

# Obtendo Interoperabilidade Semântica em Sistemas Heterogêneos de Informação com Metamorphosis

Giovani Rubert Librelotto, José Carlos Ramalho, and  
Pedro Rangel Henriques

Universidade do Minho, Departamento de Informática  
4710-057, Braga, Portugal  
gr1@di.uminho.pt

**Abstract.** Hoje em dia, os dados manipulados pelas instituições estão dispersos nos mais variados recursos de informação, tais como bases de dados e documentos de diferentes tipos. Com isso, a integração da informação contida nessas fontes é uma tarefa árdua.

Diferentes abordagens podem ser seguidas para conseguir a integração, porém o problema principal será conquistar a interoperabilidade semântica entre as fontes de informação, garantindo a manipulação do conjunto sem forçar a conversão das partes em um formato único. Ontologias surgem como uma solução para este problema e Topic Maps, como uma forma de representação de ontologias, afirmam-se como um caminho para alcançar esta solução.

Neste contexto, este artigo apresenta o Metamorphosis – um ambiente orientado a Topic Maps composto por módulos capazes de extrair dados de recursos heterogêneos de informação, construir uma ontologia a partir dos mesmos (baseado em uma especificação), validá-la e gerar um sistema de navegação conceptual – que permite atingir a interoperabilidade semântica desejada.

## 1 Introdução

Diariamente, uma grande quantidade de dados é produzida em instituições. Para satisfazer os requisitos de armazenamento, estas organizações geralmente utilizam bases de dados relacionais, as quais são eficientes para guardar e manipular dados estruturados. Quando se trata de dados semi-estruturados, o armazenamento é realizado em documentos textuais ou anotados.

Há um problema quando estas organizações requerem uma visão integrada desses sistemas heterogêneos de informação, pois é necessário interrogar/extrair cada fonte de dados. Contudo, o acesso para cada sistema de informação é diferente. Nesta situação, há a necessidade de uma abordagem que permita extrair a informação contida nestes recursos e a disponibilize como uma visão única sobre todo o sistema.

Topic Maps são uma boa solução para organizar conceitos e os relacionar entre si, porque eles seguem uma notação normalizada – ISO/IEC 13250 [2] –

para a representação de conhecimento intercambiável. Topic Maps são compostos por tópicos e associações dando origem a uma rede semântica estruturada que concentra a informação relacionada com um certo domínio. Esta rede hierárquica de tópicos pode representar uma ontologia. Esta é a razão pela qual se usa, há alguns anos, esta tecnologia para classificação e integração de documentos na área de Arquivos Digitais, com sucesso.

Contudo, o processo de desenvolvimento de ontologias baseado em Topic Maps é complexo, consome muito tempo e requer uma quantidade significativa de recursos humanos e financeiros, porque elas podem envolver um enorme conjunto de tópicos (conceitos) e associações (relacionamentos entre os conceitos).

Para resolver este problema, propomos o sistema *Metamorphosis* que torna possível a extracção, validação, armazenamento e navegação de Topic Maps. Ele é composto por três módulos principais: (1) *Oveia*, que extrai dados de sistemas heterogêneos de informação, de acordo com a especificação de uma ontologia e armazena-a em um topic map; (2) *XTche*, que valida o topic map gerado, de acordo com uma especificação de restrições; e (3) *Ulisses*, que navega sobre o topic map, dando uma visão conceptual integrada sobre o domínio e os recursos.

Deste modo, o *Metamorphosis* permite obter a interoperabilidade semântica em sistemas heterogêneos de informação porque os dados relevantes são extraídos e armazenados em um topic map, de acordo com uma especificação da ontologia desejada. O ambiente valida o topic map gerado de acordo com um conjunto de regras definido numa linguagem para descrição de restrições. Este topic map fornece fragmentos de informação (os dados propriamente ditos) conectados por relações específicas para outros conceitos, em diferentes níveis de abstracção. A navegação sobre o topic map é realizada sobre uma rede semântica e proporciona uma visão homogênea sobre os recursos – o que justifica a decisão de chamar a este processo de *interoperabilidade semântica*.

