROS        -->
<!--   ############################ OBSERVACAO ##############################  -->
<!ELEMENT AUTORES EMPTY>
<!ATTLIST AUTORES
```

```
        NOME-COMPLETO-DO-AUTOR CDATA #IMPLIED
        NOME-PARA-CITACAO CDATA #IMPLIED
        CPF CDATA #IMPLIED
>
<!ELEMENT INFORMACOES-ADICIONAIS EMPTY>
<!ATTLIST INFORMACOES-ADICIONAIS
        DESCRICAO-INFORMACOES-ADICIONAIS CDATA #IMPLIED
>
<!ELEMENT ARTIGOS-PUBLICADOS (ARTIGO-PUBLICADO+)>
<!ELEMENT ARTIGO-PUBLICADO (DADOS-BASICOS-DO-ARTIGO, DETALHAMENTO-DO-ARTIGO,
AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?,
INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-ARTIGO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-ARTIGO
        NATUREZA (COMPLETO | RESUMO) #REQUIRED
        TITULO-DO-ARTIGO CDATA #IMPLIED
        ANO-DO-ARTIGO CDATA #IMPLIED
        PAIS-DE-PUBLICACAO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DO-ARTIGO EMPTY>
<!ATTLIST DETALHAMENTO-DO-ARTIGO
        TITULO-DO-PERIODICO-OU-REVISTA CDATA #IMPLIED
        ISSN CDATA #IMPLIED
        VOLUME CDATA #IMPLIED
        FASCICULO CDATA #IMPLIED
        SERIE CDATA #IMPLIED
        PAGINA-INICIAL CDATA #IMPLIED
        PAGINA-FINAL CDATA #IMPLIED
        LOCAL-DE-PUBLICACAO CDATA #IMPLIED
>
<!ELEMENT LIVROS-E-CAPITULOS (LIVROS-PUBLICADOS-OU-ORGANIZADOS?, CAPITULOS-DE-
LIVROS-PUBLICADOS?)>
<!ELEMENT LIVROS-PUBLICADOS-OU-ORGANIZADOS (LIVRO-PUBLICADO-OU-ORGANIZADO+)>
<!ELEMENT CAPITULOS-DE-LIVROS-PUBLICADOS (CAPITULO-DE-LIVRO-PUBLICADO+)>
<!ELEMENT LIVRO-PUBLICADO-OU-ORGANIZADO (DADOS-BASICOS-DO-LIVRO, DETALHAMENTO-DO-
LIVRO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?,
INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-LIVRO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-LIVRO
        TIPO (LIVRO_PUBLICADO | LIVRO_ORGANIZADO_OU_EDICAO) #REQUIRED
        NATUREZA (COLETANEA | TEXTO_INTEGRAL | VERBETE | ANAIS | CATALOGO |
ENCICLOPEDIA | LIVRO | OUTRA | PERIODICO) #REQUIRED
        TITULO-DO-LIVRO CDATA #IMPLIED
        ANO CDATA #IMPLIED
        PAIS-DE-PUBLICACAO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DO-LIVRO EMPTY>
<!ATTLIST DETALHAMENTO-DO-LIVRO
        NUMERO-DE-VOLUMES CDATA #IMPLIED
        NUMERO-DE-PAGINAS CDATA #IMPLIED
        ISBN CDATA #IMPLIED
```

```
        NUMERO-DA-EDICAO-REVISAO CDATA #IMPLIED
        NUMERO-DA-SERIE CDATA #IMPLIED
        CIDADE-DA-EDITORA CDATA #IMPLIED
        NOME-DA-EDITORA CDATA #IMPLIED
>
<!ELEMENT CAPITULO-DE-LIVRO-PUBLICADO (DADOS-BASICOS-DO-CAPITULO, DETALHAMENTO-DO-
CAPITULO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-
ATIVIDADE?, INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-CAPITULO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-CAPITULO
        TIPO CDATA #IMPLIED
        TITULO-DO-CAPITULO-DO-LIVRO CDATA #IMPLIED
        ANO CDATA #IMPLIED
        PAIS-DE-PUBLICACAO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DO-CAPITULO EMPTY>
<!ATTLIST DETALHAMENTO-DO-CAPITULO
        TITULO-DO-LIVRO CDATA #IMPLIED
        NUMERO-DE-VOLUMES CDATA #IMPLIED
        PAGINA-INICIAL CDATA #IMPLIED
        PAGINA-FINAL CDATA #IMPLIED
        ISBN CDATA #IMPLIED
        ORGANIZADORES CDATA #IMPLIED
        NUMERO-DA-EDICAO-REVISAO CDATA #IMPLIED
        NUMERO-DA-SERIE CDATA #IMPLIED
        CIDADE-DA-EDITORA CDATA #IMPLIED
        NOME-DA-EDITORA CDATA #IMPLIED
>
<!ELEMENT TEXTOS-EM-JORNAIS-OU-REVISTAS (TEXTO-EM-JORNAL-OU-REVISTA+)>
<!ELEMENT TEXTO-EM-JORNAL-OU-REVISTA (DADOS-BASICOS-DO-TEXTO, DETALHAMENTO-DO-
TEXTO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?,
INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-TEXTO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-TEXTO
        NATUREZA (JORNAL_DE_NOTICIAS | REVISTA_MAGAZINE) #REQUIRED
        TITULO-DO-TEXTO CDATA #IMPLIED
        ANO-DO-TEXTO CDATA #IMPLIED
        PAIS-DE-PUBLICACAO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DO-TEXTO EMPTY>
<!ATTLIST DETALHAMENTO-DO-TEXTO
        TITULO-DO-JORNAL-OU-REVISTA CDATA #IMPLIED
        ISSN CDATA #IMPLIED
        FORMATO-DATA-DE-PUBLICACAO NMTOKEN #FIXED "DDMMAAAA"
        DATA-DE-PUBLICACAO CDATA #IMPLIED
        VOLUME CDATA #IMPLIED
        PAGINA-INICIAL CDATA #IMPLIED
        PAGINA-FINAL CDATA #IMPLIED
        LOCAL-DE-PUBLICACAO CDATA #IMPLIED
>
```

