

SBCARS 2007

Transformando Modelos da MDA com o apoio de Componentes de Software



Autores:

Marco Antonio Pereira
Antonio Francisco do Prado
Mauro Biajiz
Valdirene Fontanette
Daniel Lucrédio

Estrutura Geral

1. Contextualização
2. MDA
3. Transformando Modelos
4. Componentes de Transformação na MVCASE
5. Exemplo de Uso

[Contextualização]

Contexto



Software

[Contextualização]

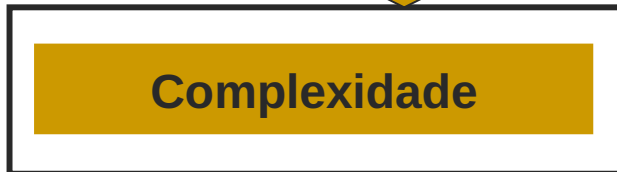
Contexto



Software



Problema



Qualidade

Reusabilidade



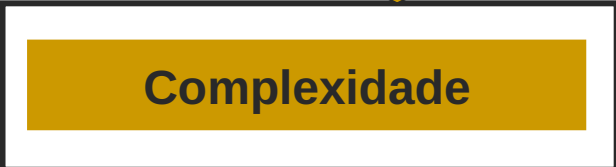
Contextualização

Contexto



Software

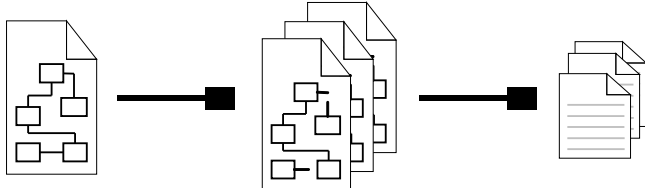
Problema



Reusabilidade

Solução

Model-Driven Development (MDD)



Modelo Independente de Plataforma

Modelo Dependente De Plataforma

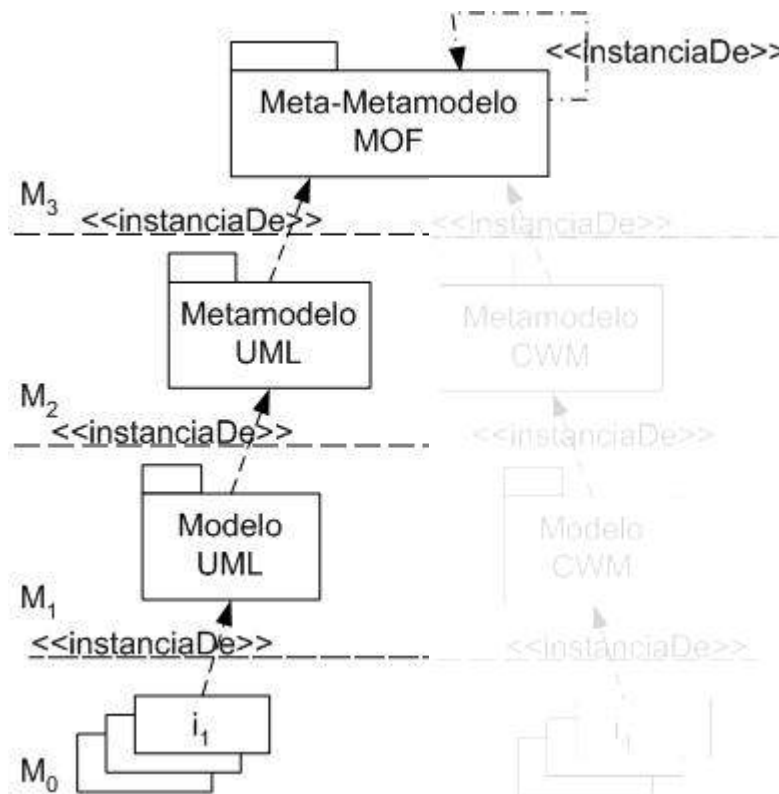
Códigos

Model Driven Architecture (MDA)

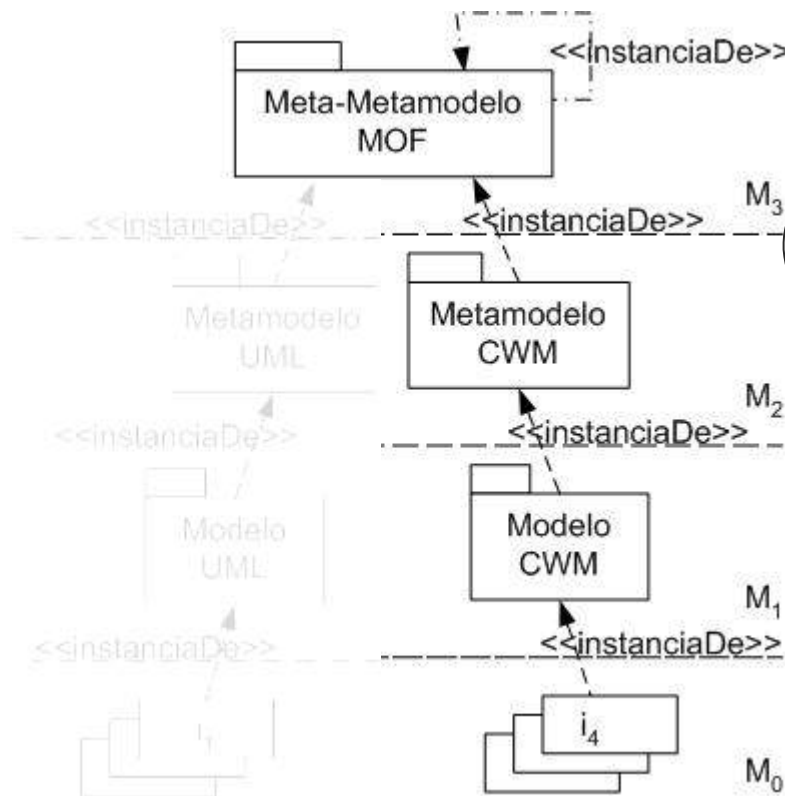
- Suporte a arquitetura por meio de linguagens padronizadas pelo OMG:
 - Meta Object Facility (MOF)
 - Unified Modeling Language (UML)
 - Common Warehouse Metamodel (CWM)
 - XML Metadata Interchange (XMI)
 - Entre outras

[Arquitetura da MDA]

