A Gillespie-based Computational Model for Integrating Event-driven and Multi-Agent Based Simulation

(Extended Abstract)

Sara Montagna Andrea Omicini Danilo Pianini
Alma Mater Studiorum–Università di Bologna via Sacchi 3, 47521 Cesena, Italy
{sara.montagna,andrea.omicini,danilo.pianini}@unibo.it

ABSTRACT

Based on two intuitions – *(i)* event-driven systems and multi-agent systems are two computational paradigms that are amenable of a coherent interpretation within a unique conceptual framework; *(ii)* integrating the two simulation approaches can lead to a more expressive and powerful simulation framework – we propose a computational model integrating Discrete-Event Simulation (DES) and Multi-Agent Based Simulation (MABS), based on an extension of the Gillespie's stochastic simulation algorithm.

Categories and Subject Descriptors

I.6 [**Computing Methodologies**]: Agent/discrete models; I.6 [**Computing Methodologies**]: Discrete-event simulation

General Terms

Theory

Keywords

Multi-agent based simulation, event-driven simulation, stochastic simulation, Gillespie algorithm

1. MOTIVATION

One crucial issue in Multi-Agent Based Simulation (MABS) is how to deal with the evolution of time. The model for relating agent actions with the dynamics of the environment can be either *continuous* or *discrete*. Continuous approaches are very rare and, in case, specifically used for modelling the endogenous dynamic of the environment coupled with discrete agent internal processes. In discrete approaches, time can evolve with respect to regular intervals (time steps) or to event executions (time is increased of any real number along a continuous time line from one event to the next one) [3]. Among such techniques, the most widely used is the discrete approach with fixed time steps (hereafter called *time-driven*) [5, 2].

1.1 Time-driven drawbacks: A comparison with DES

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

Efficiency | The time-driven approach is definitely less efficient of an event-driven approach: t requires to pass by fixed time steps even if no actions are scheduled to be executed for changing the system state. Thus, modellers are required to choose the temporal granularity of actions: this normally corresponds to the fastest events, but it could obviously be a problem in those systems with a wide spectrum of time scales, such as in stiff systems which possibly require an inefficient allocation of computational resources. In this respect, an event-driven approach can really improve the efficiency of a simulation by skipping inactive phases.

Accuracy/Validity/Coherency | To be as close as possible to the MAS modelling paradigm, agents are supposed to conduct their behaviour (actions and interactions) concurrently, so that also simultaneous actions are feasible. Moreover they must be coupled with a possibly dynamic environment. In a time-driven approach, from one instant t, the state at $t + \Delta t$ is the result of the combination of all the agent actions scheduled for the fixed interval Δt with the environment evolution expected in the same interval. However, the property of concurrency is lost: as a consequence, whereas agent and environment actions are interdependent, the order at which they are performed can radically change the overall result, *i.e.* how the state of the system changes in that interval and, in turn, it can lead to an overall dynamic behaviour of the system over time that is different from one run to the other. Whereas some works have already discussed this issue, proposing different solutions (e.g., [3]), an event-driven approach can be a very good compromise: it partially solves the problem by confining the problem just to those rare actions that are expected to be triggered simultaneously.

Congruence | The approximation of the reality in which all the entities of the system are updated simultaneously, as for the time-driven approach, seems to be very far to the real behaviour of a complex system.

Accordingly, our intuition is that integrating MABS and DES could be crucial in the simulation of complex systems.

2. A UNIFIED STOCHASTIC COMPUTATIONAL MODEL

We here adopt and extend a novel stochastic computational model [4] for *(i)* integrating a DES scheduler based

on the Gillespie's SSA – and in particular on its extensions developed in [1, 6] – into a MABS toolkit, and *(ii)* specifying the agent (internal and interacting) and environment behaviours in terms of chemical reactions.

2.1 The simulation engine

The simulation engine was enriched in two ways: *(1)* non-Markovian events can be scheduled (schedulers in [1, 6] are designed to simulate Poisson processes); *(2)* the "dependency graph" – namely, the data structure responsible of statically linking each reaction to the set of reactions whose execution speed may be influenced by the execution of the former – has been improved in order to cope with multiple, separate, situated, interacting and possible mobile entities. First of all, possibly interdependent reactions occurring on separated compartments need not be marked as interconnected within the dependency graph. Moreover, in order to manage mobility of node that lead to the creation and disruption of communication channels between compartments, the dependency graph becomes a dynamic data structure that cannot be pre-computed at the simulation start and stay unchanged. In particular, this was obtained by defining the input and output *contexts* for each reaction, namely, the places where reactions respectively "read" their reactants and "write" their products. We adopted three levels: `local`, `neighborhood` and `global`, depending on whether reactions do or do not modify the content of other compartments. Thus, events that are occur far in space are ensured not to trigger any unnecessary update.