```
<!ELEMENT DEMAIS-TIPOS-DE-PRODUCAO-BIBLIOGRAFICA (OUTRA-PRODUCAO-BIBLIOGRAFICA*, PARTITURA-MUSICAL*, PREFACIO-POSFACIO*, TRADUCAO*)>
<!ELEMENT OUTRA-PRODUCAO-BIBLIOGRAFICA (DADOS-BASICOS-DE-OUTRA-PRODUCAO, DETALHAMENTO-DE-OUTRA-PRODUCAO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?, INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DE-OUTRA-PRODUCAO EMPTY>
<!ATTLIST DADOS-BASICOS-DE-OUTRA-PRODUCAO
	NATUREZA CDATA #IMPLIED
	TITULO CDATA #IMPLIED
	ANO CDATA #IMPLIED
	PAIS-DE-PUBLICACAO CDATA #IMPLIED
	IDIOMA CDATA #IMPLIED
	MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME | HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
	HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
	FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DE-OUTRA-PRODUCAO EMPTY>
<!ATTLIST DETALHAMENTO-DE-OUTRA-PRODUCAO
	EDITORA CDATA #IMPLIED
	CIDADE-DA-EDITORA CDATA #IMPLIED
	NUMERO-DE-PAGINAS CDATA #IMPLIED
	ISSN-ISBN CDATA #IMPLIED
>
<!ELEMENT PARTITURA-MUSICAL (DADOS-BASICOS-DA-PARTITURA, DETALHAMENTO-DA-PARTITURA, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?, INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DA-PARTITURA EMPTY>
<!ATTLIST DADOS-BASICOS-DA-PARTITURA
	NATUREZA (CANTO | CORAL | ORQUESTRA | OUTRO) #REQUIRED
	TITULO CDATA #IMPLIED
	ANO CDATA #IMPLIED
	PAIS-DE-PUBLICACAO CDATA #IMPLIED
	IDIOMA CDATA #IMPLIED
	MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME | HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
	HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
	FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DA-PARTITURA EMPTY>
<!ATTLIST DETALHAMENTO-DA-PARTITURA
	FORMACAO-INSTRUMENTAL CDATA #IMPLIED
	EDITORA CDATA #IMPLIED
	CIDADE-DA-EDITORA CDATA #IMPLIED
	NUMERO-DE-PAGINAS CDATA #IMPLIED
	NUMERO-DO-CATALOGO CDATA #IMPLIED
>
<!ELEMENT PREFACIO-POSFACIO (DADOS-BASICOS-DO-PREFACIO-POSFACIO, DETALHAMENTO-DO-PREFACIO-POSFACIO, AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?, INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DO-PREFACIO-POSFACIO EMPTY>
<!ATTLIST DADOS-BASICOS-DO-PREFACIO-POSFACIO
	NATUREZA (PREFACIO | POSFACIO | APRESENTACAO | INTRODUCAO) #REQUIRED
	TITULO CDATA #IMPLIED
	ANO CDATA #IMPLIED
	PAIS-DE-PUBLICACAO CDATA #IMPLIED
	IDIOMA CDATA #IMPLIED
	MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME | HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
	HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
	FLAG-RELEVANCIA (SIM | NAO) "NAO"
```

