



Universidade do Minho
Departamento de Informática

Engenharia de Aplicações

Desenvolvimento Baseado em Modelos
(MDSE)

António Nestor Ribeiro
anr@di.uminho.pt

Motivação

- A utilização de modelos, reconhecida por todos os intervenientes no processo de construção de sistemas software como sendo valiosa para se representar e pensar o problema, era considerada como sendo desligada do processo de desenvolvimento.
- A separação entre as tarefas de modelação e de desenvolvimento não beneficia a qualidade final do produto
 - nada obriga a que a transformação em código respeite as indicações do modelo
- A Model Driven Development surge como:
 - Uma abordagem ao desenvolvimento de software
 - Uma abordagem onde os modelos assumem papel preponderante
 - Uma abordagem em que a geração de código final é obtida através de transformações

Motivação

- O desenvolvimento baseado em modelos tem como entidades principais do processo:
 - os modelos
 - as tarefas de modelação
 - a Model Driven Architecture, que é uma instanciação da Model Driven Development
- A abordagem seguida pela MDA separa a funcionalidade do sistema das questões relativas ao desenvolvimento e à tecnologia utilizada
 - manter a modelação ao nível dos requisitos e dos conceitos de negócio
 - através de mecanismos de software obter a solução software

Artefactos da MDA

- De que precisamos para transformar um modelo em código?
 - High-level models, written in a standard language, that are consistent, precise, and unambiguous.
 - One or more standard, well-defined transformation rules.
 - Definitions of how a Platform Independent Model (PIM) is transformed into a specific PSM (Platform Specific Model).

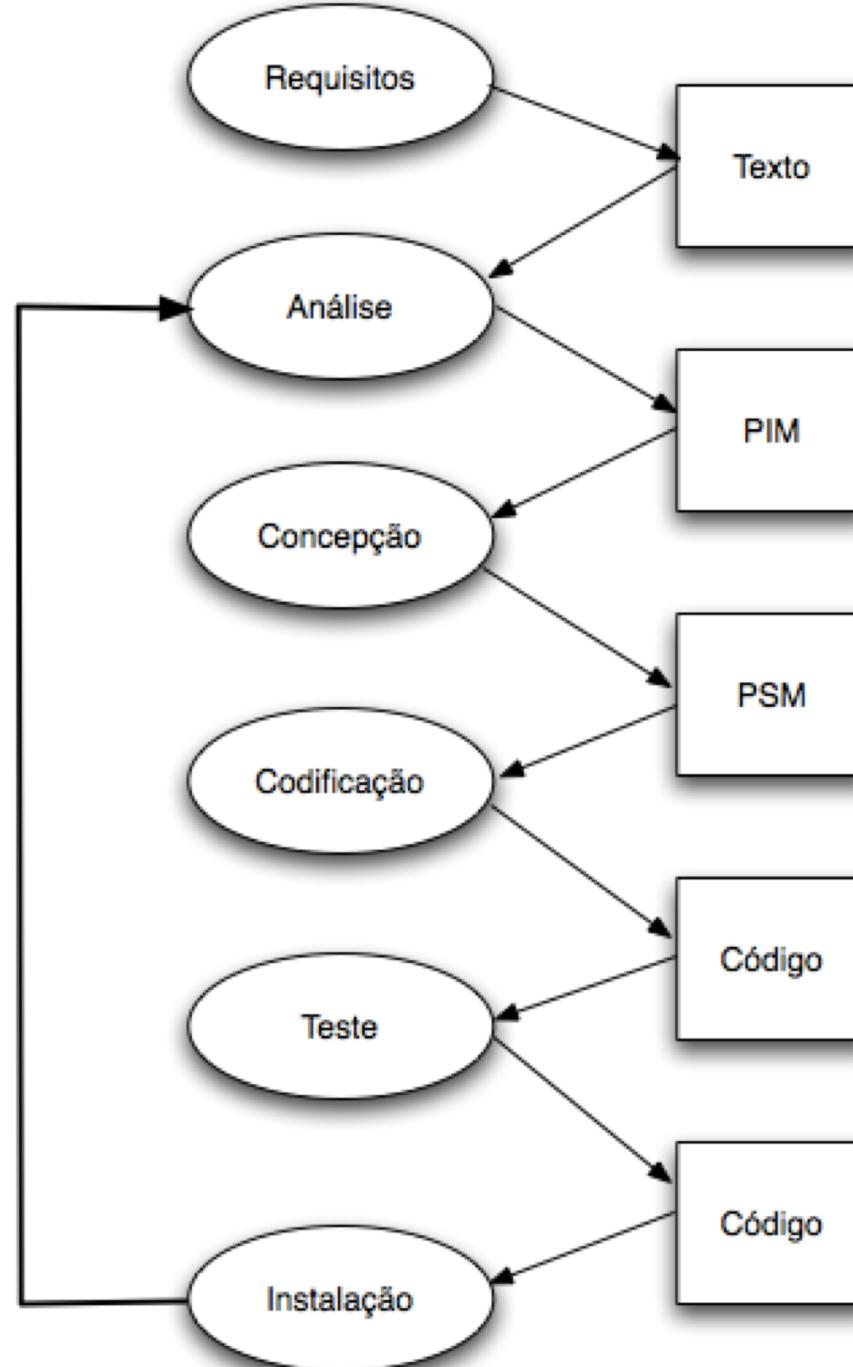
Modelos de alto nível, escritos numa linguagem padrão e bem definida, que são consistentes, precisos e ambigüos. Alguns dessas definições podem ser flexíveis, dependendo do ambiente de execução. Algumas definições seriam invariáveis, enquanto outras seriam tunáveis para o indivíduo.
 - A language in which transformations can be specified and interpreted by the transformation tools. This language is called the transformation language.
 - Tools that implement the transformation language. Preferavelmente, essas ferramentas oferecem flexibilidade para ajustar o passo de transformação de acordo com as necessidades específicas.
 - Tools that implement the execution of the transformation of a PSM to code.