Habilitando a criação de um mapa virtual de informação, os recursos de informação são mantidos em seus estados originais; ou seja, não são modificados. Então, um mesmo recurso pode ser usado pelo *Metamorphosis* de diferentes formas, para a criação de diferentes topic maps. Com isso, a reutilização de fontes de informação é conquistada.

A secção 2 apresenta uma visão geral sobre Topic Maps. A arquitectura proposta e suas características principais encontram-se descritas na secção 3; as seguintes secções descrevem os módulos principais de *Metamorphosis*: *Oveia* (secção 4), *XTche* (secção 5) e *Ulisses* (secção 6). Por fim, a conclusão fornece uma síntese deste artigo.

## 2 Topic Maps

Topic Maps – ISO/IEC 13250 [2] – é uma norma para organização e representação de conhecimento sobre um domínio, a qual permite a especificação de temas e de relacionamentos entre temas. Steve Pepper [13] define *tema* como um termo usado para designar alguma coisa do mundo real, ou seja, algo que

um tópicos possa representar. Portanto, um *tópico* pode ser qualquer coisa: uma pessoa ou objecto, uma entidade ou uma organização, um conceito, etc.

Um topic map é composto por tópicos e associações que dão origem a uma rede semântica estruturada que agrupa informações relacionadas sobre um certo domínio.

Um topic map pode ser visto como um conjunto organizado de tópicos (representação formal de temas), contendo:

- uma estrutura hierárquica de tópicos (definido pelas relações *é-um* ou *contém* – relações classe-instância);
- vários nomes para cada tópico (ou tema de um índice);
- ponteiros (ocorrências) entre tópicos e documentos externos (recursos de informação);
- relacionamentos semânticos (associações) entre tópicos.

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. A informação sobre cada tópico pode ser inferida ao examinar as associações e ocorrências ligadas a esse tópico.

Topic Maps podem ser expressados em XML, através da especificação XML Topic Maps (XTM) 1.0 [14], a qual é um dialecto para escrita de topic maps e foi desenvolvido para aplicar a norma Topic Maps [2] para a web.

### 3 Metamorphosis

A principal ideia do *Metamorphosis* é integrar a especificação de redes de conceitos ou ontologias, com sua navegação e armazenamento, assim como sua extracção automática e sua validação, a partir de recursos heterogêneos de informação.

A motivação para o desenvolvimento do *Metamorphosis* também veio de duas situações que surgiram no contexto de alguns projectos de desenvolvimento de software:

- Muitas vezes, para se testar algumas funcionalidades de um sistema que se está a desenvolver é necessário criar uma interface Web para realizar esses testes. Normalmente, estas interfaces não têm grandes requisitos quanto à aparência, o que se pretende é que sejam desenvolvidas o mais rapidamente possível.
- Quer-se expor na Web um sistema de informação. Este sistema é composto por bases de dados, documentos XML, documentos PDF, etc. Quer-se que toda a informação esteja acessível via Web e para isso constroem-se os primeiros índices. Estes índices acabam por ser enormes excedendo a capacidade dos browsers actuais. Podia pensar-se em fraccioná-los alfabeticamente, mas há situações em que isso não é possível nem recomendável. Mas,

é sempre possível arranjar um método para fraccionar a informação conceptualmente. É aqui que se começa a discutir a organização dos recursos de informação e é aqui que se introduz o *Metamorphosis*.

O *Metamorphosis* toma como entrada:

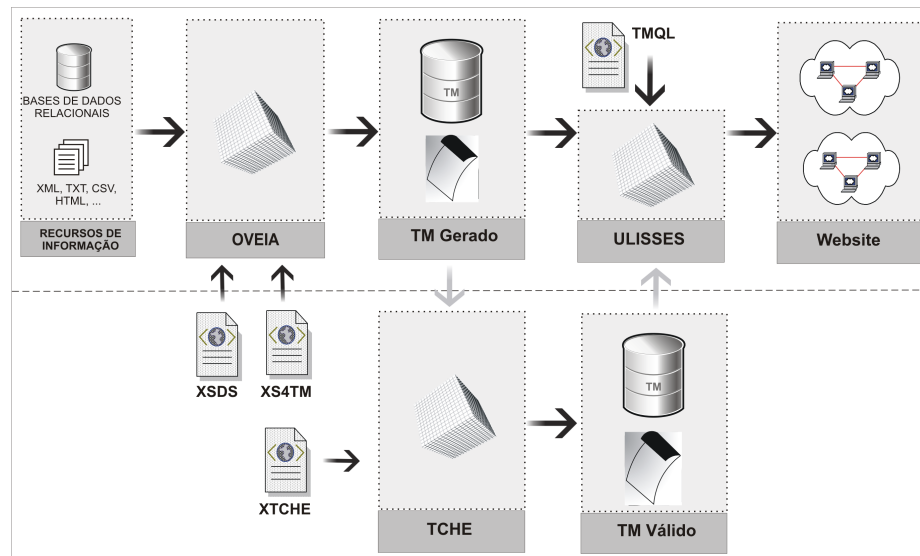
**Recursos de Informação:** compostos por um ou mais recursos de dados: documentos XML, páginas HTML, bases de dados, etc. O *Metamorphosis* não modifica nenhuma fonte, apenas usa parte de seus dados para construir a rede semântica, através de uma ontologia definida para tais fontes de dados;

**Especificações XML:** a descrição das fontes de dados (escrita em XSDS – *XML Specification for DataSources*); a descrição da ontologia a ser construída (escrita em XS4TM – *XML Specification for Topic Maps*); e a descrição das regras a serem obedecidas por um topic map (estrita em XTche – *Topic Maps Schema and Constraint Language*).

e gera, como saída:

**Um Website Conceptual:** O website gerado permite a navegação através do sistema de informação dirigido por conceitos organizados em uma rede semântica.

A figura 1 vem reforçar esta ideia e dá uma visão pictórica da arquitectura do *Metamorphosis*.



**Fig. 1.** Metamorphosis

Esta arquitectura pode ser descrita da seguinte forma:

- (1) **Camada de recursos de informação:** Este componente é composto pelos vários recursos de informação: documentos XML, páginas Web, bases de dados, ... O *Metamorphosis* não interfere com nenhum deles, apenas utiliza parte da informação de cada um para construir a ontologia ou rede semântica.
- (2) **Especificações XSDS e XS4TM:** São documentos XML que fornecem informações precisas sobre onde aceder para extrair as partes da informação necessárias para a construção da ontologia; este componente é descrito em detalhes em [9].
- (3) **Oveia:** Este componente utiliza a especificação XS4TM para ir aos recursos de informação buscar a informação de que necessita para construir a ontologia. O *Oveia* [9] é um extractor de topic maps em sistemas heterogéneos de informação.
- (4) **TM gerado:** Este é o topic map de acordo com a sintaxe *XML Topic Maps*. Além da possibilidade de armazenamento dos topic maps em formato XTM, os topic maps gerados pelo *Oveia* também podem ser armazenados em uma representação relacional. Para isso, a norma Topic Maps foi mapeada para um modelo relacional; este modelo é chamado de *BD Ontologia*.
- (5) **Especificação XTche:** É uma linguagem de especificação de restrições para topic maps, baseada nos requisitos de TMCL (*Topic Map Constraint Language*) [10] definidos pela ISO/IEC. Permite especificar regras para a validação semântica de topic maps.
- (6) **Processador XTche:** Este componente é responsável pela validação de topic maps de acordo com um conjunto de restrições especificado em XTche. Se os topic maps a serem validados estiverem de acordo com a especificação XTche, nada acontece; caso apresentem alguma irregularidade, mensagens de erros serão mostradas, indicando os pontos em que o topic map não está correto.
- (7) **Ulisses:** Toma como entrada um *topic map* e produz uma visualização na web, de acordo com algumas regras. O *Ulisses* tanto fornece a navegação conceptual a partir da sintaxe XTM, como a partir do modelo relacional apresentado no *Oveia*.
- (8) **Website:** É o website gerado através do qual é possível navegar semanticamente pelos vários recursos de informação que compõem o sistema de informação original.

Nas próximas secções, os três módulos principais desta arquitectura são discutidos: *Oveia* (secção 4), *XTche* (secção 5) e *Ulisses* (secção 6).

