Ferramenta para Extração de Ontologias a Partir de Bancos de Dados Relacionais

André A.Vieira [1]
Astério K.Tanaka [2]
Ana Maria de C. Moura [1]

Departamento de Sistemas e Computação, Instituto Militar de Engenharia [1]
Rio de Janeiro, RJ – Brasil – 22.290-270
<aaccioly, anamoura>@ime.eb.br

Escola de Informática Aplicada, UNIRIO - Universidade do Rio de Janeiro [2]
Rio de Janeiro, RJ – Brasil – 22.290-240
tanaka@uniriotec.br

Abstract

According to Tim Berners-Lee, the Semantic Web aims at developing languages for expressing information in a machine understandable form. In this context ontologies play a fundamental role as they enable to establish mechanisms to build up the *semantic* part of the Web. However, ontology development is known as an expensive and resource consuming task. Database schemas and ontologies are closely related. Legacy database systems are large sources of ontological information, promptly available to be converted to ontologies. Reverse engineering extraction tools may accomplish part of this task, reducing costs and saving often-expensive resources. In this paper we analyze and compare two of these tools and propose a tool with some new functionalities.

Keywords: Semantic Web, Ontologies, Reverse Engineering, Data Extraction, Relational Databases.

Resumo

De acordo com Tim Berners-Lee, a Web Semântica almeja desenvolver linguagens para expressar a informação de maneira a ser entendida por máquinas. Nesse contexto, ontologias terão um papel fundamental uma vez que irão prover o mecanismo para construir a parte *semântica* da Web. Entretanto, o desenvolvimento de ontologias é reconhecidamente complexo e consumidor de recursos. Bancos de dados de sistemas legados são uma fonte substancial de informações passíveis de serem transformados em ontologias. Ferramentas de engenharia reversa poderão contribuir na execução dessas tarefas, reduzindo custos e economizando recursos freqüentemente escassos. Este trabalho analisa e compara duas dessas ferramentas e apresenta uma proposta de ferramenta com algumas novas funcionalidades

Palavras Chaves: Web Semântica, Ontologias, Engenharia Reversa, Extração de Dados, Bancos de Dados Relacionais.

1 Introdução

1.1 A Web Semântica

Sob qualquer parâmetro que se queira avaliar, a *World-Wide-Web* (WWW) [31], ou simplesmente *Web*, é sem dúvida um dos mais impressionantes sucessos na história dos empreendimentos humanos, contando com bilhões de usuários, milhões de páginas e uma quantidade sem precedentes de informação.

Apesar da complexidade crescente da Web, isto não é refletido no estado atual das tecnologias utilizadas para sua manipulação. A maior parte das tarefas de acessar, extrair, interpretar e manter a informação disponível ainda é deixado a cargo dos usuários humanos.

Engenhos de busca são pobres quando se trata de fazer inferências complexas e correlacionar assuntos aparentemente disjuntos. A simples anotação de páginas HTML por intermédio das *tags* <META> ou mesmo o emprego de padrões de metadados não é suficiente para incluir a semântica desejada, possibilitando a execução de tarefas mais sofisticadas e mais úteis do que as existentes hoje.

Na visão de Tim Berners-Lee [4], esse tipo de construção, orientada para entendimento humano, leva a limitações e a um tratamento trivial por parte dos computadores, do conteúdo das páginas Web – limita-se a um

cabeçalho (*header*), *links* para outras páginas – mas, em geral, as máquinas não possuem uma forma confiável de processar o significado, ou seja, o conteúdo semântico das informações contidas em uma página.

Com base nessas premissas, surgiu a idéia da Web Semântica, na qual o conhecimento do significado de recursos da Web é armazenado por meio da utilização de (meta) dados processáveis por máquinas. Pretende-se que a Web Semântica não seja separada da Web, mas uma extensão da tecnologia corrente. Basicamente os mecanismos a serem desenvolvidos para o estabelecimento da Web Semântica compreendem duas grandes vertentes: a disponibilização de um conjunto de coleções estruturadas de informações e regras de inferência associadas a esses conjuntos; e a criação de agentes de *software* capazes de percorrer a Web realizando tarefas complexas com base nessas estruturas de conhecimento.

A fim de prover o primeiro mecanismo necessário à Web Semântica, algumas tecnologias vêm sendo reconhecidas como relevantes: a *eXtensible Markup Language* (XML)[1] e o *Resource Description Framework* (RDF)[2] [6]. XML permite representar dados em um formato mais próximo da realidade uma vez que dados semi-estruturados ocorrem com freqüência no mundo real. Entretanto, XML por si mesmo, não permite acrescentar significado a tais estruturas. Em que pese a capacidade de XML como sintaxe para transmissão de dados semi-estruturados, a expressão de significados deve ficar a cargo de RDF codificado em conjuntos de triplas. Cada tripla RDF é composta por um sujeito, um predicado e um objeto usando a sintaxe XML [1].

Na verdade, o componente responsável por representar e manter tais coleções estruturadas são as *ontologias*. Estas incluem estruturas que permitem manipular termos de uma forma muito eficiente e útil para consumo humano e mecanismos de validação para comunicação inter-agentes. A importância de seu uso é a possibilidade de representar hierarquias de classes de objetos (taxonomias), seus relacionamentos e realizar inferências acerca dessas propriedades. O desenvolvimento de ontologias irá prover o mecanismo de construção da parte *semântica* da Web Semântica [26]. O modelo em camadas proposto por Berners-Lee[3] tem sido aceito geralmente como representação para a arquitetura da Web Semântica [18].

O desenvolvimento de tais mecanismos depende, obrigatoriamente, de linguagens que expressem a informação de maneira a ser entendida por máquinas. O desafio é proporcionar uma linguagem que manipule igualmente bem dados e regras para deduções sobre esses dados e que permita que regras existentes em qualquer sistema de representação de conhecimento possam ser exportadas para Web [16].