Sobre os modelos

- Num mesmo projecto podem existir diferentes tipos de modelos, de acordo com as diferentes vistas que se pretende ter do modelo.
- Algumas questões pertinentes:
 - In what part of the software development process is the model used? Is it an analysis or design model?
 - Does the model contain much detail? Is it abstract or detailed?
 - What is the system that the model describes? Is it a business model or software model?
 - What aspect of the system does the model describe? Is it a structural or dynamic model?
 - Is the model targeted at a specific technology? Is it platform independent or platform specific?
 - At which platform is the model targeted? Is it an EJB, ER, C++, XML, or other model?

Ciclo de Vida

- Processo proposto pela MDA

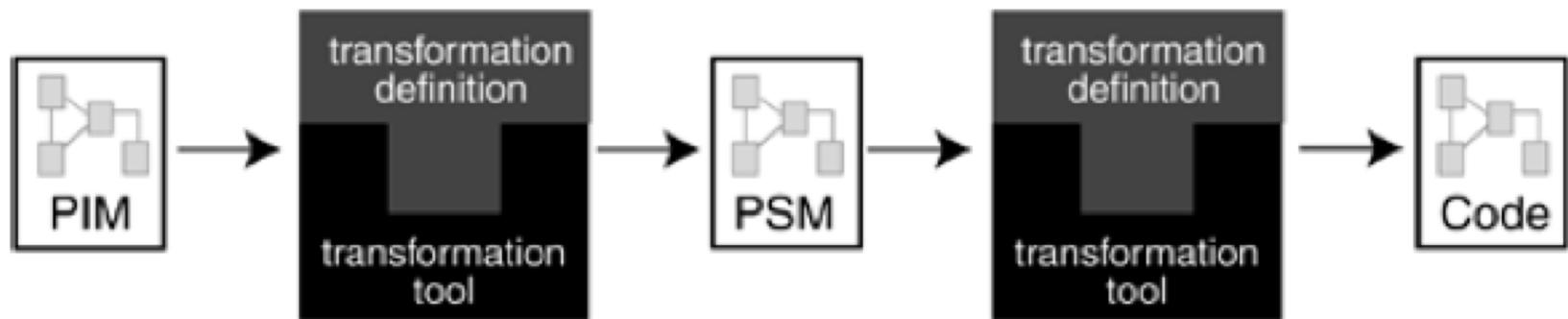


Fases do ciclo de vida MDA

- O processo assenta em quatro fases, depois de recolhidos os requisitos:
 1. Criação de um modelo independente da camada computacional obtida a partir da informação do negócio. Este modelo designa-se por **Computation Independent Model (CIM)** e é a concretização do modelo de domínio
 2. Criação de um modelo da aplicação independente da tecnologia. O **Platform Independent Model (PIM)** descreve o núcleo dos componentes e serviços oferecidos pela lógica de negócio.
 3. Desenvolvimento de modelos específicos para determinada tecnologia, acrescentando ao modelo independente os componentes e tecnologia que seja necessário instanciar. A este modelo chama-se **Platform Specific Model (PSM)** e é um refinamento do PIM para uma determinada tecnologia. Note-se que para um PIM podem existir vários PSM.
 1. Geração de código, no qual se transforma o PSM em código final.

