

USANDO ONTOLOGIAS NA CONSTRUÇÃO DE MODELOS MDA (*MODEL-DRIVEN ARCHITECTURE*)

Jaguaraci Batista Silva¹ <jaguaracisilva@gmail.com>

Milton Sampaio² <miltionsampaio@terra.com.br>

Juliana Pezzin³ <Juliana_pezzin@hotmail.com>

Universidade Salvador (Unifacs) – Curso de Pós-graduação em Engenharia de Software – Campus Costa Azul
Rua Doutor José Peroba, 215 – Stiep– CEP 41770-235 – Salvador – BA

RESUMO

O domínio de uma aplicação, durante a identificação de requisitos, utiliza um modelo de classes independente da plataforma ao qual se destina, abstraindo boa parte da especificação de um sistema sem uma verificação formal, seja sintática ou semântica. Este artigo propõe uma arquitetura para utilização de ontologia na construção de modelos MDA, formalizando seus conceitos e reduzindo o tempo de manutenções em modelos durante a sua especificação para um determinado domínio.

Palavras-chaves: MDA; Ontologia; Modelo.

ABSTRACT

Title: “Using ontologies in the MDA (model-driven architecture) models constructions”

The domain of an application, during the requirements identification, uses a platform independent class model the which that it destines, abstracting good part of the specification of a system without a formal verification, be either syntactic or semantic. This article proposes an architecture for use of ontology in the construction of MDA models formalizing its concepts and reducing the time of maintenances in models during its specification for one determined domain.

Key- words: Publications; MDA, Ontology, Model.

1 INTRODUÇÃO

A MDA (Model- Driven Architecture), ou Arquitetura Orientada a Modelos, propõe a criação de modelos em diferentes níveis de abstração, separando os interesses de implementação da arquitetura a ser implementada. A MDA utiliza o modelo conceitual de uma aplicação para geração de outro em uma plataforma específica, reduzindo drasticamente o número de implementações de sistemas para o mesmo domínio de aplicação. O objetivo deste trabalho é propor a aplicação do uso da ontologia na construção de um modelo MDA, devido à insuficiência formal com a combinação do uso da UML e MOF.

Existem vários problemas encontrados na utilização da MDA. Dentre eles, os discutidos em fóruns são: a mistura da parte conceitual da aplicação com os comportamentos específicos nas regras de transformação de um metamodelo; a utilização de padrões arquiteturais; a sincronização de modelos com artefatos de código e a falta de formalização deste. A UML é a linguagem utilizada para construção de modelos MDA. Segundo Freitas (2003), a UML não pode ser considerada um formalismo de representação, devido à ausência de declaratividade, de um motor de inferência e de uma semântica formal, ou seja, não se pode medir a satisfatibilidade dos requisitos funcionais sobre um domínio conceitual. Durante a fase de elicitação de requisitos, os modelos de classes são genéricos, abstraindo boa parte da especificação de um sistema.

1 Analista de Sistemas da Cia de Ferro Ligas da Bahia, pós-graduado em engenharia de Software pela Universidade Salvador (Unifacs) em maio de 2006.

2 Professor de cursos de extensão, graduação e pós-graduação na Universidade Salvador (Unifacs), Faculdade de Tecnologia Empresarial (FTE) e de graduação nas Faculdades Jorge Amado.

3 Professora da disciplina de gestão do conhecimento no curso de pós-graduação em engenharia de software na Universidade Salvador (Unifacs).

O presente artigo propõe a aplicação do uso da ontologia na construção de modelos MDA, com o objetivo de formalizar os requisitos de aplicação, reduzindo o tempo de manutenções em modelos durante a especificação conceitual de um domínio de aplicação. Para isso, a estrutura textual é composta por cinco partes: A parte inicial refere-se aos problemas encontrados no uso da MDA, ao objetivo do trabalho proposto e à organização do estudo. A segunda fornece uma visão geral dos seus conceitos e níveis de abstrações propostos. Na terceira parte, há uma breve apresentação do que é uma ontologia e sua linguagem de representação. O quarto item constitui-se da apresentação de um modelo MDA construído a partir de uma ontologia. As conclusões e considerações finais encontram-se na quinta e última parte deste artigo.

