Aplicação de Ontologias a Dados Semi-Estruturados

Ronaldo dos Santos Mello[1,2]
Carlos Alberto Heuser[1]

[1]Universidade Federal do Rio Grande do Sul
Instituto de Informática
Caixa Postal 15064
Av. Bento Gonçalves, 9500 - Bloco IV - Prédio 43412
CEP: 91501-970 - Porto Alegre - RS - Brasil
e-mail: {ronaldo,heuser}@inf.ufrgs.br

[2]Universidade Federal de Santa Catarina
Centro Tecnológico
Departamento de Informática e de Estatística
Campus Universitário Trindade - Caixa Postal 476
CEP: 88040-900 - Florianópolis - SC - Brasil
e-mail: ronaldo@inf.ufsc.br

Abstract

When semistructured data are considered, it is possible to have different instances representing the same information. This heterogeneity makes difficult the prescription of a database schema as well as query processing. Data in the *Web* are the most common example of semistructured data. An ontology, in a database point of view, is a partial specification of a domain, that basically describes entities and their relationships. Ontologies are very useful as a consensual semantic support for efficient access and manipulation of semistructured data. This work presents a comparative of proposals that deals with these two recent research topics.

Keywords: databases, semistructured data, ontologies, *Web*

1 Introdução

A *Web* vem se tornando um vasto repositório eletrônico de dados. Porém, grande parte destes dados não estão mantidos em bancos de dados por terem uma organização bastante heterogênea, que pode variar de um texto informal até um conjunto de registros bem formatados [Abi97b, Nes97]. A alta heterogeneidade destes dados torna complexa as operações de manipulação uma vez que não existe uma estrutura que sirva de base para a execução das mesmas. Consultas são, em geral, realizadas através de navegação, que é uma atividade exaustiva, ou busca por palavras-chaves, que apresenta diversas desvantagens, como o grande volume de dados a ser investigado e o custo de manutenção de índices de documentos [Ash97, Cat98, Atz97b]. Dados que se enquadram nestas características são denominados dados semi-estruturados. O gerenciamento desses dados traz novos desafios à comunidade científica de bancos de dados, pois não é possível definir um esquema *a priori* e, consequentemente, consultas declarativas.

Ontologias vêm sendo utilizadas na Ciência da Computação, mais especificamente na área de inteligência artificial, para descrever vocabulários conceituais utilizados por uma comunidade de agentes para compartilhamento de conhecimento [Gru93]. Na área de banco de dados, ontologias vêm sendo aplicadas como esquemas conceituais para integração de bancos de dados heterogêneos.

A aplicação de ontologias no gerenciamento de dados semi-estruturados é recente em termos de pesquisa científica. Este artigo tem o objetivo de analisar comparativamente várias propostas de aplicação do conceito de ontologia à manipulação de dados semi-estruturados.

O restante deste artigo está organizado da seguinte forma: o capítulo dois caracteriza dados semi-estruturados e alguns tópicos de pesquisa relacionados e relevantes. O capítulo três define ontologias de um ponto de vista de banco de dados e mostra suas classificações. O capítulo quatro

apresenta algumas propostas de aplicação de ontologias a dados semi-estruturados e o capítulo cinco é reservado à análise das mesmas. O capítulo seis é dedicado à conclusão.

2 Dados Semi-Estuturados

Dados semi-estruturados são dados que não são estritamente tipados nem totalmente não-estruturados. Dados presentes na *Web* são um exemplo típico: em alguns casos existe uma estrutura predefinida (tabelas ou registros), em outros a estrutura está total ou parcialmente embutida no dado (documentos no formato de artigo) ou então, quase não apresentam informações descritivas (imagem) [Abi97b]. Em geral, o esquema de representação está presente juntamente com o dado, ou seja, o mesmo é auto-descritivo, sendo necessário um procedimento analítico para identificação e extração de estrutura [Bun97].

Os três tópicos mais relevantes na pesquisa relativa a dados semi-estruturados são: modelagem, consulta e extração de dados.

Modelos de dados semi-estruturados são grafos direcionados rotulados, onde os vértices representam objetos identificáveis e as arestas são arcos para objetos componentes [Flo98, Bun97]. Como cada ocorrência de dado pode ter uma estrutura diferente, não há restrição no número de arcos que partem de um dado objeto. OEM (*Object Exchange Model*) [Pap95, Ham97] é o modelo mais popular. Objetos OEM podem ser atômicos (*integer*, *string*, *gif*, etc) ou complexos (conjunto de referências a objetos). Todos os vértices folha do grafo apresentam tipos atômicos. A figura 1 mostra um objeto semi-estruturado representado no modelo OEM. O objeto em questão é um departamento de um hospital (*Cardiologia*). A heterogeneidade dos dados pode ser percebida através das diferentes representações que objetos do tipo *paciente* apresentam.

Linguagens de consulta a dados semi-estruturados devem permitir pesquisas em grafos [Fer97, Abi97a, Bun96, Atz97a, Li99, Deu99]. Alguns requisitos para tais linguagens são: (i) *a especificação de expressões de caminho* que percorram a estrutura de objetos semi-estruturados para obter informação ou formular uma condição sobre um atributo. A forma mais usual de especificação é a concatenação de nomes de rótulos. Tais expressões podem ainda indicar padrões de busca, quando combinadas com certos caracteres especiais. Essa facilidade é útil quando se desconhece total ou parcialmente a estrutura dos objetos; (ii) *o uso de coerção* (critérios de conversão de tipos) na comparação entre atributos de objetos, para lidar com as heterogeneidades de tipo e estrutural dos dados; (iii) *consulta ao esquema de classes de dados semi-estruturados*, quando este existir, para a identificação de características a serem consultadas.

O processo de *extração de dados* tem por objetivo construir uma visão estruturada de dados semi-estruturados para facilitar a sua manipulação [Dor00]. Este processo envolve mecanismos que identificam, recuperam e estruturam dados relevantes, transformando, em geral, dados de uma fonte de dados como a *Web* para dados adequados a um modelo de dados, seja ele semi-estruturado (OEM) ou estruturado (esquema relacional), ou algum formato com maior estruturação (XML) que possa ser entendido por uma gramática de processamento de consultas. Nesse processo, *wrappers* e mediadores atuam como módulos intermediários entre aplicações e fontes de dados semi-estruturadas, sendo responsáveis, respectivamente, pelo acesso e integração de dados.