Transformações

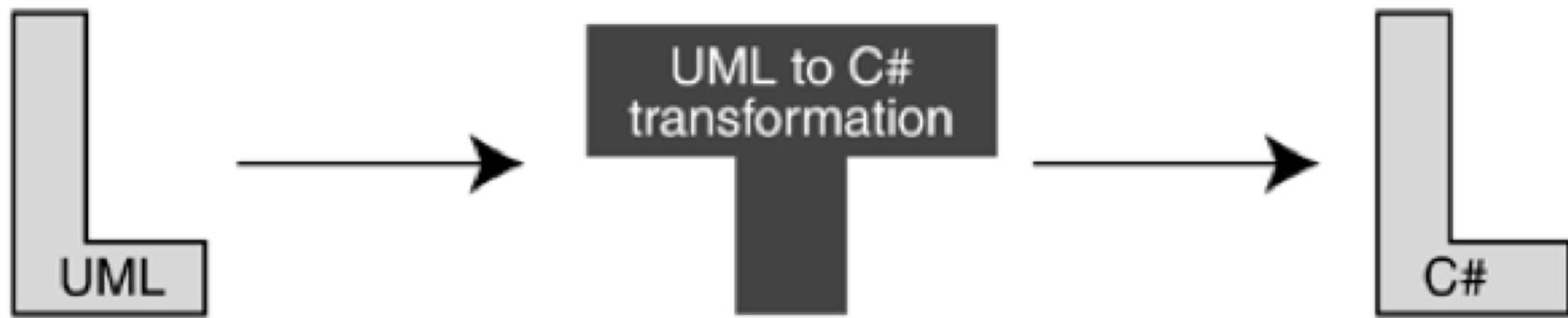
- Uma transformação baseia-se num modelo e gera o modelo seguinte no ciclo de vida apresentado.
 - Exemplo: PIM -> PSM ou PSM -> código final



- Para que seja possível é necessário que a linguagem em que se descreve o PIM e o PSM tenha um meta-modelo conhecido: **UML**

Transformações

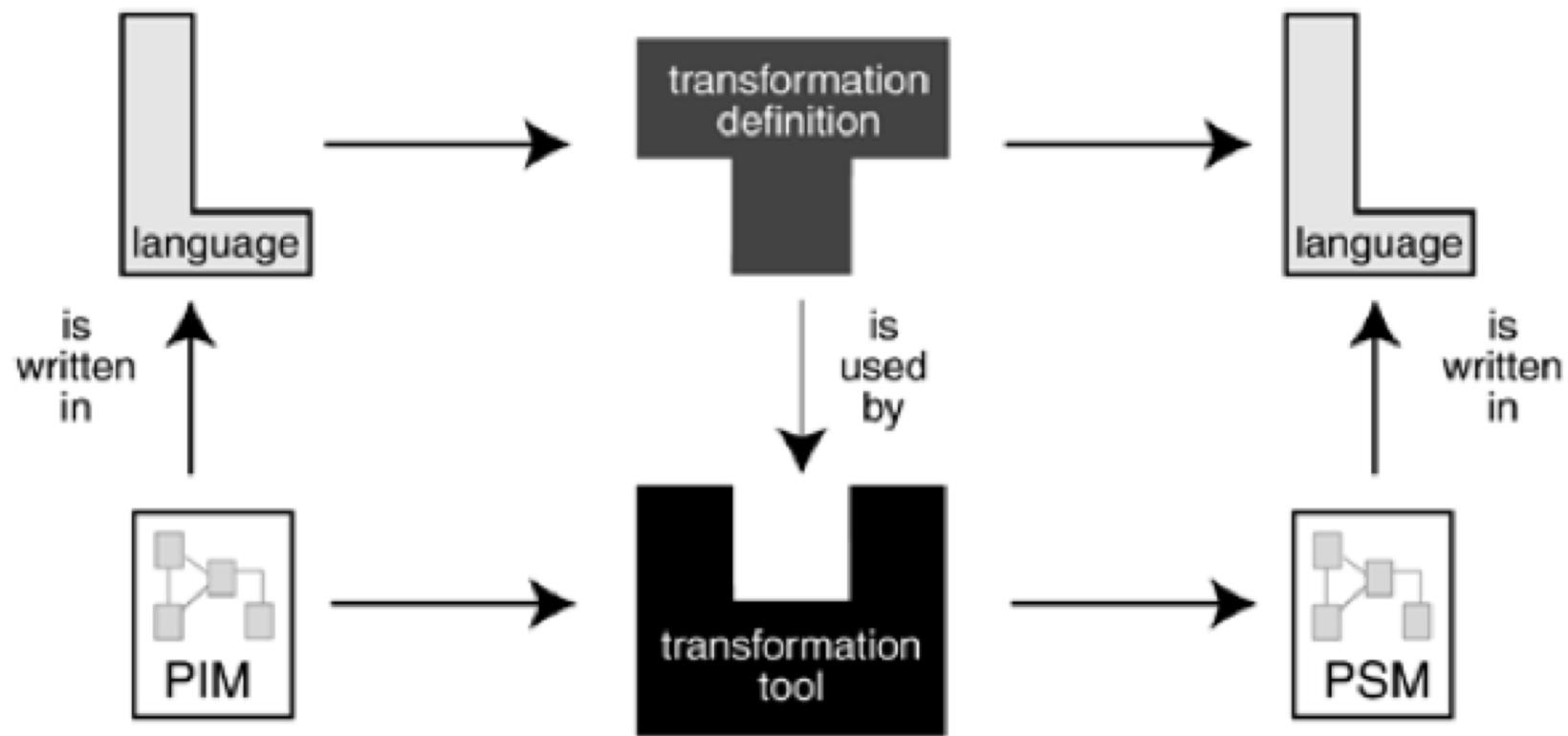
- É necessário que exista uma correspondência entre os metamodelos das linguagens das entradas e saídas de uma transformação.