Suporte a Modelagem de Objetos



[Arquitetura



Suporte a Modelagem Data Warehouse

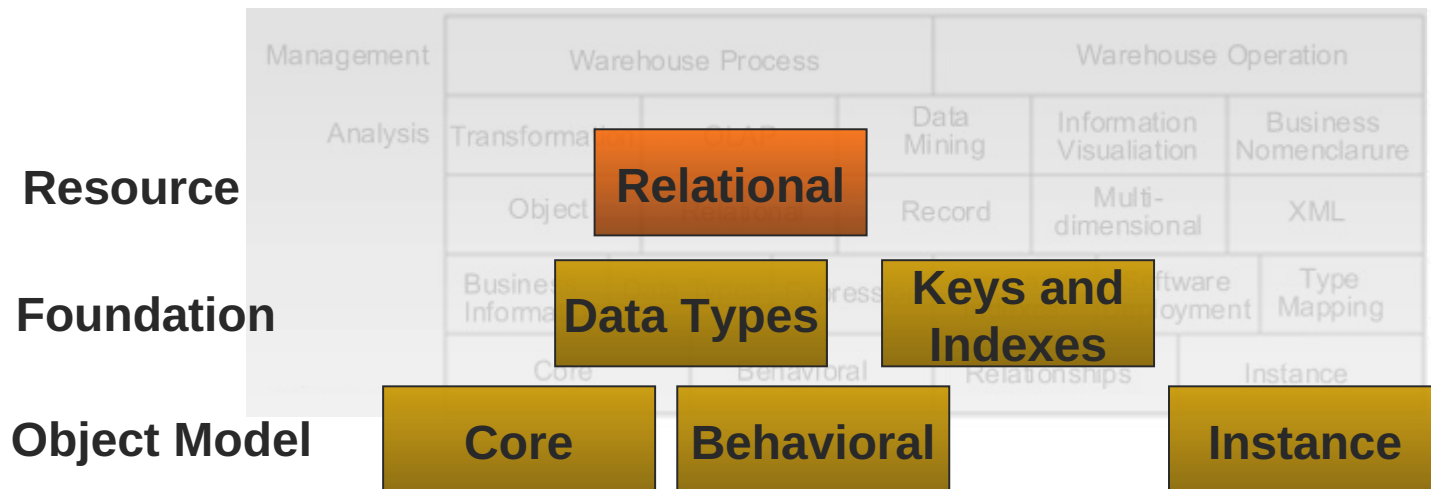
Metamodelo CWM

Management	Warehouse Process			Warehouse Operation		
Analysis	Transformation	OLAP	Data Mining	Information Visualiation	Business Nomenclarure	
Resource	Object	Relational	Record	Multi-dimensional	XML	
Foundation	Business Information	Data Types	Expression	Keys and Indexes	Software Deployment	Type Mapping
Object Model	Core		Behavioral	Relationships		Instance

Metamodelo CWM

Pacote Relational e suas dependências

- O pacote Relational suporta as características Orientadas a Objetos do Padrão SQL3
- O pacote Relational depende dos pacotes Data Types, Keys and Indexes, Core, Behavioral e Instance do CWM

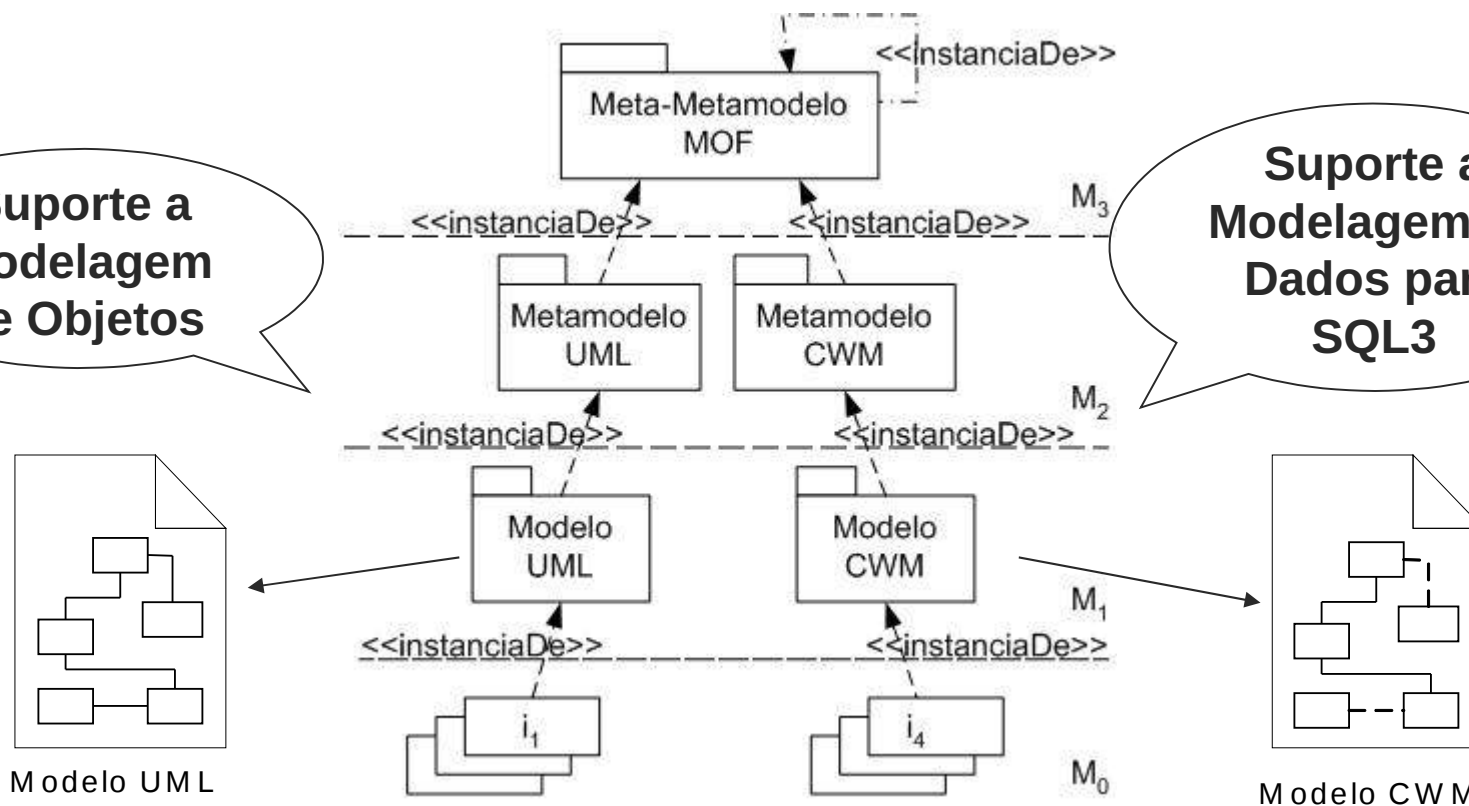


“O Modelo Relacional pode ser estendido sem comprometer seus princípios fundamentais”
Darwen Date - The third manifesto (1995)

