

MAPEAMENTO DA NORMA ISO 13250 *TOPIC MAPS* PARA BANCO DE DADOS RELACIONAIS

MAPPING OF TOPIC MAP ISO 13250 NORM FOR RELATIONAL DATA BANK

Lucas Indrusiak¹, Renato Azevedo² e Giovani R. Librelotto³

RESUMO

Este artigo propõe um Modelo Entidade-Relacionamento que suporte a estrutura de arquivos XTM (*XML Topic Maps*), o qual foi base no desenvolvimento de um banco de dados. Foi implementado um aplicativo capaz de importar as informações de arquivos XTM para o banco de dados e exportar *topic maps* armazenados no banco novamente para arquivo XTM, além de gerar dinamicamente uma interface *Web* para a navegação pelos *topic maps* armazenados.

Palavras-chave: Modelo Entidade-Relacionamento, arquivo XTM, interface *web*.

ABSTRACT

In this article we suggest an Entity-Relationship Model that may hold the structure of XTM files (XML Topic Maps), which was the base in the development of a data bank. An application software was implemented, and it was able to import information from XTM files for the data bank and to export topic maps stored in the bank back to XTM files, besides generating dynamically a Web interface for the navigation by the stored topic maps.

Keywords: *Entity-Relationship Model, XTM file, web interface.*

¹ Acadêmico do Curso de Sistemas de Informação - UNIFRA.

² Acadêmico do Curso de Ciência da Computação - UNIFRA.

³ Professor do Curso de Ciência da Computação - UNIFRA.

{indrusiak.red,rpa.renato}@gmail.com,giovani@unifra.br

INTRODUÇÃO

Topic Maps é uma especificação formalizada conforme a norma ISO/IEC 13250 que provê um modelo para representação de estruturas de conhecimento sobre temas determinados, que são representados por tópicos. XML *Topic Maps* (XTM) é um padrão desenvolvido que suporta a norma *Topic Maps* em arquivos escritos em linguagem XML. O presente artigo propõe a implementação de um aplicativo capaz de importar e exportar arquivos XML *Topic Maps* (PEPPER; MOORE, 2001) para um banco de dados relacional e a navegação *Web* gerados dinamicamente pelo aplicativo através de consultas aos dados já armazenados no banco.

Para que o padrão XTM fosse compreendido, inicialmente foram estudadas as suas origens, como a linguagem XML (*Extensible Markup Language*) (BRAY et al., 2000), pois sua sintaxe é baseada nessa linguagem. Foi estudado o paradigma *Topic Maps*, definido pela norma ISO/IEC 13250 (BIEZUNSKY et al., 1999), sua semântica, suas estruturas, seus elementos e sua hierarquia. Optou-se pela linguagem de programação orientada a objetos Java para a implementação do aplicativo, pois além de ser uma linguagem muito poderosa, gratuita e portátil, que é uma característica presente em todos os assuntos abordados, possui uma grande comunidade e oferece suporte a programadores.

O primeiro passo para a implementação do aplicativo foi a criação de um Modelo Entidade-Relacionamento (ER) capaz de armazenar os arquivos XTM, preservando a sua estrutura e a sua hierarquia entre os elementos. Após definido o diagrama ER, começou a implementação do aplicativo: foram definidas as classes que servem como interface das informações entre os arquivos e o banco de dados, os métodos de importação, o tratamento das informações presentes nos arquivos XTM e inserção no banco de dados, os métodos de exportação das informações contidas nas tabelas do banco de dados, a formação de um novo arquivo XTM e, por fim, métodos para gerar os arquivos HTML permitindo navegar e consultar as informações dos *topic maps* contidos no banco de dados.

Este artigo inicia com uma apresentação dos conceitos básicos sobre *Topic Maps* na seção 2. A seção 3 descreve o processo de desenvolvimento do esquema relacional e do banco de dados para o armazenamento de *Topic Maps*. Os trabalhos relacionados estão na seção 4 e uma síntese do artigo é apresentada na seção 5.

TOPIC MAPS

Topic Maps é uma norma mantida pela ISO/IEC, utilizada para facilitar o intercâmbio de grandes volumes de informação e melhorar a eficiência do acesso e da manipulação dessa informação. Nesta seção, serão abordados temas relacionados à norma, como, por exemplo, sua origem, suas especificações, sua sintaxe, sua estrutura, entre outros (BIEZUNSKY et al., 1999).