- No caso do UML é possível escrever as regras que fazem a transformação para linguagens como C#, C++, Java, Hibernate, etc.

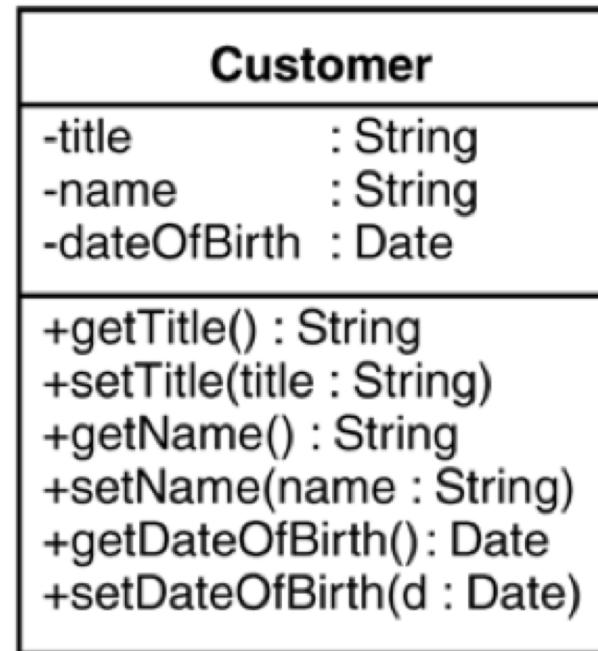
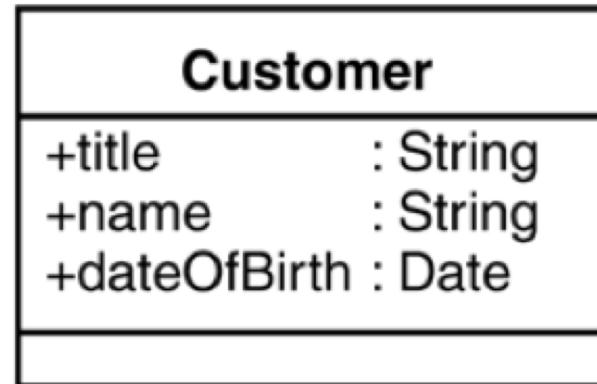
Transformações