Arquitetura

Suporte a Modelagem de Objetos

Suporte a Modelagem de Dados para SQL3

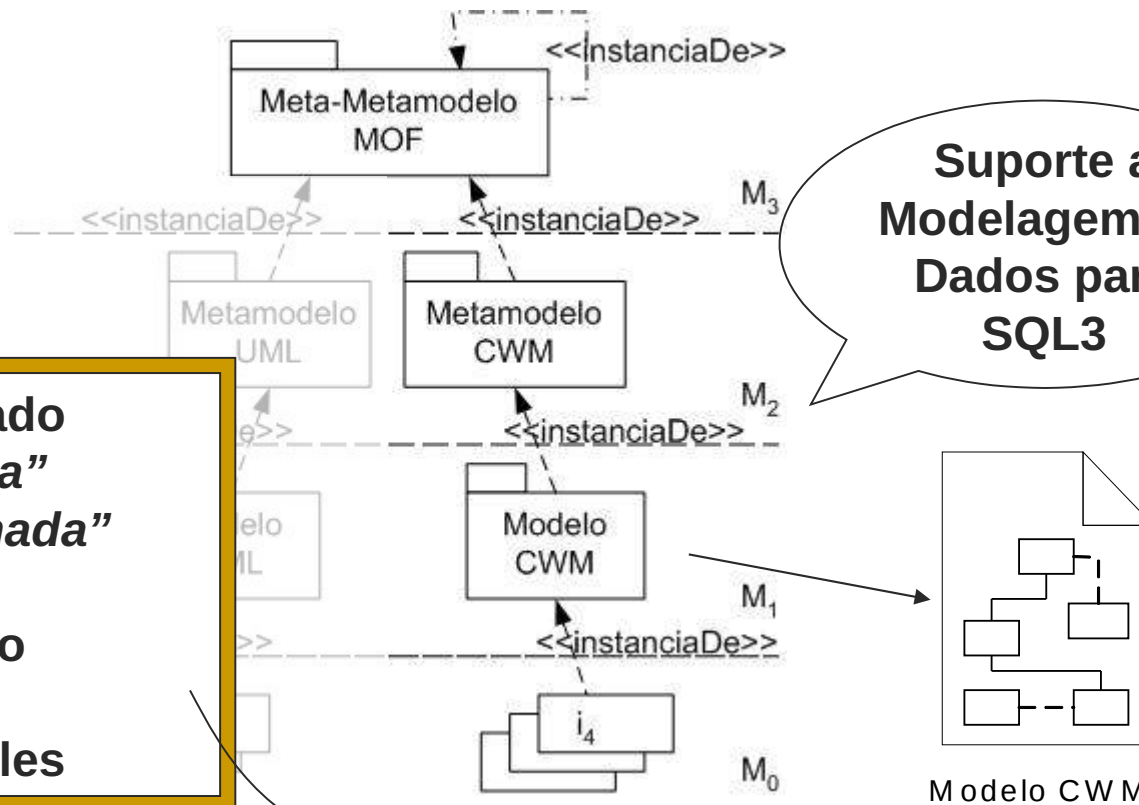


Arquitetura

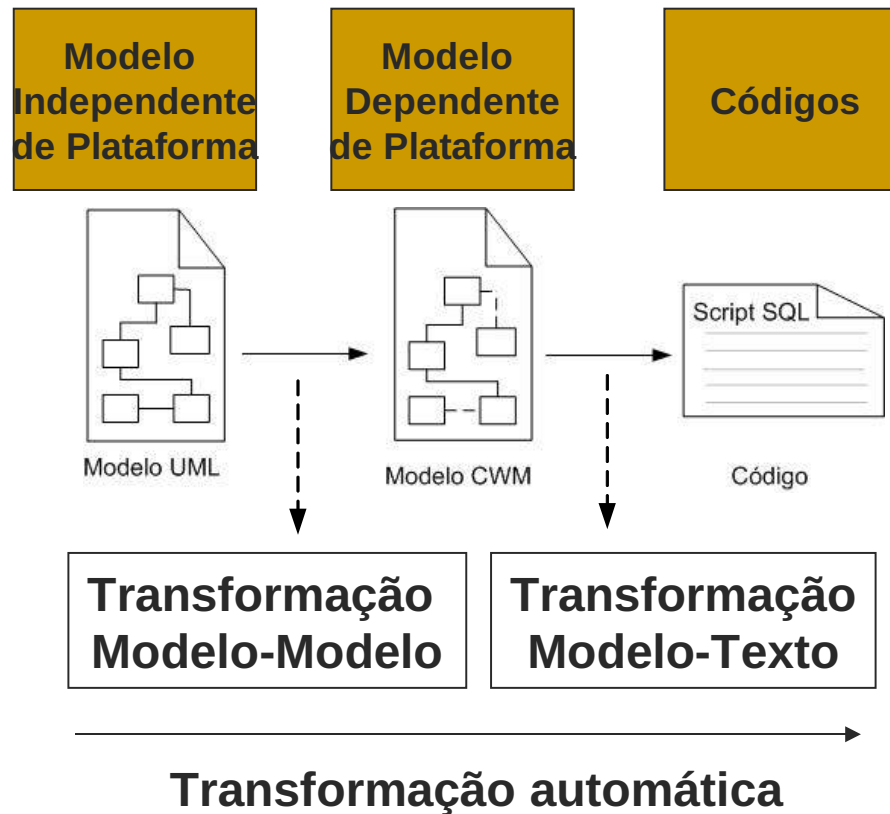
Suporte a Modelagem de Objetos

Tipo Estruturado
Tabela "Tipada"
Tabela "Aninhada"
Varray
Tipo Complexo
Referência
Herança Simples

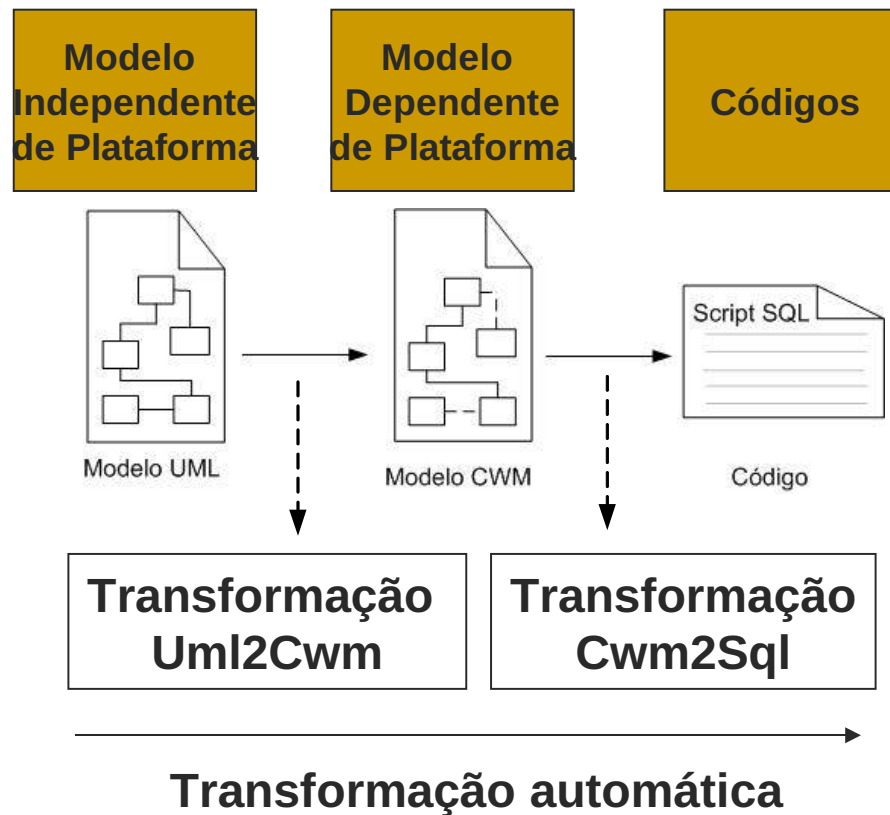
Suporte a Modelagem de Dados para SQL3



[Transformando Modelos]