Topic Maps fornece um modelo para representação de estruturas de conhecimento sobre temas determinados, que são representados por tópicos. Tópico é a representação de qualquer coisa abstrata ou concreta que tenha necessidade de manter informações sobre as características de tópicos, como nomes, recursos e relacionamentos, que podem se limitar a contextos específicos, conforme o desejado. Um ou mais documentos, segundo esse modelo, é chamado *topic map*.

Um tópico é um recurso computacional que representa (reifica) qualquer coisa relacionada com um tema do mundo real, que pode ser algo concreto ou abstrato. Por exemplo, sobre o tema *móveis*, podem ser tópicos *cadeira*, *mesa*, *cama*.

As principais características do tópico são: (a) *identificador*: todo tópico obrigatoriamente possui um identificador único que irá diferenciá-lo dos demais tópicos e é usado pelas aplicações que utilizam os *topic maps*; (b) *tipo*: um tópico pode representar uma instância de outro tópico; (c) *nomes*: são usados para auxiliar a compreensão humana do tópico em questão; (d) *identificador de tema*: indica o tema que o tópico representa do mundo real.

Os tópicos podem estar relacionados a recursos de informações que são relevantes a eles de alguma forma. Esses recursos são chamados ocorrências (*occurrences*) do tópico. Uma ocorrência sobre o tópico *Dinossauro*, por exemplo, poderia ser um texto científico sobre répteis primitivos, uma foto de um dinossauro, o filme “Jurassic Park”, ou qualquer outro recurso de informação que tenha alguma relação com o tópico em questão. Existem dois tipos de ocorrências:

- Referência ao recurso (*resourceRef*): encontram-se externas ao *topic map*, mas possuem uma conexão de acordo com o sistema usado. Isso permite que o padrão *topic map* possa ser aplicado em um sistema de informação já existente sem a alteração dos documentos em si;
- Recurso de dado (*resourceData*): ocorrências internas, utilizadas para atribuir um valor específico a um tópico. Pode ser usado para associação de meta-dados a tópicos. Não há limitações para tipos de dados a serem usados e para sua manipulação e é responsabilidade do aplicativo que interpreta o *topic map*.

Uma associação define alguma relação qualquer entre dois ou mais tópicos. A habilidade de classificar associações de acordo com o seu tipo aumenta bastante o poder de expressão dos *topic maps*, tornando possível agrupar vários tópicos que possuam o mesmo relacionamento com determinado tópico. Isso é muito importante para prover interfaces amigáveis para navegar grandes volumes de informações (PEPPER, 2000).

ARMAZENAMENTO DE *TOPIC MAPS* EM MODELO RELACIONAL

O propósito deste trabalho é, primeiramente, conseguir criar um modelo ER (entidade-relacionamento) que possibilite armazenar todo o conteúdo de um arquivo XTM. Para isso, foram tomados alguns cuidados para que o modelo preservasse todas as características do arquivo XTM, pois seu objetivo, além de ter a função de importação de arquivos XTM externos para o banco, também deve possibilitar a exportação de cada *topic map* armazenado no banco novamente para arquivos XTM.

Como a compreensão de um arquivo XTM, sem o auxílio de nenhum aplicativo especializado para a utilização desse formato, pode ser um pouco difícil, foi implementado um módulo que permite navegar pelas informações contidas nos *topic maps*, deixando bastante simples a tarefa de pesquisar qualquer coisa sobre esses arquivos, suas estruturas, hierarquias e quaisquer recursos ligados ao mesmo.

MODELO ER PARA *TOPIC MAPS*

Existem diversas maneiras e regras pré-definidas para mapear DTDs para modelo ER, mas nenhuma será usada nesta pesquisa. Existem outros trabalhos relacionados a esse assunto que serão comentados na penúltima seção deste artigo que utilizou, inclusive, tais regras.

Como foi visto anteriormente, a estrutura dos *topic maps* é uma árvore contendo todos os elementos em níveis definidos e fixos, embora alguns elementos, como o *topicRef*, possam apresentar em níveis diferentes, essas particularidades que devem ser observadas ao se criar o modelo ER, pois ao contrário da linguagem XML, ele não é muito bom para armazenamento de informações que não possuem uma estrutura bastante rígida. Todos os elementos de um *topic map* possuem atributos e/ou algum parentesco com outro elemento. No modelo ER,

tabelas possuem atributos e/ou relacionamentos com outras tabelas, o que leva à indubitável conclusão de que cada elemento de um *topic map* será representado por uma tabela no modelo ER.