```
>
<!ELEMENT DETALHAMENTO-DO-PREFACIO-POSFACIO EMPTY>
<!ATTLIST DETALHAMENTO-DO-PREFACIO-POSFACIO
        NOME-DO-AUTOR-DA-PUBLICACAO CDATA #IMPLIED
        TITULO-DA-PUBLICACAO CDATA #IMPLIED
        ISSN-ISBN CDATA #IMPLIED
        NUMERO-DA-EDICAO-REVISAO CDATA #IMPLIED
        VOLUME CDATA #IMPLIED
        SERIE CDATA #IMPLIED
        FASCICULO CDATA #IMPLIED
        EDITORA-DO-PREFACIO-POSFACIO CDATA #IMPLIED
        CIDADE-DA-EDITORA CDATA #IMPLIED
>
<!ELEMENT TRADUCAO (DADOS-BASICOS-DA-TRADUCAO, DETALHAMENTO-DA-TRADUCAO,
AUTORES*, PALAVRAS-CHAVE?, AREAS-DO-CONHECIMENTO?, SETORES-DE-ATIVIDADE?,
INFORMACOES-ADICIONAIS?)>
<!ELEMENT DADOS-BASICOS-DA-TRADUCAO EMPTY>
<!ATTLIST DADOS-BASICOS-DA-TRADUCAO
        NATUREZA (ARTIGO | LIVRO | OUTRO) #REQUIRED
        TITULO CDATA #IMPLIED
        ANO CDATA #IMPLIED
        PAIS-DE-PUBLICACAO CDATA #IMPLIED
        IDIOMA CDATA #IMPLIED
        MEIO-DE-DIVULGACAO (IMPRESSO | MEIO_MAGNETICO | MEIO_DIGITAL | FILME |
HIPERTEXTO | OUTRO | VARIOS) #IMPLIED
        HOME-PAGE-DO-TRABALHO CDATA #IMPLIED
        FLAG-RELEVANCIA (SIM | NAO) "NAO"
>
<!ELEMENT DETALHAMENTO-DA-TRADUCAO EMPTY>
<!ATTLIST DETALHAMENTO-DA-TRADUCAO
        NOME-DO-AUTOR-TRADUZIDO CDATA #IMPLIED
        TITULO-DA-OBRA-ORIGINAL CDATA #IMPLIED
        ISSN-ISBN CDATA #IMPLIED
        IDIOMA-DA-OBRA-ORIGINAL CDATA #IMPLIED
        EDITORA-DA-TRADUCAO CDATA #IMPLIED
        CIDADE-DA-EDITORA CDATA #IMPLIED
        NUMERO-DE-PAGINAS CDATA #IMPLIED
        NUMERO-DA-EDICAO-REVISAO CDATA #IMPLIED
        VOLUME CDATA #IMPLIED
        FASCICULO CDATA #IMPLIED
        SERIE CDATA #IMPLIED
>
```

Anexo 2 Código de Extração Lattes

Código referente ao processo de extração das publicações contidas no currículo Lattes.

```
Option Explicit

'VARIAVEIS BASICAS DO CURRICULO // globais
Dim Nome_Citacao As String
Dim Sexo As String
Dim VetorTabela() As Long
Dim DES_TIPO As String 'recebe o tipo de publicacao

Private Sub Start_Click()

'VARIAVEIS DO BANCO DE DADOS
Dim Conn As ADODB.Connection
Dim Reg As ADODB.Recordset

'VARIAVEL PARA RECEBER O CODIGO HTML
Dim CurSource As String

'Variaveis para importacao de dados
Dim NumTabelas As Integer
Dim i As Integer
Dim Tipo As String

Dim Pause As Integer
Pause = 0

'abrindo conexao com o banco de dados PlataformaLattes com ODBC Lattes

    Set Conn = New ADODB.Connection

    Conn.Open "Lattes"

'executando SQL para aquisicao do nome e cod_cnpq dos professores com curriculos

    Set Reg = New ADODB.Recordset

    Reg.Open "Autor", Conn, adOpenKeyset, adLockOptimistic, adCmdTable

'configurando tempo de espera de resposta do componente ITC

    WebConn.RequestTimeout = 240

'iniciar leitura do registro

    Reg.MoveFirst

    NumProfessor.Text = 1

    While Not Reg.EOF

        NumProfessor.Text = NumProfessor.Text + 1

        If Trim(Reg!COD_CNPq) <> "" Then 'se existe curriculo para o professor

            CNPq.Text = Reg!COD_CNPq

            'requisicao ao site cnpq
```

```
        CurSource = WebConn.OpenURL("http://genos.cnpq.br:12010/dwlattes/owa/prc_imp_cv_ext?f_cod=" &
Trim(Reg!COD_CNPq))

        'extrair dados basicos

        ExtractName CurSource

        'gravar dados basicos

        Reg!Des_Nom_Citacao = UCase(Replace(Nome_Citacao, Chr(13), ""))
        Reg!DES_SEXO = UCase(Replace(Sexo, Chr(13), ""))
        Reg.Update

        'verifica a existencia de publicacoes no curriculo Lattes
        If TemPublicacoes(CurSource) Then

           LocalizarTabelas (CurSource) 'localiza todas as tabelas

           NumTabelas = UBound(VetorTabela)

           For i = 1 To NumTabelas - 1
              'identifica o tipo de tabela {TIPO | CONTEUDO }
              Tipo = IdentificarTabela(CurSource, VetorTabela(i), VetorTabela(i + 1))