1.2 Tratamento da Heterogeneidade Semântica

Uma das soluções para inclusão e reconhecimento da semântica de termos é o uso de ontologias. A semântica de conceitos de diversos domínios é capturada por suas ontologias, isto é, os termos existentes e as relações entre eles [23]. Entretanto, seu desenvolvimento não é uma atividade trivial. O desenvolvedor deve combinar criatividade com o treinamento apropriado na tecnologia de representação utilizada e um sólido domínio do campo de conhecimento no qual esteja trabalhando. O processo de validação de uma ontologia requer a classificação de um conjunto representativo de objetos, sendo normalmente longo e complexo [30].

Em aplicações fortemente acopladas (bancos de dados, por exemplo), o significado pretendido de um termo (ou seja, a semântica dos metadados) é freqüentemente implícito, tendo por base um acordo mútuo entre seus desenvolvedores. Em ambientes abertos, obter acordos mútuos pode ser muito difícil, senão mesmo impossível [22]. Dessa forma é crucial que o vocabulário usado para descrever o modelo do domínio seja especificado e mantido de maneira que outros sistemas possam processá-lo com um mínimo de intervenção humana [5].

A semântica de um termo pode variar de um contexto para outro, de um lugar para outro e mesmo de uma pessoa para outra. Diversos mecanismos foram propostos para lidar com o problema de integração de dados heterogêneos, tais como mediadores, *middlewares*, dicionários de metadados, etc [9,28]. Uma solução para o problema da heterogeneidade tem sido o emprego de *Document Type Definitions*[4] (DTD) com a sintaxe XML e de *XML Schema*[5]. A principal dificuldade é que, em todos os casos, o tratamento da heterogeneidade restringe-se ao nível sintático.

Amith Sheth [27] classifica a heterogeneidade em quatro categorias: *sistêmica, sintática, estrutural* e *semântica*. Heterogeneidade sistêmica diz respeito ao hardware e sistemas operacionais utilizados; heterogeneidade sintática

[1] eXtensible Markup Language - Language Syntax Specification, http://www.w3.org/TR/REC-xml
[2] Resource Description Framework - Model and Syntax Specification, http://www.w3.org/TR/REC-rdf-syntax/
[3] Disponível em http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html
[4] Document Type Definition – http://www.w3.org/TR/REC-xml#dt-doctype
[5] XMLS – XML Schemas – http://www.w3.org/XML/Schema

refere-se a utilização de diferentes linguagens; heterogeneidade estrutural está ligada ao uso de diferentes modelos de dados e finalmente, a heterogeneidade semântica compreende a semântica de consultas de usuários e das fontes de informação.

Neste ponto cabe questionarmos qual seria então a melhor forma de se tratar o problema da heterogeneidade semântica. Um raciocínio imediato seria a utilização de dicionários, léxicos ou enciclopédias que contivessem informações úteis. Entretanto, muitas informações encontram-se subentendidas em estruturas e modelos de negócio e não estão disponíveis imediatamente sem um tratamento prévio. Aparentemente a melhor solução é obtida com o desenvolvimento de ontologias. Evidentemente uma ontologia pode ser desenvolvida desde o princípio, entretanto, uma vasta quantidade de informação encontra-se disponível em fontes existentes (p.ex. na Internet, em bancos de dados legados, etc). O problema volta-se para a extração de informações ontológicas dessas fontes.

1.3 Ontologias e Bancos de Dados Relacionais

Muito embora bancos de dados e ontologias sejam aparentemente distintos, o esquema conceitual de um banco de dados contém muita informação semântica que pode ser extraída por ferramentas de software e utilizada na construção de uma ontologia. De fato, ontologias e esquemas de bancos de dados são intimamente correlacionados e, na verdade, a diferença entre uns e outros está no propósito de sua utilização [23].

Partindo desse princípio, bancos de dados de sistemas legados são uma fonte potencial de informações, passíveis de serem transformados em ontologias. Para tal é necessário o emprego de ferramentas que possibilitem extrair ontologias a partir de esquemas tradicionais de bancos de dados, reduzindo o custo e os recursos empregados em seu desenvolvimento.

Nas ontologias a linguagem é um mecanismo mais expressivo para representação do pensamento do que os sistemas de bancos de dados podem permitir. Freqüentemente as ontologias não oferecem garantias quanto à forma de representação do pensamento e muito trabalho é deixado a cargo do construtor. Apesar dessas limitações, a percepção de restrições, que normalmente são deixadas de lado ou simplesmente ignoradas no projeto de bancos de dados, pode ser encontrada nas ontologias. Por outro lado, no universo de bancos de dados, a maior preocupação concentra-se em garantir a durabilidade, segurança e o gerenciamento de transações. Os usuários de sistemas de bancos de dados esperam eficiência, completeza, durabilidade, integridade e suporte a concorrência de múltiplas transações [8].

Bancos de dados podem ser vistos como uma abstração de um domínio de uma área de conhecimento ou assunto, nos quais as entidades e os relacionamentos entre elas são relevantes para o problema da extração de ontologias. Sistemas de bancos de dados relacionais baseiam-se em restrições de integridade para garantir que o conteúdo do banco represente fielmente as ocorrências do mundo real. Dessa forma, a extração de ontologias a partir de fontes existentes deve se concentrar mais nos aspectos semânticos do que no nível de representação dos modelos.

O desenvolvimento de ferramentas que permitam automatizar o processo de geração de ontologias traz benefícios imediatos tais como: menor tempo e emprego de especialistas para o desenvolvimento de uma ontologia primitiva, acesso a termos de uso consagrado no domínio para o qual se deseja desenvolver a ontologia e possibilidade de manutenção da “inteligência” do negócio em uma forma inteligível por máquinas. Evidentemente que ontologias geradas por processos automatizados ou semi-automatizados devem passar por refinamentos posteriores, mas este trabalho sempre será bem menor do que o despendido para sua construção com base em uma simples “folha em branco”.

Algumas ferramentas com essa finalidade foram desenvolvidas no escopo de projetos maiores tais como a REVERSE, ferramenta para extração de ontologias do projeto KAON [13] e XRA, uma ferramenta de extração a partir de engenharia reversa do projeto DOME [23]. Em ambos, abordagens distintas foram utilizadas demonstrando, contudo, a viabilidade e oportunidade do conceito.

