

# Uma Ferramenta MDA para Modelagem de Banco de Dados Relacionais

André S. Rosa<sup>1</sup>, Carlos Eduardo Pantoja<sup>1</sup>

<sup>1</sup>CEFET/RJ - UnED Nova Friburgo, Rio de Janeiro, Brasil

andre\_souza.rosa@hotmail.com, pantoja@cefet-rj.br

***Abstract.** This paper proposes a Tool for relational databases modeling that uses the Model-Driven Architecture (MDA) approach. The generic meta-model used in this approach enables the designer to choose among different languages and notations. Moreover, it allows the SQL automatic code generation. The tool for the Entity-Relationship notation will be constructed using the Graphical Modeling Framework (GMF). A simple work in progress example is presented the tool's main purpose.*

***Resumo** Este artigo propõe uma ferramenta para modelagem de bancos de dados relacionais que usa a abordagem Model-Driven Architecture (MDA). O meta-modelo utilizado na abordagem permite ao projetista escolher entre diferentes linguagens e notações. Além disso, permite a geração do código na linguagem SQL de forma automática. A ferramenta para a notação Entidade-Relacionamento (ER) vai ser construída usando o Graphical Modeling Framework (GMF). Um simples exemplo do andamento do trabalho é apresentado para garantir a finalidade principal da ferramenta.*

## 1. Introdução

A modelagem conceitual é uma descrição concisa dos requisitos de dados do usuário, que inclui detalhes específicos do banco de dados e é usada para garantir que tais requisitos sejam atendidos e que não estejam em conflito entre si [Elmasri et al., 2005]. Existem diversos modelos para modelagem conceitual, voltados para banco de dados relacionais como o Modelo Entidade-Relacionamento (ER) [Chen, 1976], o *Crow'sFoot* [Simsion, 2007] e o diagrama de classes da UML, que apesar de ser um recurso da linguagem de modelagem para sistemas orientados a objetos também pode ser utilizado para modelagem de banco de dados relacionais.

Algumas ferramentas auxiliam o projetista na modelagem conceitual automatizando o projeto de banco de dados, como a *Xcase* [Xcase, 2013], que possui suporte para engenharia reversa e abrange grande parte dos Sistemas Gerenciadores de Banco de Dados (SGBD); o *ER/Studio* [ER/Studio, 2013], que também possui suporte a engenharia reversa e recursos para manutenção da base de dados; e o *brModelo* [Cândido, 2004], que é uma ferramenta *freeware* voltada para o ensino, usa a notação Extended Entity-Relationship (EER), e ainda, faz a conversão do modelo conceitual para o modelo lógico e geração de código para o modelo físico.

Porém, as ferramentas apresentam algumas limitações, como o caso da *Xcase*, que possui o atrelamento a uma notação específica, a *Crow'sFoot*, além de ser uma ferramenta privada. O *ER/Studio* possui suporte a duas notações, a *Crow'sFoot* e a IDEF1X, e a modelagem pode ser direcionada a um grande número de SGBD, porém é

uma ferramenta privada. O brModelo, apesar de ser um *freeware*, tem como foco apenas a notação EER.

Este artigo tem como objetivo propor uma ferramenta, baseada na Arquitetura Orientada por Modelos (*Model-Driven Architecture - MDA*), para modelagem conceitual de banco de dados relacionais. A MDA é uma abordagem de desenvolvimento de *software* dirigida por modelos em diversos níveis de abstração, onde um sistema é modelado usando um modelo independente de plataforma, que será transformado em um modelo específico de plataforma, dado um modelo de plataforma. A utilização de modelos direciona o entendimento, *design*, construção, teste, operação, manutenção e modificação do sistema [Mellor et al., 2005].

A ferramenta utilizará um meta-modelo genérico para as linguagens de modelagens relacionais e o gerador de código de núcleo comum com os padrões ANSI/SQL 93/99/03, proposto por [Rosa et al. 2013], integrado a um ambiente gráfico desenvolvido no *Graphical Modeling Framework (GMF)*, permitindo ao projetista utilizar qualquer linguagem de modelagem existente. O GMF disponibiliza a infraestrutura necessária para o desenvolvimento de editores gráficos para modelos e foi construído baseado no *Eclipse Modeling Framework (EMF)* [Steinberg et al. 2008].

