MODEL-DRIVEN DEVELOPMENT FOR AGENT-BASED MODELING AND SIMULATION

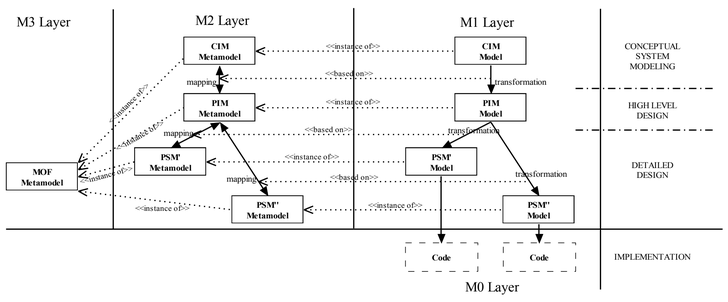
Alfredo Garro, Francesco Parisi, Wilma Russo
DEIS – Università della Calabria, Via P. Bucci, cubo 41C, 87036 Rende (CS), Italy
{garro, fparisi, russow}@deis.unical.it

1. Introduction and Proposal

- **ABMS** (Agent Based Modeling and Simulation) can effectively represent a system at different levels of complexity through the use of **autonomous, goal-driven and interacting entities** (agents) organized into societies which exhibit emergent properties. The agent-based model of a system can then be executed to simulate the behavior of the complete system so that knowledge of the behaviors of the entities (**micro-level**) produce an understanding of the overall outcome at the system-level (**macro-level**).
- There is a **lack of processes** able to guide and support ABMS practitioners in the definition of agent-based simulation models starting from a conceptual, and domain-expert-oriented modeling of the system and without excessively focusing on simulation implementation details.
- To address these issues, we propose a **Model-Driven approach for ABMS** which conforms to the OMG Model-Driven Architecture (**MDA**) and then allows to (automatically) produce **Platform-Specific simulation Models (PSMs)** starting from a **Platform-Independent simulation Model** (**PIM**) obtained on the basis of a starting **Computation Independent Model** (**CIM**).
- PIMs are produced by exploiting the AMF framework defined in the **AMP (Agent-Modeling Platform) Eclipse Project** which currently represents the only effort able to provide automatic generation of Platform-Dependent simulation Models and related code for several ABMS platforms.

2. An MDA-based approach for ABMS

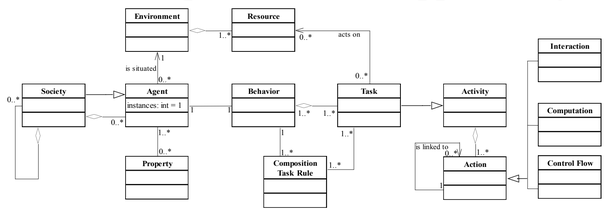
- The MDA architecture can be effectively exploited in the ABMS domain to obtain platform-independent agent-based models which are not only easy to verify, modify and update but also require significantly reduced programming and implementation efforts.



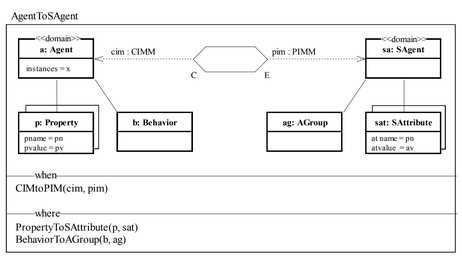
THE MDA-BASED PROCESS

- **To map the basic MDA concepts**, which have been specifically conceived for the Software Engineering domain, **into the ABMS counterparts**, we have defined the following components:
  i. **a reference CIM metamodel** for the definition of Computation Independent Models which supports the agent-based conceptual system modeling carried out through both abstract and domain-expert oriented concepts;
  ii. **a PIM metamodel** for the definition of Platform-Independent ABMS Models based on the AMF framework;
  iii. a **mappings** among these metamodels so to enable CIM to PIM model transformations.

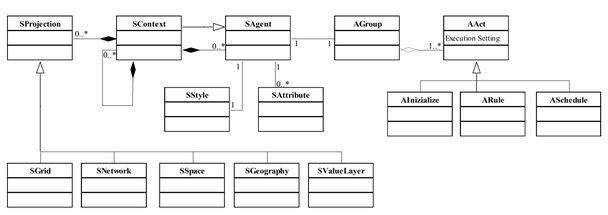
  The final transformations from the PIM to the PSMs of the target ABMS simulation platforms are supported by the AMF framework.