A idéia básica desta proposta é tentar mapear o domínio de representação de bancos de dados relacionais para o sistema de informação de ontologias utilizando mecanismos de engenharia reversa e então serializar os dados obtidos sob forma de uma linguagem de representação de ontologias.

A fim de apresentar uma estratégia para a extração de ontologias a partir de modelos relacionais, o restante deste artigo foi estruturado da seguinte forma: na seção 2 são apresentadas algumas considerações sobre ontologias. Na seção 3 são analisados trabalhos correlatos de ambientes para manipulação de ontologias e realizada uma comparação das ferramentas de extração existentes nos mesmos. Na seção 4 são descritas uma proposta de

arquitetura funcional para uma ferramenta de extração protótipo, uma comparação entre tecnologias relevantes para o trabalho com ontologias tais como, *XML Topic Maps (XTM)*[6] e a *Darpa Agent Markup Language + Ontology Inference Layer* (DAML+OIL)[7] e a arquitetura técnica da ferramenta proposta. Finalmente, a seção 5 apresenta algumas conclusões e propostas de trabalhos futuros.

2 Ontologias

As ontologias têm sido usadas principalmente por comunidades de pesquisadores há vários anos e, na verdade, o conceito remonta à época de Aristóteles que primeiro enunciou os princípios de conceito[8], espécie e termo [29], demonstrando preocupação quanto à forma da mente humana de observar e classificar os objetos do mundo real e os relacionamentos entre eles. Normalmente tem-se usado diferenciar Ontologia (com "O" maiúsculo) como ramo da filosofia, enquanto ontologia (com "o" minúsculo) como um artefato produzido por uma seção particular de código. Sob este enfoque, uma ontologia é:

Um conjunto de informações, representadas em uma forma que possa ser manipulada por componentes de software. [17]

Ironicamente, não existe um consenso em ciência da computação sobre a definição do que uma ontologia realmente é. Uma definição freqüentemente citada é a de Gruber [11]:

Uma ontologia é uma especificação explícita de uma conceitualização.

O problema com esta definição reside em estabelecer o que uma conceitualização é. Guarino [12] tem discutido extensivamente o fato do termo ser um tanto quanto vago, propondo sua própria definição:

Uma ontologia é uma teoria lógica para relacionar o significado pretendido de um vocabulário formal, isto é, seu comprometimento com uma conceitualização particular do mundo.

Uma ontologia, de acordo com tal interpretação, é uma teoria lógica cujo modelo restringe uma conceitualização particular, sem especificar exatamente qual. Em muitos casos os axiomas de uma ontologia expressam relacionamentos de submissão (é-um ou ISA) entre predicados unários. Evidentemente que uma axiomatização mais detalhada freqüentemente é necessária a fim de excluir interpretações indesejadas.

Handschuh e Maedche (op.cit.) definem formalmente uma ontologia como uma quíntupla $O := \{C, R, H^c, rel, A^0\}$ consistindo de:

- Dois conjuntos disjuntos C e R cujos elementos são denominados *conceitos* e *relações* respectivamente;
- Uma ***hierarquia de conceitos*** $H^c : H^c$ é uma relação direcionada $H^c \subseteq C \times C$ a qual é chamada de *hierarquia de conceitos* ou *taxonomia*. H^c (C_1, C_2) significa que C_1 é um subconjunto de C_2;
- Uma ***função*** $rel : R \rightarrow C \times C$ que relaciona conceitos não taxonomicamente. A função $dom : R \rightarrow C$ com $dom(R) := \prod_1(rel(R))$ fornece o domínio de R enquanto a função $val: R \rightarrow C$ com $val(R) := \prod_2(rel(R))$ fornece seu conjunto de valores. Notemos que para $rel(R) = (C_1, C_2)$ pode-se escrever $R(C_1, C_2)$;
- Um conjunto de ***axiomas*** ontológicos A^0, expressos em uma linguagem lógica apropriada como p.ex. lógica de primeira ordem (*first-order-logic* - FOL).

O desenvolvimento de ontologias deverá representar uma parcela significativa de esforço no desenvolvimento de qualquer aplicação no futuro. Dessa forma, o desenvolvimento de ambientes para construção e manipulação de ontologias é fundamental. Genericamente tais ambientes devem ser compostos de um repositório de ontologias que possa ser manipulado por desenvolvedores, usuários e programas de aplicação e que também permita a navegação, pesquisa e reuso de termos. Quando novos termos forem acrescidos à ontologia, o ambiente deve verificar a consistência do repositório. Durante a fase de operação, programas de aplicação acessam o repositório via API's que permitem consultar a estrutura de entidades e seus relacionamentos com outras entidades ou converter um termo em outro caso necessário. Tais ambientes devem ainda possuir um conjunto de ferramentas que permitam extrair informações ontológicas incorporadas nos sistemas legados existentes.

---

[6] XML Topic Maps – http://www.topicmaps.org/
[7] Darpa Agent Markup Language – http://www.daml.org
[8] O conceito reúne as características comuns ao conjunto de seres da mesma espécie, distinguindo-os dos seres constitutivos de outra(s) espécie(s). Enquanto representação mental, o conceito distingue-se do termo, isto é, a sua expressão verbal. Assim, o conceito de *ser humano* (animal racional) pode exprimir-se pelos termos *homem*, *hombre*, *homme*...

3 Ambientes de Manipulação e Ferramentas para Extração de Ontologias

Como ambientes para edição de ontologias podemos citar dentre outros o Protégé[9], OntoEdit[10], Webonto[11] e OilEd[12]. Protégé e OntoEdit são editores de ontologias que permitem a aquisição de ontologias e conhecimentos a partir de um único usuário. Ambos são capazes de gerar ontologias usando OIL [24]. Webonto foi especificamente projetado para o desenvolvimento conjunto de ontologias via Web. OilEd é um editor simples de ontologias que permite a construção de ontologias usando OIL.

É relevante mencionar que os editores de ontologias não podem ser considerados como ambientes de manipulação completos, uma vez que podem não abranger aspectos tais como extração de ontologias de diversas fontes, manipulação de ontologias (união, interseção, extração, validação, etc) ou outras operações.

