Um Extrator de Topic Maps a partir de Recursos Heterogêneos de Informação

Giovani Rubert Librelotto[1]*,
José Carlos Ramalho[1], and Pedro Rangel Henriques[1]

University of Minho, Computer Science Department
4710-057, Braga, Portugal
{grl, jcr, prh}@di.uminho.pt

Abstract. O processo de desenvolvimento de ontologias baseadas em Topic Maps é complexo, consumidor de tempo e requer grande quantidade de recursos humanos e financeiros, devido ao fato de qualquer topic map (por mais simples que seja) possuir um conjunto significativo de tópicos e associações; além disso, para que a ontologia extraída seja realmente significativa, pode envolver um grande número de recursos de informação que poderão ser de tipos diferentes. Para resolver este problema, este artigo propõe um extrator de ontologias, chamado *Oveia*, que constrói topic maps a partir de recursos heterogêneos de informação, onde a ontologia a ser extraída é definida em uma linguagem de especificação denominada XS4TM (XML Specification for Topic Maps). O topic map extraído pode ser armazenado em uma base de dados relacional (permitindo que as ontologias possam crescer, sem restrições), ou em um documento no formato XML Topic Maps (XTM). Essa dupla capacidade de manipular várias fontes de informação e de poder armazenar o resultado em um suporte diferente é vantagem na comparação com a primeira versão do extrator, chamado *TM-Builder*.

1 Introdução

No funcionamento normal de uma organização, tipicamente são produzidos grandes volumes de dados. Normalmente, para satisfazer os seus requisitos de armazenamento, estas organizações utilizam bases de dados relacionais que são bastante eficientes para lidar com esta situação. Entre outras razões, os seus sistemas de indexação estão otimizados para suportar grandes volumes de dados.

Quando há necessidade de uma estruturação de mais alto nível dessa informação, o paradigma Topic Maps – ISO/IEC 13250 – mostra ser uma excelente opção, por ser suportado em um número reduzido de elementos simples (tópicos e associações). Com um mapa conceitual da informação encontrada nas bases de dados, habilita-se uma navegação através dos conceitos e das relações. Para isso, é necessária a definição de uma ontologia adequada ao universo em que esse sistema se insere, na qual estejam especificados os conceitos e as relações entre as entidades do próprio sistema. Um mapa de tópicos pode ser visto como uma coleção de índices interconectados. Tópicos e associações permitem definir, de forma estruturada, a semântica de um conjunto de recursos de informaçção. Esta rede hierárquica de tópicos é chamada ontologia.

* Bolsista CNPq - Brasil

Um extrator de ontologias baseado em XTM (*XML Topic Maps*) foi apresentado em [Librelotto et al., 2003]. Este ambiente processa um conjunto de documentos XML – pertencentes à mesma família, ou seja, documentos que respeitem o mesmo DTD ou XML Schema – com um extrator (*TM-Builder*), criado a partir de uma especificação da ontologia em XSTM (*XML Specification for Topic Maps*). O TM-Builder produz um documento XTM, o qual contém a ontologia extraída de acordo com a especificação XSTM. Essa plataforma é, portanto, totalmente baseada em XML.

Porém, quando os recursos de informação não são documentos XML (como acontece em vários projetos concretos), é necessário proceder, previamente, a uma conversão das fontes em causa para XML. Esta tarefa não é a melhor escolha, pois a sincronização dos dados é complexa; quando o recurso original é modificado é necessário gerar novamente o documento XML, atualizando o seu conteúdo.

Como solução para essa dificuldade prática, apresenta-se aqui um novo construtor de Topic Maps, chamado *Oveia*. O *Oveia* evita a transformação de recursos de informação não-XML para documentos XML, pois ele extrai as informações diretamente dos recursos.

A ontologia a ser extraída é especificada em XS4TM (XML Specification for Topic Maps). Esta linguagem tem o mesmo objetivo que XSTM, porém está um nível acima: XS4TM cobre todos os elementos da especificação XTM, além de ser projetada para a extração de ontologias em recursos de informação heterogêneos.