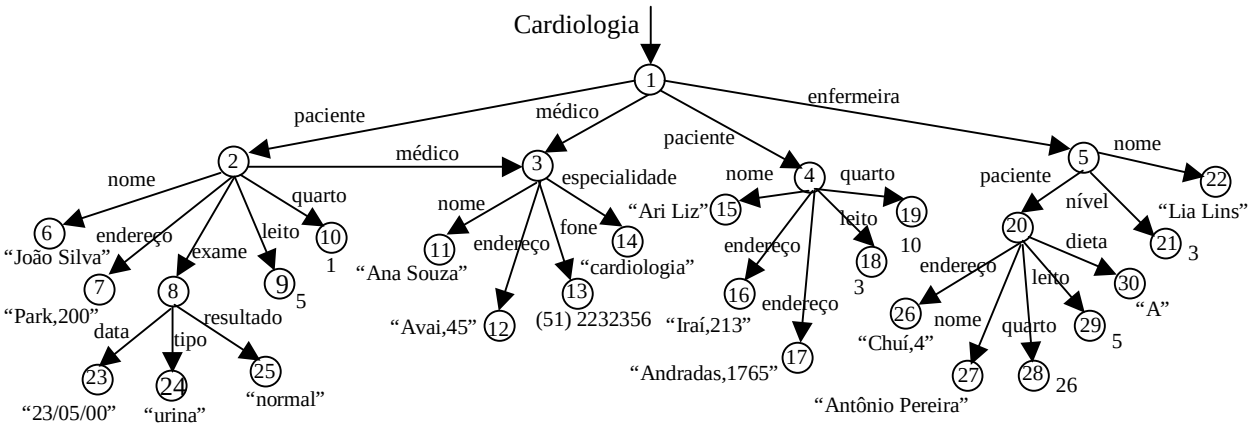

Figura 1- Exemplo de um objeto semi-estruturado representado no modelo OEM [Ber99]

3 Ontologias

O termo “*ontologia*”, na Ciência da Computação, vêm sendo aplicado desde o início da década de 90 na área de inteligência artificial para representação computacional de conhecimento em áreas como engenharia de conhecimento e processamento de língua natural. Neste contexto, podem ser entendidas como uma especificação formal e explícita de uma conceitualização consensual [Gru93].

Recentemente, ontologias vêm sendo utilizadas nas áreas de banco de dados e recuperação de informação como suporte à interoperabilidade de fontes de dados distribuídas e heterogêneas. Do ponto de vista de banco de dados, uma ontologia pode ser considerada uma especificação parcial de um universo de discurso que descreve basicamente conceitos, relações entre conceitos e regras de integridade. Porém, uma ontologia não é estritamente um esquema conceitual de dados. Um esquema conceitual descreve, dentre outras coisas, a estrutura de um banco de dados em um alto nível de abstração. Já uma ontologia não representa a estrutura das fontes de dados associadas a ela, apenas propõe uma estrutura de consenso para grupos de usuários, sendo essa estrutura instanciada através de mecanismos de integração de dados. Assim, uma ontologia é um mecanismo de interpretação parcial ou total do universo de dados de uma ou mais fontes, não existindo obrigatoriamente uma correspondência direta entre possíveis estruturas implícitas ou explícitas dessas fontes e a estrutura da ontologia.

Ontologias podem ser classificadas em [Gua9?]: (i) *ontologia de nível superior*, que descreve conceitos gerais, como espaço, tempo, objeto, assunto, ação e/ou metadados como entidades, relações e atributos; (ii) *ontologias de domínio e de tarefa*, que podem ser especializações da ontologia de nível superior, descrevendo, respectivamente, um vocabulário para um universo de discurso (medicina ou automóveis) e para uma tarefa genérica (diagnóstico ou venda); (iii) *ontologia de aplicação*, que depende tanto de um domínio quanto de uma tarefa particular, podendo ser uma especialização de ambas. Corresponde a regras impostas por conceitos do domínio quando executam uma certa tarefa, como por exemplo, substituição de uma unidade sobressalente de um automóvel.

4 Propostas de Aplicação de Ontologias a Dados Semi-Estruturados

Grande parte das linguagens de consultas a dados semi-estruturados baseiam-se na especificação de caminhos a serem percorridos em um grafo, considerando apenas a forma de apresentação dos dados em um documento e não a sua semântica. Esta limitação pode ser eliminada com o uso de ontologias, que podem atuar como esquemas conceituais de um conjunto de fontes de dados semi-estruturados, a partir do qual consultas declarativas podem ser formuladas sobre as fontes

de dados. Ontologias podem também guiar processos de extração de dados através da manutenção de informações que auxiliem na identificação de atributos de conceitos e relações.

Algumas propostas de aplicação de ontologias a dados semi-estruturados são descritas a seguir.

4.1 Observer

Observer é um sistema de informação global, em desenvolvimento na Universidade Politécnica de Madrid, que utiliza ontologias para o processamento de consultas a fontes de dados heterogêneos estruturados e semi-estruturados [Nie98]. Ontologias descrevem conceitos e regras organizados em uma hierarquia de especialização que captura a semântica do conteúdo de uma ou mais fontes.

Dois tipos de relacionamento existem neste contexto: (i) *relacionamento entre ontologias*, onde cada conceito de uma ontologia pode ter uma informação de mapeamento para conceitos da mesma ou de outras ontologias. Os tipos de relacionamento suportados são: *sinônimos*, *mais geral que*, *menos geral que*, *interseção*, *disjunção* e *cobertura*; (ii) *relacionamento entre ontologia e fonte de dados*, onde cada conceito da ontologia possui uma informação de mapeamento que o relaciona com estruturas das fontes de dados através de uma álgebra relacional estendida. Uma fonte é alcançada por apenas uma ontologia.

