Desenvolvimento Dirigido por Modelos: Ferramentas

Eduardo Figueiredo

http://www.dcc.ufmg.br/~figueiredo

Existe MDD na prática?

- Poucos sistemas ainda são desenvolvidos usando a filosofia MDD
  - A expectativa é de aumentar a adoção nos próximos anos a medida que MDD amadurece
- Desde que MDD foi proposta, vários ferramentas afirmam apoiar MDD
  - Na verdade, as ferramentas apóiam alguns aspectos de MDD

Dificuldades de Automação

- Ferramentas MDD devem considerar casos particulares
- Ambiente de execução inclui
  - Plataforma de programação (ex. J2EE)
  - Bibliotecas específicas da empresa ou do domínio
  - Bibliotecas específicas de interface com o usuário, etc.

Tipos de Ferramentas

- Transformação de PIM para PSM
- Transformação de PSM para Código
- Transformação de PIM para Código
- Ferramentas para definir transformações
  - Outras

Ferramentas: PIM para PSM

- Tipo de ferramenta que recebe PIM de alto nível e transforma em um ou mais PSM
  - Ferramentas deste tipo quase não existem

PSM para Código

- As ferramentas mais conhecidas para suporte a MDD
  - Recebem um ou vários modelos como entrada
  - Geram código em uma determinada linguagem (modelo de código)
- Algumas ferramentas mantém a consistência entre modelos e código

PIM para Código

- Tipo de ferramenta que suporta
  - Transformação de PIM para PSM
  - Transformação de PSM para Código
- Os usuários podem ver somente a transformação PIM para Código
- UML é geralmente usada como uma linguagem para PIM
  - Comportamento nem sempre é expresso em UML (manual ou OCL)