A literatura pesquisada aponta duas propostas de ambientes para manipulação de ontologias. O *Karlsruhe Ontology and Semantic Web Infrastructure* (**KAON**) e o *Domain Ontology Management Environment* (**DOME**) já referenciados. Esses projetos podem ser considerados como ambientes mais completos para manipulação de ontologias em cujo escopo são propostas ferramentas para extração de ontologias a partir de esquemas de bancos de dados relacionais.

3.1 Domain Ontology Management Environment (DOME)

O projeto DOME reconhece que o problema central na construção de serviços dinâmicos é a falta de métodos e ferramentas que suportem a integração de modelos de processo e sistemas de informação de múltiplas organizações em ambientes de processos compartilhados pelas empresas.

O objetivo do DOME é proporcionar um ambiente no qual os desenvolvedores de ontologias possam utilizar ferramentas para desenvolver ontologias ou executar processos de engenharia reversa sobre fontes de dados e que usuários ou agentes de *software* possam utilizar as ontologias desenvolvidas para, dinamicamente, integrar múltiplos sistemas de informação.

Uma ontologia no DOME consiste de termos denotando conceitos, relacionamentos e restrições. Um conceito é caracterizado por atributos que são condições necessárias, mas não suficientes. Esta definição é intencionalmente similar à definição de esquemas de bancos de dados relacionais e classes de bancos de dados orientados a objetos, uma vez que o DOME é centrado na construção de ontologias para fontes de dados estruturados.

O DOME distingue duas classes de termos de ontologias: termos primitivos e termos definidos. A semântica de termos primitivos não é especificada. Quando o mapeamento entre duas ontologias é necessário, termos primitivos têm que ser mapeados manualmente. Nesse ambiente uma ontologia forma uma hierarquia de especialização com os termos de mais baixo nível tendo ligações estreitas com os modelos entidade-relacionamento e esquemas dos bancos de dados. A Figura 1 mostra o relacionamento entre ontologias, modelos ER e esquemas de bancos de dados.

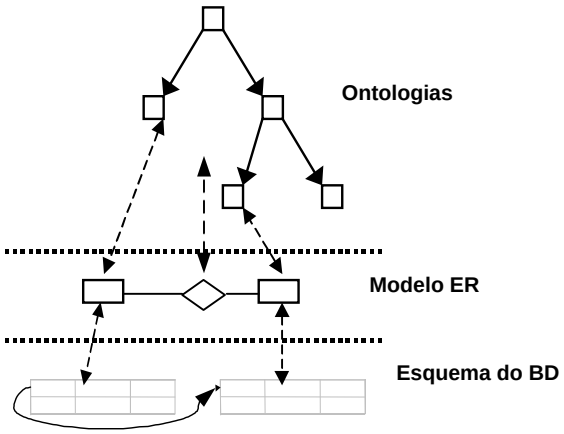

Figura 1 – Relacionamento entre Ontologias, modelo ER e esquema de BD (Extraído de [23])

Uma ontologia é vista como um conjunto de conceitos orientados a um domínio. Isso inclui conceitos abstratos e restrições no nível de domínio, necessários para inferências no nível de conhecimento. Esquemas e classes são conceitos no nível de dados, dependentes de implementação e são requisitos importantes para otimizar operações procedurais. Restrições nesse nível são restrições consideradas operacionais e muitas das restrições de domínio não são representadas. Termos de uma ontologia podem ser usados para definir esquemas de bancos de dados e uma ontologia pode ser usada para definir diferentes esquemas.

[9] Protégé, http://protege.stanford.edu
[10] OntoEdit, http://ontoserver.aifb.uni-karlsruhe.de/ontoedit/
[11] WebOnto, http://kmi.open.ac.uk/projects/Webonto/
[12] OilEd, http://img.cs.man.ac.uk/oil/

O DOME possui uma estrutura modular onde cada componente é responsável pela execução de uma tarefa específica comunicando-se via *interfaces* com os demais. Os componentes são divididos em:

- *Fontes de Dados*: São os sistemas legados dos quais serão extraídas as ontologias. O DOME foi originalmente projetado para a manipulação de fontes de dados estruturados. Uma ampliação proposta seria incluir as fontes de dados semi-estruturadas tais como páginas Web através de XML/ DTDs.

- *Ferramenta de Extração de Ontologias*: O módulo de extração do DOME é composto por uma ferramenta denominada XRA que utiliza engenharia reversa para extração de uma ontologia inicial a partir de fontes de dados e seus programas de aplicação.

A ferramenta XRA permite extrair entidades e relacionamentos a partir de esquemas de bancos de dados, podendo ainda extrair relacionamentos semânticos e funções a partir de programas de aplicação. É importante salientar que muitos relacionamentos são determinados pelos programas de aplicação de tal forma que o uso correto dos dados de origem freqüentemente não está documentado. A ontologia inicial precisa posteriormente ser refinada por projetistas de ontologias.

3.2 Karlsruhe Ontology and Semantic Web Infrastructure (KAON)

O conceito central do projeto KAON é baseado na arquitetura funcional proposta por Berners-Lee mencionada anteriormente. A arquitetura conceitual do KAON está baseada em três camadas distintas:

- *Camada Cliente*: Clientes podem ser (*i*) componentes de aplicações baseadas no KAON *Integrated Developing Environment* (IDE) ou (*ii*) aplicações que estendem o *framework* do portal Web KAON. Clientes se conectam com a segunda camada (camada de gerenciamento) via API. A API KAON é uma interface de programa de aplicação para aplicações baseadas em ontologias.

- *Camada de Gerenciamento*: A camada de gerenciamento é dividida em duas partes: A primeira é um mapeamento direto para uma API RDF (um modelo gráfico para processamento de RDF) e a segunda é mapeada para o servidor KAON. A diferença entre esses dois componentes é que o mapeamento para a API-RDF permite somente o uso transitório de modelos e suporta apenas acessos por um único usuário. O servidor KAON tem capacidade multi-usuário, permite o armazenamento persistente de modelos e utiliza funcionalidades de segurança e controle de transações definidas pelo Java Enterprise Edition (J2EE)[13] utilizando a implementação *open-source* JBOSS[14].

