

Model Driven Development Persistency

Rui Couto; António Nestor Ribeiro

Outline

- Introduction
- ORM
- Process
- Tutorial

Introduction

- One of the advantages of model driven development is the tool support.
- Some repetitive and laborious tasks can be automated resorting to these tools.
- An example is the automatic generation of persistency for the model entities.

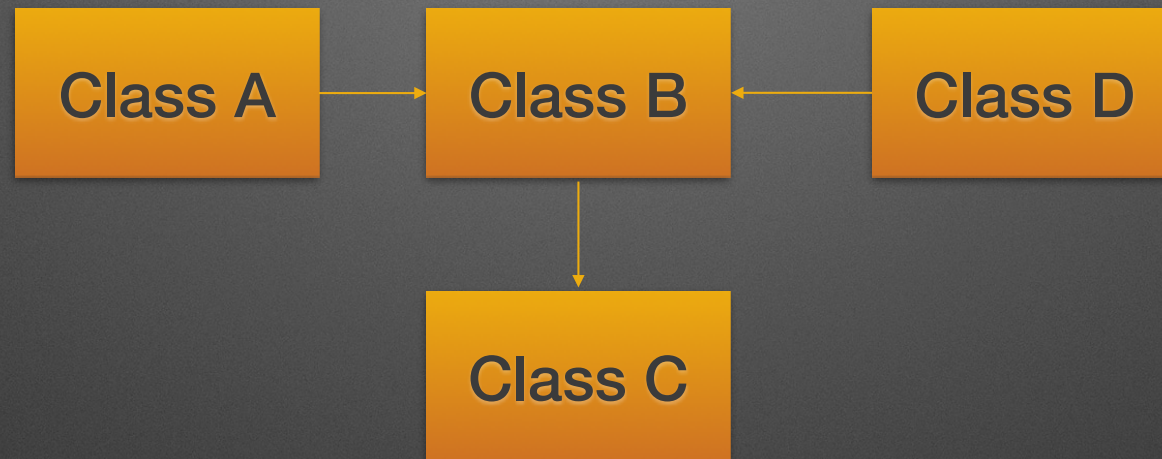
- Visual Paradigm supports the model driven development process.
- One of the provided features is the generation of source code.
- Supports also the generation of the *persistence code*.

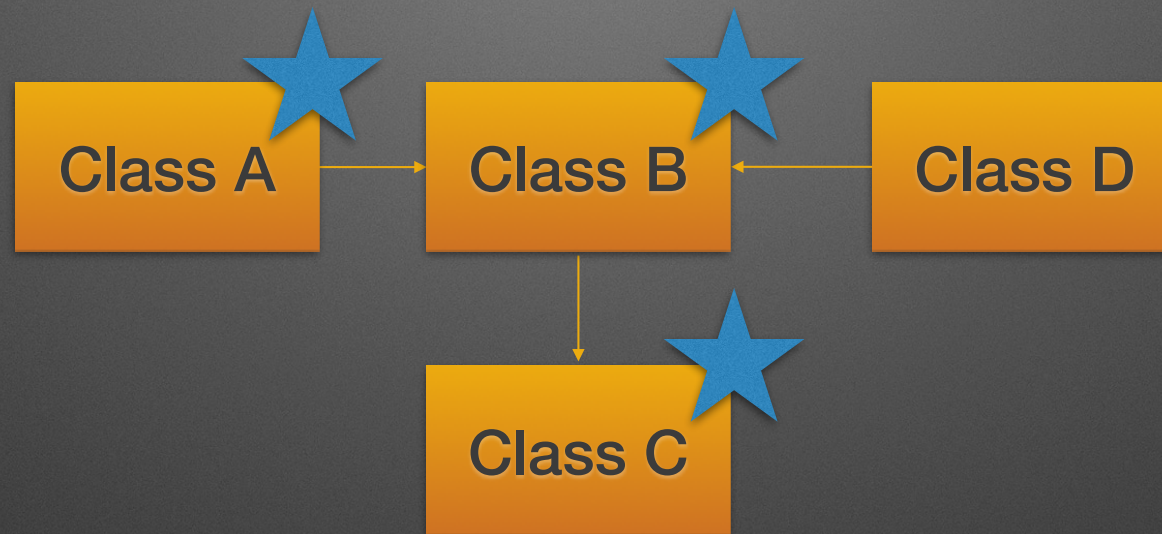
VP and ORM

- As a model driven development process, generating persistency requires:
 - A model of the system;
 - Information regarding the entities to persist.
- The expected output is composed of:
 - The source code;
 - Persistency classes.