Um exemplo de mapeamento de um conceito de uma ontologia chamado *Livro* para fontes de dados é o seguinte:

```
CONCEPT Livro:
< [Union [Selection pub-BDI.publication [= pub-BDI.publication.formato “Livro”]]
            [Selection bib-BDI.ref [= bib-BDI.ref.type “Livro”]]],
  pub-BDI.publication.código,
  string>
```

Nesta descrição de mapeamento, tem-se uma expressão de álgebra relacional estendida como primeiro argumento, seguida do(s) atributo(s) que identifica(m) seus objetos e o(s) tipo(s) deste(s) atributo(s). O conceito *Livro* está presente em duas fontes de dados: *pub-BDI* e *bib-BDI*. A álgebra atua como uma linguagem de consulta intermediária entre a especificação textual da consulta do usuário e as linguagens de consulta ou métodos de acesso utilizados pelos *wrappers*. Expressões de álgebra são otimizadas e divididas em expressões individuais para cada fonte de dados antes de serem traduzidas para expressões de consulta compreendidas por cada *wrapper*.

A arquitetura do sistema é composta por servidores de ontologias, um gerenciador de relacionamentos inter-ontologias (GRI), *wrappers* e um processador de consultas. O processador de consultas é o núcleo da arquitetura, realizando os seguintes passos na execução de uma consulta: construção, acesso e expansão. Os dois últimos passos são cíclicos, terminando quando o usuário estiver satisfeito com o resultado.

No primeiro passo, através de uma interface gráfica, o usuário seleciona uma ontologia alvo e edita a consulta. Para o acesso a dados, o processador de consultas invoca o servidor da ontologia, que recupera e correlaciona os dados, com o auxílio de informações de mapeamento e dos *wrappers*, passando o resultado de volta ao processador. Caso o usuário queira enriquecer a consulta, ocorre o que se chama de expansão: uma nova ontologia alvo é selecionada e a consulta é reescrita na linguagem (vocabulário de conceitos) desta nova ontologia alvo, com o auxílio do GRI, preservando-se o máximo possível a semântica da mesma. Caso esta reescrita não seja completa (considerando o número de conceitos da consulta que foram mapeados com sucesso), o percentual de perda de informação não pode ser superior a um limite máximo de perda estipulado pelo usuário. Não ocorrendo este problema, o acesso a essa nova ontologia é feito e o ciclo se repete.

4.2 SHOE

SHOE (*Simple HTML Ontology Extensions*) é uma linguagem, desenvolvida na Universidade de Maryland, Estados Unidos, que estende a linguagem HTML com *tags* orientadas a conhecimento, provendo uma estrutura para aquisição deste conhecimento [Hef99]. Taxionomias de conceitos e regras de inferência disponíveis em ontologias são referidas em SHOE e habilitam a descoberta de conhecimento implícito em documentos da *Web*.

SHOE faz parte de um ambiente cuja arquitetura é composta por ontologias, uma ferramenta de marcação, que produz documentos no formato SHOE, um *browser* chamado *Exposé* e uma base de conhecimento chamada *Parka*. Conhecimento SHOE é descoberto em documentos através de *Exposé* e adicionado à *Parka*.

Ontologias são criadas também através da sintaxe SHOE, que especifica hierarquias de conceitos, relacionamentos e regras de inferência. Relacionamentos geralmente compreendem navegação em *links*, muito comum em pesquisas na *Web*. Atributos de conceitos e relacionamentos são definidos através de regras que associam valores a instâncias de conceitos. Na seqüência, uma especificação de ontologia em SHOE (a) e uma instância gerada a partir de uma marcação SHOE baseada nesta ontologia (b) são mostradas.

```
<HTML>
. . .
<BODY>
<ONTOLOGY ID="Pessoas" VERSION="1.0">
<USE-ONTOLOGY ID="Base Ontology" VERSION="1.0"
PREFIX="base">
. . .
<DEF-CATEGORY NAME="Pessoa" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Estudante" ISA="Pessoa">
<DEF-CATEGORY NAME="City" ISA="base.SHOEEntity">
. . .
<RELATION NAME="localEstudo">
        <ARG POS="1" TYPE="Estudante">
        <ARG POS="2" TYPE="City">
</RELATION>
. . .
</ONTOLOGY>
</BODY>
. . .
</HTML>
```

(a)

```
<HTML>
. . .
<BODY>
<INSTANCE KEY="http://www.examplo/shoe/examplo.html">
<USE-ONTOLOGY ID="Pessoas" VERSION="1.0"
PREFIX="pessoas" URL="http://www.exemplo/shoe/pessoas.html">
. . .
<CATEGORY NAME="pessoas.Estudante">
<RELATION NAME="pessoas.name">
        <ARG POS="TO" VALUE="João Silva">
</RELATION>
<RELATION NAME="pessoas.localEstudo">
        <ARG POS="TO"
VALUE="http://www.exemplo/shoe/examplolUniversidade.html">
</RELATION>
</INSTANCE>
...
</BODY>
...
</HTML>
```

(b)

A ferramenta de marcação adiciona a semântica da ontologia a páginas HTML. Páginas SHOE descrevem uma ou mais instâncias de conceitos. A descrição de uma instância consiste da ontologia que esta faz referência, o conceito que a classifica e suas propriedades. A inserção de marcações semânticas é facilitada por uma interface gráfica onde um documento HTML pode ser criado e o usuário pode selecionar ontologias, conceitos e relações e associar valores a regras. Conceitos de diversas ontologias podem ser associados a dados de um mesmo documento. No caso de páginas HTML criadas externamente que possuam dados relevantes, definem-se páginas de sumário com *tags* SHOE e *links* para essas páginas.

4.3 Ontobroker

Ontobroker é uma ferramenta baseada em ontologias da Universidade de Karlsruhe, Alemanha, que processa documentos especificados em linguagens de marcação como HTML e XML, provendo recuperação inteligente de informação [Erd99, Fen99]. A ferramenta possui uma arquitetura formada pelos seguintes componentes que interagem com ontologias: (i) uma *máquina de consulta*, que recebe consultas e as responde, verificando o conteúdo de uma *base de conhecimento*; (ii) um *agente de informação*, responsável pela coleta de conhecimento factual da *Web* e armazenamento no banco de dados, lidando com vários estilos de meta-anotações diretas, como XML e HTML-A; (iii) uma *máquina de inferência*, que usa fatos, a terminologia e os axiomas das ontologias para derivar conhecimento factual adicional que também é armazenado na base de conhecimento. As ontologias são o princípio geral de estruturação de dados: o agente de informação as utiliza para extrair fatos, a máquina de inferência para inferir fatos, o gerenciador do banco de dados para estruturar os dados e a máquina de consulta para formular consultas.