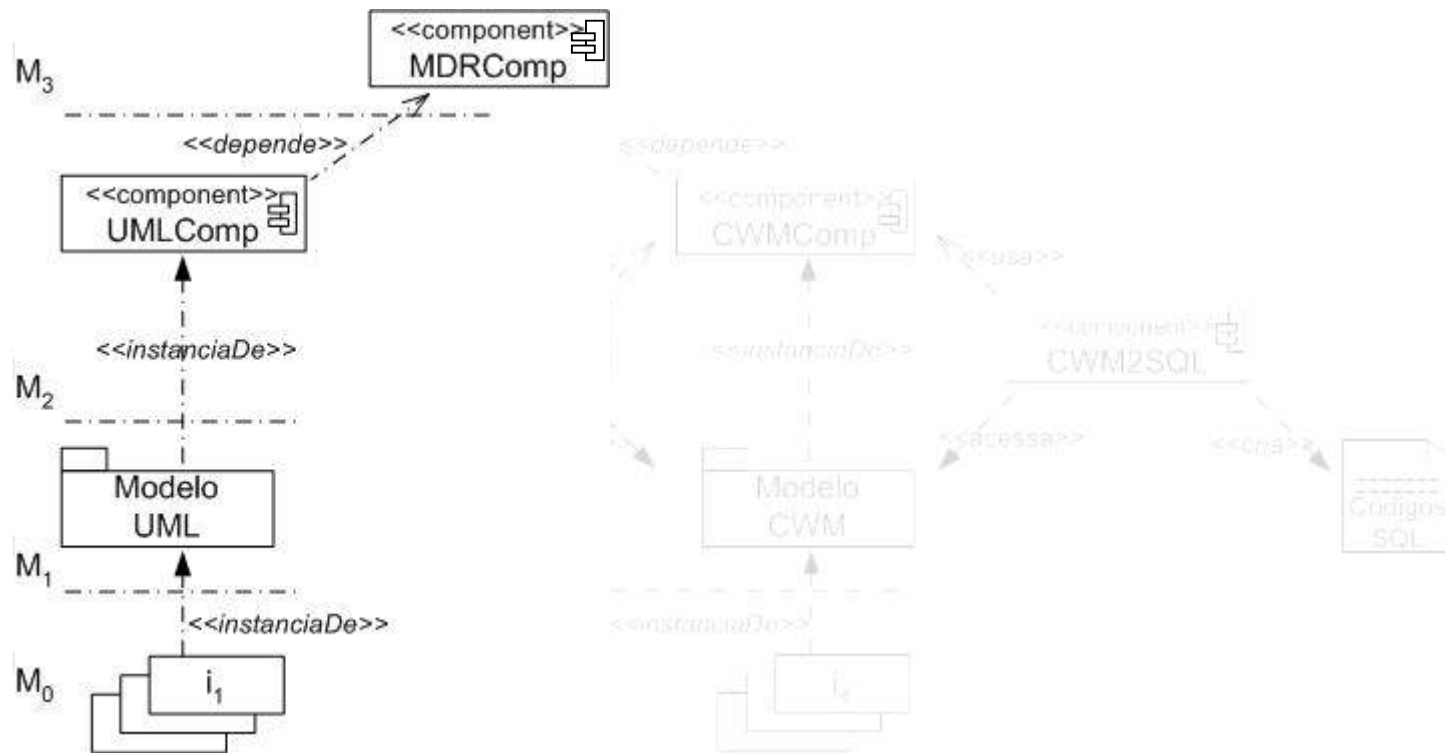
[Transformação de Modelos]



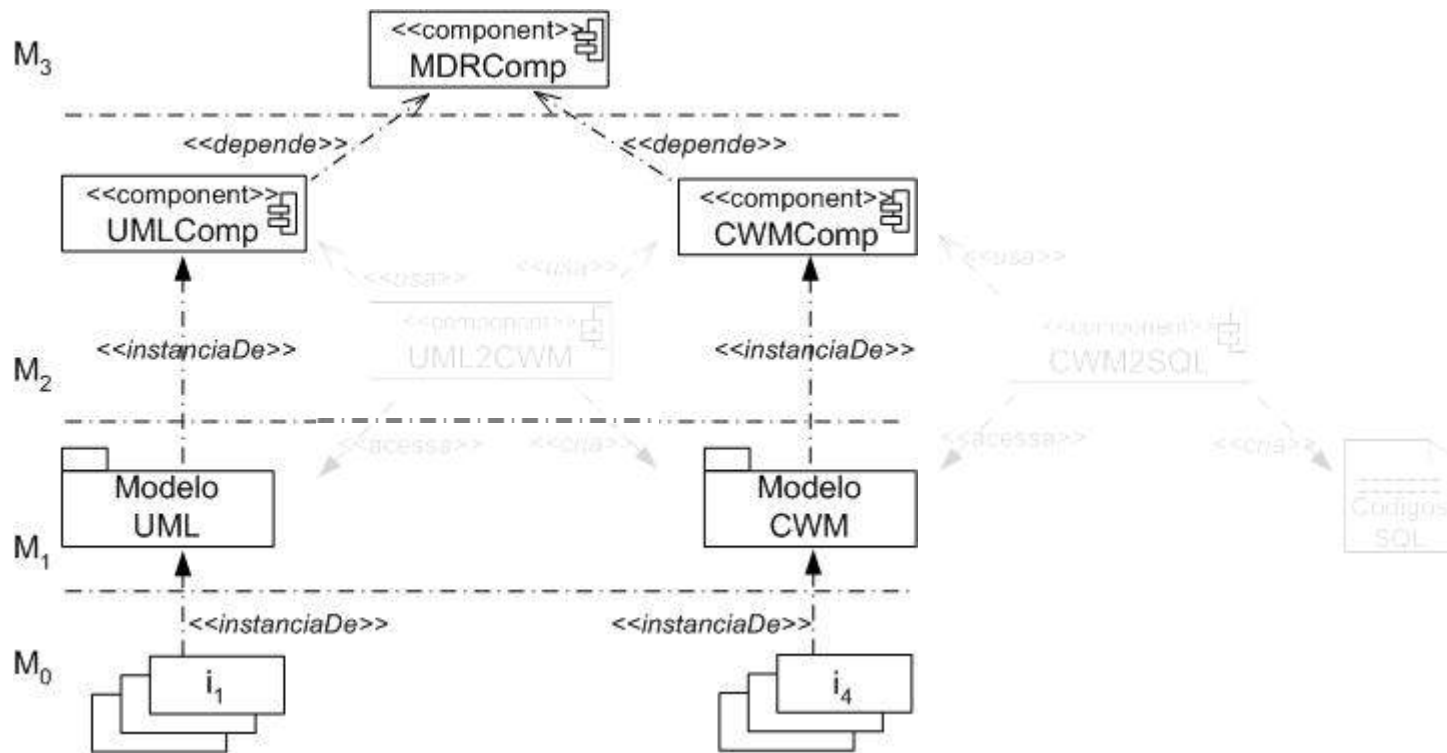
Operacionalização dos conceitos da MDA

- MVCASE
 - É uma ferramenta que suporta a modelagem de sistemas de software com a notação UML.
 - Os sistemas podem ser especificados segundo 4 visões:
 - Diagrama de Casos de Uso;
 - Diagrama de Classes;
 - Diagrama de Seqüência ;
 - Diagrama de Componentes.