Exemplos de Ferramentas

- xUML-Compiler
- IBM Rational Rhapsody
- AndroMDA

xUML Compiler

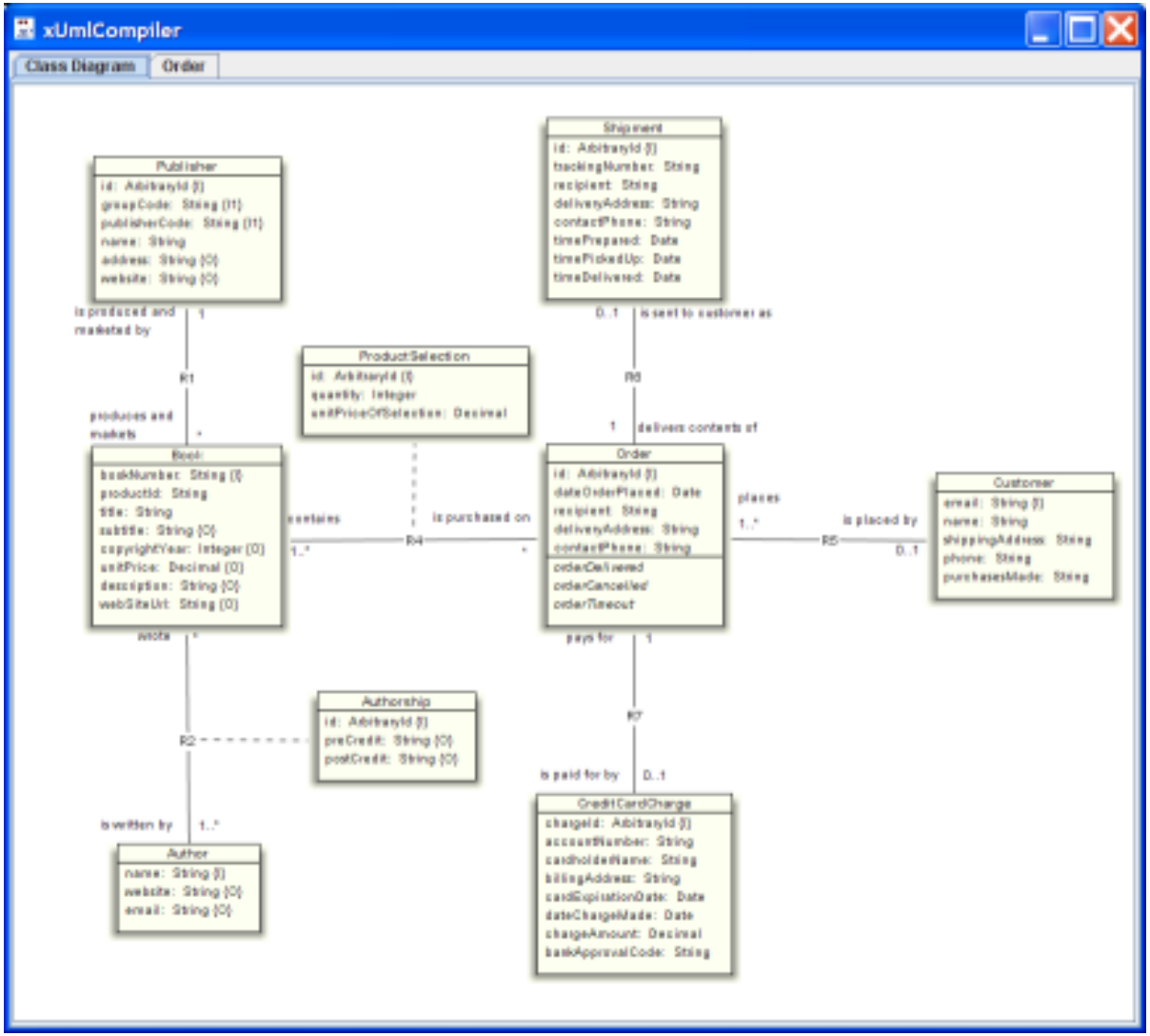
xUML Compiler

- É um projeto *open source*
- Visão gráfica
  - Diagrama de Classes
  - Diagrama de Estados
- Gera código executável e testável (Java)
  - Gera documentação parcial (Java Doc)

Linguagem de Ações

- É escrita em sintaxe Java
  - Linguagem de Ações é tão poderosa e expressiva quanto Java
- Elementos da Linguagem de Ações são inseridos nas classes

Diagrama de Classes



IBM Rational Rhapsody

Rhapsody

- Ambiente de desenvolvimento para MDD
  - Suporta C, C++ e Java
- Funcionalidades principais
  - Permite analisar e verificar rastreabilidade entre requisitos
  - Validar as funcionalidades antecipadamente no desenvolvimento
  - Conduzir testes nos modelos

Família Rhapsody

- A família de software Rhapsody inclui diversos produtos, entre eles:
  - Rational Rhapsody Architect for Software: IDE para desenvolvimento gráfico de aplicações Java, C++ e C# usando UML
  - Rational Rhapsody Developer: apoia reutilização de código e engenharia reversa além de sincronizar modelos e código

Demonstração

- Alguns vídeos de demonstração estão disponíveis
  - Vide “IBM Rational Rhapsody product demos” no website da ferramenta

http://www.ibm.com/developerworks/rational/products/rhapsody/

AndroMDA

AndroMDA

- Framework para geração de código
  - Adere aos principais conceitos de MDD
- Modelos UML são transformados em componentes implantáveis
- Gera código compatível com tecnologias atuais
  - J2EE, Spring, Struts, JSF, Spring and Hibernate, etc.

Modelos de Entrada (UML)

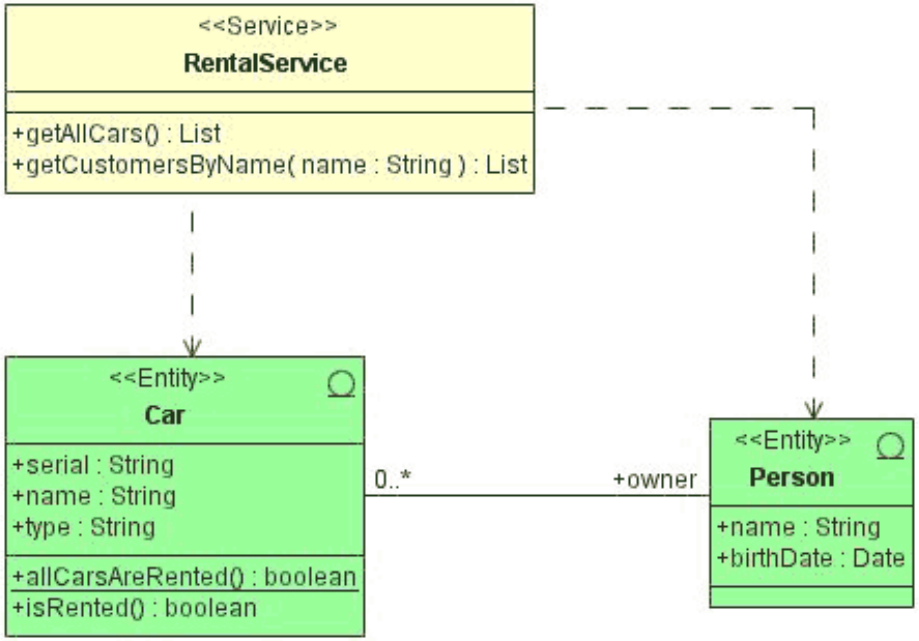
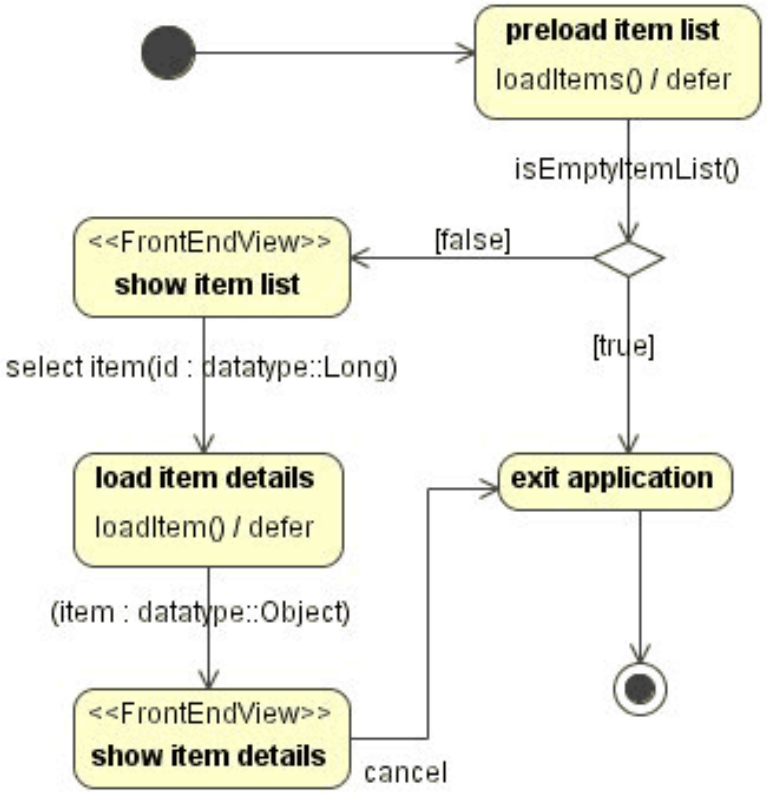