O *Oveia* cria uma base de dados com a estrutura definida de acordo com o paradigma Topic Maps [Biezunsky et al., 1999] contendo todos os tópicos e as associações entre eles, chamada *BD Ontologia*. Na verdade, o *Oveia* pode armazenar os topic maps extraídos tanto na BD Ontologia, como no formato XTM[1].

O artigo inicia apresentando o paradigma Topic Maps na seção 2; Topic Maps é um formalismo para representar conhecimento sobre um recurso de informação, organizando por tópicos. A descrição do sistema que propõe-se, o extrator de Topic Maps a partir de recursos heterogêneos de informação – *Oveia* – é feita na seção 3. A definição da linguagem XS4TM será encontrada na seção 3.5. Por fim, uma síntese do artigo e os trabalhos futuros são apresentados na conclusão.

2 Topic Maps

Topic Maps [Biezunsky et al., 1999] é um formalismo para representar conhecimento acerca da estrutura de um conjunto de recursos de informação e para o organizar em *tópicos*. Esses tópicos têm ocorrências e associações que representam e definem relacionamentos entre os tópicos. A informação sobre cada tópico pode ser inferida ao examinar as associações e ocorrências ligadas ao tópico. Uma coleção desses tópicos e associações é chamada *topic map*. Também pode ser visto como um paradigma que permite organizar, manter e navegar pela informação, permitindo transformá-la em conhecimento. Falar sobre Topic Maps, é falar sobre estrutura de conhecimento.

[1] O que basicamente é um documento XML onde diferentes elementos são usados para representar: tópicos, ocorrências de tópicos e relacionamentos (ou associações) entre os tópicos [Pepper, 2000].

Um mapa de tópicos expressa a opinião de alguém sobre o que os tópicos são, e quais as partes do conjunto de informação que são relevantes para cada tópico.

Permitindo a criação de um mapa virtual da informação, os recursos de informação mantém-se em sua forma original e não são modificados. Então, o mesmo recurso de informação pode ser usado de diferentes maneiras, por diferentes mapas de tópicos. Como é possível e fácil modificar um mapa, a reutilização da informação é conquistada.

Tópicos são o ponto principal de Topic Maps [Park et al., 2003]. Em um sentido mais genérico, um tópico pode ser qualquer coisa: uma pessoa, uma entidade, um conceito. Eles constituem a base para a criação de Topic Maps (TM). Cada tópico tem um tipo de tópico (*topic type*), ou talvez múltiplos tipos. Cada tipo de tópico pode ser visto como uma típica relação classe-instância.

Ao analisar Topic Maps, identificam-se duas camadas distintas: os tópicos e as ocorrências. Os tópicos podem ser divididos em duas partes: os que representam conceitos abstratos e os que representam conceitos concretos. A ontologia é definida pelos conceitos abstratos, ou seja, os que serão instanciados por outros tópicos; por exemplo: tipo de tópico, tipo de associação e pelo tipo de papel de atuação em ocorrências.

Os tópicos restantes formam a base de conhecimento associada à ontologia, os quais compõem um conjunto de objetos de informação que permite organizar e indicar os reais recursos de informação (um objeto pode ter múltiplas ocorrências nos recursos de informação). A figura 1 dá uma representação esquematizada desta visão.