              If Tipo = "Tipo" Then 'extrair o tipo de publicacao

                 DES_TIPO = TipoPublicacao(Mid(CurSource, VetorTabela(i), VetorTabela(i + 1) - VetorTabela(i)))

              Else 'extrair os itens

                 ItemPublicacao (Mid(CurSource, VetorTabela(i), VetorTabela(i + 1) - VetorTabela(i)))

              End If

           Next

           'extrai os elemento das ultima tabela no vetor VetorTabela()
           Tipo = IdentificarTabela(CurSource, VetorTabela(NumTabelas), Len(CurSource) - VetorTabela(NumTabelas))

           If Tipo = "Tipo" Then

              ItemPublicacao (Mid(CurSource, VetorTabela(NumTabelas), Len(CurSource) - VetorTabela(NumTabelas)))

           End If

           CriarVinculoAutorObra (Reg!ID_autor)

        End If


     End If

     Reg.MoveNext

  Wend


End Sub

Sub ExtractName(CodFonte As String) 'Extrai nome para publicacao e sexo do autor
Dim PosInicial As Long
Dim PosFinal As Long
Dim Tabela As String

'localizar primeira tabela

  PosInicial = InStr(1, CodFonte, "<TABLE")

'localizar primeira tabela
```

```
    PosInicial = InStr(PosInicial + 1, CodFonte, "<TABLE")

'localizar primeira tabela

    PosInicial = InStr(PosInicial + 1, CodFonte, "<TABLE")
    PosFinal = InStr(PosInicial, CodFonte, "</TABLE>")

'selecionando tabela para exportação

    Tabela = Mid(CodFonte, PosInicial, PosFinal - PosInicial)

'localizar quarto elemento da tabela

    PosInicial = InStr(1, Tabela, "<FONT FACE") 'primeiro
    PosInicial = InStr(PosInicial + 1, Tabela, "<FONT FACE") 'segundo
    PosInicial = InStr(PosInicial + 1, Tabela, "<FONT FACE") 'terceiro
    PosInicial = InStr(PosInicial + 1, Tabela, "<FONT FACE") 'quarto

'localizar conteudo do quarto elemento da tabela

    PosInicial = InStr(PosInicial, Tabela, ">")
    PosFinal = InStr(PosInicial, Tabela, "<")
    Nome_Citacao = Mid(Tabela, PosInicial + 1, PosFinal - PosInicial - 1)

'localizar sexto elememento

    PosInicial = InStr(PosInicial + 1, Tabela, "<FONT FACE") 'quinto
    PosInicial = InStr(PosInicial + 1, Tabela, "<FONT FACE") 'sexto

'localizar conteudo do sexto elemento

    PosInicial = InStr(PosInicial, Tabela, ">")
    PosFinal = InStr(PosInicial, Tabela, "<")
    Sexo = Mid(Tabela, PosInicial + 1, PosFinal - PosInicial - 1)

End Sub

'Verifica se existe publicacoes no curriculo Lattes
Function TemPublicacoes(Texto As String) As Boolean

Dim PosInicial As Long
Dim PalavraChave As String

    PalavraChave = "<A NAME=" & Chr(34) & "Produção bibliográfica" & Chr(34)

    PosInicial = InStr(1, Texto, PalavraChave)

    If PosInicial = 0 Then

        TemPublicacoes = False

    Else

        TemPublicacoes = True
        ReDim Preserve VetorTabela(1)
        VetorTabela(1) = PosInicial

    End If

End Function

'Cria um mapa de posicoes de tabelas
Sub LocalizarTabelas(Texto As String)
Dim PosTabela As Long

    PosTabela = VetorTabela(1) 'posicao inicial das publicacoes

    PosTabela = InStr(PosTabela, Texto, "<TABLE") 'primeira tabela
```

```
    VetorTabela(1) = PosTabela

    'acrescenta mais uma celula no vetor

    While PosTabela <> 0

       PosTabela = InStr(PosTabela + 1, Texto, "<TABLE")

       If PosTabela <> 0 Then 'se o arquivo não chegou ao fim

          ReDim Preserve VetorTabela(UBound(VetorTabela) + 1)

          VetorTabela(UBound(VetorTabela)) = PosTabela

       End If

    Wend

End Sub

Function IdentificarTabela(Texto As String, x As Long, y As Long) As String
Dim PalavraChave As String
Dim z As Long

    PalavraChave = " </FONT></FONT>"

    z = InStr(x, Texto, PalavraChave)

    If z > y Then

       IdentificarTabela = "Tipo"

    Else

       IdentificarTabela = "Conteudo"

    End If

End Function

'Extrai o tipo de Publicacao
Function TipoPublicacao(Texto As String) As String
Dim PosInicial As Long
Dim PosFinal As Long
Dim PalavraChave As String