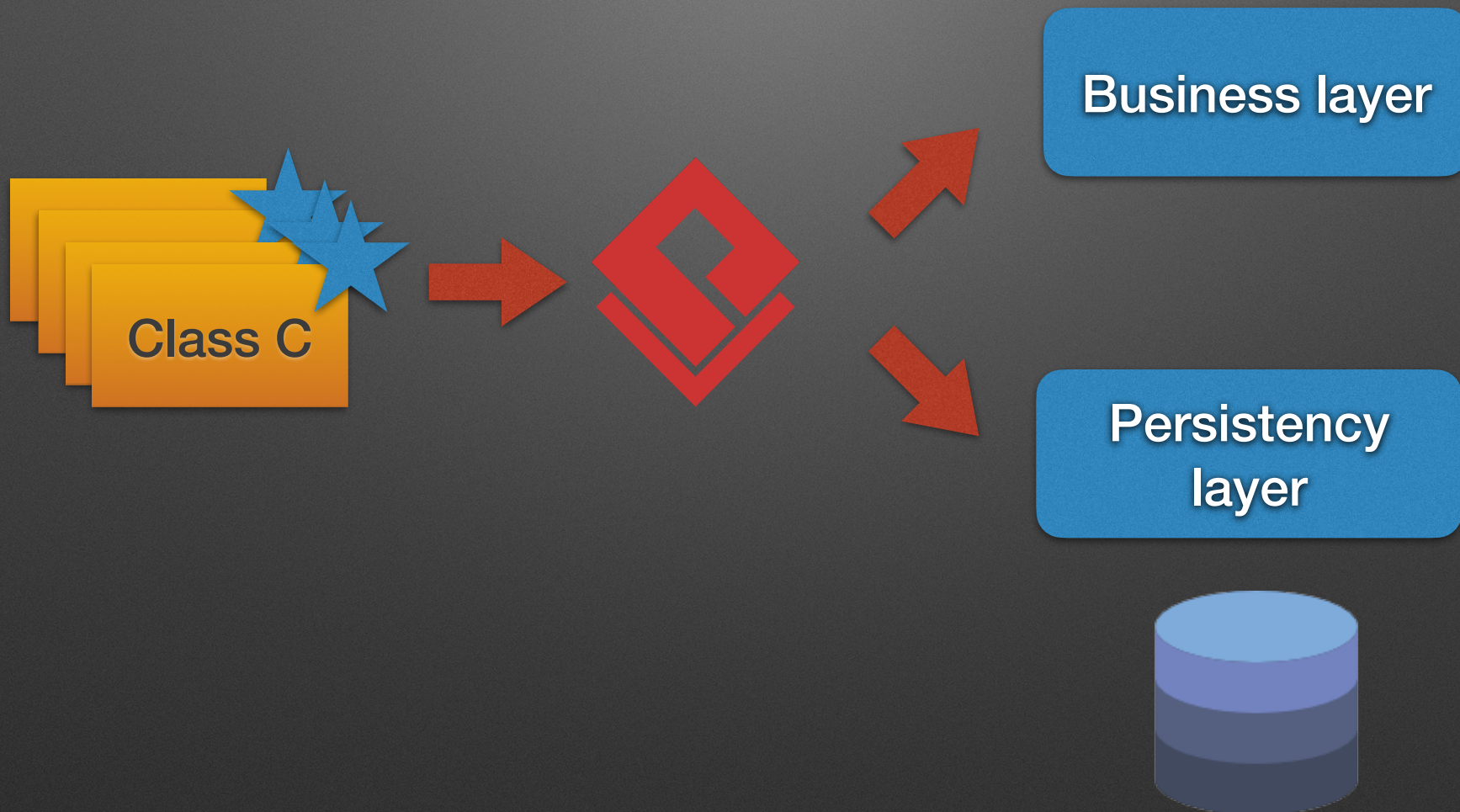
Process

- The process to generate the persistency consists in:
 - Creating the model;
 - Define the entities to persist;
 - Generate the source code and supporting database schema.