O parentesco entre o nodo *topicMap* e os nodos *topic* ou *association* (oriundos do formato XTM) será representado no modelo ER por relacionamentos do tipo 1-n (lê-se um para n), ou seja, todo elemento do tipo *topicMap* poderá estar relacionado com zero ou mais elementos do tipo *topic*, enquanto todo elemento do tipo *topic* deverá estar relacionado com apenas um elemento do tipo *topicMap*. Esse tipo de relacionamento vale para os elementos *topicMap* e *association*.

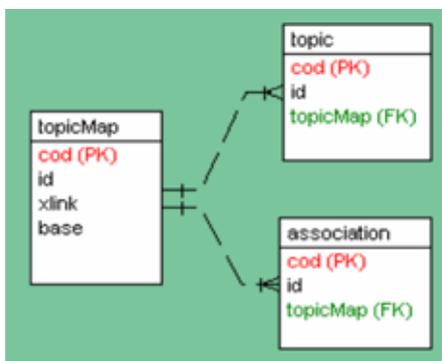


Figura 1 – Trecho do modelo ER para *Topic Maps*.

Como demonstra a figura 1, foi criada uma tabela para os elementos *topicMap*, *topic* e *association*. Cada tabela conterà os atributos que constam no DTD do modelo XTM mais outros atributos que foram adicionados no modelo ER, para que seja possível manter a hierarquia da árvore.

Por exemplo, a tabela *topic* conterà os seguintes atributos: *cod*, *id* e *topicMap*. Foi adicionado o atributo chamado *cod* para todas as tabelas, que será sua chave primária em toda a base de dados, já que serão armazenados *topic maps* ilimitados na mesma base e cada um deles apresentará identificadores únicos para cada elemento dentro do próprio *topic map*. O atributo *id* é importado do arquivo *topic map*, porém o atributo *topicMap* representa a chave estrangeira na tabela *topic*, fazendo com que ela seja relacionada à tabela *topicMap* contendo a chave primária (*cod*) do elemento da tabela *topicMap* com que o elemento atual da tabela *topic* se relaciona.

Como na figura 1, todos os outros elementos de *Topic Maps* relacionam-se da mesma forma, possibilitando a modelagem completa do diagrama ER, segundo a figura 2.

A FERRAMENTA XTM-BD

Com base no diagrama ER previamente modelado, foi desenvolvida a base de dados para o uso da ferramenta que foi implementada. O banco de dados escolhido foi o PostgreSQL, pois é um banco de dados livre, de código aberto e que atualmente vem sendo comparado com os melhores bancos de dados proprietários do mundo, como Oracle, DB2, entre outros. A partir do modelo ER, foram criadas todas as tabelas e seus relacionamentos no banco de dados. Nas figuras 3 e 4 encontram-se exemplos dos comandos SQL para a criação das tabelas *topicMap* e *topic*.

```
CREATE TABLE "topicMap"
(
  cod int4 NOT NULL DEFAULT nextval('"topicMap_cod_seq"'::regclass),
  id varchar(50),
  xlink varchar(50),
  base varchar(50),
  CONSTRAINT "topicMap_pkey" PRIMARY KEY (cod)
)
WITHOUT OIDS;
ALTER TABLE "topicMap" OWNER TO postgres;
```

Figura 3 – Comando SQL para criação da tabela *topicMap*.

```
CREATE TABLE topic
(
  cod serial NOT NULL,
  id varchar(50),
  "topicMap" numeric NOT NULL,
  CONSTRAINT cod PRIMARY KEY (cod),
  CONSTRAINT "topic_topicMap_fkey" FOREIGN KEY ("topicMap")
  REFERENCES "topicMap" (cod) MATCH SIMPLE
  ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITHOUT OIDS;
ALTER TABLE topic OWNER TO postgres;
```

Figura 4 – Comando SQL para criação da tabela *topic*.

Dessa forma, o banco de dados suporta o conteúdo dos arquivos XTM. O próximo passo será o desenvolvimento de um *software* que seja capaz de interagir com o banco. Foi utilizada a linguagem de programação orientada a objetos Java para a implementação do aplicativo.