- Correspondência entre modelos e transformações

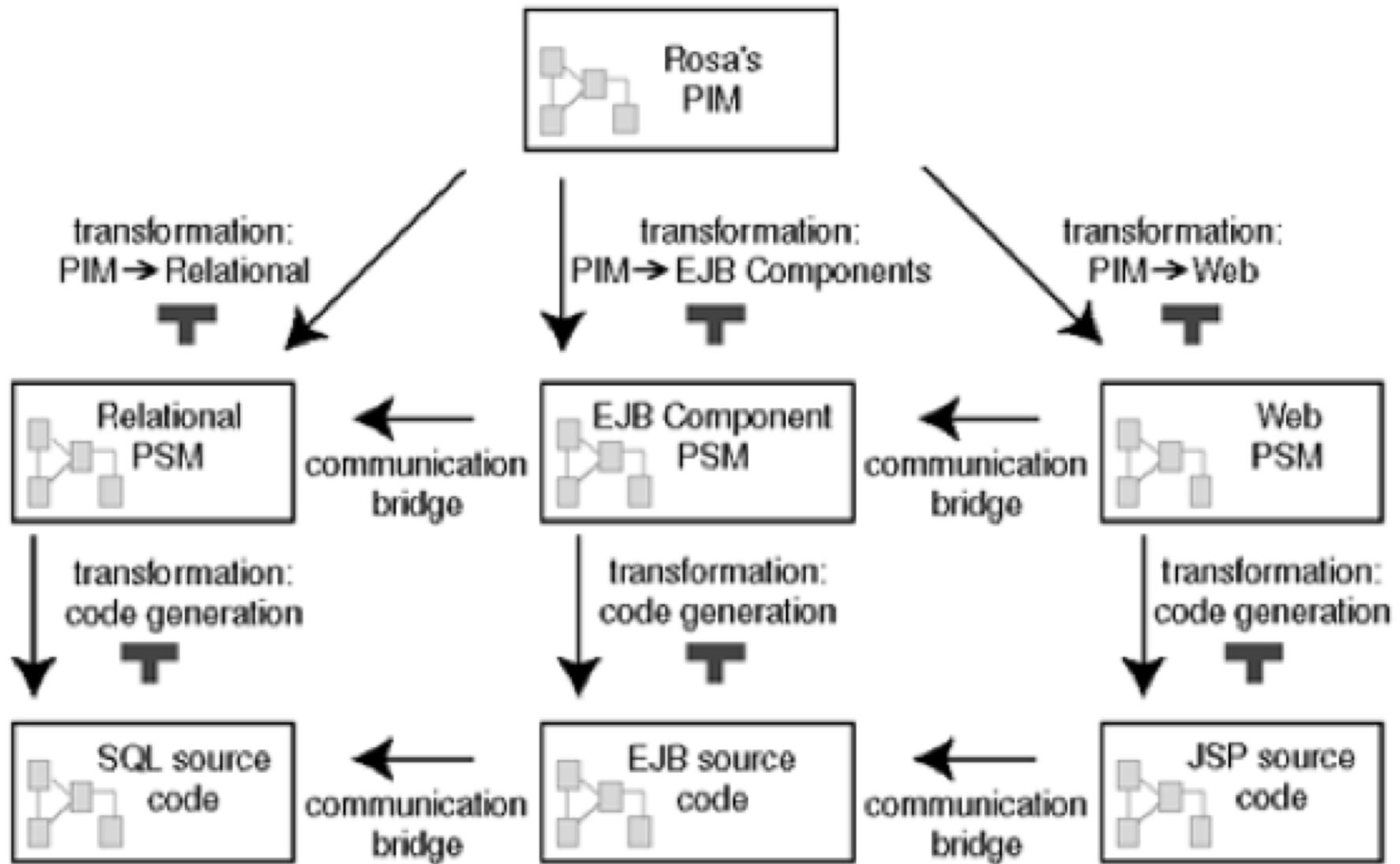


PIM e PSM