    PosInicial = InStr(1, Texto, "></TD>") 'posicao inicial

    PosInicial = InStr(PosInicial + 1, Texto, "=-1>") 'inicio de conteúdo

    PosFinal = InStr(PosInicial + 1, Texto, "</FONT") 'fim de conteúdo

    TipoPublicacao = Mid(Texto, PosInicial + 4, PosFinal - PosInicial - 4)

End Function

Sub ItemPublicacao(Texto As String)  'Extrai um item de publicacao
Dim PosInicial As Long
Dim PosFinal As Long
Dim Final1 As Long
Dim Final2 As Long
Dim PalavraChave As String
Dim ItemPub As String

'variaveis de banco de dados
Dim ConnPub As ADODB.Connection
Set ConnPub = New ADODB.Connection
```

```
ConnPub.Open "Lattes"

   PalavraChave = " </FONT></FONT>"
   PosInicial = 1
   Final1 = 1

   While Final1 <> 0

      PosInicial = InStr(PosInicial, Texto, PalavraChave)
      PosInicial = InStr(PosInicial + 1, Texto, "<FONT SIZE=-1>")
      PosInicial = InStr(PosInicial + 1, Texto, ">")

      Final1 = InStr(PosInicial + 1, Texto, "<TR")
      Final2 = InStr(PosInicial + 1, Texto, "</TABLE")

      If Final1 <> 0 Then
         PosFinal = Final1
      Else
         PosFinal = Final2
      End If

      ItemPub = Replace(Mid(Texto, PosInicial, PosFinal - PosInicial), Chr(13), "")

      ItemPub = Replace(ItemPub, """, "")

      ConnPub.Execute ("insert into Publicacao (DES_TIPO,MEGADATA) values ('" & DES_TIPO & "','" & ItemPub & "')")

   Wend

End Sub

Sub CriarVinculoAutorObra(Cod_Autor As Long)
'variaveis de banco de dados
Dim ConnAutorObra As ADODB.Connection
Dim Registro As ADODB.Recordset

   Set ConnAutorObra = New ADODB.Connection

   ConnAutorObra.Open "Lattes"

   Set Registro = ConnAutorObra.Execute("select id_publicacao from publicacao where id_publicacao not in (select
ID_publicacao from PublicacaoAutor )")

   Registro.MoveFirst

   While Not Registro.EOF

      ConnAutorObra.Execute ("insert into PublicacaoAutor (ID_PUBLICACAO, ID_AUTOR) values ('" &
Registro!ID_Publicacao & "','" & Cod_Autor & "')")