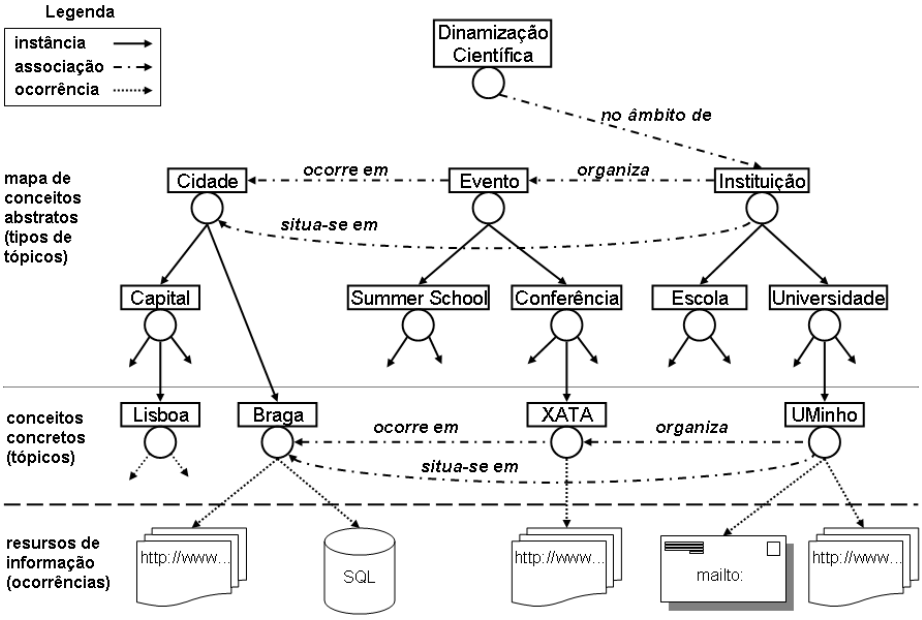

Fig. 1. O Mapa do Conceito *Dinamização Científica*.

O conceito de associação (*association*) permite descrever relações entre tópicos. Uma associação é (formalmente) um elemento de vínculo que define uma relação entre dois ou mais tópicos. Um ilimitado número de tópicos podem ser relacionados por uma associação.

3 Oveia – Um Extrator de Topic Maps a partir de Recursos de Informação

O *Oveia* é um extrator de ontologias em sistemas heterogêneos de informação baseado em Topic Maps. O *Oveia* foi desenvolvido com o objetivo de suprir as deficiências encontradas pelas atuais ferramentas de extração de ontologias. O *Oveia* é uma seqüencia do projeto que resultou no *TM-Builder* [Librelotto et al., 2003], o qual fornece um modelo de extração que consiste de uma especificação da ontologia a ser extraída.

Um fato que representa a evolução do *Oveia* em relação à sua versão inicial é a capacidade de extrair ontologias a partir de fontes de dados diversas. No *TM-Builder*, quando a fonte de informação é diferente de um documento XML, há uma necessidade uma conversão desta fonte para um arquivo XML. Portanto, considerando que a maior parte dos recursos de informação em empresas e instituições estão armazenadas em base de dados, para realizar uma extração de uma ontologia a partir delas, inicialmente seria necessário a geração de documentos XML com o conteúdo da base de dados.

Assim como o *TM-Builder*, o *Oveia* faz uso de uma linguagem de especificação de extração (XS4TM), a qual permite extrair Topic Maps de forma genérica e adaptativa. A especificação de extração de ontologias em XS4TM tornou-se mais flexível e completa, contemplando todos os elementos do padrão Topic Maps [Biezunsky et al., 1999]. Isso garante maior flexibilidade de especificação para propósitos diversos de extração.

Na fase de especificação dos recursos de informação, é possível fazer transformações e filtros nessas fontes de dados, pois é utilizada a linguagem de consulta de cada recurso para sua especificação.

A linguagem de especificação de extração foi inspirada no modelo XTM. Isso significa que a especificação da ontologia a ser extraída (em XS4TM) é feita em um esquema XML similar ao esquema de XTM. Essa característica permite maior facilidade de compreensão da especificação proposta, pois o modelo XTM é um padrão que vem sendo adotado amplamente pela comunidade acadêmica. Assim, o projetista da ontologia apenas deve conhecer a sintaxe de XTM e a estrutura das fontes de dados, para estar habilitado a especificar extrações de ontologias em XS4TM.

A arquitetura do *Oveia* pode ser expressa conforme demonstra a figura 2: inicialmente, é feita em uma especificação XSDS (*XML Specification for DataSources/DataSets*), a qual define quais dados que devem ser recuperados pelo *Extrator de Datasets*; a informação extraída é armazenada em um formato intermediário, chamado *Datasets*. O próximo passo é a especificação da ontologia em XS4TM; essa fase determina o que é relevante para a extração dos tópicos e associações, assim como clarifica os limites que devem ser impostos ao topic map. O processador XS4TM recebe os *datasets* gerados e a especificação da ontologia na linguagem XS4TM (*XML Specification for Topic Maps*) e gera o topic map final. Por fim, o *Oveia* armazena o topic map gerado na BD Ontologia ou no formato XTM.

