Extending the Gillespie's Stochastic Simulation Algorithm for Integrating Discrete-Event and Multi-Agent Based Simulation

Sara Montagna, Andrea Omicini, **Danilo Pianini**

{sara.montagna, andrea.omicini, danilo.pianini}@unibo.it

Alma Mater Studiorum—Università di Bologna a Cesena

The XVI International Workshop on Multi-Agent Based Simulation

May 5, 2015 - Istanbul, Turkey

1. Background
   - Discrete Event Simulation (DES)
   - Agent Based Modelling (ABM) and Simulation (MABS)
2. Motivation
3. A Unified Stochastic Computational Model
   - Model
   - Engine
4. Tools: The ALCHEMIST Platform
5. Conclusion

DES

. . . The most used approach in the simulation mainstream

- Instantaneous events responsible for the changes in the system state
- In between events, no change to the system is assumed to occur
- Different events cannot be simultaneous
- It is normally very efficient since it allows to jump in time from one relevant event

Defining a DES means to model the behaviour of a system as an ordered sequence of non-continuous events, by specifying for each of them the perturbations in the system state it provokes, and the exact point in time when it has to be triggered.

MABS I

MABS (Multi-Agent Based Simulation) was introduced as a novel and alternative approach to Discrete Event Simulation (DES) and to Systems Dynamic (SD)

- Rather than a simulation method, it is a simulation **paradigm**
- ABM models the system at the *micro-level*
  - active entity are autonomous and interacting agent
  - dynamic environment
  - global system-level behaviour as a result of the agent-to-agent interactions

MABS II

The simulation platform

- Various platforms were developed for the purpose [All10],
  - tools for developing, editing, and executing ABM, as well as for visualising the simulation dynamic.
- How do they operate over a timeline?
- How are agents and environment behaviours coupled and scheduled?

Crucial Issue: Time

How to deal with the evolution of time?

Continuous approaches

- Rare
- Specifically used for modelling the endogenous dynamic of the environment coupled with discrete agent internal processes.

Discrete approaches

- Time-driven
  - most widely used
  - fixed time steps
- Event-driven

Motivation: Why not Time-driven?

Efficiency

- Time passes fixed time steps
  - even if no action changes the state happen in between
- Carefully choice of temporal granularity
  - low granularity may ruin results
  - high granularity may lead to a waste of computational resources

Accuracy, validity, coherency

- All the events happening in the same Δt are executed together
  - possible loss of order and changes on the system outcome
  - to be as close as possible to the MAS paradigm, actions and interactions should be conducted concurrently

Congruence

- Simultaneous updates of system entities is far from reality

From the Time-driven Drawbacks to Our Contribution

Based on two intuitions

- a unified conceptual framework [Omi15]
  - coherent interpretation of event-driven and multi-agent systems
- powerful simulation framework [Mey14]
  - from the integration of DES and MABS

. . . and one observation

- Gillespie's SSA [Gil77] as an event-driven algorithm

Our proposal

A computational model integrating DES and MABS based on an extension of the Gillespie's stochastic simulation algorithm.

Gillespie's SSA as an Event-driven Algorithm

Gillespie's algorithm

- Gillespie first proposed an event driven stochastic simulation algorithm (SSA) for the exact stochastic simulation of chemical systems
- Gibson and Bruck [GB00] improved its performance
  - Next reaction selection not by propensity (function of concentration of reagents and a markovian rate) but by generated putative times
  - Dependency graph meant to update only the events whose scheduling time might have changed because of other events
- Building on their work, we extended the algorithm in order to be able to shift from the world of chemistry to the richer MABS world

Generalised Chemistry

Pure chemistry vs. agent-based systems

- Single, static compartment versus multiple, possibly mobile, and interconnected agents whose ability to communicate may depend on environmental and technological factors
- Molecules are described by concentrations (an integer), agents may carry and process any kind of data
- Reactions "scheduling" in nature follows a Poisson distribution whose rate equation depends on reagents' concentration [Gil77]. Events in an agent-based simulation may be influenced by any of the environment components and follow any probability distribution (triggers, timers, events with memory)
- Agents live in an environment, such abstraction is absent in chemistry