2 MODELOS E NÍVEIS DE ABSTRAÇÕES EM MDA

A MDA, arquitetura orientada a modelos, propõe a criação de modelos em diferentes níveis de abstração separando os interesses de implementação de uma arquitetura específica do modelo conceitual de uma aplicação. A MDA é baseada na arquitetura MOF que conceitua quatro níveis de modelos: M3, M2, M1, e M0.

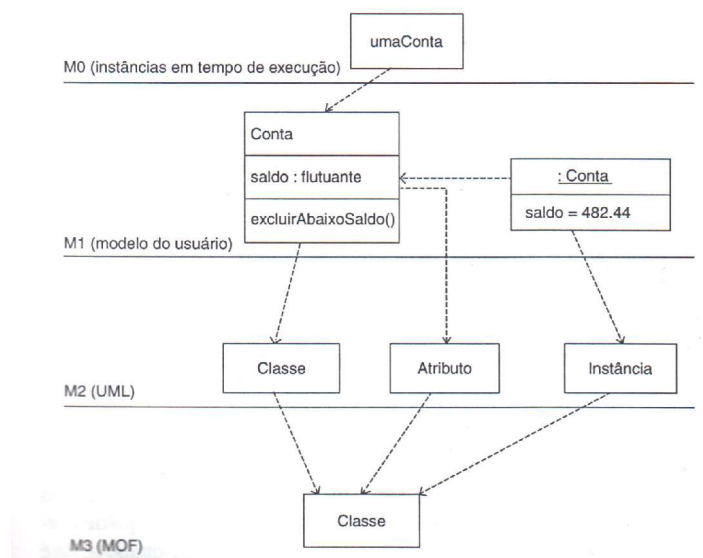


Figura 1 – Exemplo de níveis de modelos MDA (MELLOR, 2005, p. 48)

A figura acima mostra um exemplo do uso de metaníveis em MDA, umaConta (M0) é um exemplo da classe “conta” (M1) que implementou a classe “Classe” (M2), por sua vez, sendo uma instância da classe “Classe” em MOF (M3).

Cada metamodelo proposto, deve ter no mínimo uma implementação onde sua realização pode ter um ou mais modelos. Um modelo de diagramas UML por exemplo, é capturado por um metamodelo UML, que descreve como modelos UML podem ser estruturados, os elementos que eles contêm, e as propriedades de uma plataforma particular. Os modelos poderão relacionar-se usando mapeamentos sendo estes feitos automaticamente ou manualmente.

Em um mapeamento existem regras que são chamadas regras de mapeamento, as quais definem basicamente a aceitação de um ou mais modelos - que são as origens - e um modelo de destino a ser gerado. Essas regras são escritas em nível de metamodelos e aplicáveis a todos os modelos de origem que obedecem às especificações de seu metamodelo. Para que seja possível a utilização de mapeamentos em modelos UML é necessário importar o modelo em uma ferramenta MDA, através do arquivo XMI (XML Metadata Interchange), que é um artefato gerado a partir das linguagens XML (Extensible Markup Language), UML e MOF que utiliza DTD (Definição de Tipo de Documento), nos

quais são descritas regras de sintaxe dos elementos usados no modelo MDA. A declaração de um DTD pode ser feita interna ou externamente ao documento, em que são definidas e ordenadas através de tags.