O modo de funcionamento do *Metamorphosis* permite uma interdependência entre tais módulos; assim, quando o utilizador for efectuar o processamento de um conjunto de recursos de informação de acordo com especificações XSDS, XS4TM e XTche, cada etapa é realizada em separado:

1. no *Oveia*, define-se os ficheiros com as especificações das fontes de informação e da ontologia a ser aplicada. Após o seu processamento, um sumário indica o tempo total e a quantidade de tópicos e associações extraídos que serão encontrados no topic map obtido;

2. no *XTche*, indica-se o nome dos ficheiros que possuem o topic map a ser validado e o conjunto de restrições a ser aplicado. O resultado do processamento é a confirmação (ou não) da validação do topic map;
3. no *Ulisses*, basta informar o topic map que será a fonte para a construção do website semântico.

O funcionamento independente dos módulos permite que uma tarefa específica seja refeita, caso necessário.

## 4 Oveia – Um Extractor de Topic Maps

O *Oveia* é um extractor de ontologias baseadas em Topic Maps a partir de sistemas heterogéneos de informação.

Sua arquitectura é composta por duas especificações e os referentes processadores, conforme apresentada na Figura 2: a primeira, escrita na linguagem XSDS (*XML Specification for DataSources/DataSets*), especifica os dados a serem extraídos das fontes de informação; enquanto que a segunda, escrita na linguagem XS4TM (*XML Specification for Topic Maps*), é responsável por declarar como são construídos os topic maps. Com base nestas especificações, o extractor busca as informações nas fontes de informação e produz um topic map. Este topic map gerado pode ser armazenado em formato XTM (XML Topic Maps) ou em uma base de dados relacional.

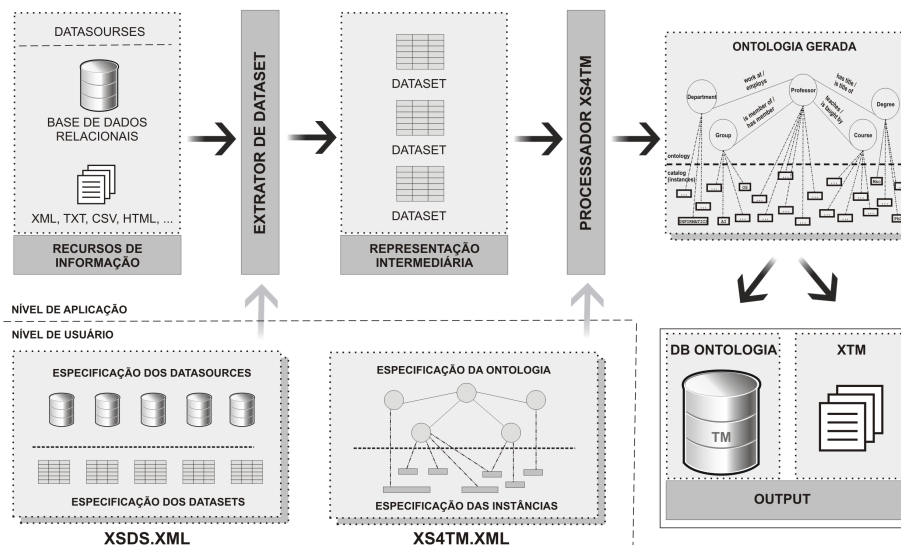


Fig. 2. Oveia

#### 4.1 XSDS – XML Specification for DataSources

A linguagem XSDS (*XML Specification for DataSources/DataSets*) – apresentada em detalhes em [9] – foi definida com o intuito de especificar que fontes de informação fornecerão dados para a criação de topic maps, de acordo com uma ontologia posteriormente especificada. Essa especificação fornece todos os mecanismos necessários para especificar as fontes de dados passíveis de extracção de informação. Assim, os dados extraídos são armazenados em *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 registo 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.

Para efeitos de demonstração da sintaxe de XSDS, apresenta-se o código abaixo:

```
<resources>
  <datasources>
    <datasource extratorDriver="br.uneb.dcet.tmbuilder.drivers.XMLFile" name="XML_DI">
      <parameter name="pathDocument">D:\\UMINHO\\DI.xml</parameter>
    </datasource>
  </datasources>
  <datasets>
    <dataset name="Pessoas" database="XML_DI">
      <columns>
        <column identify="true">CodPes</column>
        <column>nome</column>
      </columns>
      <statement>//Pessoas</statement>
    </dataset>
  </datasets>
</resources>
```

Na prática, 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*.