A huge leap, indeed

Close the Gap: Environment

Close the Gap: Reactions

Flexibility and Data Types

Abstract Concentration

- Concentration can be any data type
  - Pick integers, the result is a simulator for (bio)chemistry, with multiple intercommunicating compartments situated in an environment [MPV12]
  - Pick "set of tuples matching a tuple template", the result is a simulator of network of programmable tuple spaces
  - Pick "any object", the result is flexibile enough to simulate a network of devices running their own program [PVB15]
- For each type of concentration, a specific set of legal conditions and actions can (must) be defined
- All the other entities can be defined in a generic fashion, and reused

Extended SSA—Phase 1: Pick the Next Event

How to select the next event?

- Most high-performance SSAs presume an underlying model that only includes memoryless events [STP08]
- Gibson/Bruck's "next reaction" uses putative times instead
- We extended it adding support for addition and removal of events at runtime
  - Agents may join and leave the system at runtime, new agents may be equipped with novel behaviours

Extended SSA—Phase 2: Dependency Management

How to select the next event?

- The dependency graph is key for the high performance of SSAs [STP08]
- In general, it is very hard to build a dependency graph in an open environment composed of multiple entities
- We extended the original concept of (static) dependency graph with:
  - Events can be added and removed at runtime, the graph is dynamically updated
  - Execution contexts: `local`, `neighborhood`, `global`
  - Separation of influencing context and context of influence (input and output)
  - Overall, the dependency graph is greatly pruned, with positive impact on performance

The ALCHEMIST Platform as a Tool [1]

ALCHEMIST: A chemistry-inspired meta-simulator

- Based on the machinery already described
- Java written
- Provides out of the box support for simulating distributed programmable tuple spaces and Protelis [PVB15] devices
- Supports mobility and complex environments, both indoor and outdoor (with data from OpenStreetMap)
- Available as Maven artifact (`it.unibo.alchemist:alchemist`)

[1] `http://alchemist.apice.unibo.it`

Architecture of ALCHEMIST

Case Study: Crowd-sensitive User Steering in London

(a) Initial (b) Ongoing (c) Near-end

Figure: Three snapshots of the simulation. Our altered distance metric (the gradient) goes from green (most favourable and close to destination) to red (unpleasant or far from destination). Agents (in black) climb this structure towards greener values.

Conclusion

Integration of DES and MABS

- We adopted an extension of Gillespie's SSA as stochastic event-driven algorithm
- We extended it to support the inherent complexity of multiagent systems, still retaining the performance optimisations
- We extended the chemical model towards higher flexibility, introducing the environment, generalising the concept of reaction and allowing arbitrary data to be a "concentration"
- We implemented those concepts inside the ALCHEMIST framework
- A non-trivial example was provided: crowd steering in London

References I

Rob Allan.
Survey of agent based modelling and simulation tools.
Technical Report DL-TR-2010-007, Computational Science and Engineering Department, October 2010.

Michael A. Gibson and Jehoshua Bruck.
Efficient exact stochastic simulation of chemical systems with many species and many channels.
J. Phys. Chem. A, 104:1876–1889, 2000.

Daniel T. Gillespie.
Exact stochastic simulation of coupled crfreactions.
The Journal of Physical Chemistry, 81(25):2340–2361, December 1977.

Ruth Meyer.
Event-driven multi-agent simulation.
In *Electronic Proceedings of the Fifteenth International Workshop on Multi-Agent-Based Simulation*, 2014.

References II

Sara Montagna, Danilo Pianini, and Mirko Viroli.
A model for drosophila melanogaster development from a single cell to stripe pattern formation.
In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, SAC '12, pages 1406–1412, New York, NY, USA, 2012. ACM.

Andrea Omicini.
Event-based vs. multi-agent systems: Towards a unified conceptual framework.
In *2015 19th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD2015)*. IEEE Computer Society, May 2015.

Danilo Pianini, Mirko Viroli, and Jacob Beal.
Protelis: Practical aggregate programming.
In *30th Symposium On Applied Computing (SAC 2015)*. ACM, 2015.

Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton.
A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks.
The Journal of Chemical Physics, 128(20):205101, 2008.