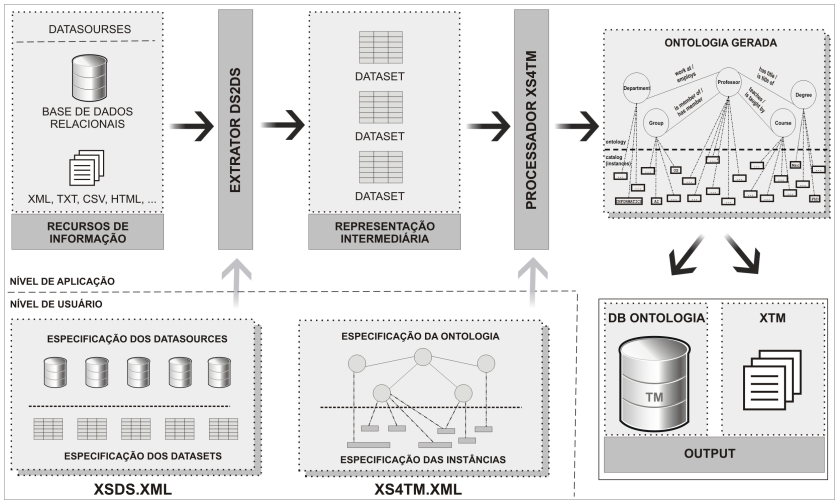

Fig. 2. Arquitetura do Oveia

As próximas sub-seções apresentam cada um dos componentes do *Oveia*.

3.1 Recursos de Informação: Os *Datasources*

Este componente é composto pelos recursos de dados: bases de dados, documentos XML, páginas HTML, etc. Ao final do processo de extração de ontologias, os recursos manterão-se inalterados, ou seja, o *Oveia* não modifica as fontes; somente copia as partes relevantes de informação para a construção do topic map. Esses recursos de dados são mapeados para uma representação intermediária, chamada *datasets*. Esse mapeamento é descrito pela linguagem XSDS.

3.2 Representação Intermediária da Informação: Os *Datasets*

Os *datasets* são a representação intermediária que contém os dados extraídos das fontes de informação. Cada *dataset* tem uma relação com uma entidade dos *datasources*, e seu conteúdo é representado na forma de uma tabela, onde cada linha é um registro segundo a estrutura definida em XSDS. Os *datasets* garantem que o *Oveia* tenha uma visão uniforme sobre a estrutura de dados que representam as fontes de dados participantes.

Cada *dataset* tem uma identidade única, a qual será usada pelo Oveia para o referenciar. A idéia fundamental é que todos os objetos tem rótulos que descrevem o seu conhecimento. Por exemplo, o seguinte objeto representa um registro da categoria de tipo de professor: <1,PhD>, onde ”1” é o identificador da categoria, enquanto que ”PhD” é um rótulo legível por humanos. Os *datasets* são simples, enquanto provém um poder de expressividade e flexibilidade necessário para integrar sistemas de informação de diferentes fontes.

3.3 XSDS: Especificação das Fontes de Dados da Extração

XSDS (*XML Specification for DataSources/DataSets*) é a linguagem definida com o intuito de especificar quais fontes de informação fornecerão dados para a criação de topic maps, de acordo com uma ontologia posteriormente especificada. Essa especificação fornece todos os elementos necessários para especificar as fontes de dados passíveis de extração de informação.

De um modo formal, a gramática de XSDS é dividida em duas partes: a definição dos *datasources* e a definição dos *datasets*. A primeira parte refere-se aos recursos físicos, ou seja, define-se quais fontes reais de informação serão usadas para a obtenção de dados; a segunda parte refere-se a quais campos de dados das fontes de informação devem ser extraídos, usando a linguagem de query de cada fonte em questão. Assim, pode-se dizer que a partir de um mesmo *datasource*, podem ser construídos vários *datasets*.