- *Camada de Armazenamento*: A camada de armazenamento é proporcionada por bancos de dados de tecnologia relacional ou sistemas de arquivos (*flat-files*), suportando diferentes sintaxes de serialização.

A arquitetura técnica do KAON inclui a *interface* com usuário, o servidor Web, um sistema de gerenciamento de documentos, sistemas operacionais, etc. Em seu estágio atual, a arquitetura foi desenvolvida em Java.

Além dos componentes do núcleo do KAON, o servidor KAON, a API RDF e a API KAON, o projeto incorpora uma família de ferramentas destinadas a realizar tarefas específicas. Tais ferramentas são desenvolvidas com base em um *framework*, permitindo obter vantagens significativas como: modularidade, extensibilidade, flexibilidade e baixo acoplamento.

Para esse estudo, analisamos a ferramenta KAON REVERSE que é o *plug-in* de conexão e mapeamento de bancos de dados relacionais para uma ontologia. Seu objetivo é extrair instâncias e relacionamentos de instâncias a partir de bancos de dados (BD). A ligação entre a ferramenta e o BD é estabelecida por meio de conexões *Java Database Connections* (JDBC) o que possibilita a utilização de praticamente qualquer banco de dados desde que exista o "*driver"* correspondente. A extração do esquema do BD é feita de forma semi-automática, cabendo ao usuário definir o mapeamento de tabelas, chaves primárias e chaves estrangeiras do BD para a ontologia usando a técnica *drag-and-drop* (arrastar e soltar). A ontologia produzida é apresentada sob forma de árvore de conceitos e gerada de forma subjacente usando-se RDF.

3.3 Comparação de Ferramentas de Extração

Nesta seção pretende-se realizar uma comparação entre as ferramentas para extração de ontologias dos ambientes de manipulação de ontologias apresentados. Independentemente do ambiente, linguagem utilizada, etc, as

[13] Java 2 Enterprise Edition (J2EE), http://java.sun.com/j2ee/
[14] JBOSS, http://www.jboss.org

ferramentas de extração seguem, de forma geral, passos semelhantes na extração de metadados, elaboração e serialização (sob forma de alguma linguagem de representação) de uma ontologia. Esses passos ou etapas podem ser enumerados como:

1. Captura de informação (metadados) através de engenharia reversa;
2. Análise da informação obtida para obtenção de quatro elementos construtivos básicos (classes, relacionamentos, funções e instâncias) de uma ontologia;
3. Construção da ontologia propriamente dita (de forma automática ou semi-automática); e
4. Validação, avaliação e documentação da ontologia (este passo é opcional).

Os dois ambientes previamente descritos, DOME e KAON, possuem ferramentas que implementam, ainda que de formas diferentes, a operação de extração. A ferramenta XRA no contexto do projeto DOME e a REVERSE, no ambiente do projeto KAON. O quadro abaixo (Figura 2) demonstra as características e funcionalidades de cada uma:

	DOME – XRA	KAON-REVERSE
Implementada	Sem informação	Sim
Linguagem	RWSL (Linguagem Proprietária)	Java
Uso de APIs	Não	Sim
Uso de engenharia reversa	Sim	Sim
Produto Final	Texto	Arquivo XML (RDF)
Fontes de extração de ontologias	Esquemas relacionais, Modelos O-O, Programas de aplicação	BDs relacionais (via JDBC)
Dependência de Uso de mediadores	Sim	Não
Independência de linguagem de manipulação	Não	Sim
GUI (Interface Gráfica)	Não	Sim

Figura 2 – Quadro Comparativo XRA x REVERSE

4 Desenvolvimento do Protótipo de Ferramenta de Extração de Ontologias

Uma limitação encontrada nos sistemas estudados diz respeito às linguagens de representação de ontologias usadas pelos mesmos. XRA gera um arquivo texto puro enquanto a ferramenta REVERSE gera um arquivo serializado em RDF. No entanto, tem havido algumas discussões[15], principalmente quanto à limitação de RDF em representar o conceito de reificação bem como quanto à capacidade dos mecanismos de inferência utilizados por DAML+OIL baseados em *description-logics* (DL) [3].

Dessa forma, a motivação para utilizar *Topic Maps* (XTM) como alternativa de linguagem de representação emergiu naturalmente durante nosso projeto. XTM possui um grau de expressividade consideravelmente superior a RDF [14,15]. Entretanto este último permanece como o padrão adotado pelo consórcio W3C. Assim sendo, esforços têm sido feitos para harmonizar e convergir esses padrões independentes, desenvolvidos por diferentes grupos, os quais podem ser usados para representar dados na Web de forma interoperável [19].

Um dos trabalhos mais recentes é a proposta de Decker e Lacher [7]. De forma geral, a idéia é tratar os modelos de dados em camadas separadas, de forma semelhante à utilizada em padrões de protocolos de redes.

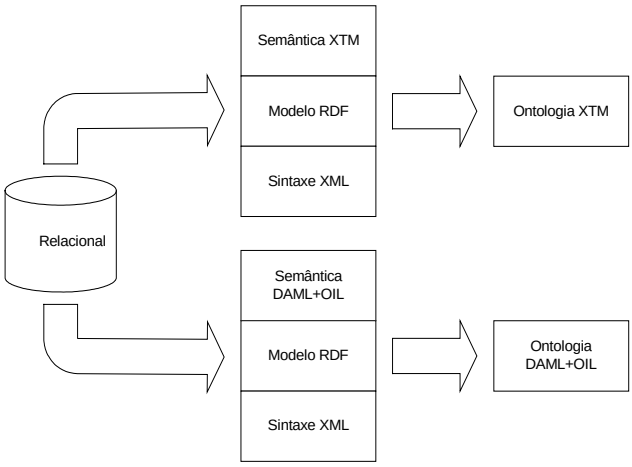

Figura 3 – Arquitetura Funcional

Verificamos então que RDF pode ser utilizado convenientemente como linguagem base para definição de objetos de ontologias, ou seja, como uma camada de modelo enquanto XTM pode ser utilizada como a linguagem de definição de ontologias, isto é, uma *(meta)ontologia*. Raciocínio análogo se aplica a DAML+OIL.