Um dos formatos de marcação semântica de *Ontobroker* é XML. O agente de informação traduz conceitos e atributos de ontologias para elementos de uma estrutura XML através de uma DTD (*Data Type Description*). As ontologias, neste contexto, funcionam como um padrão para a sintaxe de documentos XML, ao mesmo tempo em que permitem consultas com caráter semântico. Recebe-se especificações ontológicas e gera-se arquivos DTD XML. As DTDs geradas definem classes de conceitos que auxiliarão na marcação futura de documentos XML. Uma especificação parcial de uma ontologia (formalismo orientado a objetos) (a) e sua correspondente DTD (b) é mostrada a seguir:

```
Object[ ].
Pessoa:: Object.
  Empregado:: Pessoa.
        EquipeAcadêmica:: Empregado.
  Estudante:: Pessoa.
        EstudanteDoutorado:: Estudante.
Publicação:: Object.
  Livro:: Publicação.
  Artigo:: Publicação.
. . .
Pessoa[nome =>> string; email =>> string; telefone =>> string; publicação =>> Publicação, editor =>> Livro].
Publicação[autor =>> Pessoa; título =>> string; ano =>> number; abstract =>> string].
Livro[editor =>> Pessoa].
FORALL Pes1, Pub1Pub1: Livro[editor ->> Pes1] <-> Pes1: Pessoa[editor ->> Pub1]
```

(a)

```
<!ENTITY % Pessoa “Pessoa | Empregado | EquipeAcadêmica | Estudante | EstudanteDoutorado” >
<!ENTITY % Publicação “Publicação | Livro | Artigo” >
<!ELEMENT Pessoa (#PCDATA | nome | email | telefone | publicação | editor)*>
<!ELEMENT Publicação (#PCDATA | autor | título | ano | abstract)*>
<!ELEMENT Livro (#PCDATA | autor | título | ano | abstract | editor)*>
<!ELEMENT autor (#PCDATA | %Pessoa;)*>
<!ELEMENT título (#PCDATA>
<!ELEMENT ano (#PCDATA)*>
<!ELEMENT editor (#PCDATA | %Book; | %Pessoa;)*>
```

(b)

A especificação da DTD não impõe nenhuma restrição sobre qual dos elementos definidos deve ser o elemento raiz no documento XML, assim como a obrigatoriedade de existência de todos os

subelementos (atributos). Um exemplo de ocorrência do elemento *Livro* é mostrado no trecho a seguir de um documento XML que indica *ka2.dtd* como sendo o arquivo DTD que especifica os conceitos ontológicos a serem instanciados:

```
<!DOCTYPE Livro SYSTEM "ka2.dtd">
. . .
<Livro>
        <título> Sistema de Banco de Dados </título>
        <ano> 1994 </ano>
        <autor> Henry Korth </autor>
        <autor> Abraham Silberschatz </autor>
</Livro>
. . .
```

4.4 Proposta de Embley

A proposta de Embley é uma estratégia de extração de dados que utiliza ontologias como suporte semântico para a coleta e estruturação de dados semi-estruturados [Emb98]. Esta estratégia é particularmente interessante para conjuntos de documentos que seguem um certo padrão de discurso.

A ontologia é um esquema conceitual de um universo de discurso que representa os dados de um conjunto de documentos. Ela serve de entrada, juntamente com um documento semi-estruturado, para uma procedimento de extração que produz como saída um documento estruturado e filtrado a ser inserido em um banco de dados, conforme mostra a figura 2.

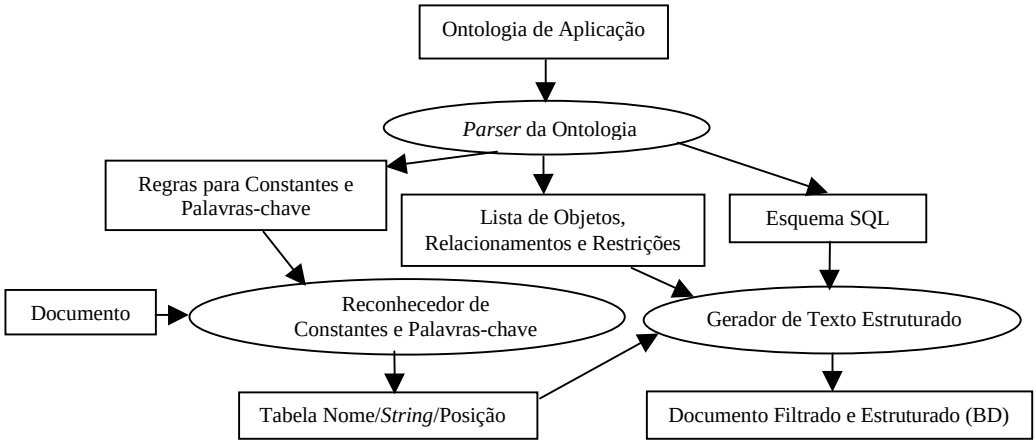

Figura 2 - Procedimento de extração de dados baseado em ontologia

O *parser* (analisador sintático) da ontologia gera três arquivos: *regras*, *lista de objetos* e *esquema SQL*. O primeiro arquivo descreve as regras para identificação de elementos da ontologia (conceitos, atributos e relacionamentos) no documento, como constantes válidas, expressões regulares que descrevem o formato de atributos e palavras-chaves associadas. O *esquema SQL* é uma seqüência de comandos de criação de tabelas resultante do mapeamento do esquema conceitual. A *lista de objetos* é uma associação entre os objetos da ontologia e as declarações das tabelas do *esquema SQL*, mais as restrições de cardinalidade.

Para cada *documento* de entrada, dois procedimentos são feitos utilizando os arquivos gerados: (i) reconhecimento de constantes e palavras-chaves e (ii) geração de texto estruturado. O *reconhecedor* produz uma *tabela nome/string/posição* com base na análise do *documento* e do arquivo de *regras*. O

gerador utiliza esta tabela para povoar o *banco de dados*, utilizando a *lista de objetos* e o *esquema SQL*. O *reconhecedor* tem por incumbência identificar *strings* no texto do documento conforme as expressões regulares, gravando o seu nome (atributo ou palavra-chave), valor (*string*) e posição no texto. O *gerador* se vale de heurísticas, baseadas na análise de restrições da *lista de objetos* e dos elementos da *tabela*, para gerar tuplas do *esquema SQL* e inseri-las no *banco de dados*.