A orientação a objetos consiste em um paradigma de programação que relaciona módulos de um *software* com um objeto que poderá apresentar determinadas características e ser capaz de executar determinadas ações. Analogamente, tais características e ações, em programação orientada a objetos, são chamadas atributos e métodos de um objeto, respectivamente.

A classe será o modelo, o esqueleto, que contém todos os atributos e métodos possíveis de todos os objetos pertencentes a essa classe. Portanto, tem-se um banco de dados composto de várias tabelas, com vários atributos em cada uma e deseja-se criar um *software* que possibilite o intercâmbio de informações de arquivos do tipo XTM para tais tabelas, e vice-versa. Como se optou por criar uma linguagem de programação orientada a objetos, uma solução que se adaptou a essa finalidade foi representar cada tabela por uma classe, em que cada tupla (linha) representa um objeto dessa classe e o conteúdo dos campos da tupla são os atributos do objeto. A figura 5 representa essa modelagem.

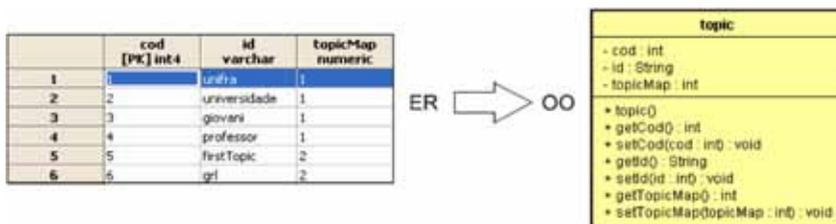


Figura 5 – Mapeamento do Modelo ER para o Modelo OO.

O modelo aplica-se para todas as outras tabelas do banco de dados. Assim, o esquema possui uma classe representando cada tabela do banco de dados com todos seus campos. Como o banco de dados suporta todas as informações dos arquivos XTM, a modelagem orientada a objetos também suportará interagir diretamente com os arquivos XTM, podendo servir tanto para importar arquivos XTM para o banco de dados quanto para exportar *topic maps* armazenados no banco de dados para arquivos XTM novamente.

NAVEGAÇÃO SOBRE *TOPIC MAPS* NO XTM-ER

A função de navegação do aplicativo XTM-ER funciona da seguinte forma: ao abrir a sua interface, conforme mostra a figura 6, haverá uma

caixa de listagem onde constam todos *topic maps* armazenados no banco de dados. Depois de pressionado o botão com o rótulo *Navegar*, são criados dinamicamente vários arquivos HTML referentes ao sistema para navegação em uma pasta temporária na unidade local de armazenamento, os quais serão removidos após o fechamento do aplicativo.



Figura 6 – Interface do XTM-ER.

Após a criação de todos os arquivos HTML, é aberto o navegador *Web* padrão do sistema no arquivo principal do qual começa a navegação. Nesse arquivo, encontram-se os dados do tópico raiz do *topic map* escolhido. Nessa mesma tela inicial encontram-se *links* que serão a base para toda a navegação do *topic map*. Na interface do arquivo HTML podem ser identificados todos os seus nomes, seu tipo de tópico e suas instâncias ou, ainda, tópicos que façam parte desse tipo.

TRABALHOS RELACIONADOS

Existem alguns trabalhos que já abordaram o assunto do uso de modelos relacionais para o mapeamento de *topic maps*. Pela necessidade de um meio mais flexível que permita intercâmbio de dados, Graauw (2003) criou um modelo relacional para *topic maps*, baseado no paradigma *Topic Maps* em que tudo é representado por um tópico, que pode ser instância de outro tópico (tipo de tópico). Essa modelagem permite que se possa saber o tipo de um tópico apenas considerando os dados da tabela tópico, o que não é possível no padrão XTM. Apesar de não ter contemplado completamente a norma *Topic Maps*, o modelo proposto pelo autor poderia suprir as necessidades que deram origem a ele.

Por sua vez, Mugnaini (2003) propõe um meio de armazenar *topic maps* permanentemente em bancos de dados relacionais, por meio de um diagrama ER, em vários aspectos semelhante ao modelo proposto neste artigo.