[15] Particularmente no âmbito do grupo de discussão de Web Semântica em http://groups.yahoo.com/group/semanticweb/

4.1 Arquitetura Funcional

Com base nessa abordagem, foi desenvolvida uma arquitetura funcional (Figura 3) para a ferramenta proposta. A finalidade desta arquitetura é apoiar conceitualmente o desenvolvimento do protótipo. O emprego deste modelo é importante pela manutenção do isolamento entre camadas. Neste caso, entretanto, o modelo idealizado difere do modelo RM/ISO-OSI[16] uma vez que as camadas inferiores não provêm serviços para as camadas superiores, servindo sim, como base de padrões.

A arquitetura prevê que o esquema do BD seja serializado de acordo com a sintaxe XML, obedecendo ao modelo estabelecido pela especificação RDF. Como mencionado, o modelo RDF não é suficiente para garantir a representação semântica de metadados do esquema, de tal forma que é necessário acrescentar camadas superiores – XTM e DAML+OIL, no caso – que irão proporcionar o tratamento semântico apropriado para estes metadados.

Neste ponto verificou-se a necessidade de comparar XTM e DAML+OIL para avaliar o real poder de expressividade de cada linguagem e garantir que ambas lidassem com conceitos equivalentes.

4.1.1 Comparação de Linguagens de Representação de Ontologias

A comparação realizada baseou-se nos trabalhos de Ratnakar [25] e Gómez-Pérez [10]. A definição dos critérios e características de comparação foge ao escopo deste trabalho, porém acreditamos serem suficientemente auto-explicativos para proporcionarem o entendimento necessário. A tabela da Figura 4 exibe a comparação realizada.

Dimensões	Detalhes	DAML+OIL	XTM
Contextos	**Contextos**	Sim	Sim <scope>
Classes	**Classes de Objetos e Propriedades**	Sim (daml:Class, daml:ObjectProperty, daml:DatatypeProperty)	Sim <topicMap>; <topic>; <instanceOf>
	Herança	Sim (Propriedades e Classes) Usa sintaxe RDF	Sim <instanceOf>; <topic>; <topicMap>
Propriedades de Restrições de Elementos	**Propriedade / Faixa de Elemento**	Sim (Global – rdfs:range Local – daml:Restriction ,onProperty, toClass)	Sim <topicRef> Permite apontar para Faixas definidas em um URI
	Propriedade / Domínio de Elemento	Sim (Global) – rdfs:domain	Sim <topicRef> Permite apontar para domínios definidos em um URI
	Propriedade / Cardinalidade de Elemento	Sim (Local – minCardinality, maxCardinality, cardinalidade Global – UniqueProperty, ou Restriction subClasse de “#Resource”)	Sim <topicRef>; <association> Permite apontar cardinalidades definidas em um URI
Tipos de Dados e Instâncias	**Tipos Básicos**	Sim Permite o uso de tipos de dados de XMLSchema	Sim <topic>; <association>; <roleEspec>; <subjectIndicatorRef> Permite apontar para Tipos de dados definidos em um URI
	Enumerados	Sim <daml:oneOf> Pode apontar para tipos enumerados de XMLSchema	Sim <topicRef> Permite apontar para Tipos de dados definidos em um URI
	Instâncias	Usa sintaxe RDF	Sim <topic>; <baseName>; <baseNameString>; <occurence>

[16] Reference Model – ISO - Open Systems Interconnection – http://www.iso.org

Conjuntos de Dados	**Listas**	Sim <daml:collection>	Sim <variant> Permite apontar para dados definidos em um URI
	Listas ordenadas	Sim <daml:list>	Sim <variantName> Permite apontar para dados definidos em um URI
Negação/ Disjunção/ Conjução	**Negação**	Sim <daml:ComplementOf>	Não diretamente
	Classes Disjuntivas	Sim <daml:disjointUnionOf> <daml:unionOf>	Não diretamente
	Classes Conjuntivas	Sim <daml:intersectionOf>	Sim <mergeMap>; <association>; <member>
Definições	**Condições Necessárias e Suficientes para afiliação**	Sim <daml:sameClassAs> <daml:UnambigousProperty>	Sim <baseName>; <baseNameString>
Tipos de Propriedades	**Inversa**	Sim <daml:inverseOf>	Sim <roleSpec>
	Transitiva	Sim <daml:TransitiveProperty>	Sim <roleSpec>
Reificação	**Reificação**	Não diretamente	Sim <subjectIdentity>

Figura 4 – Tabela Comparativa entre DAML+OIL e XTM

A comparação realizada permite concluir que DAML+OIL e XTM possuem a expressividade necessária como linguagens de representação de ontologias.

4.2 Protótipo da Ferramenta de Extração

A partir dessas conclusões, foi estabelecida uma arquitetura técnica para implementação do protótipo (Figura 5). O processo tem início por meio do estabelecimento de uma conexão JDBC com um banco de dados relacional. Os metadados do esquema do BD são extraídos preparando-se o mapeamento para a ontologia a ser gerada. O usuário deve indicar nessa fase as tabelas e os campos utilizados como chaves primárias e estrangeiras. Isso é necessário, pois cada tabela será mapeada diretamente como um conceito enquanto as chaves primárias/ estrangeiras serão utilizadas para mapeamento dos relacionamentos entre os mesmos.