4.5 InfoSleuth

O sistema *InfoSleuth* utiliza uma abordagem baseada em agentes cooperativos em um ambiente distribuído, para obtenção e análise de informação em diversos níveis de abstração. Além dos agentes, o sistema conta ainda com várias ontologias, que descrevem conceitos, análises e eventos [Per98, Unr98].

Ontologias de domínio são modeladas como diagramas entidade-relacionamento (ER), especificando hierarquias de herança de conceitos, além de relacionamentos e atributos. As ontologias de eventos provêem mudanças ou observações analíticas em dados. Modificações em instâncias ontológicas (visões) ocorrem quando dados são alterados nas fontes heterogêneas ou diretamente nas visões. Essas modificações são detectadas via investigações periódicas nas fontes ou, no sentido oposto, *triggers* para as fontes são ativados em virtude de mudanças na visão. As ontologias de análise dão suporte à intercomunicação de agentes durante tarefas de análise de dados. Toda vez que um agente realiza uma operação de análise, ele cria uma instância de uma atividade analítica da ontologia (inicializa a operação), especificando os recursos de dados, filtragens a serem feitas, pré-processamento dos dados e algoritmos de análise necessários.

InfoSleuth fornece um suporte especial para consulta a documentos semi-estruturados através de uma máquina de recuperação de informação (MRI) composta por operadores que podem ser combinados para aumentar a expressividade das pesquisas. Estes operadores são: (i) *<word> (w1)*: verifica se *w1* é uma palavra no corpo do texto; (ii) *<phrase> (w1, w2, ..., wn)*: verifica se *w1, w2, ..., wn* formam uma frase (em alguma ordem) no corpo do texto; (iii) *<sentence> (w1, w2, ..., wn)*: verifica se *w1, w2, ..., wn* aparecem em alguma sentença (em alguma ordem) no corpo do texto; (iv) *<thesaurus> (w1)*: verifica se uma expansão a nível de *Thesaurus* aparece no corpo do texto para *w1*; (v) *<stem> (w1)*: verifica se algum radical de *w1* aparece no corpo do texto; (vi) *<topic> (t1)*: indica um tópico predefinido através de uma combinação de operadores; (vii) *<accrue> (t1, t2, ..., tn)*: verifica se os tópicos *t1, t2, ..., tn* aparecem no corpo do texto, podendo cada um deles ter um peso associado.

Cada elemento da ontologia pode ter uma expressão de tópico (ET) associada a ele, que é um operador *<accrue>* utilizado para consultas a coleções de documentos baseadas em conteúdo. Supondo um universo de discurso de relações político-partidárias, um exemplo de diagrama ER e suas correspondentes ETs é mostrado na figura 3.

Agentes de recursos (semelhantes a *wrappers*), responsáveis pelo acesso a dados textuais, recebem consultas sobre entidades ontológicas em SQL e realizam um mapeamento para uma ET geral a ser submetida à MRI. Esta ET geral é uma combinação das ETs de todos os elementos da ontologia envolvidos na consulta. Por exemplo, a seguinte consulta busca documentos referentes ao presidente *Fernando Henrique Cardoso*:

select has_document
from Pessoa
where nome = ‘Fernando Henrique Cardoso’
and profissão = ‘presidente’

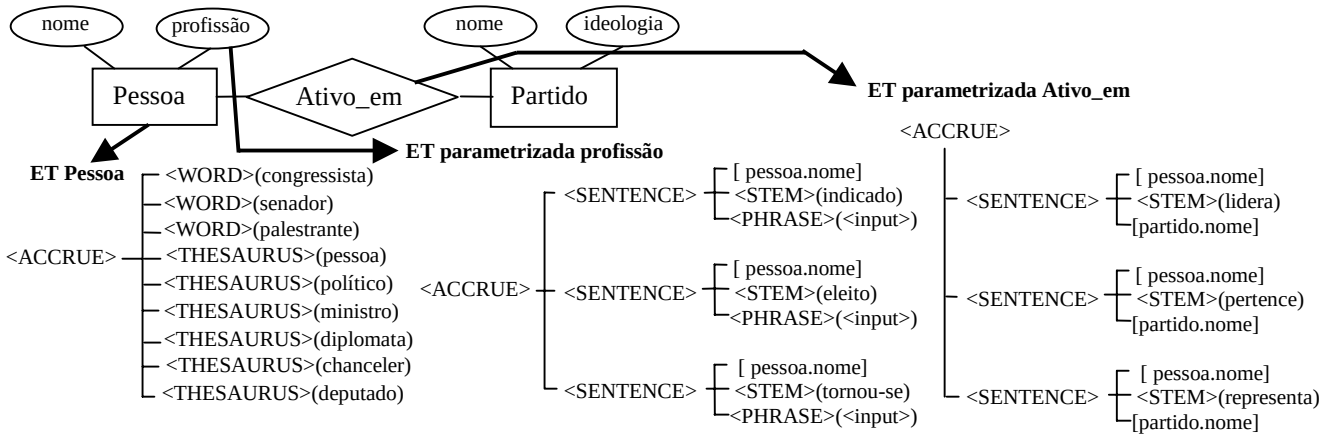

Figura 3 - Exemplos de expressões de tópicos associadas a elementos de uma ontologia

Esta consulta refere-se à entidade *pessoa* e aos atributos *nome* e *profissão*, gerando a ET geral:

<ACCRUE> (
<TOPIC> (pessoa),
<PHRASE> (<WORD> (Fernando) <WORD> (Henrique) <WORD> (Cardoso)),
<ACCRUE> (<SENTENCE> (<PHRASE> (<WORD> (Fernando) <WORD> (Henrique) <WORD> (Cardoso)), <STEM> (indicado), <WORD> (presidente)) ...)

Esse processo de mapeamento suporta ainda certos tipos de junção e heurísticas para determinação de prioridades na ordem de aplicação de predicados de seleção sobre atributos.