A Figura 2 mostra um modelo UML no formato XMI, seguindo a padronização definida pela OMG. O padrão combina os benefícios da XML para definição, validação e compartilhamento de formatos de documentos com os benefícios da linguagem de modelagem visual UML para especificação, visualização, construção e documentação. O principal objetivo da XMI é facilitar o intercâmbio de metadados, entre ferramentas de modelagem baseadas na linguagem UML e MOF especificadas pela OMG.

```

<UML:Class xmi.id = 'a45' name = 'docente' isSpecification = 'false' isRoot = 'false'
  isLeaf = 'false' isAbstract = 'false' isActive = 'false'
  <UML:GeneralizableElement.generalization>
    <UML:Generalization xmi.idref = 'a46' />
  </UML:GeneralizableElement.generalization>
  <UML:Classifier.feature>
    <UML:Attribute xmi.id = 'a47' name = 'dataAdmissao' visibility = 'private'
      isSpecification = 'false'>
      <UML:StructuralFeature.multiplicity>
        <UML:Multiplicity xmi.id = 'a48'>
          <UML:Multiplicity.range>
            <UML:MultiplicityRange xmi.id = 'a49' lower = '0' upper = '-1' />
          </UML:Multiplicity.range>
        </UML:Multiplicity>
      </UML:StructuralFeature.multiplicity>
      <UML:StructuralFeature.type>
        <UML:Class xmi.idref = 'a24' />
      </UML:StructuralFeature.type>
    </UML:Attribute>
  </UML:Classifier.feature>
</UML:Class>

```

Figura 2 – Trecho do arquivo XMI gerado usando o Protégé

3 VISÃO GERAL SOBRE ONTOLOGIA

Assim como XMI, a linguagem OWL (Ontology Web Language) é uma linguagem XML. É possível expressar a estrutura dos conceitos, relacionamentos e descrever características especiais sobre estes, através de axiomas lógicos para formar uma ontologia.

Segundo Guizzard (2000), em ciência da computação, o termo ontologia se refere a um artefato de engenharia, constituído de um vocabulário de termos organizados em uma taxonomia, suas definições e um conjunto de axiomas formais usados para criar novas relações e para restringir as suas interpretações segundo um sentido pretendido. Ainda na área da ciência da computação, uma ontologia pode ser utilizada para criar um vocabulário. Falbo (2002) cita que, neste caso, a ontologia pode ser vista como um modelo para domínio de aplicações, sendo usada basicamente para especificá-los e desenvolvê-los aumentando o reuso.

Seja qual for a categoria, uma ontologia é formada por conjuntos de conceitos, relações, propriedades, axiomas e instâncias. Um conceito pode ser abstrato ou concreto, elementar ou composto, real ou fictício. Em um conceito podem existir taxonomias. O conceito homem, por exemplo, pode ser um subconceito do conceito pessoa. Podem também existir relações entre conceitos. Por exemplo: entre os conceitos de pessoa e carro, o relacionamento “ser- dono”. As propriedades representam os relacionamentos entre os conceitos. Os axiomas são regras ou afirmações da verdade sobre conceitos. Um exemplo de axioma é afirmar que toda pessoa tem uma mãe. Existem também, na base de conhecimentos, instâncias que são um conhecimento prévio existente na ontologia.

Na base de conhecimentos podem estar inclusos os conhecimentos intencional (TBox) e extencional (ABox). São usados em DL (Description Logic) para realizar o mapeamento da linguagem em OWL, a fim de processar o conhecimento e encontrar informações implícitas numa ontologia com o uso de uma ferramenta de raciocínio.

Busca-se testar o TBox através de um sistema dedutivo a satisfatibilidade, subclassificação, equivalência e disjunção. Com relação ao ABox, é possível checar a consistência, as instâncias, o retorno e a realização. Com estes testes pode-se, por exemplo, checar a hierarquia de classes, suas relações, regras na atribuição de valores aos seus atributos. Enfim, diminui-se bastante o tempo de verificação de modelos.

4 CONSTRUÇÃO DE MODELOS MDA USANDO ONTOLOGIAS

Nesta proposta, a construção de um modelo MDA baseado em ontologia segue a padronização MOF, utilizando no intercâmbio de dados artefatos gerados baseados em XMI *schema* entre os metaníveis. Segundo a especificação da OMG, no metanível M2 é possível concretizar as classes especificadas em M3 (MOF). Dessa forma, um modelo de nível 2 pode ser utilizado para construção de modelos MDA, sendo implementado no metanível M1, criando um modelo independente de plataforma, através de uma ferramenta de construção de diagramas UML, em virtude de todas as ferramentas se basearem no metamodelo UML. A geração de mapeamentos e regras para plataforma específica ocorre: no metanível M1, utilizando ferramentas de construção de modelos MDA; após a construção do modelo conceitual no metanível M1, usando a ferramenta de edição de ontologia Protégé (ABDALLA, 2005).