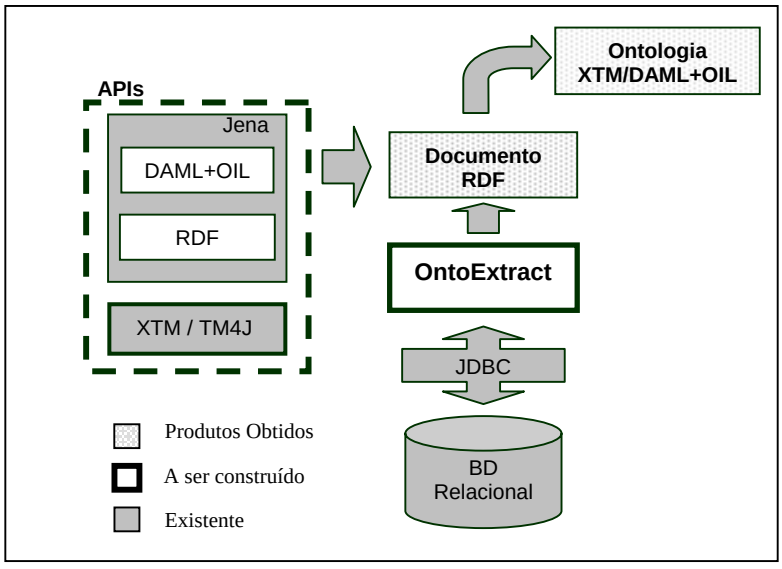

Figura 5 - Arquitetura Técnica

Durante o processo de extração de objetos do esquema do BD é preciso que cada elemento (tabela, coluna, etc) seja reconhecido e tratado individualmente. Para isso foi necessário o desenvolvimento de um (meta)metamodelo de BDs relacionais (Figura 6). Este (meta)metamodelo, adaptado de Ahmed [2] em conjunto com a especificação *Common Warehouse Metamodel* (CWM) [20], irá proporcionar o estabelecimento da semântica dos elementos constituintes do BD que serão posteriormente serializados sob a sintaxe DAML+OIL ou XTM. Dessa forma cada tabela, assim como suas colunas, é mapeada diretamente como um conceito da ontologia. O algoritmo para tratamento de relacionamentos existentes no BD leva em conta que estes podem ser de três tipos: um-para-um (1:1), um-para-muitos (1:N) e muitos-para-muitos (N:N). Em cada caso, a ferramenta tratará de forma diferente o relacionamento entre os conceitos. Nos casos de relacionamentos 1:N, a tabela (T_2) que contém a chave estrangeira do relacionamento (chave primária de T_1) será mapeada para o mesmo conceito definido pela tabela T_1. Em casos do tipo M:N., uma tabela (T_r) não determina um conceito em si mesmo, mas representa um relacionamento entre uma tabela T_1 e outra tabela T_2. Nesses casos T_r pode ser mapeada indistintamente para os conceitos de T_1 ou T_2.

Até este ponto, não existem ainda objetos RDF sob a sintaxe XML. O uso da *application-program-iterface* (API) Jena[17] é que permite a criação de uma camada de persistência dos elementos do BD, agora convertidos em objetos, que serão serializados como elementos XML de acordo com o modelo RDF. O conjunto de classes de Jena realiza a conversão com base em uma DTD para RDF e, ao final do processo, obtêm-se objetos RDF. A própria API Jena contém as classes necessárias para a serialização (usando SAX[18]) para um arquivo DAML+OIL. De forma semelhante a API TM4J[19] é empregada para a geração de uma ontologia no formato XTM.

É interessante observar que foi criada uma classe *Connection* a fim de atender ao requisito de persistência da conexão, inexistente no padrão CWM. Estas tarefas são executadas por um conjunto de classes do *Módulo de Extração*, que também será responsável pela serialização desses objetos como elementos RDF persistentes.

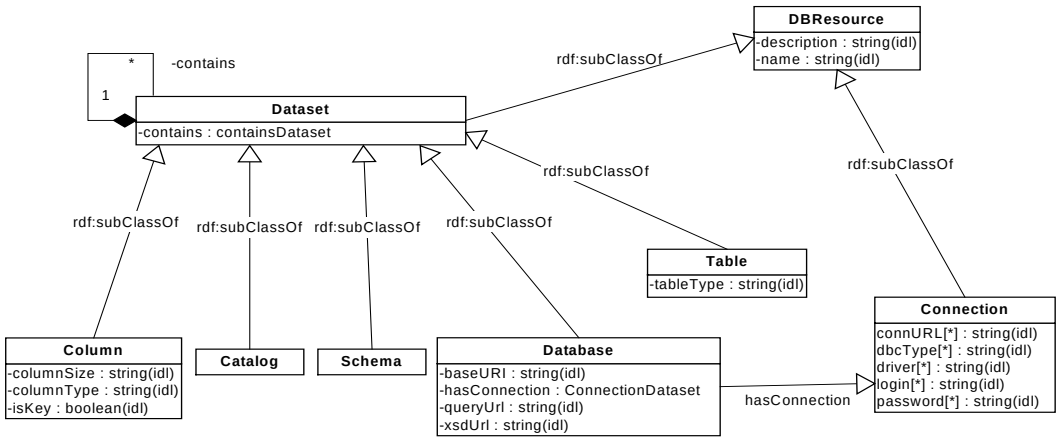

Figura 6 – Metamodelo de um BD Relacional (Adaptado de [2] e [20])

5 Conclusões

Podemos entender a Web, em seu estado atual, como um imenso repositório de dados. No entanto, os mecanismos existentes para a recuperação desses dados têm desempenho meramente sofrível se levarmos em conta dois requisitos básicos: a habilidade de interagir com os bancos de dados disponíveis e a habilidade de encontramos os bancos de dados que contenham as informações que são realmente necessárias a nosso propósito. O primeiro requisito é razoavelmente bem atendido pelas tecnologias disponíveis para Web. O segundo, contudo, permanece quase que integralmente sob responsabilidade dos usuários. Seja por intermédio de pesquisas em mecanismos de buscas, um *link* a partir de outra página, um anúncio de publicidade ou o que quer que seja para se chegar a um *Universal Resource Identifier* (URI) útil.

O reconhecimento da semântica de metadados em conjunto com "agentes" de software, permitirá que os dados sejam acessados segundo os termos do usuário ao invés de obriga-lo ao preenchimento de formulários, por exemplo. Para que tais objetivos sejam alcançados, um dos problemas é a necessidade de geração de ontologias passíveis de serem utilizadas pelos programas "agentes" que não apenas encontrem os bancos de dados necessários, mas possam reconhecer a semântica de seu conteúdo.

A fim de minimizar a tarefa de construção dessas ontologias, este trabalho buscou analisar o estado da arte das ferramentas já desenvolvidas e propor uma extensão dos mecanismos existentes, de forma a adaptar a pesquisa já realizada às novas tecnologias. Assim foi proposta uma ferramenta protótipo (*OntoExtract*) para extração de ontologias a partir de bancos de dados relacionais com capacidade de gerar ontologias em DAML+OIL e XTM.