4.6 MOMIS

MOMIS (*Mediator envirOnment for Multiple Information Sources*) é um ambiente para integração de fontes de dados estruturados e semi-estruturados, em desenvolvimento na Universidades de Modena e Milão, Itália [Ber99]. A arquitetura do ambiente apresenta os seguintes componentes: (i) um *esquema conceitual ou ontológico* orientado a objetos especificado na linguagem ODL_I^3 (linguagem de integração semântica); (ii) *wrappers* para tradução de esquemas em representações ODL_I^3; (iii) e componentes *mediador* e *processador de consultas*, baseados em duas ferramentas: ARTEMIS e ODB-Tools.

A integração de dados exige que dados de fontes semi-estruturadas sejam organizados em classes, cada classe correspondendo a um rótulo estruturado do grafo OEM. Estas classes, assim como esquemas de fontes estruturadas, são traduzidas para classes ODL_I^3. Um exemplo de especificação ODL_I^3 para a classe *Paciente* do departamento de *Cardiologia* da figura 1 é a seguinte:

Interface Paciente (source semistructured Departamento-Cardiologia)
{ *attribute string* nome;
attribute string endereço;
attribute set <Exame> exame*;
attribute integer quarto;
attribute integer leito;
attribute string dieta*;
attribute set <Médico> médico*; };

O símbolo ‘*’ indica que o atributo pode ser opcional em algumas instâncias.

Para a definição da ontologia, deve-se descrever três tipos de relacionamentos terminológicos para classes e atributos: *sinônimo-de* (t1 SYN t2), *termo-mais-geral-que* (t1 BT t2) e *termo-relacionado-a* (t1 RT t2). O relacionamento simétrico à *termo-mais-geral-que* é *termo-mais-específico-que* (t1 NT t2). A descoberta destes relacionamentos é uma tarefa semi-automática feita com o auxílio da ferramenta *ODB-Tools*.

Na seqüência, o módulo mediador de MOMIS converte a ontologia em um esquema global. Para tanto, são associados *coeficientes de afinidade* a relacionamentos entre classes ontológicas de fontes de dados distintas, com base nos seus nomes e seus atributos. Um procedimento semi-automático de clusterização hierárquica classifica as classes de acordo com estes coeficientes, gerando uma *árvore de afinidade* que tem classes como folhas e nodos intermediários com um valor de afinidade associado às classes do seu *cluster* (subárvore). A determinação dos coeficientes e da árvore de afinidade são tarefas da ferramenta ARTEMIS. Após, um procedimento semi-automático de criação de esquemas globais é executado em dois passos, nesta ordem: (i) para cada *cluster Cli* gera-se uma classe global *Ci*. No caso de atributos com relacionamento SYN, apenas um deles é selecionado como atributo global. No caso de atributos com relacionamento BT/NT, o atributo mais geral em *Cli* é selecionado como atributo global; (ii) definem-se regras de mapeamento dos atributos globais para os atributos locais das fontes de dados. Regras de integridade (*rules*) podem ser especificadas, expressando relacionamentos semânticos entre fontes de dados distintas.

Um exemplo parcial de classe global é a classe *Hospital_Paciente*, considerando DC a fonte semi-estruturada referente ao departamento de *Cardiologia* e DT um banco de dados relacional com dados do departamento de *Tratamento Intensivo*:

```
Interface Hospital_Paciente {
        attribute nome
                mapping_rule (DT.Paciente.primeiro_nome and DT.Paciente.sobrenome),
                DC.Paciente.nome;
        attribute médico
                  mapping_rule DC.Paciente.médico, DT.Paciente.id_doutor;
        attribute departamento
                   mapping_rule DC.Paciente = 'Cardiologia',
                                DT.Paciente = 'Tratamento Intensivo',
                                DT.PacienteDispensado = 'Tratamento Intensivo'}
rule R1 forall X in Hospital_Paciente:
        (X.Exame.resultado = 'problemas coração') then X.departamento = 'Cardiologia';
```

4.7 WebKB

WebKB é uma ferramenta para indexação, anotação e exploração de conhecimento em documentos da *Web*, em desenvolvimento na Universidade de Griffith, Austrália [Mar99]. Utiliza uma linguagem de representação de conhecimento que define associações e especializações entre termos predefinidos em uma única e ampla ontologia, projetada para facilitar a criação de ontologias de aplicação. A ferramenta possui operadores que podem ser aplicados a diversos formatos de documentos, como HTML, textos indentados e inglês formalizado.

WebKB utiliza abordagens léxica, estrutural e baseada em conhecimento para consulta e definição de conhecimento. A abordagem léxica processa documentos através de um comando padrão *Unix* que lista documentos acessíveis direta ou indiretamente a partir de um certo número de *links*. As abordagens estrutural e baseada em conhecimento são empregadas na linguagem de representação de conhecimento. Através de *tags* especiais, embutem-se comandos de especificação, indexação ou consulta em documentos. Os termos da ontologia são utilizados nestes comandos para definir

ocorrências de relações ou conceitos. Essa estratégia aproxima a definição do conhecimento da fonte de dados em que ele se encontra.

Indexação pode ser aplicada a qualquer elemento de um documento textual ou HTML, como uma sentença, seção, referência, imagem ou o documento completo. O trecho de código a seguir define uma indexação sobre a segunda ocorrência da sentença "*o veículo vermelho*" presente em um documento HTML, referente aos conceitos *cor* e *veículo* da ontologia:

```
$ (Indexation
        (Context: Language: GC; Ontology: http://www.bar.com/topLevelOntology.html;
                Repr_author: phmartin; Creation_date: Mon sep 14 02:32:21 PDT 1998;
                Indexed_doc: http://www.bar.com/exemplo.html;)
        (DE:    (2nd occurence) o veículo vermelho)
        (Repr: [Cor: vermelha] <- (Cor) <- [Veículo]) )$
```

A interface gráfica de *WebKB* permite a formulação de consultas através da seleção de termos da ontologia e especificação de predicados de seleção, trazendo como resultado uma estrutura de dados ou mesmo um documento completo. Graças ao embutimento de comandos de consulta em documentos, pode-se associar uma consulta a um *link*. A consulta é processada quando este *link* é ativado, gerando um documento HTML (documento virtual) com a estrutura do resultado da consulta. Documentos virtuais adaptam o conteúdo de documentos da *Web* à semântica da ontologia.

5 Análise das Propostas

A tabela 1 compara alguns aspectos do uso de ontologias pelas propostas.