A definição da arquitetura do extrator foi idealizada para suportar extensão a diversos recursos de informação como fontes de dados. Para isso, essa arquitetura baseia-se no conceito de *drivers* de extração.

Especificação dos Datasources e Datasets em XSDS: Os *datasources* definem a localização física do recurso de informação. A declaração de cada uma das fontes de informação é feita no elemento <*datasource*>. Este elemento possui um atributo, chamado *extractorDriver* que indica qual *driver* de extração será utilizado: de acordo com o tipo de fonte de informação. Por exemplo: no caso de uma base de dados, além da localização da mesma, são passados parâmetros como o usuário e a senha a ser utilizada nesta base de dados, juntamente com o *driver* de extração que fará este processo; no caso da fonte de informação ser um documento XML, é necessário apenas o nome do arquivo com o seu caminho na árvore de diretórios do sistema operacional.

Para cada conjunto de dados dos recursos de informação (*datasets*) que se queira mapear a partir dos recursos de informação, é necessária a declaração do elemento <*dataset*>. Neste elemento, é necessário indicar qual fonte de dados provém os dados para a construção do *dataset* em questão.

O conteúdo do elemento <*dataset*> é uma expressão na linguagem de consulta referente ao tipo da fonte de informação. Caso esta fonte seja uma base de dados, o conteúdo deste elemento será uma expressão SQL para recuperar os dados referentes ao *dataset* em questão. Se a fonte de informação é um documento XML, o conteúdo deste elemento será uma expressão XPath, indicando o caminho para a informação deste *dataset*.

3.4 O Extrator DS2DS

O *Extrator DS2DS* (*DataSource to DataSet*) é um processador que extrai dados de recursos de informação e faz a criação dos *datasets*, de acordo com a especificação XSDS. Este componente processa uma especificação XSDS, a qual especifica a fonte dos dados a serem extraídos (*datasources*) e o destino das informações extraídas, as quais definem a representação intermediária (*datasets*).

Esta representação intermediária é composta por um conjunto de tabelas que contém a informação extraída dos datasources. Estas tabelas contém somente os dados selecionados nos elementos *datasets* da especificação XSDS em questão.

O *Extrator DS2DS* possui diversos *drivers* de extração que, como dito anteriormente, são os módulos responsáveis pela extração de informação das fonte de dados; portanto, há um *driver* desenvolvido para cada tipo de recurso de informação.

Atualmente, o protótipo do Oveia possui dois drivers implementados: para conectar com base de dados relacionais (br.uneb.dcet.tmbuilder.drivers.DataBase) e para recuperar informação de documentos XML (br.uneb.dcet.tmbuilder.drivers.XMLFile). A implementação de novos *drivers* para outros recursos de informação é um processo relativamente fácil e pode ser realizado conforme a necessidade.

3.5 XS4TM: Uma linguagem XML para especificar a extração de Topic Maps

A linguagem XSTM, proposta em [Librelotto et al., 2003], foi inicialmente definida como sendo um dialeto XML para especificar o topic map que se pretende construir ao analisar documentos anotados pertencentes a um mesmo esquema XML. Por essa definição, o XSTM está diretamente ligado a extrações a partir de documentos XML. Por outro lado, a necessidade de abranger novas fontes de dados fez com que se propusesse uma nova arquitetura para extração de ontologias. Dessa forma tornou-se necessário repensar e redesenhar o XSTM; o qual passa a ser denominado por XS4TM.

Cada especificação XS4TM é uma instância XML. Portanto, na prática a linguagem XSTM é definida por um DTD (e/ou um XML-Schema), de modo a permitir o uso de todos os ambientes de processamento XML.

A linguagem XS4TM tem por objetivo tornar a especificação da extração de Topic Maps mais completa e flexível. XS4TM é caracterizado por transformar o atual padrão XTM em um subconjunto da sua especificação.