      Registro.MoveNext

   Wend

End Sub

Private Sub Stop_Click() 'Fecha o programa
   End
End Sub
```

Bibliografia

[ABI 97] ABITEBOUL, S. et al. The Lorel Query Language for Semistructured Data. **International Journal on Digital Libraries**, [S.l.], v.1, n.1, Apr.1997.

[ABI 97a] ABITEBOUL, S. **Queryng Semi-Structured Data**. 1997. Disponível em: <http://www-rocc.infira.fr/~abiteboul/pub/icdt97.ps>. Acesso em: 20 dez. 2001.

[ABI 98] ABITEBOUL, S. et al. Incremental Maintenance for Materialized Views Over Semistructured Data. In: CONFERENCE ON VERY LARGE DATA BASES, VLDB, 24., 1998. **Proceedings ...** [S.l.:s.n.],1998.

[ADE 98] ADELBERG, B. Nodose - A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents. In: SPECIAL INTEREST GROUP ON MANAGEMENT OF DATA, SIGMOD, 1998, Seatle. **Proceedings...** [S.l.:s.n.], 1998.

[ARO 98] AROCENA, G. O.; MENDELZON, A. O. WebOQL: Restructuring Documents Databases in Web. In: INTERNACIONAL CONFERENCE ON DATA ENGINEERING, 1998. **Proceedings...** [S.l.:s.n.],1998.

[ARO 97] AROCENA, G.; MENDELZON, A. O. Applications of a Web Query Language. **Computer Networks and ISDN Systems**, Amsterdam, 1997.

[ASH 97] ASHISH, N.; KNOBLOCK, C. Wrapper Generation for Semi-Structured Internet Sources. **SIGMOD Record**, New York, v. 26, n. 4, Dec. 1997.

[ATZ 97] ATZENI, P.; MECCA, G. Cut and Paste. In: International Symposiun of Principles of Data Base System, 16., 1997. **Proceedings**…[S.l.:s.n.], 1997.

[ATZ 98] ATZENI, P.; MECCA, G.; MERIALDO, P. Semistructured and Structured Data in the Web. In: WORKSHOP ON MANAGEMENT OF SEMISCTRUCTURED DATA, 1997. **Proceedings**…[S.l.:s.n], 1997.

[ATZ 98a] ATZENI, P. et al. **A Logival Model for Metadata in Web Bases**. 1998. Disponível em: <http://www.difa.unibas.it/Araneus/publications/ercim98.ps.gz>. Acesso em: 20 dez. 2001.

[BER 99] BERGAMASCHI, S.; CASTANO, S.; VINCINI, M. Semantic Integration of Semisctructured and Structured Data Sources. **SIGMOD Records,** New York [S.l.], v. 36, n. 3, 1999.

[BUN 96] BUNEMAN, P. et al. A Query Language and Otimization Techniques for Unstructured Data. In: SIGMOD CONFERENCE, 1996, Canada. **Proceedings…**[S.l.:s.n], 1996.

[BUN 97] BUNEMAN, P. **Semistructured Data**. Disponível em: <ftp.cis.upenn.edu/pub/peter/semistructured-papers.ps>. Acesso em: 20 dez. 2001.

[BUS 90] BUSSE, S. et al. **Federated Information Systems:** Concepts, Terminology and Architectures. Berlim: Technische Universitat Berlin, 1990. (Relatório n.99-9).

[BRA 94] BRA, P. et al. Information Retrieval in Distribute Hypertext. 1994. **Journal on Computer Networks and ISDN Systems,** [S.l.], n. 27, p. 183-192, 1994.

[CHA 94] CHAWATHE, S. et al. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In: ANIVERSARY MEETING OF THE INFORMATION PROCESSING SOCIETY OF JAPAN, 100.,1994, Tokyo, Japan. **Proceedings...** [S.l.:s.n.], 1994.

[CHA 96] CHANDY, K. M.; WEISMAN, L. **A World Wide Distributed System using Java and the Internet**. 1996. Disponível em: <http:// www.infospheres.caltech.edu/papers/hpdc96.ps>. Acesso em: 20 dez. 2001.

[CHI 97] CHIDLOVSKI B. et al. Towards Sophisticated Wrapping of Web-based Information Repositories. In: INTERNATIONAL RIAO CONFERENCE, 5., 1997, Montreal - Canada. **Proceedings...** [S.l.:s.n.], 1997.

[CHR 99] CHRISTOPHIDES, V.; SIMEON. J.; CLUET, S. **Semistructured and Structured Integration Reconciled**. 1999. Disponível em: <http://www.ics.forth.gr/~christop/>. Acesso em: 21 dez. 2001.

[COL 97] COOLEY, R.; MOBASHER, B.; SRIVASTAVA, J. Web Mining: Information and Pattern Discovery on the World Wide Web. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS, 9., 1997. **Proceedings...** [S.l.:s.n.], 1997. Disponível em: <http://maya.cs.depaul.edu/%257Emobasher/papers/webminer-tai97.ps>. Acesso em: 20 dez. 2001.

[CON 97] CONRAD, S. et al. Schema Integration with Integrity Constraints. In: BRITISH NATIONAL CONFERENCE ON DATABASES, BNCOD 15., 1997. **Proceedings...** [S.l.:s.n.], 1997.

[CRA 97] CRAIG, A. K.; MINTON, S.; TEJADA, S. Modeling Web sources for Information Integration. In: AMERICAN ASSOCIATION FOR ARTIFICIAL INTELLIGENCE, Madison, WI. **Proceedings...** [S.l.:s.n.], 1997. Disponível em: <http://citeseer.nj.nec.com/knoblock98modeling.html>. Acesso em: 21 dez. 2001.