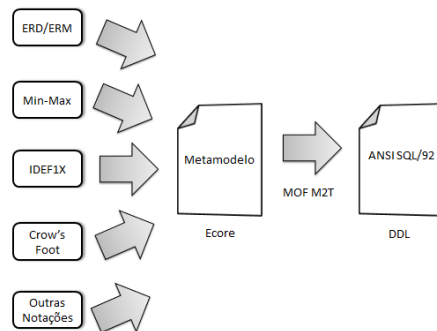
O artigo está estruturado da seguinte forma: na seção 2 será apresentada a ferramenta desenvolvida; na seção 3 serão apresentados os resultados obtidos; na seção 4 serão apresentados alguns trabalhos relacionados; e por fim, na seção 5 a conclusão.

## 2. A Ferramenta MDA

A ferramenta, que consiste em um conjunto de *plug-ins* integrados para o ambiente de desenvolvimento *Eclipse*, utiliza uma metodologia MDA para modelagem conceitual de banco de dados relacionais. A codificação automática ANSI SQL 92/99/03 é gerada a partir de um conjunto de regras de transformações na linguagem *Model To Text (M2T)*.

A M2T [OMG, 2008] é uma linguagem utilizada para transformar instâncias de modelos em artefatos de texto a partir de regras de transformações estruturadas na forma de *templates*, onde estes são responsáveis por determinada transformação. A metodologia [Rosa et al., 2013] tem como núcleo comum o meta-modelo genérico para as linguagens de modelagens ER, IDEF1X, *Crow'sFoot* e UML; além de permitir a utilização da notação Min-Max nas instâncias de seus modelos. A metodologia ainda permite a integração de outras linguagens de modelagens, desde que essas sejam aderentes aos conceitos previstos no meta-modelo. A metodologia utilizada pela ferramenta pode ser vista na figura 1.

Para a construção da interface gráfica é necessário ter um modelo de domínio como base, que será o meta-modelo da metodologia adotada. O meta-modelo é representado pelo *Ecore Model (.ecore)*, que é o arquivo utilizado pelo EMF na construção de meta-modelos. Será criado o arquivo contendo o *Domain Generator Model (.gen)*, utilizado para geração do código Java referente a cada elemento presente no modelo de domínio. Também será gerado a partir desse arquivo um projeto de extensão *.edit*, que terá o papel de possibilitar a customização de alguma das partes que precisam ser definidas como imagens e ícones utilizados na representação de recursos. Esse projeto dará origem a parte dos *plug-ins* que conterão as configurações da ferramenta para utilização no *Eclipse*.



**Figura 1. Metodologia adotada [Rosa et al. 2013].**

Os arquivos contendo o *GMF Tool Model* (.gmftool) e o *GMF Graph Def Model* (.gmfgraph) serão criados onde o .gmftool conterá as definições e as customizações da paleta de recursos disponíveis para modelagem. Já o .gmfgraph irá conter a definição gráfica de cada elemento dos recursos para modelagem conceitual. Os arquivos .gen, .gmftool e gmfgraph serão submetidos a uma combinação através do GMF e darão origem ao *Mapping Model* (.gmfmap). O arquivo .gmfmap fará o mapeamento associando os elementos do modelo de domínio às classes; as classes aos elementos gráficos; e esses elementos a suas representações na paleta de recursos.

Será realizada uma transformação sobre o arquivo .gmfmap para gerar o *Diagram Editor Gen Model* (.gmfgen). Através do .gmfgen será possível gerar um projeto executável que conterá as classes necessárias para o funcionamento da interface gráfica e os *plug-ins* utilizados no *Eclipse*. A interface gráfica proposta dará suporte, inicialmente, para modelagens utilizando o modelo ER. O modelo foi escolhido por ser uma linguagem de modelagem amplamente utilizada.