2.2 The computational model

The simulation engine, despite its improved flexibility, is still bound to a world made of molecules, reactions, and compartments. Typical ABM models require instead the specification of higher-level concepts, such as agent, internal, and interactive behaviour and environment. In our vision, a compartment represents an agent, and all the possible events in the model must break down to a set of reactions. We argue that there are three main issues to be faced in order to offer a model that can be considered agent based, but still suitable to be simulated by the extended SSA engine presented in the previous section. *(1)* As a first step, we introduce *environment* as a first-class abstraction [7]. The environment, based on arbitrarily complex rules, has the responsibility to provide, for each compartment, a set of compartments that are its neighbours. *(2)* Then, we define a reaction as "a set of conditions that, when matched, triggers a set of actions on the environment". A condition is a function that associates a numeric value ranging from zero to positive infinity to each possible state of the environment. If such a value is zero, the event can not be scheduled; otherwise, it is up to the reaction to interpret the number: it can either influence or not the time at which the reaction will be scheduled, depending on the specific reaction implementation. An action is an arbitrary change of the environment. Examples are: changes in the reactant and product number into local compartments; molecule transfer to neighbouring nodes; node movements. Both conditions and actions should expose the set of possible data items (molecules) that they may read or modify: this is required in order to build the dependency graph. Also, both conditions and actions should expose a context of type `local`, `neighborhood` or `global`; it will be used internally to determine the input and output contexts for the reaction itself. *(3)* Finally, we improve the low expressive power of the concept of the classical concentration. We propose to define "concentration" as "type of data items agents can manipulate", de facto making "concentration" depend on the actual meta-model. Besides the trivial example of chemistry, where data items are integer numbers, a distributed tuple spaces model could be considered: in this case, molecules would be tuples, and concentration would be defined as "the set of tuples matching a certain tuple".

3. CONCLUSIONS

We adopt the Gillespie's SSA, a stochastic event driven algorithm from which we derive a novel model that enables the integration of DES and MABS. The model impacts both the way the scheduling module of a MABS platform is implemented, and the way agents and environment behaviours must be defined. Given its inspiration into the chemistry world, it is grounded on the concepts of chemical reactions and biological compartments. However, such abstractions are suitably extended to properly fill the gap with the typical ABM abstractions, as well as to fit even more general scenarios. To the best of our knowledge, this is one of the first attempts to bring Gillespie into the MABS scientific field. Even though our meta-model is inspired by chemical systems, it could actually be adapted so as to model other kind of systems that feature suitable description in terms of atomic events.

REFERENCES

[1] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *The Journal of Physical Chemistry A*, 104(9):1876–1889, March 2000.

[2] R. Meyer. Event-driven multi-agent simulation. In F. Grimaldo and E. Norling, editors, *Multi-Agent-Based Simulation XV*, Lecture Notes in Computer Science, pages 3–16. Springer International Publishing, 2015.

[3] F. Michel, J. Ferber, and A. Drogoul. Multi-Agent Systems and Simulation: a Survey From the Agents Community's Perspective. In Danny Weyns and Adelinde Uhrmacher, editor, *Multi-Agent Systems: Simulation and Applications*, Computational Analysis, Synthesis, and Design of Dynamic Systems, page 47. CRC Press - Taylor & Francis, May 2009.

[4] D. Pianini, S. Montagna, and M. Viroli. Chemical-oriented simulation of computational systems with Alchemist. *Journal of Simulation*, 2013.

[5] S. F. Railsback, S. L. Lytinen, and S. K. Jackson. Agent-based simulation platforms: Review and development recommendations. *Simulation*, 82(9):609–623, Sept. 2006.

[6] A. Slepoy, A. P. Thompson, and S. 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.

[7] D. Weyns, A. Omicini, and J. Odell. Environment as a first-class abstraction in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, Feb. 2007. Special Issue on Environments for Multi-agent Systems.