#### 4.2 O Extrator DS2DS

O *Extractor 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 seleccionados nos elementos *datasets* da especificação XSDS em questão.

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

### 4.3 XS4TM – XML Specification for Topic Maps

A linguagem XS4TM – também apresentada em detalhes em [9] – tem por objectivo especificar a construção de Topic Maps a partir dos dados contidos nos *datasets*. XS4TM torna a definição da extracção de Topic Maps mais completa e flexível, pois ela caracteriza-se por transformar a norma XTM em um sub-conjunto da sua especificação; em outras palavras, XS4TM estende a linguagem XTM, como pode ser percebido neste extracto de código XS4TM abaixo:

```
<xstm xmlns="http://www.topicmaps.org/xtm/1.0/" xmlns:xlink="http://www.w3.org/1999/xlink">
  <ontologies>
    <topic id="Pessoas">
      <baseName>
        <baseNameString>Pessoas</baseNameString>
      </baseName>
    </topic>
    ...
  </ontologies>
  <instances>
    <topic dataset="Pessoas">
      <instanceOf>
        <topicRef xlink:href="#Pessoas"/>
      </instanceOf>
      <baseName>
        <baseNameString>@Nome</baseNameString>
      </baseName>
    </topic>
    ...
  </instances>
</xstm>
```

A especificação XS4TM é subdividida em duas partes distintas, as quais seguem o esquema de XTM 1.0 DTD:

**Ontologia:** nesta primeira parte são declarados os elementos responsáveis pela definição da ontologia, como os tipos de tópicos, os tipos de associações, ou qualquer outra definição de acordo com o modelo XTM que possa ser utilizada para expressar a estrutura de conhecimento a ser extraída – ou seja, a hierarquia de tópicos, relações super-tipo/sub-tipo e associações entre os tópicos;

**Instâncias:** nesta segunda parte representa as instâncias de tópicos e associações. Nesse momento, os *Datasets* serão utilizados para expressar quais recursos de informação fornecerão dados para a construção de cada tópico e associação.

A linguagem XS4TM pretende tornar a especificação de extracção de Topic Maps mais flexível. Para isso, ela utiliza a sintaxe de XTM para seus dois elementos principais: <Ontologies> e <Instances>; isto significa que tanto a declaração da ontologia, quanto das instâncias, seguem o formato XTM: uma sequência dos elementos <topic> e <association>. Portanto, se o utilizador conhece a sintaxe de XTM, ele terá muita facilidade em definir uma especificação XS4TM.



#### 4.4 Processador de XS4TM

Este componente utiliza a especificação XS4TM para seleccionar 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.

#### 4.5 Base de Dados de Ontologias

Uma das diferenças desta ferramenta é o armazenamento dos Topic Maps extraídos em uma base de dados relacional. Baseado nos métodos de mapeamento de documentos XML para o modelo relacional apresentados em [16], adoptou-se na *BD Ontologia* o modelo de mapeamento por estrutura.

Conforme o mapeamento por estrutura se caracteriza, 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 representá-los no modelo relacional.

A facilidade de compreensão desse modelo é garantida principalmente pelo facto de que este segue a norma XTM, a qual é bastante conhecida pela comunidade académica. 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.

### 5 XTche – Um Validador Semântico para Topic Maps

Esta secção apresenta uma linguagem para a definição de restrições sobre Topic Maps, chamada XTche [8], a qual permite garantir que um conjunto de topic maps de uma mesma família são semanticamente válidos de acordo com uma especificação. Antes de descrever a linguagem e seu processador (um gerador de validador), será dado uma motivação de seu desenvolvimento, além de uma discussão sobre restrições.

#### 5.1 Uma Linguagem para definição de Esquemas e Restrições em Topic Maps

No desenvolvimento de topic maps reais é altamente conveniente utilizar um sistema para validá-los; isto é, verificar a correcção de uma instância real de acordo com uma especificação formal de uma respectiva família de topic maps (de acordo com a intenção de seu criador).

Adoptando-se o formato XTM, a validação sintáctica de um topic map é garantida por um parser XML porque a estrutura de XTM é definida por um

DTD [15]. Contudo, sabe-se que validação estrutural não significa uma completa correcção – a semântica também deve ser garantida.