O XS4TM possui dois elementos principais: *ontologies* e *instances*. Cada um destes elementos têm a estrutura de acordo com a especificação XTM. A única diferença está nos subelementos de *instances*, pois os elementos *topic* e *association* possuem um novo atributo. Esse atributo é denominado *dataset* e é utilizado para identificar que o determinado elemento (tópico ou associação) será construído a partir de um *dataset* em específico. Este atributo faz uma referência ao nome do *dataset*, declarado no documento de especificação XSDS; a figura 3 demonstra essa referência entre o *dataset DS_aluno* declarado em XSDS e o seu uso na especificação XS4TM.

Para o preenchimento das informações referentes a cada tópico, é necessário buscar tal informação no *dataset* que a contém. Assim, identifica-se essas propriedades com a expressão:

@ + ”dataset” + ”.” + ”atributo”

Mais detalhadamente, isto significa:

- O @ apenas indica que esta declaração é referente a uma propriedade de um *dataset*;
- Após o @, encontra-se o identificador do *dataset* (especificado em XSDS) ao qual deseja-se recuperar a informação. No exemplo da figura 3, o *dataset* selecionado é o *DS_aluno*.

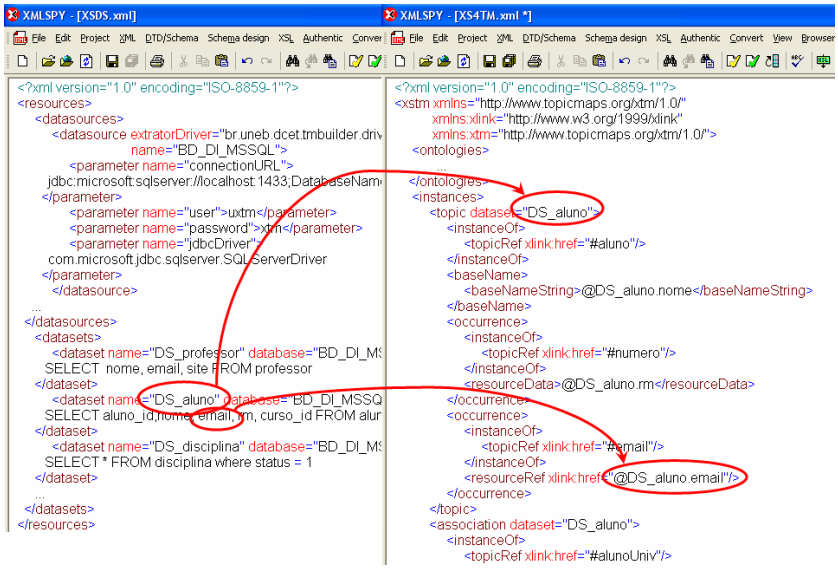

Fig. 3. Relações entre XSDS e XS4TM

– O *atributo* é uma referência ao campo do *dataset* que contém a informação desejada. Na figura 3, o atributo recuperado para a construção da ocorrência do tipo *email* é o campo *email* extraído pelo *dataset DS_aluno*.

Desta forma, cria-se uma forma de habilitar o uso das informações contidas nos *datasets*.

3.6 Processador de XS4TM

Este componente utiliza a especificação XS4TM para selecionar quais campos dos *datasets*, extraídos dos recursos de informação, são necessários para a formação do topic map. Este processador é um interpretador que tira vantagem da organização das informações em um formato uniforme.

O seu processo de execução pode ser resumido em três passos: (1) ler a especificação XS4TM e extrair os dados especificados que encontram-se nos *datasets*; (2) criar o topic map baseado na própria especificação XS4TM; (3) armazenar o topic map gerado na BD Ontologia ou em um documento no formato XTM.

3.7 Base de Dados de Ontologias

Um dos diferenciais desta ferramenta é o armazenamento dos Topic Maps extraídos em uma base de dados relacional. De acordo com [Williams et al., 2000], referente aos métodos de mapeamento de documentos XML para o modelo relacional, adotou-se na *BD Ontologia* o modelo de mapeamento por estrutura.

Conforme o mapeamento por estrutura, foi criada uma tabela para cada elemento de XTM 1.0 DTD. Esse processo consiste em identificar as características e os tipos de associações entre os elementos do DTD e representa-los no modelo relacional.