As ontologias diferem principalmente em termos de tipos e elementos definidos. Ontologias de domínio estão presentes em todas as propostas, pois refletem diretamente os requisitos de um universo de discurso. *WebKB* apresenta uma ontologia de nível superior predefinida como base para novas ontologias e permite a definição de todos os tipos de ontologias. *InfoSleuth* é a mais sofisticada em termos de elementos ontológicos suportados, permitindo ontologias de tarefas compostas por procedimentos analíticos e eventos. Outras propostas definem regras associadas a conceitos e relações. *SHOE* e *Ontobroker*, por exemplo, descrevem regras para inferência de conhecimento. Em MOMIS, o projetista pode especificar restrições de integridade.

Quanto à relação ontologia-fonte de dados, a alternativa mais natural e flexível é a possibilidade do conhecimento de uma fonte de dados estar associado a diversas interpretações semânticas. Algumas propostas, porém, restringem esta relação para o caso um-para-muitos, considerando apenas documentos específicos de um certo domínio, como *MOMIS* e *Ontobroker*. *Observer* mantém o mapeamento do conhecimento de uma fonte de dados para apenas uma ontologia, porém, este conhecimento pode ser traduzido em termos de outras ontologias relacionadas.

A definição de ontologias segue, na maioria dos casos, princípios básicos como uma especificação de requisitos do domínio alvo e o reuso de ontologias pré-existentes. *Observer* não leva em conta integração de ontologias no processo de definição. *MOMIS* constrói ontologias a partir das definições de fontes de dados semi-estruturados e estruturados. O uso imposto de um vocabulário global faz com que a maioria das propostas prefira lidar com várias pequenas ontologias e com a possibilidade de reuso de conhecimento ao invés de uma única e ampla ontologia.

Quanto ao propósito de aplicação de ontologias, é comum a todas as propostas a utilização de ontologias como *esquemas conceituais* a partir do qual consultas com caráter semântico podem ser formuladas. *Observer*, *InfoSleuth* e *MOMIS* empregam ontologias no *processamento de consultas* para, respectivamente, expandir uma consulta a várias conceitualizações semanticamente relacionadas,

realizar consultas por conteúdo a documentos fracamente estruturados e restringir fontes de dados a serem pesquisadas através da aplicação de restrições de integridade. *SHOE*, *Ontobroker* e *WebKB* utilizam ontologias para a *anotação de conhecimento* em documentos e posterior armazenamento em um banco de dados. *SHOE* e *Ontobroker* realizam inferência de conhecimento armazenado através da investigação de relacionamentos de herança. *WebKB* permite a especificação de *índices* de elementos ontológicos em trechos de documentos. A principal fraqueza destas abordagens é que apenas documentos novos ou com permissão de escrita são passíveis de anotação, o que não se aplica a maioria dos documentos já existentes. *SHOE* tenta contornar este problema através da criação de páginas de sumário com anotações semânticas para páginas já existentes. *InfoSleuth* emprega ontologias como suporte para a *dinâmica de aplicações*, definindo eventos e procedimentos de análise associados a conceitos.

Tabela 1 – Comparação entre as propostas de aplicação de ontologias a dados semi-estruturados

	Observer	***SHOE***	***Ontobroker***	***Proposta de Embley***	***InfoSleuth***	***MOMIS***	***WebKB***
Tipos de ontologia suportados	Nível superior; Domínio	Domínio	Domínio	Domínio	Domínio; Tarefa; Aplicação	Domínio	Nível superior; Domínio; Tarefa; Aplicação
Elementos ontológicos mantidos	Conceitos; Relações; Regras de mapeamento para fontes de dados	Conceitos; Relações; Regras de inferência	Conceitos; Relações; Regras de inferência	Conceitos; Relações	Conceitos; Relações; Processos de análise; Eventos; Padrões para consulta a documentos	Conceitos; Relações; Regras de mapeamento para fontes de dados; Restrições de integridade	Conceitos; Relações
Funções para as quais a ontologia fornece suporte semântico	Processamento de consultas; Esquema conceitual	Anotação de conhecimento; Inferência de conhecimento; Esquema conceitual	Anotação de conhecimento; Inferência de conhecimento; Esquema conceitual	Extração de conhecimento; Esquema conceitual	Processamento de consultas; Processamento de atualizações e análises; Esquema conceitual	Processamento de consultas; Esquema conceitual	Anotação de conhecimento; Indexação de conhecimento; Esquema conceitual
Base para definição de ontologias	Requisitos do domínio	Requisitos do domínio; Ontologias existentes	Requisitos do domínio; Ontologias existentes	Requisitos do domínio; Ontologias existentes	Requisitos do domínio; Ontologias existentes	Fontes de dados	Requisitos do domínio; Ontologia única
Relação ontologia – fonte de dados	Um-para-muitos	Muitos-para-muitos	Um-para-muitos	Muitos-para-muitos	Muitos-para-muitos	Um-para-muitos	Um-para-muitos

6 Conclusão

Ontologia é um conceito que vem se tornando popular na Ciência da Computação pois provê um entendimento comum e compartilhado de um universo de discurso, possibilitando reuso de conhecimento. Na prática, ontologias são basicamente modelos estáticos de conhecimento, representados através de formalismos orientados a objetos ou baseados em lógica de primeira ordem. Poucas propostas levam em conta aspectos dinâmicos e a noção de ontologia como mantenedora de conhecimento universal, independente de domínio. O uso pioneiro de ontologias na área de inteligência artificial, motivou a sua aplicação na área de dados semi-estruturados como um esquema conceitual que dá suporte semântico a consultas, facilitando a legibilidade e a precisão de acesso. A

conceitualização definida em uma ontologia ou é anotada diretamente nos dados das fontes ou é extraída através de *wrappers* e/ou mediadores.

Este artigo mostra um panorama da aplicação de ontologias no tratamento de dados semi-estruturados para diversos fins: modelagem conceitual; processamento de consultas, atualizações e procedimentos analíticos; especificação de metadados; e anotação, inferência e indexação de conhecimento.

Uma tendência que surge é a utilização de ontologias como meio de busca eficiente de dados na *Web* através de sistemas gerenciadores de bancos de dados semi-estruturados. Esta é uma importante motivação para que soluções efetivas em termos de manipulação e metodologias de construção de ontologias sejam encontradas.