Usando um XML Schema para a validação de Topic Maps, ao invés de DTD, aprimora-se o processo de validação porque alguns requisitos de semântica (domínio, número de ocorrências, etc.) podem ser adicionados à especificação estrutural; parsers XML tratam desta tarefa. Contudo, outros requisitos de semântica permanecem não-especificados. Assim, torna-se necessária uma linguagem de especificação que permite a definição de esquema e restrições numa família de Topic Maps.

Uma lista de requisitos para uma nova linguagem foi recentemente estabelecida pelo *ISO Working Group* – projecto ISO JTC1 SC34 – para uma *Topic Map Constraint Language* (TMCL) [11]. Esta lista é exhaustiva e cobre praticamente todos os objectos de Topic Maps.

A linguagem XTche atende todos os requisitos desta lista; para este propósito, XTche possui um conjunto de construtores para a descrição de restrições em Topic Maps, o qual será detalhado nas próximas sub-secções.

Porém, a novidade da proposta é que a linguagem permite a definição de estrutura para topic maps em um estilo baseado em XML Schema; portanto, não é mais necessário uma descrição sintáctica em separado. Uma especificação XTche une o esquema (definição da estrutura e da semântica básica) com restrições (descrevendo a semântica contextual) para todos os topic maps de uma família em particular.

Cabe ressaltar que as especificações XTche definidas de acordo com a sintaxe XML Schema não poderão ser usadas directamente por um parser XML para validar um topic map em formato XTM. Isto porque, apesar de usar tal sintaxe, o código XTche possui apenas a definição das restrições a ser aplicadas a topic maps, portanto necessita ser processado para a obtenção de um validador específico.

O uso da sintaxe XML Schema justifica-se, basicamente, por possibilitar uma edição e visualização gráfica da especificação XTche (pois há um vasto conjunto de ferramentas que possibilitam a edição gráfica de XML Schema) e por possuir uma sintaxe bem conhecida pela comunidade XML.

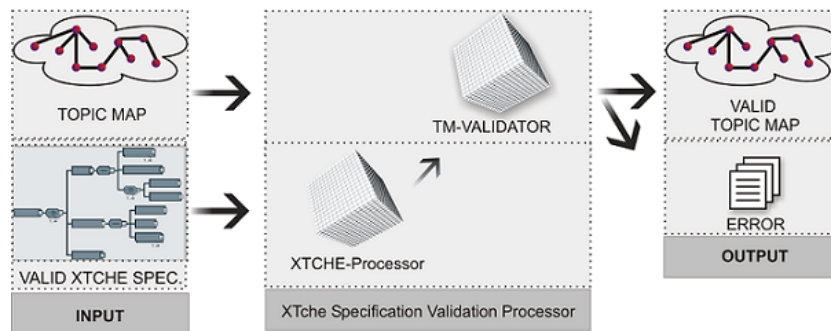
## 5.2 Processador XTche e TM-Validator

As frases escritas na linguagem XTche, referentes a um conjunto de restrições – listando as condições (envolvendo tópicos e associações) que devem ser verificadas – especificam um processo de validação em topic maps (um *TM-Validator*).

Aliado ao facto de XTche ser uma linguagem XML, o formato de intercâmbio mais utilizado pela comunidade académica é o XTM; portanto, tanto a especificação, quanto os topic maps, são anotados em XML. Com isso, é possível habilitar a codificação sistemática, em XSL, desta tarefa de verificação.

Entende-se que a partir destas circunstâncias, é possível gerar automaticamente um validador para cada especificação. Para este propósito, desenvolveu-se o *Processador XTche* (uma folha de estilos XSL) que transforma uma especificação XTche em um *TM-Validator* (outra folha de estilos XSL).

O *Processador XTche* é o gerador de *TM-Validator*; comporta-se precisamente como um gerador de compilador e é o núcleo desta arquitectura, como pode ser visto na Figura 3. Ele toma uma especificação de esquema e restrições em Topic Maps válida (um documento XML escrito de acordo com a linguagem XTche) e gera o *TM-Validator* (uma folha de estilos XSL) que irá processar o topic map para validá-lo ou gerar mensagens de erros, à medida que os mesmos forem sendo detectados. A resposta do *TM-Validator* é a confirmação de que o topic map fornecido como entrada cumpre, ou não, as restrições definidas na especificação XTche.