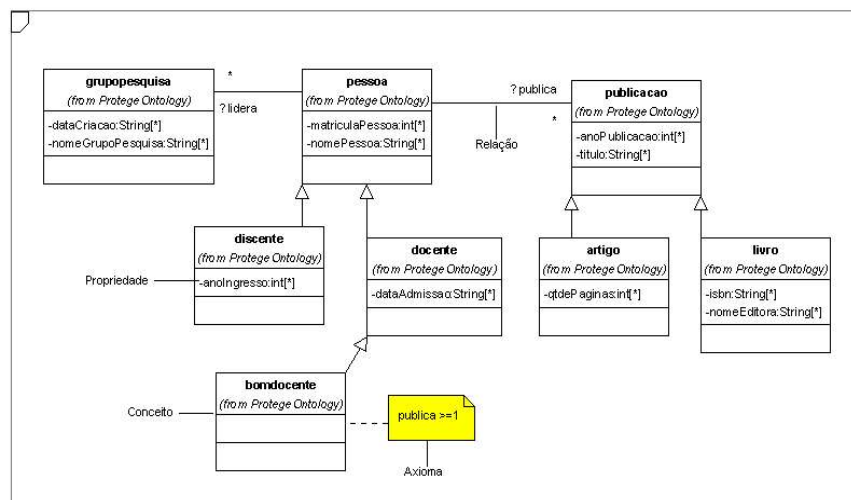


Figura 3 – Diagrama UML da ontologia criada como exemplo

Os conceitos, relações e propriedades identificados na ontologia são definições formais e semânticas. Na ontologia criada na Figura 3, uma pessoa, por exemplo, significa um professor ou um aluno que trabalha em uma universidade, as definições dos conceitos não podem ser ambíguas. O axioma “publica ≥ 1 ”, significa que um bom docente tem mais de uma publicação, seja um artigo ou livro. A ontologia é construída seguindo os requisitos funcionais da aplicação, e seu modelo será verificado quanto à satisfação destes.

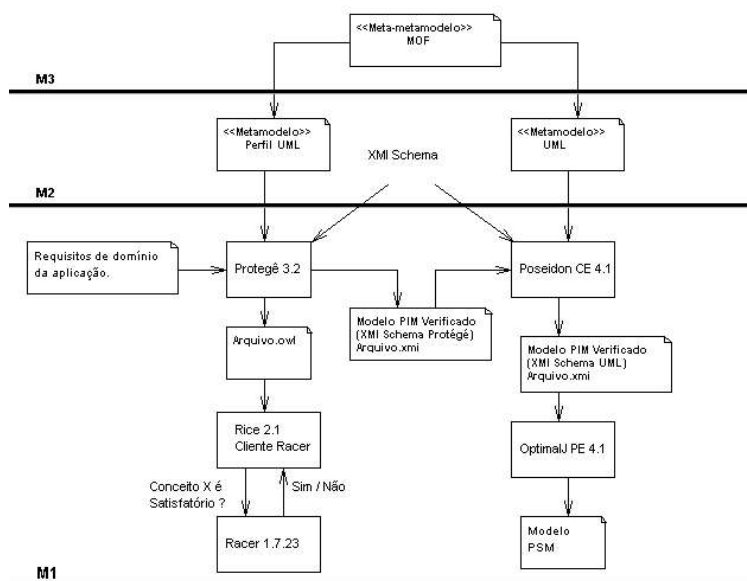


Figura 4 – Arquitetura utilizada para construção de modelos a partir de uma ontologia

A Figura 4 apresenta a arquitetura proposta, começando a partir da construção de uma ontologia para o domínio da aplicação até transformação do modelo formalmente verificado em outro para uma plataforma específica.