Bibliografia

[Abi97a] ABITEBOUL, S. et al. The Lorel Query Language for Semistructured Data. **International Journal on Digital Libraries**, v.1, n.1, p.68-88, Apr. 1997.

[Abi97b] ABITEBOUL, S. Querying Semistructured Data. In: International Conference on Database Theory, 1997, Delphi, Greece. **Proceedings**... [S.l.:s.n.]: 1997. p.1-18.

[Ash97] ASHISH, N.; KNOBLOCK, C. Wrapper Generation for Semi-structured Internet Sources. **SIGMOD Record**, v.26, n.4, p.8-15, Dec. 1997.

[Atz97a] ATZENI, P.; MECCA, G.; MERIALDO, P. To Weave the Web. In: International Conference on Very Large Databases (VLDB'97), 23., 1997, Athens. **Proceedings**... [S.l.]: Morgan Kaufmann, 1997, p.206-215.

[Atz97b] ATZENI, P.; MECCA, G.; MERIALDO, P. Semistructured and Structured Data in the Web: Going Back and Forth. **SIGMOD Record**, v.26, n.4, p.16-23, Dec. 1997.

[Ber99] BERGAMASCHI, S.; CAETANO, S.; VINCINI, M. Semantic Integration of Semistructured and Structured Data Sources. **SIGMOD Record**, v.28, n.1, p.54-59, Mar. 1999.

[Bun96] BUNEMAN, P. et al. A Query Language and Optimization Techniques for Unstructured Data. In: ACM SIGMOD International Conference on Management of Data (SIGMOD'96), Montreal, Canadá. **Proceedings**... New York: ACM Press, 1996, p.505-516.

[Bun97] BUNEMAN, P. Semistructured Data. In: SIGMOD International Symposium on Principles of Database Systems (PODS'97), 16., Tucson, Arizona, USA. **Proceedings**... [S.l.:s.n.]: 1997. p.117-121.

[Cat98] CATARCI, T. et al. Accessing the Web: Exploiting the Data Base Paradigm. In: Interdisciplinary Workshop on Building, Mantaining and Using Organizational Memories (OM'98), 1., 1998. Brighton, UK. **Proceedings**... [S.l.:s.n.]: 1998.

[Deu99] DEUTSCH, A. et al. A Query Language for XML. In: World Wide Web Conference (WWW8), 1999, Toronto, Canada. **Proceedings**... Toronto: University of Toronto, 1999.

[Dor00] DORNELES, C. F. **Extração de Dados de Semi-Estruturados Baseados em uma Ontologia**. Porto Alegre: PGCC da UFRGS, 2000. (Dissertação de Mestrado)

[Emb98] EMBLEY, D.W. et al. A Conceptual-Modelling Approach to Extracting Data from the Web. In: International Conference on Conceptual Modeling (ER'98), 17., 1998, Singapore. **Proceedings**... Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1507).

[Erd99] ERDMANN, M.; STUDER, R. Ontologies as Conceptual Models for XML Documents. In: Knowledge Acquisition for Knowledge-Based Systems Workshop, 12., 1999, Banff, Canada. **Proceedings**... [S.l.:s.n.]: 1999.

[Fen99] FENSEL, D. et al. On2broker: Semantic-Based Access to Information Sources at the WWW. In: World Conference on the WWW and Internet (WEBNET 99), 1999, Honolulu, Hawai, USA. **Proceedings**... [S.l.:s.n.]: 1999.

[Fer97] FERNANDEZ, M. et al. A Query Language for a Web-Site Management System. **SIGMOD Record**, v.26, n.3, p.4-11, Set. 1997.

[Flo98] FLORESCU, D. et al. Database Techniques for the World Wide Web: A Survey. **SIGMOD Record**, v.27, n.3, p.59-74, Mar. 1997.

[Gru93] GRUBER, T.R. A Translation Approach to Portable Ontologies. **Knowledge Acquisition**, v.5, n.2, p.199-200, 1993.

[Gua9?] GUARINO, N. Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In: **Information Extraction: A Multidisciplinary Approach to an Emerging Information Technology**. Berlin: Springer Verlag, p.139-170.

[Ham97] HAMMER, J. et al. Extracting Semistructured Information from the Web. In: Workshop on Management of Semistructured Data, 1997, Tucson, Arizona, USA. **Proceedings**... New York: ACM Press, 1997, p.18-25.

[Hef99] HEFLIN, J.; HENDLER, J.; LUKE, S. Applying Ontologies to the Web: A Case Study. In: International Work-Conference on Artificial and Natural Neural Networks (IWANN'99), 1999. **Proceedings**... Berlin: Springer-Verlag, v.II, 1999, p.715-724.

[Li99] LI, WEN-SYAN et al. WebDB Hypermidia Database System. **IEICE Transactions On Information Systems**, v.E00 A, n.1, Jan 1999.

[Mar99] MARTIN, P.; EKLUND, P. Embedding Knowledge in Web Documents. In: World Wide Web Conference (WWW8), 1999, Toronto, Canada. **Proceedings**... Toronto: University of Toronto, 1999.

[Nes97] NESTOROV, S.; ABITEBOUL, S.; MOTWANI, R. Inferring Structure in Semistructured Data. **SIGMOD Record**, v.26, n.4, p.39-43, Dec. 1997.

[Nie98] NIETO, E.M. **OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies**. Zaragoza: Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, 1998. 200p. (Tesis Doctoral).

[Pap95] PAPAKONSTANTINOU, Y.; GARCIA-MOLINA, H.; WIDOM, J. Object Exchange Across Heterogeneous Information Sources. In: International Conference on Data Engineering, Taipei, Taiwan. **Proceedings**... [S.l.:s.n.]: 1995, p.251-260.

[Per98] PERRY, B.; TAYLOR, M.; UNRUH, A. **Information Aggregation and Agent Interaction Patterns in InfoSleuth**. Austin: Microeletronics and Computer Technology Corporation, Texas, 1998. (Technical Report MCC-INSL-104-98).

[Unr98] UNRUH, A.; MARTIN, G.; PERRY, B. **Getting Only What You Want: Data Mining and Event Detection Using InfoSleuth Agents**. Austin: Microeletronics and Computer Technology Corporation., 1998. (*Technical Report* MCC-INSL-113-98).