**Fig. 3.** Arquitectura de XTche. Fonte: [8].

Ambas folhas de estilo XSL (o gerador e o validador) são interpretados por um processador XSL padrão, como o Saxon<sup>1</sup>, o que é um dos benefícios desta proposta.

## 6 Ulisses – Um Navegador Conceptual baseado em Topic Maps

Para navegar em uma ontologia expressada em Topic Maps, esta deve ser vista como um grafo, onde os nodos representam os tópicos, as ligações entre os nodos representam associações e as ligações para os recursos físicos de informação representam as ocorrências. Portanto, a partir de um grafo o processo de criação de um website para a navegação é óbvio: uma página para cada conceito (não somente o nome de cada tópico, mas com todas as suas características) associadas com os nodos relacionados, onde cada ligação representa uma ligação para outra página. Assim, quando uma ligação é escolhida, uma página relacionada é visualizada, apresentando outro conceito ou um recurso de informação.

A ideia principal da navegação conceptual pode ser descrita como: quando se está posicionado sobre um certo conceito, a ferramenta de navegação mostrará

<sup>1</sup> <http://saxon.sourceforge.net/>

as informações associadas a este conceito em particular; se for escolhido algum dos outros conceitos relacionados, a posição muda para o conceito e a visão muda de acordo com a escolha; se for escolhido algum dos recursos de informação, o sistema mostrará o próprio recurso. Esta, portanto, é a ideia sobre a qual se baseou o *Ulisses*.

Assim como as outras duas ferramentas apresentadas (*Oveia* e *XTche*), *Ulisses* – um navegador conceptual – é um processador genérico que pode ser usado independentemente. O *Ulisses* permite navegação sobre topic maps gerados por ferramentas (como o *Oveia*) ou gerados manualmente. Adicionalmente, estes topic maps podem ser validados semanticamente ou não, de acordo com uma especificação *XTche*.

Este browser ainda suporta navegação tanto sobre documentos XTM, como sobre a *BDOntologia*. Assim, quando o topic map estiver no formato XTM, este componente navegacional é implementado como um conjunto de transformações XML através de folhas de estilo XSL; quando a fonte é a *BDOntologia*, ferramentas baseadas em SQL apropriadas são utilizadas para navegar sobre ela.

## 7 Conclusão

Este artigo descreve a integração de sistemas heterogêneos de informação usando o paradigma de Ontologias, para gerar uma visão homogênea destes recursos. A proposta é um ambiente, chamado *Metamorphosis*, para a construção automática de Topic Maps com dados extraídos de várias fontes de dados e a disponibilização de uma navegação semântica sobre a informação extraída.

Embora o *Metamorphosis* tenha sido desenvolvido para o uso em nossas áreas de trabalho principais – processamento de documentos XML aplicado a Arquivos Públicos e Museus Virtuais – estamos convencidos que o *Metamorphosis* pode ser aplicado com sucesso na área geral de sistemas de informação para a integração de dados, análise e exploração de conhecimento.

O *Metamorphosis* tem sido utilizado em vários projectos de pequena e média dimensão. As interfaces Web são criadas rapidamente e sem grandes dificuldades – em termos de especificação – por parte dos utilizadores.

Em termos de trabalhos relacionados, até o presente momento não é do conhecimento dos autores uma ferramenta com as mesmas características do *Metamorphosis*. Contudo, pode-se comparar, individualmente, cada um dos seus módulos com ferramentas conhecidas pela comunidade académica. Uma comparação entre o *Oveia*, o TSIMMIS [4] e o KAON REVERSE [5] é apresentada em [9]. O *XTche* foi comparado em [8] com o *AsTMa!* [1] e com uma proposta de Eric Freese [3]. Por fim, uma comparação entre o *Ulisses* e o Ontopia Omnigator [12] será encontrada em [6].