Os conceitos, relações e propriedades identificadas na ontologia são definições formais e semânticas. Todo o modelo conceitual da ontologia deverá ser verificado formalmente criando instâncias de classes na ferramenta de ontologia e usando lógica de descrição através da ferramenta Racer (ABDALLA, 2005). Segundo Ribeiro (2005), o Racer (Reasoner for ABoxes and Concept Expressions Renamed) pode realizar um raciocínio automático usando lógica de descrição, realizando o mapeamento da DL (Description Logics) sobre o arquivo OWL gerado a partir da ontologia. Essas inferências podem ser submetidas utilizando o Rice (Racer Interactive Client Environment) (ABDALLA, 2005), que faz o uso de uma interface gráfica para importar arquivos OWL.

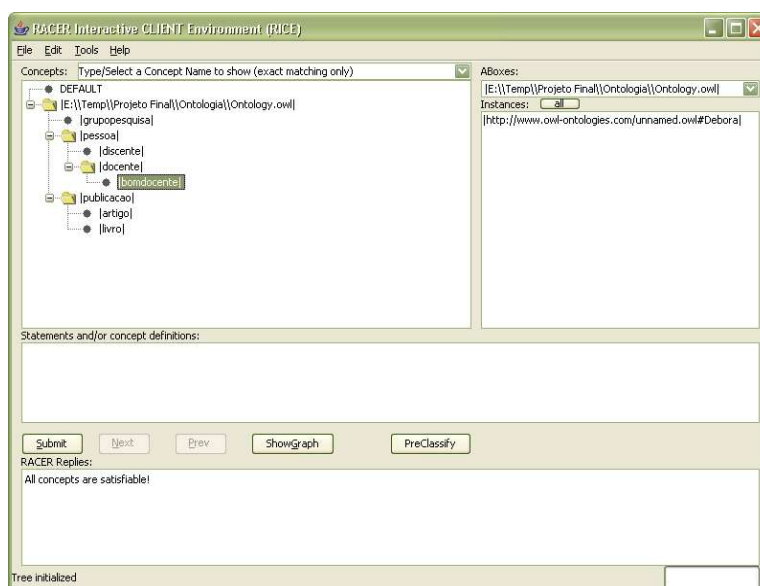


Figura 5 – Verificação formal do domínio da aplicação através do Rice

Na Figura 5 é possível visualizar o resultado da inferência sobre um domínio da aplicação, neste exemplo, todos os conceitos foram satisfeitos. O modelo exemplificado,

será intercambiado para uma ferramenta UML, o que possibilitará a atribuição de tipos, esteriótipos e outros elementos do metamodelo UML e outros perfis disponíveis no metanível M2.

Segundo a especificação da OMG, o padrão de intercâmbio de dados na arquitetura orientada a modelos (MDA) é o XMI, a ferramenta UML utilizada para gerar um modelo independente de plataforma (PIM), a Poseidon (POSEIDON, 2006), foi utilizada para importar o arquivo XMI gerado pelo protégé e criar um modelo da aplicação, no nível M1, isso se deve ao fato das diferenças encontradas nos arquivos XMI exportados pelas ferramentas Protégé e Poseidon, sendo o arquivo XMI gerado pelo Poseidon aceito para importação pela ferramenta MDA utilizada. A OptimalJ (OPTIMA1J, 2006), foi utilizada como ferramenta de transformações de modelos PIM em PSM.

Nesta proposta foi possível identificar dois exemplos de transformações de modelos: de PIM para PIM, sendo utilizado neste caso um mapeamento do modelo OWL para UML, e a transformação de modelos PIM para banco de dados relacional (PSM) usando a ferramenta OptimalJ. Na Figura 6 é possível identificar as tabelas com chaves primárias e secundárias e tipos de dados que são específicos para banco de dados.