Example - Source

- Visual Paradigm generates the source code, according to the model.
- Attributes, getters and setters are expected.

```
public class User {  
  
    public User() {  
    }  
  
    private int ID;  
    private String name;  
    private String email;  
    private String password;  
    private Platform platform;  
  
    private void setID(int value) {  
        this.ID = value;  
    }  
  
    public int getID() {  
        return ID;  
    }  
  
    //...  
  
    public void setName(String value) {  
        this.name = value;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setEmail(String value) {  
        this.email = value;  
    }  
}
```


Example - Mapping

- The mapping information is also generated.
- For each class and attributes, the corresponding mapping information is created.
- The information expressed as XML keeps the business code more *clear*.

```
<?xml version="1.0" encoding="utf-8" ?>
<!--
Licensee: Universidade do Minho
License Type: Academic
-->
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/
Hibernate Mapping DTD 3.0//EN" "http://
www.hibernate.org/dtd/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
    <class name="pt.uminho.di.aa.Game"
        table="Game" lazy="false">
        <id name="ID" column="ID"
            type="integer" unsaved-value="0">
            <generator class="native">
            </generator>
        </id>
        <many-to-one name="platform"
            column="PlatformID"
            class="pt.uminho.di.aa.Platform"
            cascade="save-update,lock"
            unique="true"
            not-null="true"
            lazy="proxy"
            access="field">
        </many-to-one>
        <property name="name"
            column="Name"
            type="string"
            length="255"
            not-null="false"
            lazy="false"/>
    ...
```


Example - DAO

- DAOs implement the persistency layer.
- The generated code provides a set of methods to manage objects.
- The class abstracts all the hibernate code.

```
public class GameDAO {
    public static Game loadGameByORMID(int ID) throws
                                                PersistentException {
        try {
            PersistentSession session= GamesLibraryPersistentManager
                                        .instance()
                                        .getSession();

            return loadGameByORMID(session, ID);
        } catch (Exception e) {
            e.printStackTrace();
            throw new PersistentException(e);
        }
    }

    public static Game getGameByORMID(int ID)

    public static List queryGame(String condition, String
orderBy)

    public static Game[] listGameByQuery(String condition,
String orderBy)

    public static java.util.Iterator iterateGameByQuery(String
condition, String orderBy)

    public static boolean save(pt.uminho.di.aa.Game game)

    public static boolean delete(pt.uminho.di.aa.Game game)

    public static boolean refresh(pt.uminho.di.aa.Game game)

    public static boolean evict(pt.uminho.di.aa.Game game)

    public static Game[] listGameByCriteria(GameCriteria
gameCriteria)
```


Example - Associations



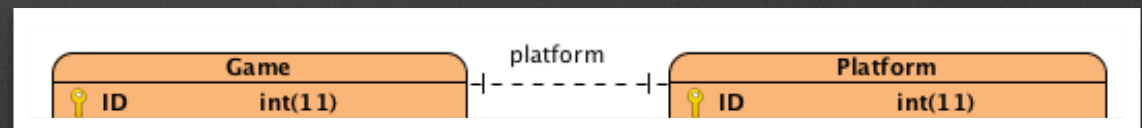
- 1-to-1 associations are represented as classes composition.

```
public class Game {
    public Game() {
    }

    ...

    private int ID;

    private Platform platform;
```



Example - Associations



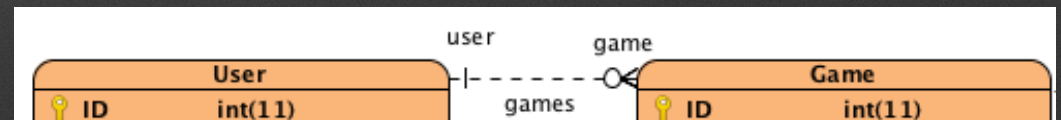
- 1-to-n associations are represented as collections

```
public class User {
    public User() {
    }

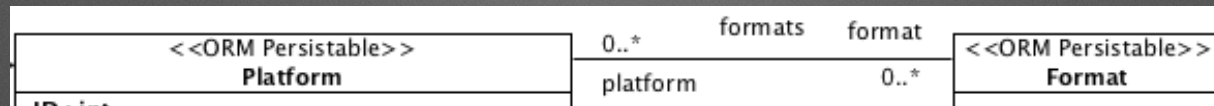
    ...

    private int ID;
    private String name;

    private java.util.Set ORM_game =
        new java.util.HashSet();
}
```



Example - Associations



- n-to-n associations are represented as collections on both ends

```
public class Platform {  
    public Platform() {  
    }  
  
    private java.util.Set ORM_format =  
        new java.util.HashSet();  
}
```



Tutorial

- Create an application to manage a games' library.
- Perform a model driven development, starting from a class diagram.
- Automatically generate the source code.

Model Driven Development Persistency

Rui Couto; António Nestor Ribeiro