Um caso de estudo sobre um departamento académico foi apresentado em [9], o qual descreveu a geração de um topic map a partir de uma base de dados MySQL de acordo com especificações XSDS e XS4TM. O resultado final, armazenado em uma *BDOntologia* em Microsoft SQL Server 2000, tinha 4420 tópicos e 4223 associações; o mesmo topic map teria aproximadamente 172490

linhas no formato XTM. O *Ulisses* fornece uma navegação completa sobre o topic map gerado. O tempo médio para a extracção de cada elemento (tópico ou associação) foi de 0,1311 segundos.

Para mais detalhes sobre o *Metamorphosis* e seus módulos, recomenda-se a leitura de [7–9].

## References

1. Robert Barta. AsTma! Bond University, TR., 2003. <http://astma.it.bond.edu.au/constraining.xsp>.
2. Michel Biezunsky, Martin Bryan, and Steve Newcomb. ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34, December, 1999. <http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>.
3. Eric Freese. Using DAML+OIL as a Constraint Language for Topic Maps. In *XML Conference and Exposition 2002*. IDEAlliance, 2002. [http://www.idealliance.org/papers/xml02/dx\\_xml02/papers/05-03-03/05-03-03.html](http://www.idealliance.org/papers/xml02/dx_xml02/papers/05-03-03/05-03-03.html).
4. H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J.D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. In *Journal of Intelligent Information Systems*, volume 8(2), pages 117–132. Kluwer Academic Publishers, 1997.
5. S. Handschuh, A. Maedche, L. Stojanovic, and R. Volz. KAON, 2001. <http://kaon.semanticWeb.org/kaon/white-paper.pdf>.
6. Giovanni Rubert Librelotto. *XML Topic Maps: da Sintaxe à Semântica*. Tese de Doutoramento, Departamento de Informática, Universidade do Minho, 2005. to be published.
7. Giovanni Rubert Librelotto, José Carlos Ramalho, and Pedro Rangel Henriques. Extração de topic maps no oveia: Especificação e processamento. In Mauricio Solar, David Fernández-Baca, and Ernesto Cuadros-Vargas, editors, *30ma Conferencia Latinoamericana de Informática (CLEI2004)*, pages 451–460. Sociedad Peruana de Computación, September 2004. ISBN 9972-9876-2-0.
8. Giovanni Rubert Librelotto, José Carlos Ramalho, and Pedro Rangel Henriques. XTche - A Topic Maps Schema and Constraint Language. In *XML 2004 Conference and Exposition*, Washington D.C., U.S.A, 2004. IDEAlliance.
9. Giovanni Rubert Librelotto, Weber Souza, José Carlos Ramalho, and Pedro Rangel Henriques. Using the Ontology Paradigm to Integrate Information Systems. In *International Conference on Knowledge Engineering and Decision Support*, pages 497–504, Porto, Portugal, 2004.
10. Mary Nishikawa and Graham Moore. Topic Map Constraint Language (TMCL) Requirements and Use Cases. ISO/IEC JTC 1/SC34 N0405rev, 2003. <http://www.isotopicmaps.org/tmcl/requirements.html>.
11. Mary Nishikawa, Graham Moore, and Dmitry Bogachev. Topic Map Constraint Language (TMCL) Requirements and Use Cases. ISO/IEC JTC 1/SC34 N0405rev, 2004. <http://www.jtc1sc34.org/repository/0548.htm>.
12. Ontopia. The Ontopia Omnigator, 2002. <http://www.ontopia.net/omnigator/>.
13. Steve Pepper. The TAO of Topic Maps - finding the way in the age of infoglut. Ontopia, 2000. <http://www.ontopia.net/topicmaps/materials/tao.html>.
14. Steve Pepper and Graham Moore. XML Topic Maps (XTM) 1.0. TopicMaps.Org Specification, August, 2001. <http://www.topicmaps.org/xtm/1.0/>.

15. Steve Pepper and Graham Moore. XML Topic Maps (XTM) 1.0 - Annex D: XTM 1.0 Document Type Declaration (Normative). TopicMaps.Org Specification, August, 2001. <http://www.topicmaps.org/xtm/1.0/#dtd>.
16. Kevin Williams, Michael Brundage, Patrick Dengler, Jeff Gabriel, Andy Hoskinson, Michael Kay, Thomas Maxwell, Marcelo Ochoa, Johnny PaPa, and Mohan Vanmane. *Professional XML Databases*. Wrox Press, 2000.