A DSL for Continuous-Time Agent-Based Modeling and Simulation

Tom Warnke

Motivation
Most state-of-the-art agent-based modeling and simulation frameworks offer a way to describe agent behavior in a programming language. Whereas these frameworks support easy development of time-stepped models, continuous-time models can only be implemented by manually scheduling and retracting events. To facilitate a separation of concerns into model- and simulation-specific code for continuous-time ABMs, we propose an embedded domain-specific language, which allows describing agent behavior concisely, and corresponding simulation algorithms, which allow executing continuous-time models.

Example SIR Model
- Agents are connected in a network
- Agents are either susceptible, infectious, or recovered
- Initially, agents are susceptible or infectious
- Susceptible agents get infected after a random waiting time that depends on the number of infectious network neighbors
- Infectious agents recover after a random waiting time

Implementing the SIR model without and with the DSL

The SIR model in vanilla Repast Simphony
138 lines of Java code in the agent class, of which 53 are behavior definition

The SIR model in Repast Simphony with the DSL
42 lines of Java code in the agent class, of which 10 are behavior definition

Results
- Compact description of continuous-time agent-based models
- Syntax inspired by rule-based modeling languages
- CTMC semantics with semantically sound simulation algorithms (two so far)
- Arbitrarily complex functions can be used inside the rules
- Event queue, observation, visualization etc. from Repast Simphony is available

References

Tom Warnke | tom.warnke@uni-rostock.de
MODELING AND SIMULATION GROUP
UNIVERSITY OF ROSTOCK