[DEA 99] DEAN, J.; HENZINGER, M. Finding Related Pages in the World Wide Web. In WWW8 – COMPUTER NETWORKS, 1999. **Proceedings...** [S.l.:s.n.], 1999. Disponível em: <http://www.research.digital.com/SRC/personal/monika/papers/monika-www8-1.ps.gz>. Acesso em: 21 dez. 2001.

[EMB92] EMBLEY, D. W. ; KURTS, B. D. ; WOOD_ELD, S. N. **Object-Oriented Systems Analysis:** A Model-Driven Approach. Englewood Cliffs: Yourdon Press, 1992. p. 335-361.

[EMB 98] EMBLEY, D. W. et al. A conceptual Modeling Approach to Extracting Data From the Web. In: INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING / THE ENTITY RELATIONSHIP APPROACH. **Proceedings...** [S.l.:s.n.], 1998. Disponível em: <http://osm7.cs.byu.edu/deg/pappers/er98.ps>. Acesso em: 21 dez. 2001.

[EMB 98a] EMBLEY, D. W. Ontology based Extraction and Sctructuring of Information from Data Rich Unstructured Documents. In: CONFERENCE INTERNATIONAL OF KNOWLEDGE MANAGEMENT, CIKM, 1998. **Proceedings...** [S.l.:s.n.], 1998. Disponível em: <http://osm7.cs.byu.edu/deg/pappers/cikm98.ps>. Acesso em: 21 dez. 2001.

[EIK 99] EIKVIL, L.1999. **Information Extraction from WWW.** Repport Nr. 945, July 1999. ISBN 82-539-0429-0, 1999 Disponível em: <http://www.nr.no/bild/PostScript/webIE-rep945.ps>. Acesso em: 21 dez. 2001.

[FER 97] FERNANDEZ, M. et al. A query Language for Web Site Management System. **SIGMOD Records**, New York, v. 26, n. 3, Sept. 1997.

[FLO98] FLORESCU, D.; LEVY, A.; MENDELZON, A. Database Techiniques for the WWW: A Survey. Special Interest Group on Management of Data, **SIGMOD Records**, v. 27, n. 3, Sept. 1998. Disponível em: <http://128.95.4.112/homes/alon/webdb.ps>. Acesso em: 21 dez. 2001.

[GAO 99] GAO, X.; STERLING, L. **Semi-structured Data Extraction from Heterogeneous Sources**. 1999. Disponível em: <http://citeseer.nj.nec.com/gao99semistructured.html>. Acesso em: 21 dez. 2001.

[GAR 99] GAROFALAKIS, M. N. et al. Data Mining andthe Web: Past, Presents and Future. In: WORKSHOP INTERNATIONAL DATABASE MANAGEMENT,1999. **Proceedings...** [S.l.:s.n.], 1999.

[GLO 2000] GLOVER, E.; LAWRENCE, L.; GILES, C.L Web Search - Your Way. 2000. **Communications of the ACM**, New York, v. 44, n. 12, Dec. 2001.

[GUP 2001] GUPTA, A.; MUMICK. Maintenance of Materialized Views Problems, Techiniques and Application. **IEEE Quarterly Bulletin on Data Engineering**, 2001. Disponível em: <http://citeseer.nj.nec.com/22323.html>. Acesso em: 21 dez. 2001.

[HAM 97] HAMMER, J. et al. Extracting Semistructured Information from the Web. In: THE WORKSHOP ON MANAGEMENT OF SEMISTRUCTURED DATA, 1997. **Proceedings...** [S.l.:s.n.], 1998. Disponível em: <http://www-db.stanford.edu/pub/papers>. Acesso em: 21 dez. 2001.

[HEI 85] HEOMBIGNER, D.; LEOD, M.C. A Federated Architecture for Information Management. **ACM Transactions on Office Information Systems**, [S.l.], v.3, n. 3, 1985.

[HSU 98] HSU, C. H.; DUNG, M. T. Generating finite-State Transducerw for Semistructured Data Extraction the Web. **Information Systems Magazine**, [S.l.], v. 23, n. 8, 1998.

[JOH 2001] JOHANNESSON, P.; WNAGLER, B.; JAYAWEERA, P. Application and Process Integration - concepts, Issues, and Research Directions. In: NUTEK – NATIONAL BOARD FOR INDUSTRIAL AND TECHINICAL DEVELOPMENT, 2001. **Proceedings...** [S.l.:s.n.], 2001. Disponível em: <http://www.nutek.se/index.htm>. Acesso em: 21 dez. 2001.

[KAN 2000] KANTORSKI, G. Z.; RIBEIRO, C. H. F. P. Heterogeneous Database Interoperability the WWW. In: BRAZILIAN SYMPOSIUM ON DATABASES, SBBD, 15., 2000, João Pessoa, Brazil. **Proceedings...** [S.l.:s.n.], 2000. p. 79-88.

[KAL 99] KALINICHENKO, L. A. **Integration of Heterogeneous SemiStructured Data Models in the Canonical One**. 1999. Disponível em: <http://www.ipi.ac.ru/synthesis/publications/inthet/inthet.ps>. Acesso em: 21 dez. 2001.

[KON 95] KONOPNICKI, D.; SHMUELI, O. W3QS: A Query Systems for the World Wide Web. In: INTERNATIONAL ON VERY LARGE DATA BASES, 21., 1995. **Proceedings...** [S.l.:s.n.], 1995.

[KRU 2000] KRUGER, A. et al. Deadliner: Building a new Niche Search Engine. In: CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENTE OF DATA, 2000, Dallas, Texas . **Proceedings...** [S.l.:s.n.], 2000.

[KUS 97] KUSHMERICK, N.; WELD, D. S.; DOORENBOS, R. Wrapper Induction for Information Extraction. In: IJCAI, 1997. **Proceedings...** [S.l.:s.n.], 1997.