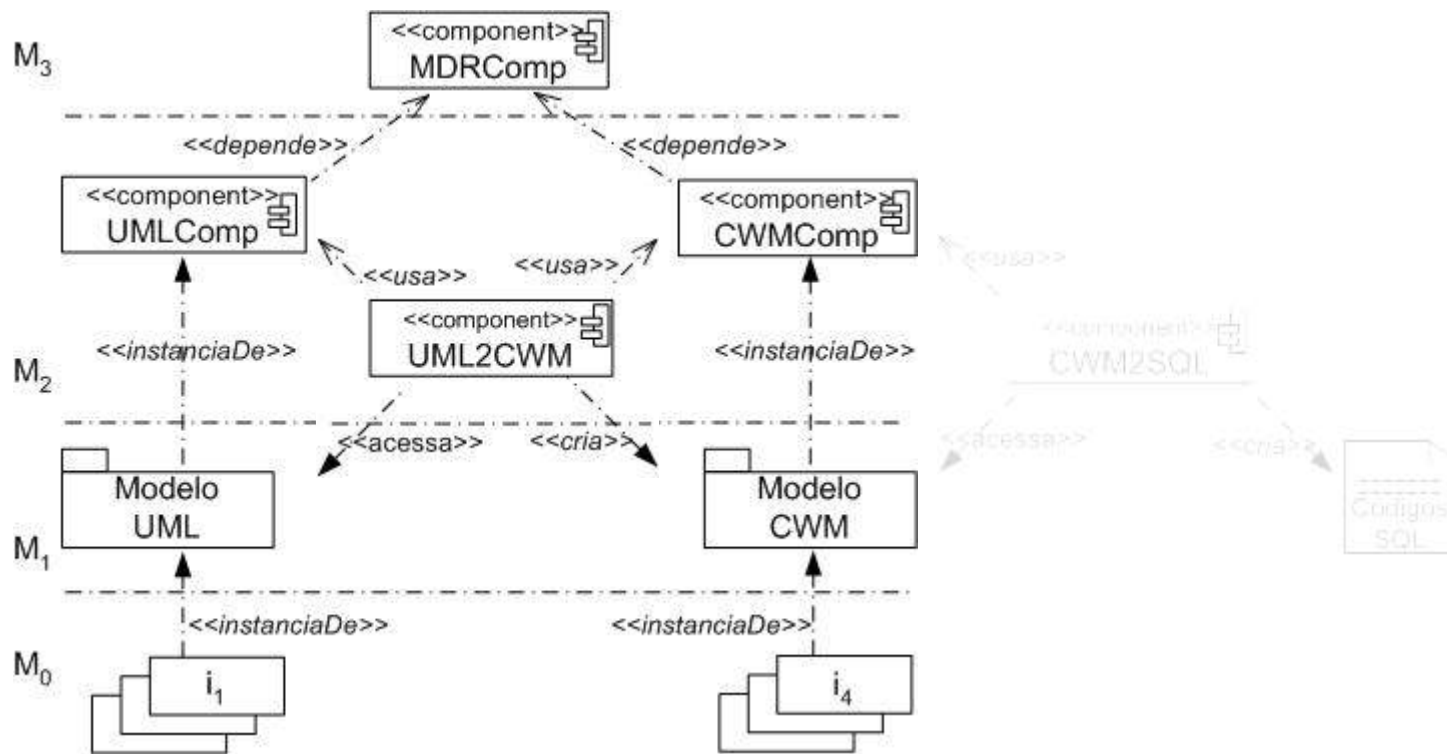
Esboço da Arquitetura Existente na MVCASE



