

DSM (RPG) to Android

Rafael Ugaz

23/01/2013

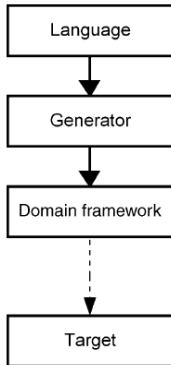
Outline

- 1 Introduction
 - What is DSM
- 2 Solution
 - metaDepth
 - EGL
 - Android
- 3 Conclusions and future work
- 4 Demo

What is Domain-Specific Modeling?

- Software engineering methodology
- Raise the level of *abstraction* (Java > Assembler)
- Focus on specific problem *domain*

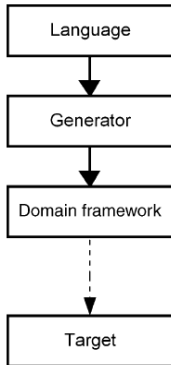
Architecture of DSM solution



- Models created in domain-specific language
- Generator produces code
- Code supported by domain framework
- Application executed in target environment

Kelly, S., Tolvanen, J.-P., 2008. Domain-Specific Modeling. p. 64.

Architecture of DSM solution



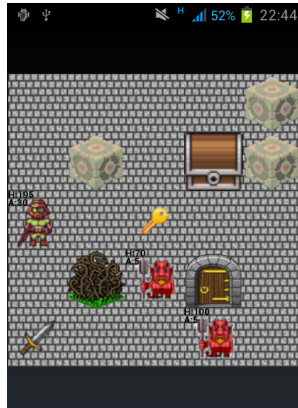
- Models created in domain-specific language
- Generator produces code
- Code supported by domain framework
- Application executed in target environment

Kelly, S., Tolvanen, J.-P., 2008. Domain-Specific Modeling. p. 64.

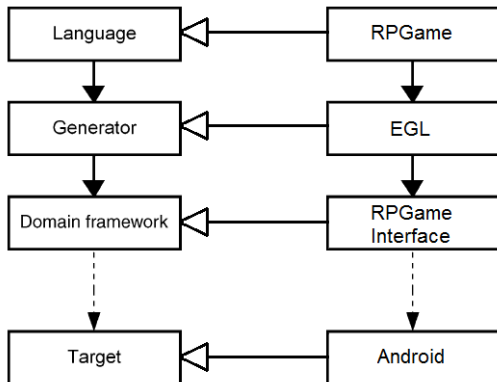
DSM RPG

Goal:

Produce RPG Android applications



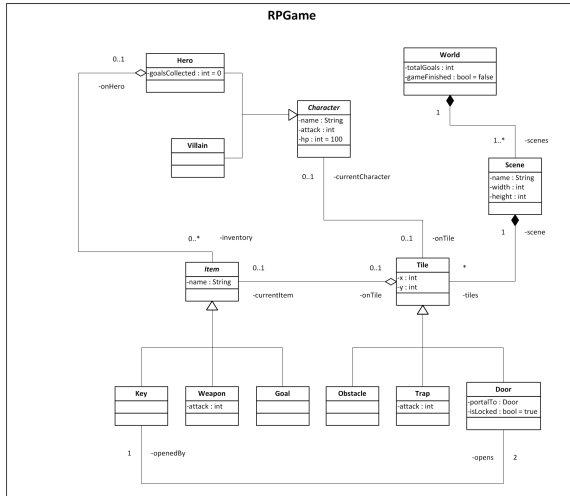
DSM RPG



Language: RPGame.mdepth

- Designed *textually* in metaDepth
- Constraints → Well formed models
- Integration with EGL

Language: RPGGame.mdepth



Generator: EGL

- Template-based language for model-to-text transformations.
- Part of Epsilon family of tools for MDE (EOL)
- Dynamic sections: [% < *EOL syntax* > %]

Generator: EGL

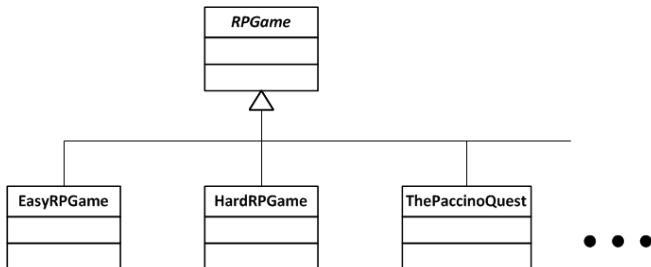
Template (EGL):

```
1 [% for (v in Villain.allInstances()) { %]  
2 Villain [%=v%] = new Villain([%=v.onTile%], "[%=v.name%", [%=v.  
   attack%], [%=v.hp%]);  
3 [%=v.onTile%].setCurrentCharacter([%=v%]);  
4 [% } %]
```

Produces (Java):

```
1 Villain bernard = new Villain(s25, "Bernard", 5, 100);  
2 s25.setCurrentCharacter(bernard);  
3 Villain kevin = new Villain(s55, "Kevin", 5, 100);  
4 s55.setCurrentCharacter(kevin);  
5 Villain donald = new Villain(f34, "Donald", 15, 100);  
6 f34.setCurrentCharacter(donald);
```

Domain framework: RPGGame.java



Target: Android

- GUI → RPGView.java
- Meta-model → Scene.java, Tile.java, ...
- Op. Semantics (EOL) → RpgEngine.java
- Layout → rpg_layout.xml

Conclusions and future work

- Easy to create own RPGGame
- Text-based DSL → Visual DSL (AToM3?)
- Future: more complex domains

Demonstration