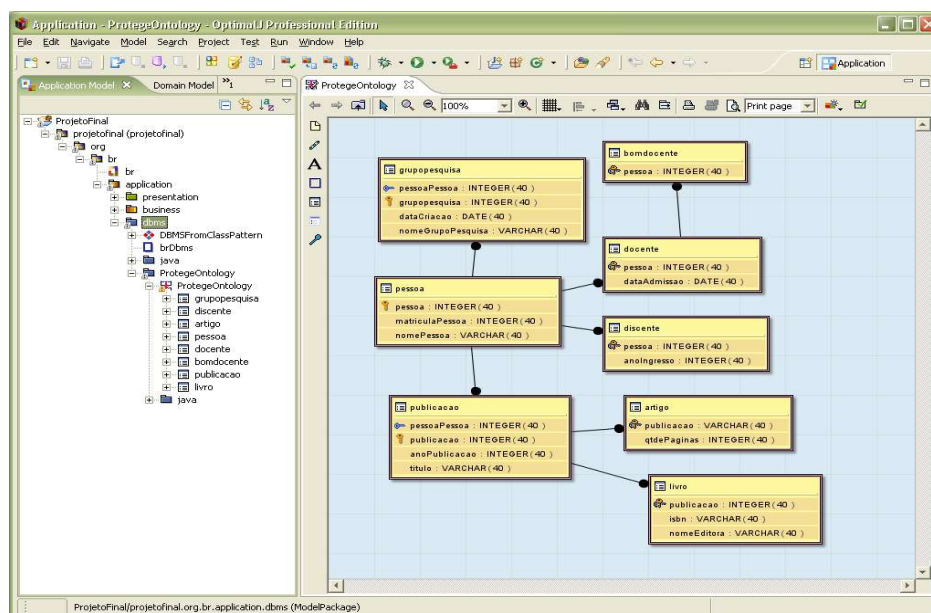


Figura 6 – Modelo específico para banco de dados gerado a partir da ontologia de domínio

5 CONCLUSÃO

A definição formal de conceitos para um domínio de aplicação é uma alternativa válida para evitar o aumento de manutenções em modelos. Durante o desenvolvimento deste trabalho pôde-se perceber uma grande quantidade de ferramentas para conceber transformações de modelos, e muitas delas apoiadas pela iniciativa de pesquisadores que as mantêm livres para uso comercial ou pessoal. Com o uso da arquitetura proposta, a ausência de definição semântica para classes, atributos e seus relacionamentos da linguagem UML puderam ser contornados. A partir da criação de um mecanismo de validação de modelos de domínio de aplicação usando lógica descritiva, pode-se garantir que as especificações contidas são verdadeiras por terem sido usadas provas matemáticas para sua satisfatibilidade.

Uma boa perspectiva com vistas a um trabalho futuro é utilizar algum modelo de processo de software para medir a qualidade dos artefatos produzidos, benefícios e comparar o uso desta arquitetura em relação a outras existentes.

REFERÊNCIAS

- ABDALLA et al. **Módulo de Gestão do Conhecimento**. Universidade Federal da Bahia, <http://twiki.im.ufba.br/bin/view/Residencia/Cursos>, abr. 2005.
- FALBO, R. A.; GUIZZARDI, G.; DUARTE, K. C. **An Ontological Approach to Domain Engineering**. Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering, SEKE'2002, p. 351- 358, Ischia, Italy, 2002.
- FREITAS, F. Ontologias e Websemântica. IV ENIA – Encontro Nacional de Inteligência Artificial, Campinas, Minicurso. **In:** Anais do XXIII Congresso da Sociedade Brasileira de Computação, 52 p, 2003.
- GUIZZARD, Giancarlo. **Uma abordagem metodológica de desenvolvimento para e com reuso, baseada em ontologias formais de domínio**. Programa de Mestrado em Informática – Universidade Federal do Espírito Santo, Vitória, <http://wwwhome.cs.utwente.nl/~guizzard/MSc/>, jul. 2000.
- MELLOR, J. Stephen et al. **MDA Destilada: Princípios da Arquitetura Orientada por Modelos**. Ciência Moderna, 2003. p. 15- 169.
- OPTIMA1J. **Model- Driven Development for Java**. <http://www.compuware.com/products/optimalj/>, jul. 2006.
- POSEIDON. **Poseidon for UML 4.2**. <http://gentleware.com/index.php>, jul. 2006.
- RIBEIRO, M. **Raciocínio em Lógica de Descrições**. Universidade Federal da Bahia, <http://twiki.im.ufba.br/bin/view/Residencia/Cursos>, abr. 2005.