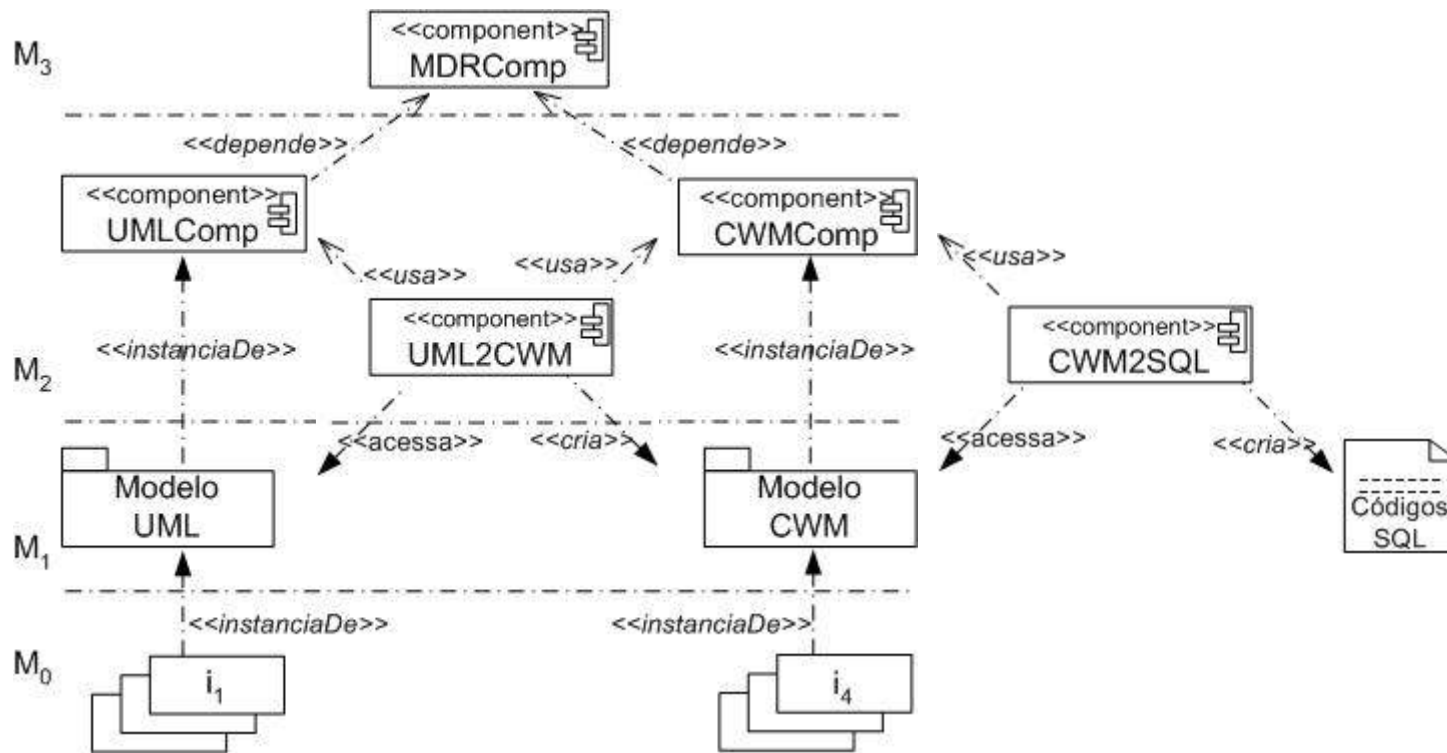
Protótipo da MDA na MVCASE



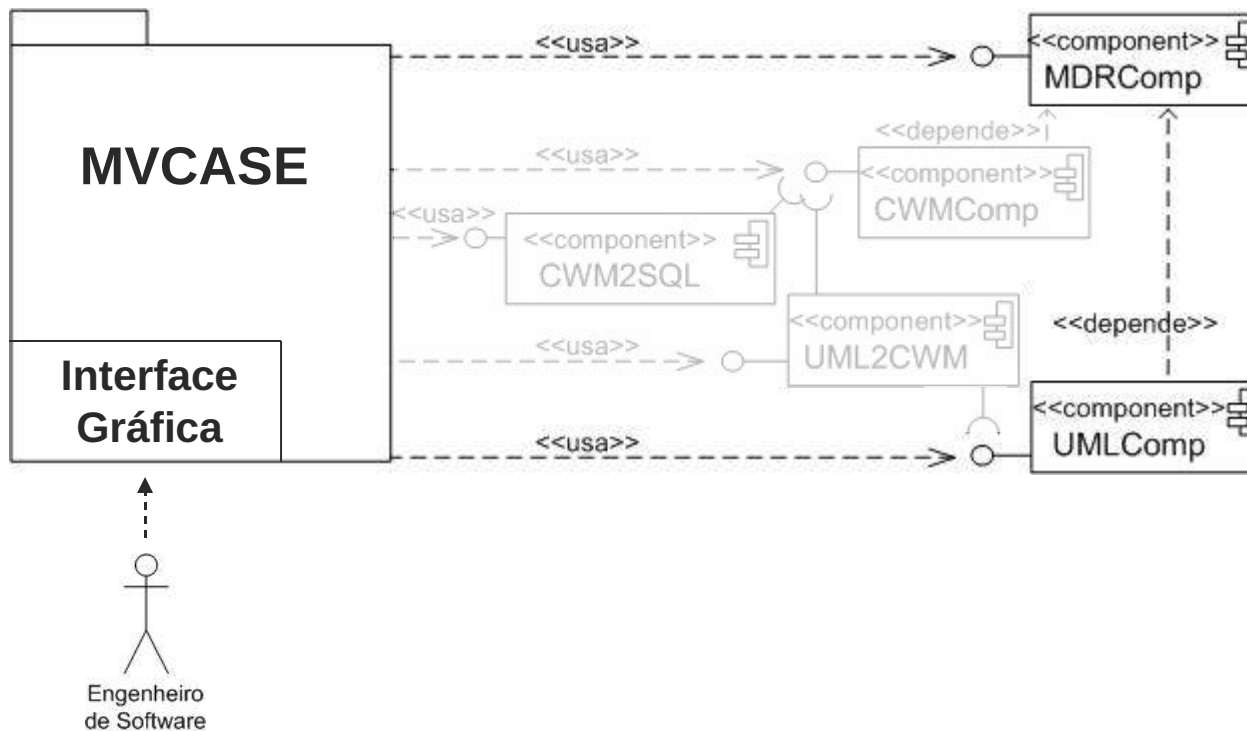
Protótipo da MDA na MVCASE



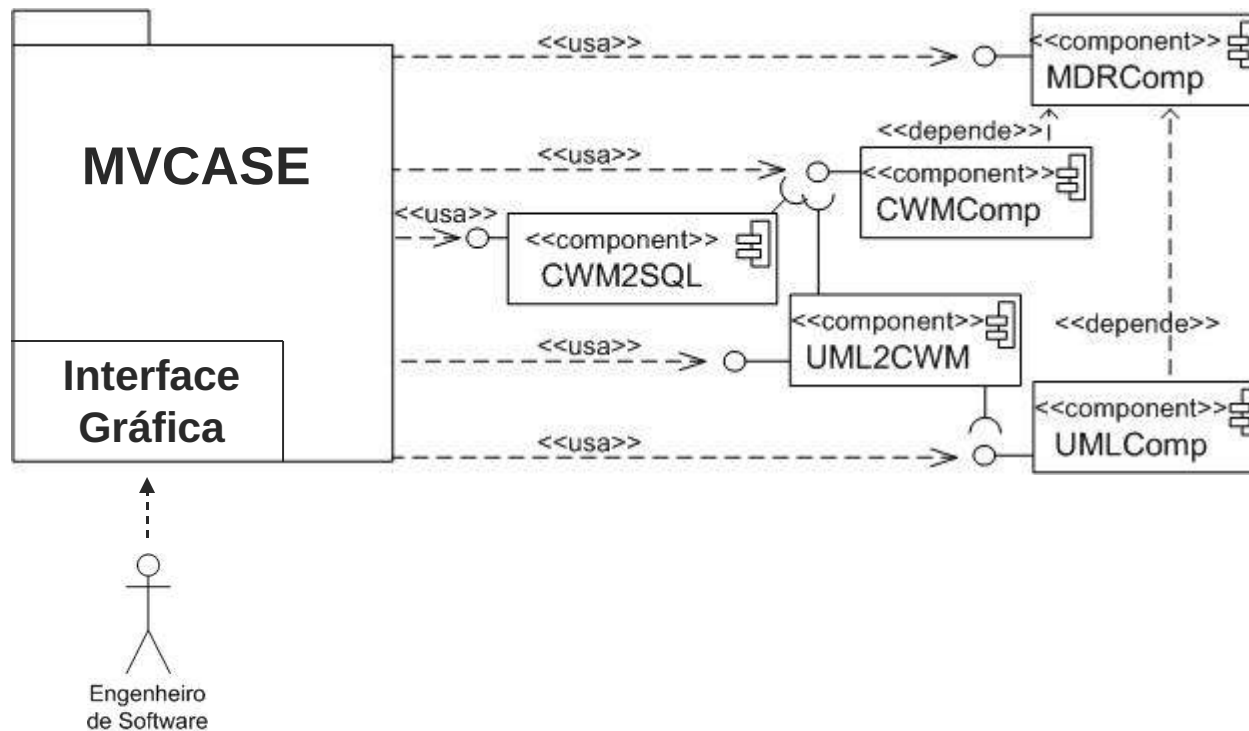
Protótipo da MDA na MVCASE



[Integração dos Componentes]



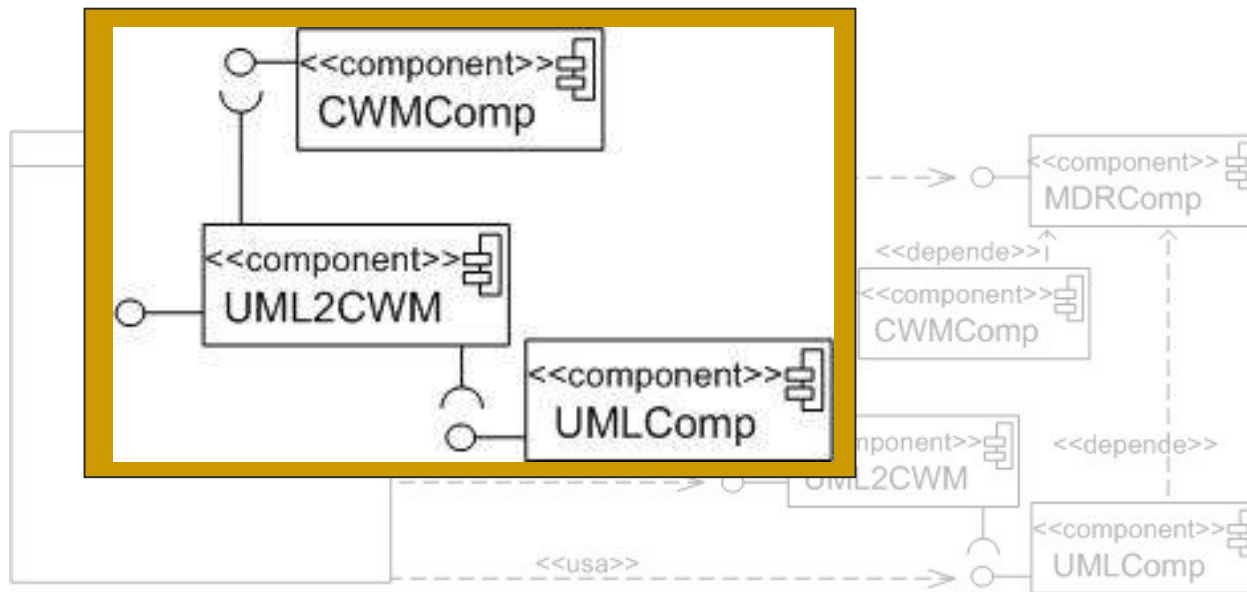
[Integração dos Componentes]