THE CIM METAMODEL



MAPPING BETWEEN THE CIM AND PIM METAMODELS: A MAPPING RULE EXPRESSED BY USING THE QVT/R GRAPHICAL NOTATION

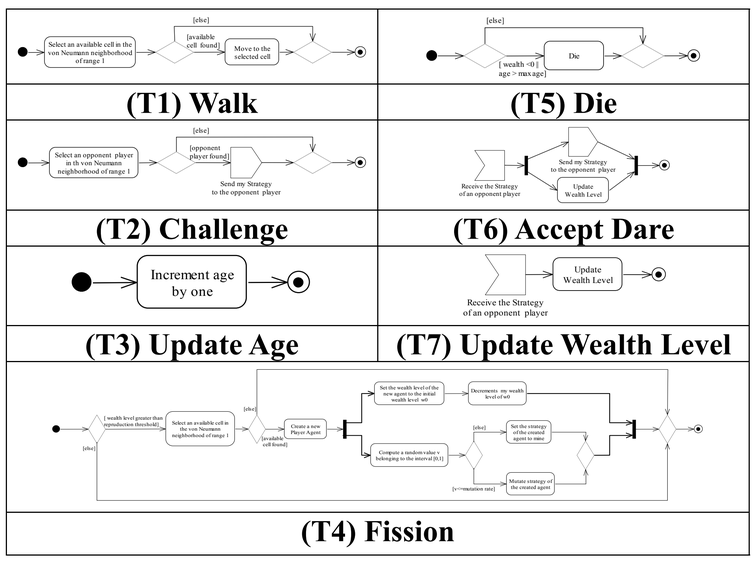


THE PIM METAMODEL

3. Exploiting the proposed MDA-based process: Agent-Based Modeling and Simulation of the Demographic Prisoner's Dilemma (*DPD*)

- The *DPD* game is able to represent several social and economic complex scenarios in which interesting issues regard the identification of starting configurations and conditions that allow initial populations to reach stable configurations (in terms of both density and geographic distribution).
- In the DPD game k players are spatially distributed over an n-dimensional toroidal grid. Each player is able to move to empty cells in its von Neumann neighborhood of range 1 (feasible cells), is characterized by a fixed pure strategy (c for cooperate or d for defect) and is endowed with a level of wealth w which will be decremented or incremented depending of the payoff earned by the player in each round of the Prisoner's Dilemma game played during its life against its neighbors.
- The player dies when its wealth level w becomes negative, whereas, when w exceeds a threshold level wb, an offspring can be produced with wealth level $w0$ deducted from the parent and plays using the same strategy as the parent unless a mutation (with a given rate m) occurs.
- A player also dies if its age exceeds a value *agemax* which was randomly fixed when the player was created.

The CIM Model of the *DPD* game

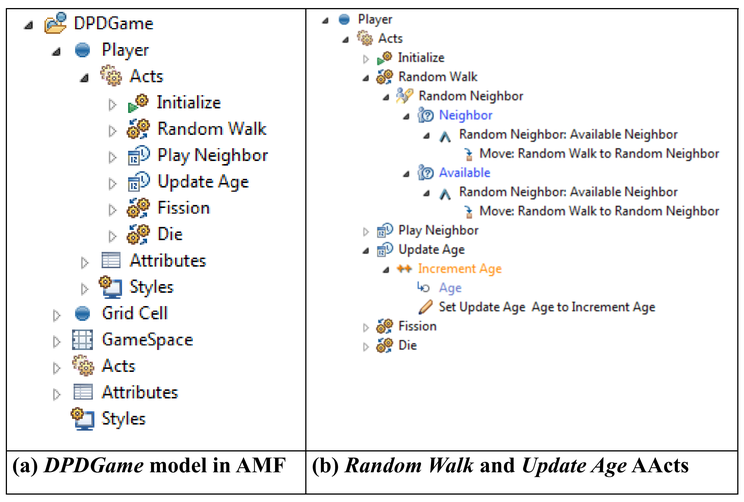


Task Id	Set of Enabling Tasks
T1	-
T2	{T1}
T3	{T1}
T4	{T7}
T5	{T3, T4}
T6	{T2}
T7	{T6}

COMPOSITION TASK RULES

THE UML ACTIVITY DIAGRAMS OF THE PLAYER AGENT TASKS

The AMF-based PIM model of the *DPD* game



(a) ***DPDGame*** **model in AMF** (b) ***Random Walk*** **and** ***Update Age*** **AActs**

AAct	AAct Execution Setting	Tasks
Random Walk	<1,1, a>	T1
Play Neighbor	<1,1, b>, with b<a	T2, T6, T7
Update Age	<1,1, c>, with c<a	T3
Fission	<1,1, d>, with d<c & d<b	T4
Die	<1,1, e>, with e<d	T5

GROUP OF ACTS (AGROUP) FOR THE PLAYER AGENT

- Starting from this definition of the PIM model, **AMF is able to automatically generate the PSM models and the related code** for the ABMS platforms which are currently supported: **Repast Simphony**, **Ascape** and **Escape**.
- The simulation of the system can then be executed in a target simulation environment and **simulation results can be thoroughly analyzed by exploiting several analysis tools** (as Matlab, R, VisAd, iReport, Jung) which can be directly invoked from the environment.

4. Conclusions and Future Work

- The proposed OMG MDA-based process enables ABMS practitioners to obtain **platform-independent agent-based models** which are **easy to verify** (as they conform to meta-models), **modify and update** (as they are mainly constituted by visual diagrams based on the UML notation) with **significantly reduced programming and implementation efforts** (as the simulation code is automatically generated from the defined models).
- Ongoing research efforts are underway devoted to the definition and extensive experimentation of a **full-fledged ABMS methodology** which, by exploiting the MDA-based approach, is **able to seamlessly guide domain experts from the analysis of a complex system to its agent-based modeling and simulation**.