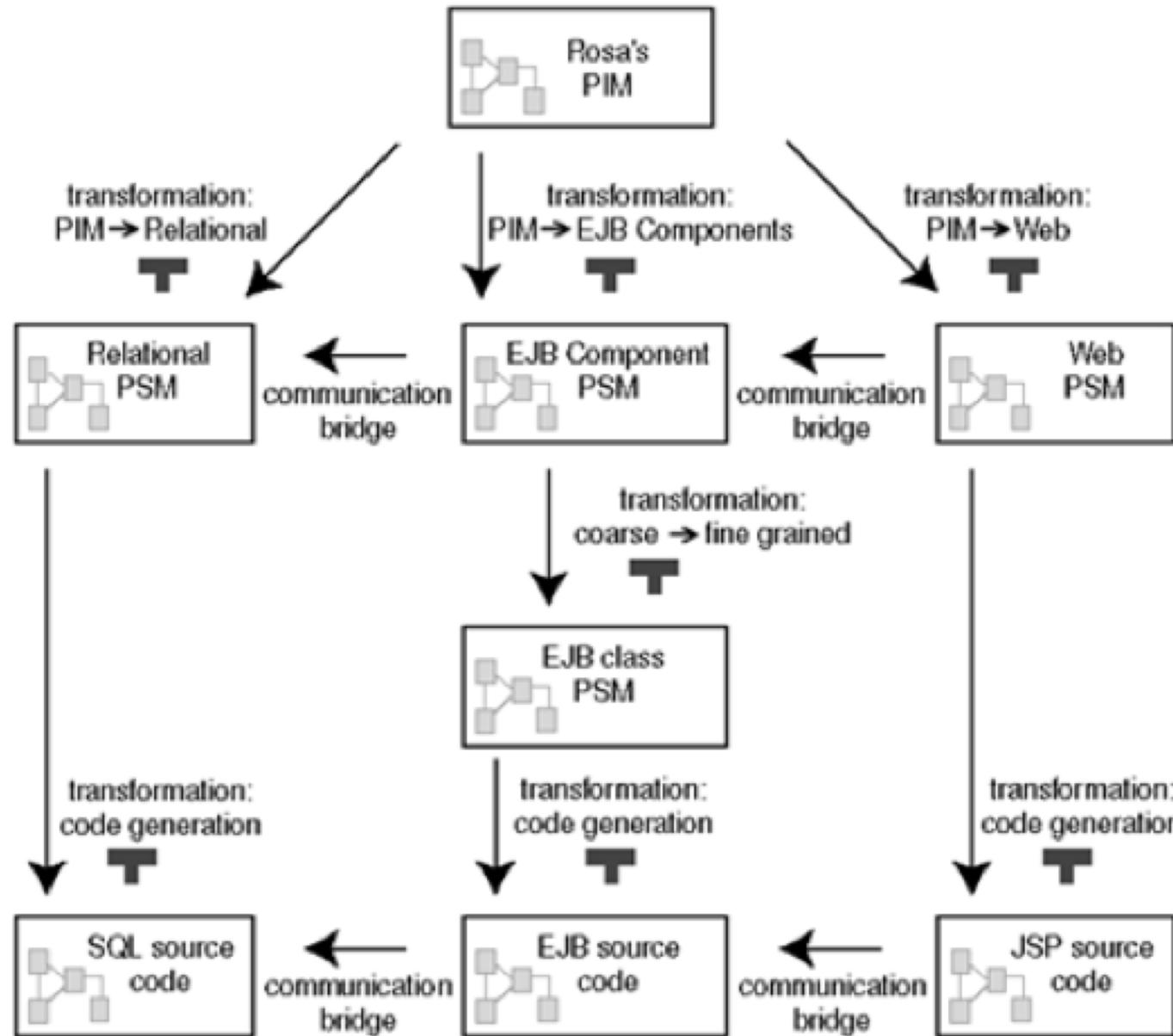
- Modelo abstracto (PIM)
- Modelo Específico (PSM)



Aplicação a um exemplo (PIM vs PSM)