Integração dos Componentes

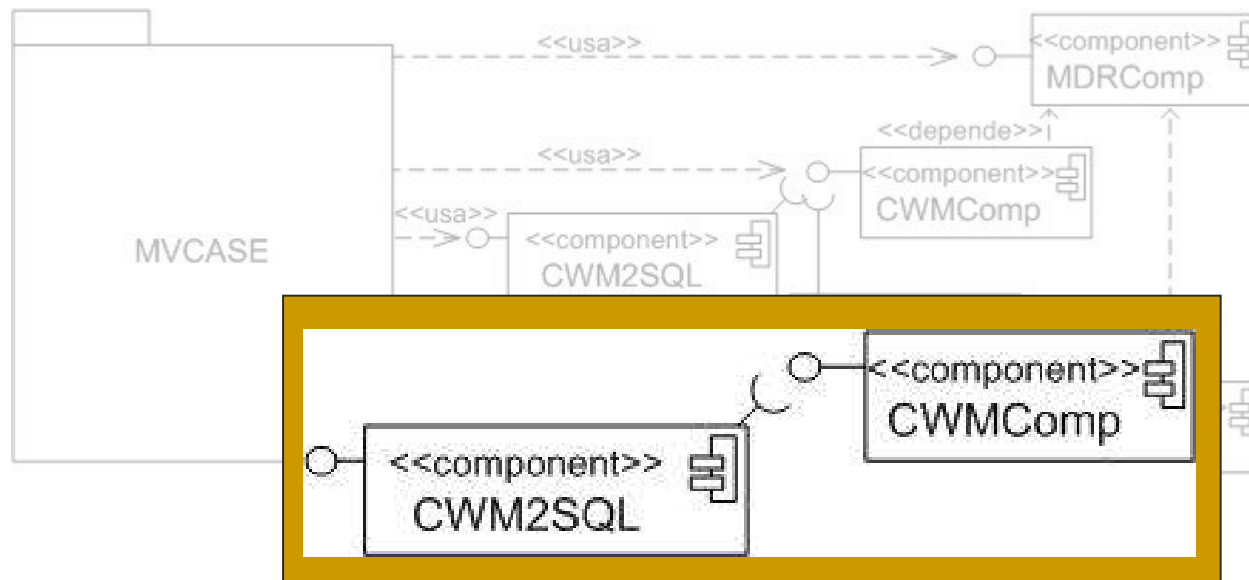
Componentes de Transformação

Componente UML2CWM



Integração dos Componentes

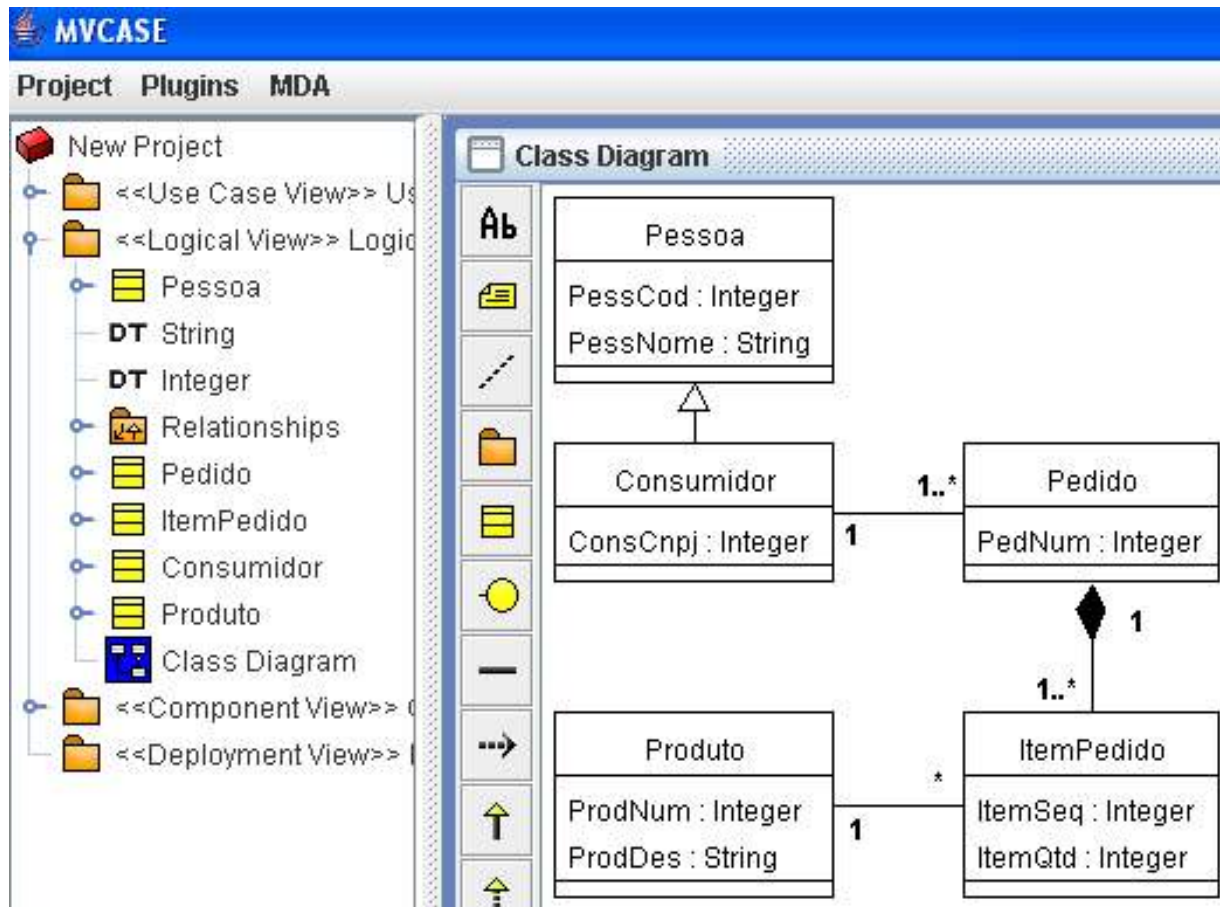
Componentes de Transformação



Componente CWM2SQL

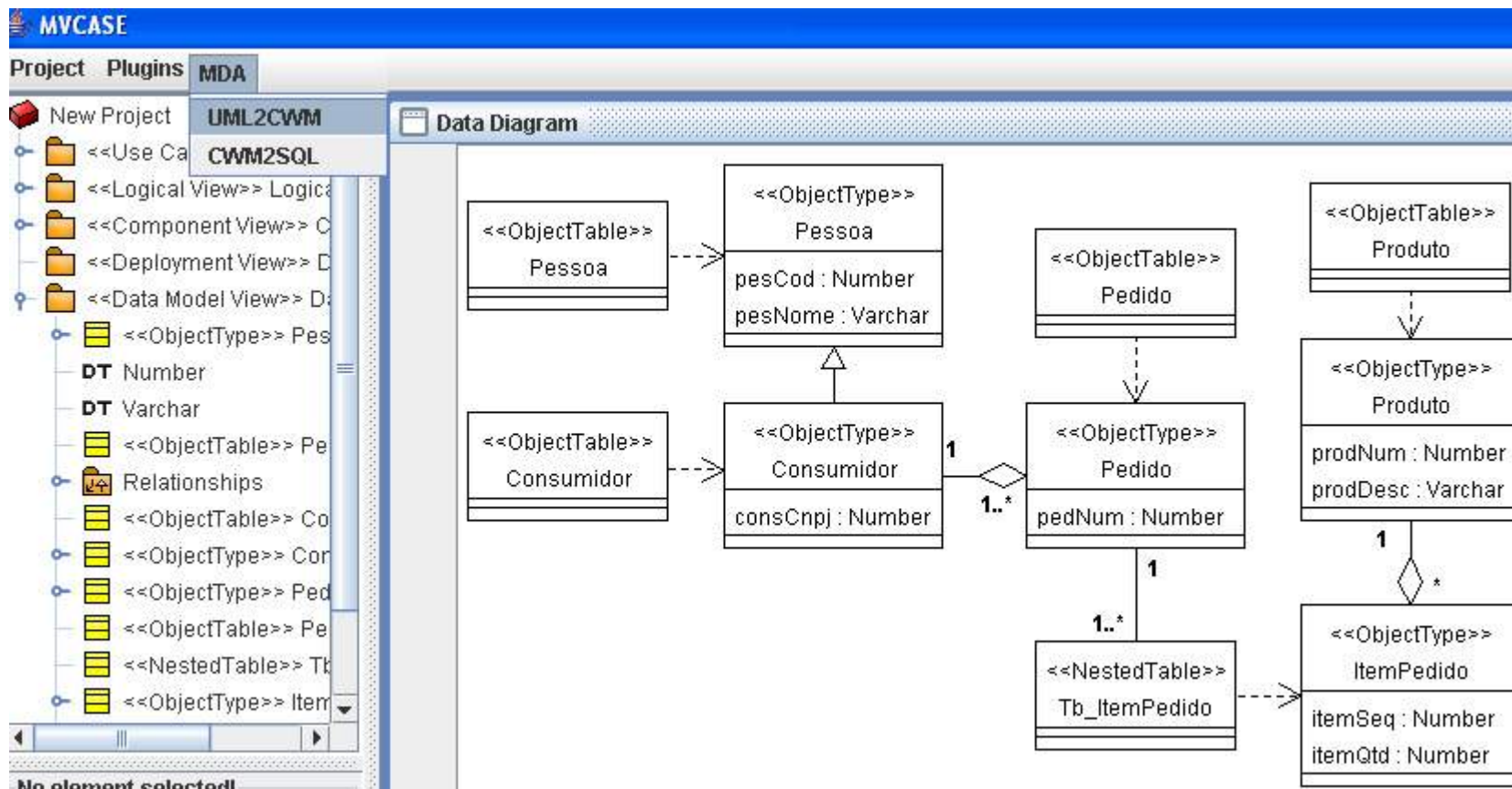
Exemplo de uso

Modelo de Classe



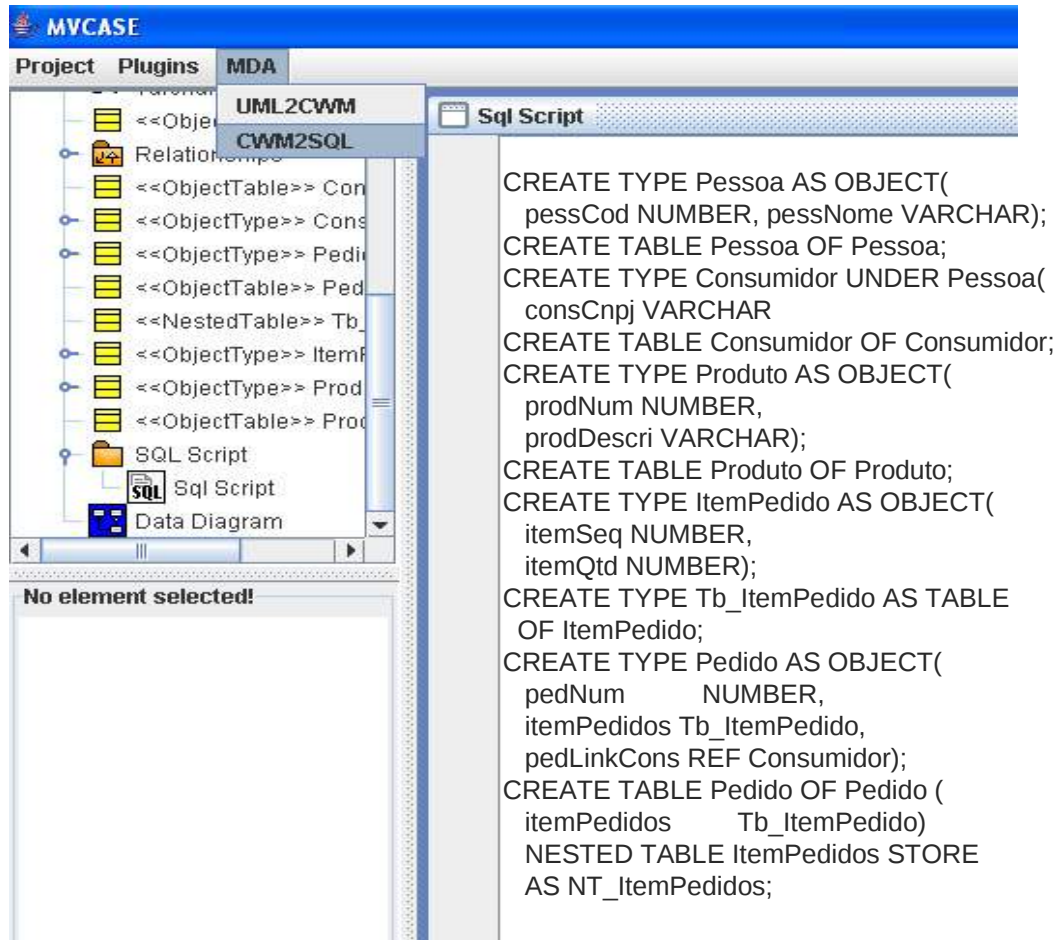
Exemplo de uso

Modelo de Dados Objeto-Relacional



Exemplo de uso

Códigos Sql



The screenshot shows the MVCASE software interface. On the left, a project tree is visible with the following structure:

- UML2CWM
- CWM2SQL
- SQL Script
- SQL Script
- Data Diagram

The main window displays the following SQL script:

```
CREATE TYPE Pessoa AS OBJECT(  
    pessCod NUMBER, pessNome VARCHAR);  
CREATE TABLE Pessoa OF Pessoa;  
CREATE TYPE Consumidor UNDER Pessoa(  
    consCnpj VARCHAR  
);  
CREATE TABLE Consumidor OF Consumidor;  
CREATE TYPE Produto AS OBJECT(  
    prodNum NUMBER,  
    prodDescri VARCHAR);  
CREATE TABLE Produto OF Produto;  
CREATE TYPE ItemPedido AS OBJECT(  
    itemSeq NUMBER,  
    itemQtd NUMBER);  
CREATE TYPE Tb_ItemPedido AS TABLE  
OF ItemPedido;  
CREATE TYPE Pedido AS OBJECT(  