[LAB 2000] LABRINIDIS, A.; ROUSSOPOULOS, N. Webview Materialization. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2000, Dallas, Texas. **Proceedings...** [S.l.:s.n.], 1997.

[LAH 98] LAHIRI, T.; ABITEBOUL, S. WINDOW, J. Ozone: Integrating Semisctructured and Sctructured Data. In: WORKSHOP ON DATABASES PROGRAMMING LANGUAGES, 1998. **Proceedings...** [S.l.:s.n.], 1998.

[LAK 96] LAKSHAMANAN, L.; SADRI, F.; SUBRAMANIAN, I. N. A Declarative Language for Queryng and Restructuring the Web. In: INTERNATIONAL WORKSHOP ON RESEARCH ISSUES IN DATA ENGINEERING, 6., 1996. **Proceedings...** [S.l.:s.n.], 1996.

[LAW 98] LAWRENCE, S.; GILES, C. L. Context and Page Analysis for Improved Web Search. **IEEE Internet Computing**, [S.l.], v. 2, n. 4, p. 38 – 46, 1998.

[LAW 99] LAWRENCE, S.; GILES, C. L.. Accessibility of Information on the Web. **Nature Magazine**, [S.l.], v. 400, July 1999.

[LAW 99a] LAWRENCE,S.; GILES, C. L. Searching the Web: General and Scientific Information Access. **IEEE Communications**, [S.l.], v. 37, n. 1, p. 116-122,1999.

[LAW 2000] LAWRENCE, S. 2000. Context in Web Search. **IEEE Data Engineering Bulletin**, [S.l.], v. 23, n. 3, p. 25-32, 2000.

[LIT 90] LITWIN, W.; MARK, L.; ROUSSOPOULOS, N. Interoperability of Multiple Autonomous Database. **ACM Computing Surveys**, New York, v. 22, n. 3, 1990.

[LIU 99] LIU, M.; LING, T. W.; GUAN, T. Integration of Semistructured Data with Partial and Inconsistent Information. In: INTERNATIONAL DATABASE ENGINEERING AND APPLICATION SYMPOSIUM, 1999. **Proceedings...** [S.l:s.n.], 1999. Disponível em: <http://www.cs.uregina.ca/%257Emliu/papers/semi-IDEAS99.ps>. Acesso em: 21 dez. 2001.

[MAN 96] MANHEIM, M. L. Beyound Groupware and Workflow. In: PRIISM, 1996. **Proceedings...** [S.l.:s.n.], 1996.

[MEN 96] MENDELZON, A.; MIHAILA, G.; MILO, T. Queryng the World Wide Web. In: INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED INFORMATION SYSTEMS, 1., 1996. **Proceedings...** [S.l.:s.n.], 1996.

[NES 98] NESTOROV, S.; ABITEBOUL, S.; MOTWANI, R. Extracting Schema from Semistructured Data, 1998. Disponível em: <http://theory.stanford.edu/~rajeev/postscripts/semi.ps>. Acesso em: 21 dez. 2001.

[PAP 96] PAPAKONSTANTINOU, Y.; ABITEBOUL, S. GARCIA-MOLINA, H. Object Fusion in Mediator Systems. In: VLDB CONFERENCE, 22., 1996, Munbai (Bombay), India. **Proceedings...** [S.l.:s.n.], 1998

[PAP 99] PAPAKONSTANTINOU, Y.; VELIKHOV, P. Enhancing Semistructured Data Mediators wtih Document Type Definitions. In: INTERNATIONAL CONFERENCE DATABASE ENGINEERING, 1999. **Proceedings...** [S.l.:s.n.], 1999. Disponível em: <http://www.db.ucsd.edu/publications/icde99.ps>. Acesso em: 21 dez. 2001.

[ROS 2000] ROSA, M. et al. Materializing the Web. In: CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS, 2000. **Proceedings...** [S.l.:s.n.], 2000. Disponível em: <ftp://ftp.dis.uniroma1.it/pub/iocchi/publications/web-coopis98.ps.gz>. Acesso em: 21 dez. 2001.

[SAT 99] SATTLER, K.; SAAKE, G. Supporting Information Fusion with Federated Database Technologies. In: INTERNATIONAL WORKSHOP ON ENGINEERING FEDERATED INFORMATION SYSTEMS, EFIS, 2., 1999. **Proceedings...** [S.l.:s.n.], 1999.

[SIN 98] SINDONI, G. Incremental Maitenance of Hipertext Views. In: INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES, 1998. **Proceedings...** [S.l.:s.n.], 1998. Disponível em: <http://www.difa.unibas.it/Araneus/publications/webdb98.ps.gz>. Acesso em: 21 dez. 2001.

[SHE 90] DHETH, A. P.; LARSON, J. A. Federated Database Systems for Managing Distributed, Heterogeneous and Automomous Database. **ACM Computing Surveys**, New York, v. 22, n. 3, 1990.

[SMI 97] SMITH, D.; LOPEZ, M. Information Extraction for SemiSctructured Documents. In: WORKSHOP ON MANAGEMENT OF SEMISTRUCTURED DATA. 1997. **Proceedings...** [S.l.:s.n.], 1997. Disponível em: <http://www.research.att.com/%257Esuciu/WORKSHOP-PAPERS/paper09.ps>. Acesso em: 21 dez. 2001.

[TEJ 98] TEJADA, S.; KNOBLOCK, C.A.; MINTON, S. 1998. Handing Inconsistency for Multi-source Integration. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELIGENCE, 16. ; CONFERENCE ON INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE, 19., 1999, Orlando, Florida. **Proceedings...** [S.l.:s.n.], 1998