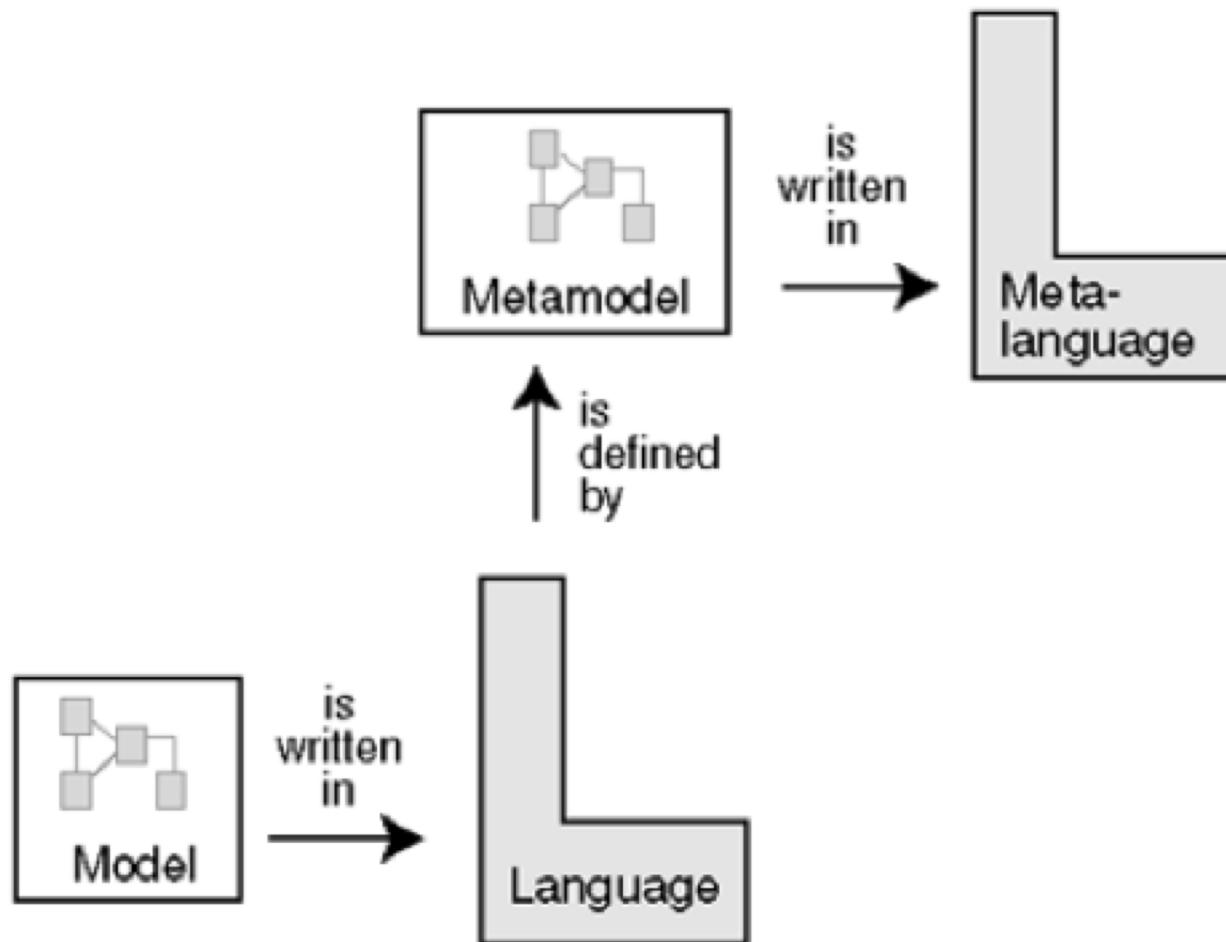


Os diversos modelos da MDA

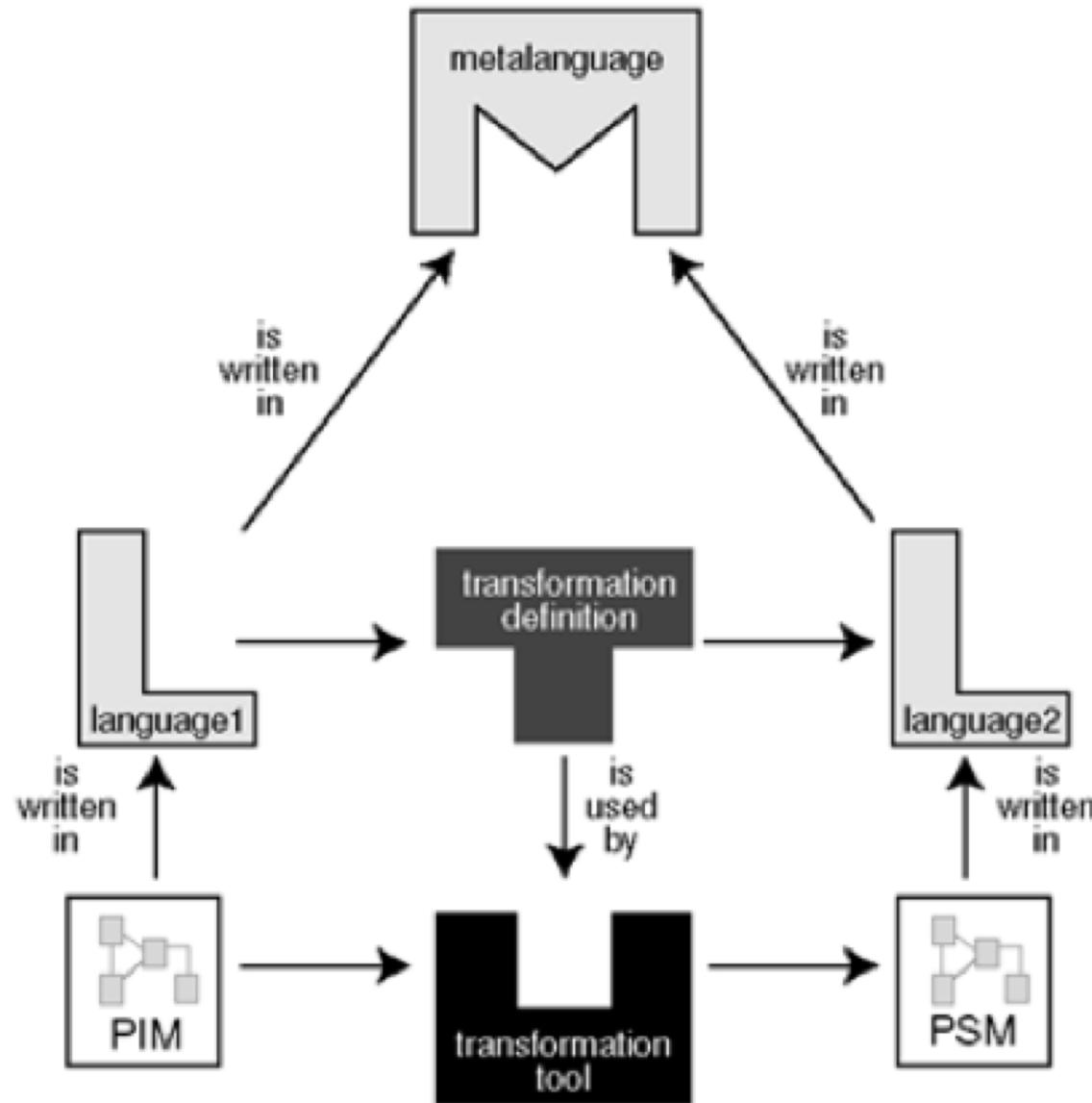


Meta-modelos

- Modelos, Linguagens, Meta-modelos e Meta-linguagens



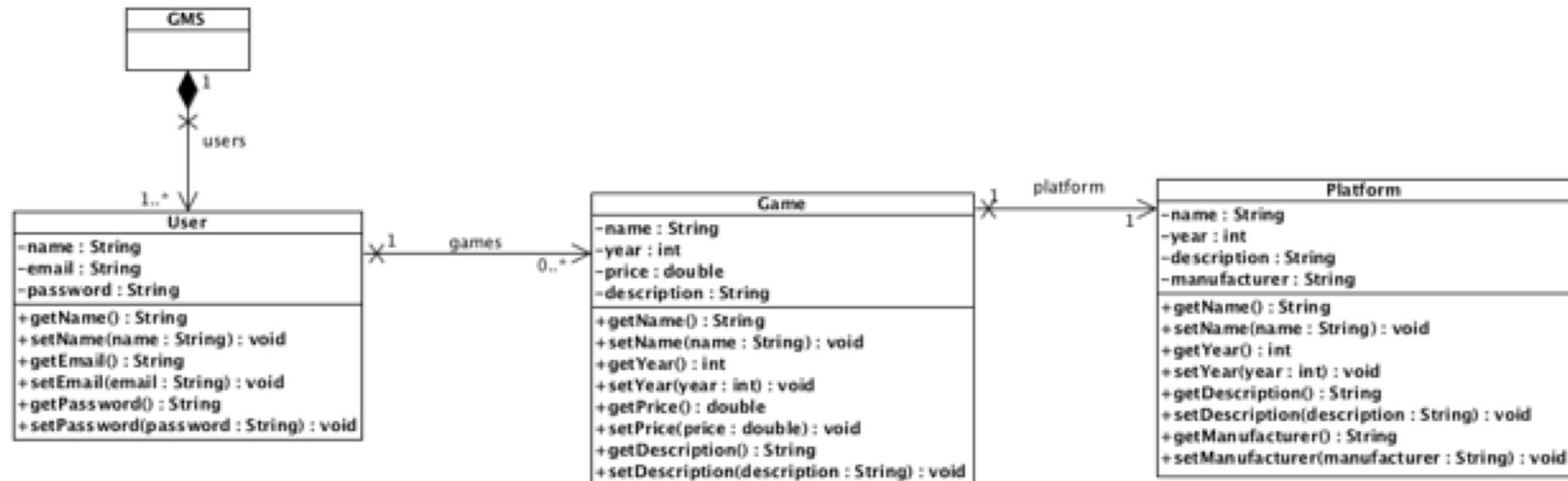
Meta-Modelação



Exercício aula 2020.03.24

- Enunciado:
 - The Game Management System allows the users to create a digital library of games. Users can create an account, and then associate with their account a videogame. The objective is to provide a tool for users to more easily manage large collections of videogames. As it is well known, a videogame has an associated platform. While each game belongs only to a platform, several games can have the same platform.
 - The proposal is to develop a system to support this digital library of video games, with the most common features:
 - Register a user;
 - Register a game;
 - Register a platform;
 - List user games;
 - List all games;
 - Search a game;
 - Delete a game

- Proposta de arquitectura:



- Conceba e crie a camada de dados para acesso à informação dos objectos que serão persistidos.

O PSM Hibernate

- As anotações Hibernate (ou EJB/JPA) são colocadas no código fonte das classes que representam conceitos do modelo da aplicação.
- Tal impede que as classes criadas ao nível do modelo PIM possam ser realmente independentes e reutilizadas por outros modelos de implementação.
- Como tornar reutilizáveis as classes definidas, para que possam prever diferentes tipos de mecanismos de persistência?
 - Hibernate
 - Camada de Dados com escrita de SQL
 - Streams de Objectos
 - Etc.