Pode ser entendido facilmente tomando um trecho desse mapeamento, como apresenta o segmento do XTM 1.0 DTD abaixo e a figura 4.

```
<!ELEMENT topic (instanceOf*, subjectIdentity?, (baseName | occurrence)*)>
<!ATTLIST topic
    id ID #REQUIRED
>
<!ELEMENT baseName (scope?, baseNameString, variant*)>
<!ATTLIST baseName
    id ID #IMPLIED
>
```

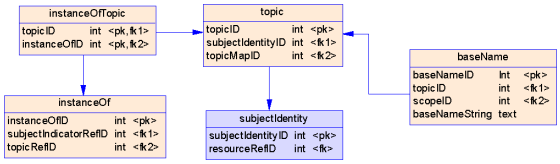

Fig. 4. Trecho do Modelo ER do BD Ontologia

A facilidade de compreensão desse modelo é garantida principalmente pelo fato de que este modelo segue o padrão XTM, o qual é bastante conhecido pela comunidade acadêmica. Essa foi umas das vantagens trazidas por essa opção de modelagem, preservando o padrão Topic Maps. Assim, a partir dessa base de dados, é possível navegar no Topic Maps utilizando consultas SQL.

4 Trabalhos Relacionados

O KAON[2] é um projeto *open-source* que fornece uma infra-estrutura para gestão de ontologias, voltado para aplicações de negócios. A ferramenta KAON REVERSE é um *plug-in* do *framework* KAON. O KAON REVERSE permite o mapeamento de bases de dados relacionais para uma ontologia, com o objetivo de extrair instâncias e relacionamentos entre instâncias, a partir da base de dados. Dentre as ferramentas conhecidas, esta é a que mais se aproxima do *Oveia*.

A tabela 1 mostra as características e funcionalidades do *Oveia* e do KAON REVERSE[3].

Analisando a tabela 1, é difícil dizer qual a melhor ferramenta, visto que é clara a complexidade entre ambas; apetece até dizer que o melhor dos mundos resultaria de sua fusão.

[2] Mais informações em: http://kaon.semanticweb.org/

[3] Os dados presentes nesta tabela foram utilizados a partir da análise feita em [Vieira, 2002].

	KAON REVERSE	OVEIA
Linguagem	Java	Java
Uso de APIs	Sim	Sim
Uso de Engenharia Reversa	Sim	Não
Especificação	Árvore(GUI)	Documento XML
Fontes de Extração de Ontologias	BDs relacionais via JDBC	BDs relacionais via JDBC, XML, extensível a outras fontes
Padrão de Representação de Ontologias	RDF	Topic Maps
GUI (Interface Gráfica)	Sim	Não
Resultado Gerado	Documento RDF	Base de Dados de Ontologias

Table 1. Comparativo entre KAON REVERSE e Oveia.

Partindo deste ponto de vista, destaca-se as vantagens de cada uma. Por um lado, a KAON REVERSE apresenta vantagens em relação ao uso de uma interface gráfica para a especificação de ontologias e o uso de engenharia reversa das fontes de dados para auxiliar o mapeamento. Por outro lado, o *Oveia* destaca-se por ser mais flexível em relação às fontes de dados passíveis de extração (mesmo que todos os drivers de extração não estejam implementados) e em relação ao processo de especificação. Além disso, o *Oveia* diferencia-se por gerar uma base de dados de ontologias capaz de manter os dados extraídos.

Ao fazer uso de uma linguagem de especificação (XS4TM) semelhante à linguagem padrão que é usada para escrever os XML Topic Maps, o *Oveia* facilita o processo de extração ao projetista da ontologia, o que é uma clara vantagem, visto ser este o foco do sistema. Assim, torna-se muito simples e conciso criar uma nova visão conceitual diferente sobre as mesmas fontes.

5 Conclusão

O objetivo deste artigo foi a apresentação de uma arquitetura para a construção automática de Topic Maps, a partir da extração de informação de fontes de dados diversas. Esse sistema, designado por *Oveia*, resultou de uma proposta inicial denominada *TM-Builder*, o qual aborda um extrator de ontologias totalmente baseado em XML.