### 3. Resultados Obtidos

Nesta seção são apresentados os resultados parciais obtidos no desenvolvimento da ferramenta gráfica. Na figura 2 há um simples exemplo da notação ER utilizando a interface gráfica desenvolvida, onde ilustra um caso de proprietários e seus respectivos veículos com a cardinalidade (1,N).



**Figura 2 - Exemplo de modelagem conceitual utilizando a ferramenta proposta.**

Os atributos e a cardinalidade não estão sendo graficamente representada nessa versão inicial. Portanto é necessário que estes sejam instanciados no modelo de domínio manualmente. Após a modelagem no ambiente gráfico, o gerador de código ANSI/SQL de Linguagem de Definição de Dados pode ser utilizado para gerar o código do modelo construído nos padrões 92/99/2003.

### 4. Trabalhos Relacionados

A ferramenta proposta com base na abordagem MDA nesse artigo, apesar de ainda estar com o ambiente gráfico em desenvolvimento, será construída tendo em seu núcleo um meta-modelo já existente que reúne características comuns às notações de modelagem

conceitual relacional. Por possuir em seu núcleo esse meta-modelo, a ferramenta estará livre do atrelamento a uma única notação, como é o caso do *Xcase*, que suporta apenas a notação de *Crow'sFoot*, do *brModelo*, que suporta apenas o Entidade-Relacionamento e do *ER/Studio*, que apesar de ser mais versátil as demais ferramentas, suporta apenas as notações de *Crow'sFoot* e IDEF1X.

## 5. Conclusão

O artigo apresentou uma ferramenta para modelagem conceitual de banco de dados relacionais que utiliza a arquitetura MDA e permite a utilização de diferentes linguagens de modelagem e notações de banco de dados para automatização do projeto de banco de dados. Foi apresentada a parte gráfica da ferramenta para o MER, com um simples exemplo de seu funcionamento, permitindo a geração de codificação automática ANSI SQL 92/99/03 a partir de seu núcleo comum utilizando um conjunto de regras M2T.

A ferramenta oferece flexibilidade ao projetista na escolha da linguagem de modelagem a ser utilizada para a modelagem conceitual do banco de dados, pois independente da linguagem escolhida para o projeto, a ferramenta irá gerar a codificação padrão ANSI SQL 92/99/03, que é compatível com a maioria dos SGBD. Como trabalhos futuros, serão desenvolvidas extensões gráficas e integrações com ferramentas já existentes para as linguagens de modelagem *Crow'sFoot*, IDEF1X e UML; além de permitir a utilização da notação Min-Max.

## Referências

- CÂNDIDO, C. H. (2004). *brModelo: Ferramenta de Modelagem Conceitual de Banco de Dados*. Dissertação, Centro Universitário UNIVAG.
- CHEN, P. P. (1976). The entity-relationship model - toward a unified view of data. *ACM Trans. Database Syst.*, v. 1, n. 1, p. 9–36.
- ELMASRI, R., NAVATHE, S. B. (2005). *Sistemas de banco de dados*. Editora Pearson.
- ER/Studio (2013). <http://www.embarcadero.com/br/products/er-studio>.
- MELLOR, S. J., SCOTT, K., UHL, A. e WEISE, D. (2005). *MDA Destilada: Princípios de Arquitetura Orientada por Modelos*. Ciência Moderna Ltda.
- OMG (Object Management Group) (2008). MOFModel To Text Transformation Language (MOFM2T), 1.0. <http://www.omg.org/spec/MOFM2T/1.0>.
- ROSA, A., GONÇALVES, I. and PANTOJA, C. E. (2013). A MDA Approach for Database Modeling. *Lecture Notes on Software Engineering*, v. 1, n. 1, p. 26–30.
- SIMSION, G. (2007). *Data Modeling: Theory and Practice*. Technics Publications Llc.
- STEINBERG, D., BUDINSKY, F., MERKS, E. and PATERNOSTRO, M. (2008). *Emf: Eclipse Modeling Framework*. Pearson Education.
- Xcase Database Design Software (2013). <http://www.xcase.com/>.