Finalmente, como proposta de trabalhos futuros, visualizamos o desenvolvimento de outras ferramentas para a manipulação das ontologias geradas, permitindo a edição, consulta (com a correspondente inclusão de uma linguagem de consulta) e realização de operações (união, combinação, separação, etc) sobre as mesmas, de forma a construir um ambiente completo para manipulação de ontologias.

[17] Jena API, http://www.hpl.hp.com/semweb/jena-top.html
[18] Simple API for XML - http://www.saxproject.org/
[19] TM4J API, http://www.techquila.com/tm4j.html

Referências

[1] Abiteboul,S.Buneman,P.,Suciu D., Gerenciando Dados na Web, Campus, 2000
[2] Ahmed, K. et all., Professional XML Meta Data, Wrox Press, 2001, Birmingham, UK
[3] Bechhofer, S., Goble, C., Horrocks, I., DAML+OIL is not Enough, http://www.semanticweb.org/SWWS/program/full/paper40.pdf , 2001
[4] Berners-Lee, T., Hendler, J., Lassila, O., The Semantic Web, Scientific American, May, 2001, http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html
[5] Cox, M., Jones,D., Zhan C., An Environment for Managing Enterprise Domain Ontology, http://www.btexact.com/projects/ibsr/papers/Infobook.doc
[6] Decker, S., Frank, H., Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., Klein, M., Melnick, S., The semantic Web – on the respective Roles of XML and RDF, http://www.ontoknowledge.org/oil/downl/IEEE00.pdf
[7] Decker, S., Lacher, M.S., On the Integration of Topic Maps and RDF Data, www.semanticweb.org/SWWS/program/full/paper53.pdf
[8] Elmasri,R., Navathe, S.B., Fundamentals of Database Systems, 3rd.Edition, Addison-Wesley, 2000
[9] Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., et all., The TSIMMIS Approach to Mediation: Data Models and Languages, http://citeseer.nj.nec.com/12944.html
[10] Gómez-Pérez, A., Corcho, O., Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages, http://delicias.dia.fi.upm.es/articulos/ocorcho/corchoetal_ecai00_ws.pdf
[11] Gruber, T.R., Toward Principles for the Design of Ontologies Used for Knowledge Sharing (1993) , http://citeseer.nj.nec.com/cache/papers/cs/490/http:zSzzSzwww-ksl.stanford.eduzSzknowledge-sharingzSzpaperszSzonto-design.pdf/gruber93toward.pdf
[12] Guarino, N., Formal Ontology and Information Systems, Proceedings of Conference on Formal Ontology (FOIS98), http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/FOIS98.pdf
[13] Handschuh, S., Maedche, A., Stojanovic, L., Volz, R., KAON – The Karlsruhe Ontology and Semantic Web Infrastructure, http://kaon.semanticWeb.org/kaon/white-paper.pdf, 2001
[14] Holger Rath, H., Semantic Resource Exploitation with Topic Maps, Proceedings of the GLDV-Spring meeting 2001, Giessen University, March, 2001, pp.3-15, http://www.uni-giessen.de/fb09/ascl/gldv2001/
[15] Holger Rath, H., Topic Maps and the Ontological World, http://onto2001.aifb.uni-karlsruhe.de/tm-talk-hhr.pdf
[16] Mädche, A., Schurr, -P., Staab.S, Studer, S., Representation Language-Neutral Modeling of Ontologies, http://www.ontoprise.de/download/ontoedit_paper.pdf
[17] Martin, P., Ontology Related Concepts, Griffith University, Australia, http://meganesia.int.gu.edu.au/~phmartin/WebKB/kb/ontology.html
[18] Melnick.S., Decker, S., A Layered Approach to Information Modeling and Interoperability on the Web, Sep/2000, http://www-db.stanford.edu/~melnik/pub/sw00/
[19] Moore, G., RDF and Topic Maps. An Exercise in Convergence, http://www.topicmaps.com/topicmapsrdf.pdf
[20] Object Management Group - Common Warehouse Metamodel , CWM, http://www.omg.org/cwm/
[21] O'Brian, P, Yang,H, Zhan Cui., Extracting Ontologies from Legacy Systems for Understanding and Re-Engineering, Proceedings of the 33rd Annual International Computer Software and Applications Conference, www.computer.org/proceedings/compsac/0368/03680021abs.htm
[22] O'Brien.P, Jones,D., Zhan C., Issues in Ontology-based Information Integration, http://www.csd.abdn.ac.uk/ebiWeb/papers/cui.pdf, 2000
[23] O'Brien.P, Zhan C., Domain Ontology Management Environment, Proceedings of 33rd Hawaii International Conference on Systems Sciences, Jan/2000, IEEE Computer Sciences pg.9, Vol. I, http://dlib.computer.org/conference/hicss/0493/pdf/04938015.pdf
[24] Oil Specification Description, http://www.ontoknowledge.org/oil/
[25] Ratnakar, V., Gil, Y., A Comparison of (Semantic) Markup Languages, http://trellis.semanticweb.org/expect/web/semanticweb/paper.pdf
[26] Rivers-Moore, D. et all, Professional Java XML, Wrox Press, 2001, http://www.wrox.com/
[27] Sheth, A.P., Changing Focus on Interoperability in Information Systems: from system syntax, structure to semantics, Interoperating Geographic Information Systems, http://www.sfu.ca/gis/Web452/icons/Q_lec9.pdf
[28] Tanaka, A.K., Valduriez, P. & project members. "The Ecobase Project: Database and Web Technologies for Environmental Information Systems", SIGMOD Record, v. 30, n. 3, pp. 70-75, 2001.
[29] Vocabulário de Filosofia, Compilado por A.R. Gomes, http://www.terravista.pt/ancora/2254/lex19.htm
[30] Wiederhold, G., Jannick, J., Composing Diverse Ontologies, IFIP Working Group on Database, 8th Working Conference on Database Semantics, DS-8, Rotorua, New Zealand, 1999, http://www-db.stanford.edu/SKC/publications/ifip99.html
[31] World Wide Web Consortium, http://www.w3.org