A extração de informação de fontes heterogêneas de informação é especificada pela linguagem XS4TM, a qual define quais os conceitos e relações encontradas nestas fontes de informação que serão mapeadas para tópicos e associações, respectivamente, no Topic Maps gerado pelo *Oveia*.

XS4TM é a linguagem para especificar a extração de Topic Maps a partir de recursos de informação. XS4TM classifica os tópicos, dando-lhes uma semântica mais concreta, através da associação de um tipo de tópico, um tipo de associação ou um tipo de papel de atuação em ocorrências. Então, do ponto de vista de descrição da ontologia, obtém-se ganho por se dispor de uma semântica mais precisa.

O *Oveia* é uma evolução do *TM-Builder*, o qual estava limitado ao processamento de documentos XML. A fim de evitar esse problema, foi construída uma nova arquitetura

que provê uma abstração dos tipos de fontes de dados baseada em *drivers* de extração. O resultado obtido foi uma camada de independência da fonte de dados, que torna possível a extração em fontes de dados diversas, de forma transparente e em uma única especificação.

A principal vantagem desta proposta é a possibilidade de adaptação do processo de especificação e extração de ontologias a um maior número de casos reais, sem acarretar em mudanças de formato do recurso de informação.

Outra vantagem é o que eventuais modificações nas fontes de informação (obviamente mudanças ao nível de seu conteúdo, e não estruturais – incluindo-se novos dados ou excluindo-os), não torna necessária uma modificação da especificação XS4TM; basta voltar a executar o extrator com a fonte de informação com os novos dados inseridos (ou retirados).

Um dos projetos a ser desenvolvido em breve será um módulo que permita a conversão de um documento XTM para uma BD Ontologia, assim como possibilite a extração de um Topic Map da BD Ontologia para um documento XTM. Com este módulo, o utilizador pode rapidamente ter um conjunto de topic maps armazenados no formato desejado, seja em documentos XML (no padrão XTM) ou em base de dados relacionais (em uma BD Ontologia).

A fusão entre Topic Maps, conhecida como *merge*, não foi considerada nesta versão de XS4TM. Isso implicaria um tratamento mais complexo no processo de extração, o que não seria viável no presente momento.

O *Oveia* não possui um ambiente amigável para o utilizador criar suas especificações. Aparentemente o processo de criação de um documento XS4TM ou XSDS mostra-se trabalhoso. Isso exige que o utilizador conheça o padrão dos documentos de especificação, que é o padrão XTM. Por outro lado, é sabido que esta tarefa pode ser auxiliada por ferramentas de edição de documentos XML, como exemplo o XMLSpy[4], mas essa função de criar uma interface de especificação agradável e fácil de usar será um outro tópico de investigação futura.

References

[Biezunsky et al., 1999] Biezunsky, M., Bryan, M., and Newcomb, S. (December, 1999). ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34. http://www.y12.doe.gov/sgml/sc34/document/0129.pdf.

[Librelotto et al., 2003] Librelotto, G., Ramalho, J. C., and Henriques, P. R. (2003). TMBuilder: Um Construtor de Ontologias baseado em Topic Maps. In *XXIX Conferencia Latinoamericana de Informática – CLEI, La Paz, Bolívia*.

[Park et al., 2003] Park, J., Hunting, S., and Engelbart, D. C. (2003). *XML Topic Maps: Creating and Using Topic Maps for the Web*. Prentice Hall.

[Pepper, 2000] Pepper, S. (2000). The TAO of Topic Maps - finding the way in the age of infoglut. Ontopia. http://www.ontopia.net/topicmaps/materials/tao.html.

[Vieira, 2002] Vieira, A. A. (2002). Extração de Ontologias a partir de Esquemas Relacionais. Instituto Militar de Engenharia. Mestrado em Sistemas e Computação.

[4] Ferramenta para construção de documentos XML, desenvolvido pela ALTOVA. Mais informações em: http://www.altova.com

[Williams et al., 2000] Williams, K., Brundage, M., Dengler, P., Gabriel, J., Hoskinson, A., Kay, M., Maxwell, T., Ochoa, M., PaPa, J., and Vanmane, M. (2000). *Professional XML Databases*. Wrox Press.