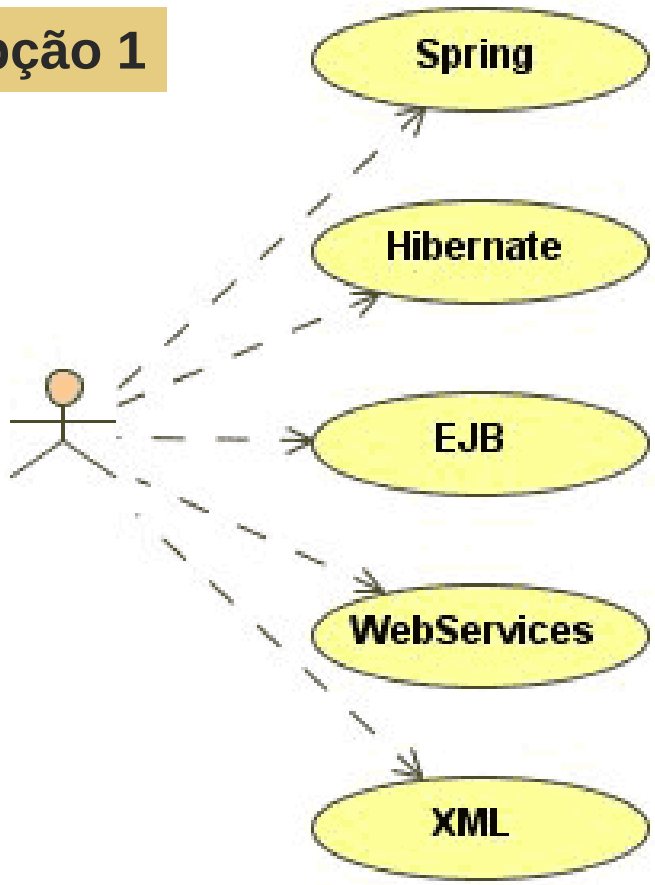
Diagrama de Classes

Diagrama de Atividades

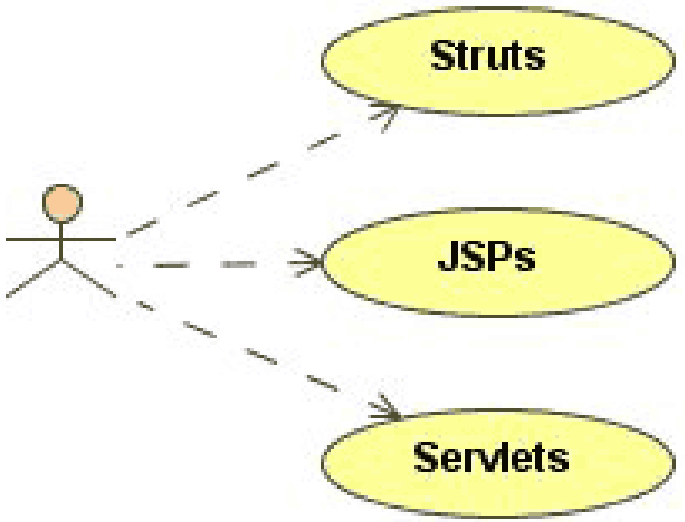


Tecnologia de Implementação

Opção 1



Opção 2



Bibliografia da Aula

- A. Kleppe, J. Warmer, W. Bast. **MDA Explained: The Model Driven Architecture: Practice and Promise**. Addison-Wesley, 2003.
  - Capítulo 2
- Website das ferramentas
  - http://code.google.com/p/xuml-compiler/
  - http://www.ibm.com/developerworks/rational/products/rhapsody/
  - http://www.andromda.org/