    pedNum NUMBER,  
    itemPedidos Tb_ItemPedido,  
    pedLinkCons REF Consumidor);  
CREATE TABLE Pedido OF Pedido (  
    itemPedidos Tb_ItemPedido)  
NESTED TABLE ItemPedidos STORE  
AS NT_ItemPedidos;
```

[Conclusão]

- Os componentes de transformação:
 - Possibilitam a obtenção de modelos dependentes de plataforma a partir de modelos independentes de plataforma e posterior geração de códigos
 - São independentes e podem ser incluídos em outras ferramentas desde que sejam respeitadas suas interfaces.
- Adoção de linguagens padrões como mecanismo facilitador para transferência de informações entre ferramentas.
- **MDA não resolve todos os seus problemas... Mas é totalmente viável...**

[Trabalhos Futuros]

- Criação de um metamodelo para rastreamento das instâncias Origem e Destino
- Manutenção e refino dos modelos de dados de forma gráfica possibilitando maior abstração semântica para no modelos de dados
- Desenvolvimento de componentes transformadores para outros domínios

SBCARS 2007

Transformando Modelos da MDA com o apoio de Componentes de Software



Obrigado pela Paciência...

Marco Antonio Pereira
pemarco@gmail.com

Campinas-SP, 31 de Agosto de 2007