Os referidos trabalhos apresentam diferentes soluções para problemas semelhantes. O DER (Diagrama ER) de Mugnaini mostra uma estrutura muito parecida

com o modelo proposto nesta pesquisa, em que cada elemento do esquema XTM é representado por uma tabela cuja estrutura hierárquica dos elementos é mantida por relacionamentos.

Já o modelo proposto por Graauw baseia-se na explicitação dos tipos de tópicos, ocorrências e associações em tabelas separadas, permitindo a adição de novos dados também no modelo ER pela simples inserção de dados nessas tabelas, já que tudo é representado por tópicos e a diferença está apenas no tipo (instâncias). Tudo traduz o paradigma *Topic Maps* para o modelo ER, porém este oferece a possibilidade de identificar quais os tipos, observando apenas a tabela original, o que não é possível no padrão *Topic Maps*.

CONCLUSÃO

O armazenamento de *topic maps* em um banco de dados relacional oferece inúmeras opções para quem necessita da interação dessas duas formas de estruturar informações. O diagrama ER apresentado neste artigo tem a possibilidade de armazenar todos os dados que constam em um documento segundo o padrão XTM (PEPPER; MOORE, 2001), preservando a hierarquia das suas estruturas e as particularidades dos seus elementos. Com isso, qualquer sistema que precise de alguma forma do intercâmbio de dados, entre qualquer banco de dados relacional e arquivos XTM, poderia aplicar o modelo proposto criando um banco compatível para o armazenamento e a interação com esses arquivos.

O aplicativo desenvolvido auxiliaria como meio entre o sistema de arquivos XTM e o modelo relacional, já que foram implementados métodos que permitem a importação desses arquivos para o banco de dados e a exportação de *topic maps* armazenados no banco para arquivos XTM. Com isso, é o possível armazenamento de ilimitados *topic maps* em um único banco de dados, facilitando a consulta e o acesso a grandes volumes de informações e aumentando ainda mais a portabilidade de sistemas baseados em *topic maps*. A interface de navegação em *browsers* por meio dos arquivos HTML, gerados pelo aplicativo desenvolvido, demonstra o potencial dos *Topic Maps*, pois representam a estruturação do conhecimento.

A definição do que é cada elemento, o que pode ser representado por eles, a relação que estabelecem entre si, a interação com outros recursos e as aplicações práticas da norma podem tanto auxiliar em sua compreensão quanto ser úteis em casos reais nos quais seja necessária a visualização das informações contidas nos *topic maps*. Outras funções poderão ser adicionadas ao aplicativo como trabalhos futuros, para que complementem seu uso, como, por exemplo, a edição em tempo real dos *topic maps* armazenados no banco por meio da interface do *browser*.

REFERÊNCIAS

BERNERS-LEE, T.; HENDLER, J.; LASSILA O. The Semantic Web. **Scientific American**. 2001. Disponível em: <http://www.sciam.com/2001/0501issue/0501berners-lee.html>. Acesso em: nov. 2006.

BIEZUNSKY, M.; BRYAN, M.; NEWCOMB, S. (1999). ISO/IEC 13250 - Topic Maps. ISO/IEC JTC 1/SC34, December. Disponível em: <http://www.y12.doe.gov/sgml/sc34/document/0129.pdf>. Acesso em: ago. 2006.

BRAY, T. et al. **Extensible Markup Language (XML) 1.0**, 2004. Disponível em: <http://www.w3.org/TR/REC-xml/>. Acesso em: maio 2006.

GRAAUW, M. **Using Topic Maps to Extend Relational Databases**. 2003. Disponível em: <http://www.xml.com/lpt/a/2003/03/05/tmrdb.html>. Acesso em: nov. 2006.

MUGNAINI, L. **Mapping Topic Maps on Relational Databases**. 2000. Disponível em: http://www.geocities.com/xtopicmaps/mapping_xtm_on_databases.html. Acesso em: nov. 2006.

PEPPER, S. **The TAO of Topic Maps – Finding the way in the Age of Infoglut**. Oslo: Ontopia, 2000.

W3C, XML Core. **Extensible Markup Language (XML)**. 2006. Disponível em: <http://www.w3.org/XML/>. Acesso em: maio 2006.

PEPPER, S.; MOORE, G. **XML Topic Maps (XTM) 1.0**. 2000. Disponível em: <http://www.topicmaps.org/xtm/>. Acesso em: jun. 2006.

