# 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

# Two intuitions

## Unique conceptial framework

Event-driven systems and multi-agent systems are amenable of a coherent interpretation within a unique conceptual framework

## Powerful simulation framework

From the integration of Discrete Event Simulation (DES) and Multi-Agent Based Simulation (MABS)

# Motivation: why event driven?

## Efficiency

- Time passes fixed time steps, even if no action changes the state happen in between
- Modellers must carefully choose temporal granularity
- If there is a wide spectrum of time scales, a low granularity may ruin results, while a high granularity may lead to a waste of computational resources

# Motivation: why event driven?

## Accuracy, validity, coherency

- To be as close as possible to the MAS paradigm, actions and interactions should be conducted concurrently
- In a time-driven setup, all the events happening in the same $\Delta t$, are executed (along with the environment evolution) together, possibly losing ordering and changing the system outcome
- Event driven patches the problem, limiting it to those actions that happen at the exact same time.

## Congruence

- Updating all the entities of the system simultaneously is often an approximation too far from reality

# Bulding on SSAs

## Gillespie's algorith

- Gillespie [Gil77] 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 wolrd of chemistry to the richer MABS world

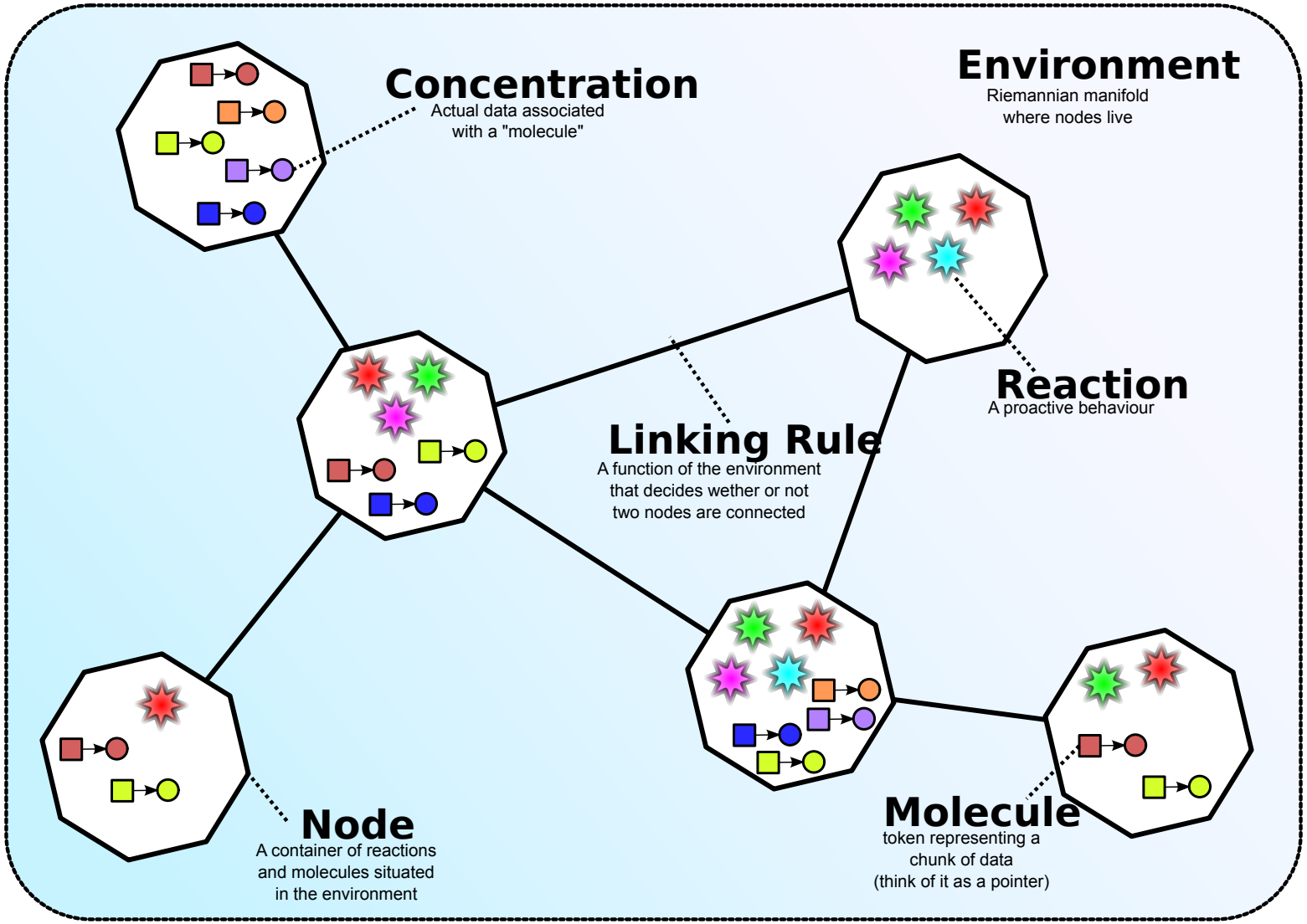
# Generalised chemistry

## Pure chemistry vs. agent-based systems

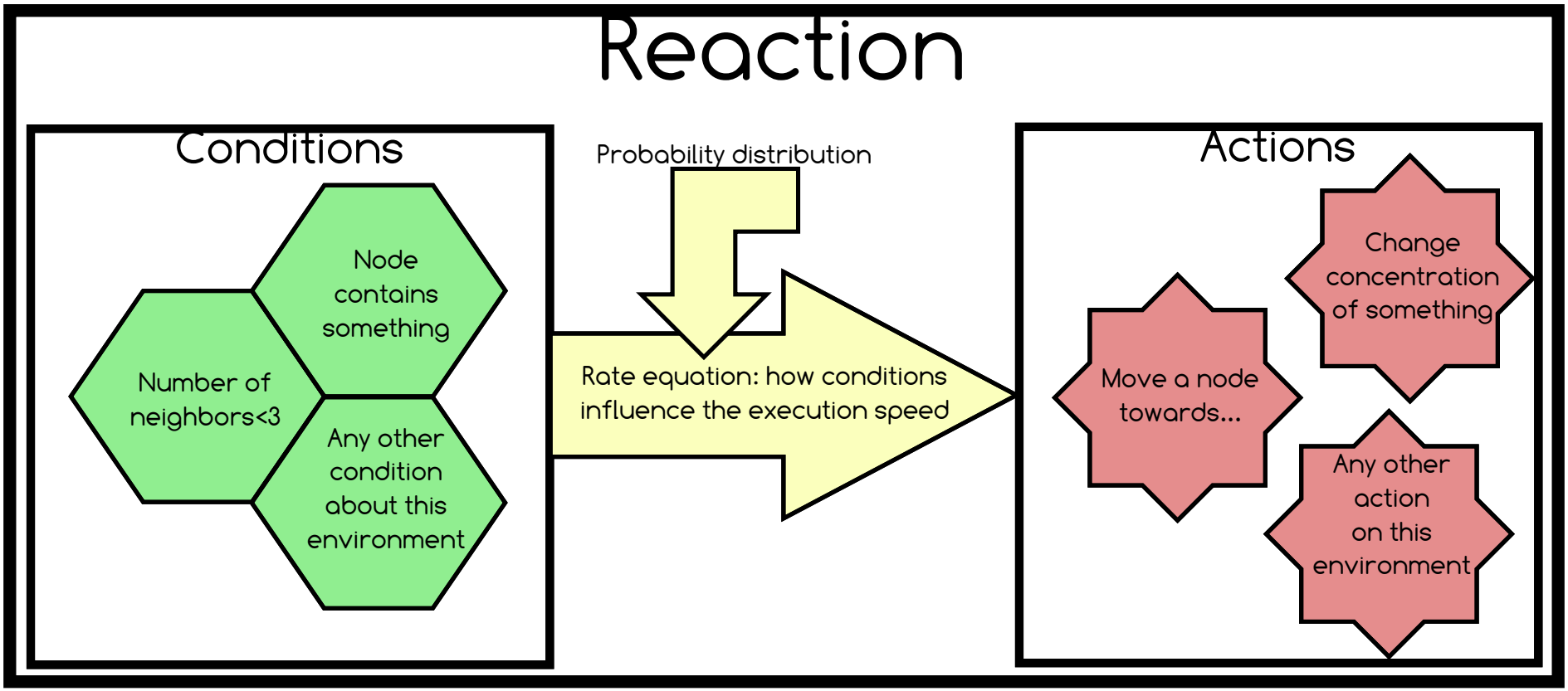
- Single, static compartment versus multiple, possibly mobile, and interconnected agents whose ability to communicate may depend on environmental and technological factors
- Molecules are described by concentrations (an integer), agents may carry and process any kind of data
- Reactions "scheduling" 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

Yes, it is a nicely big leap

# 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?

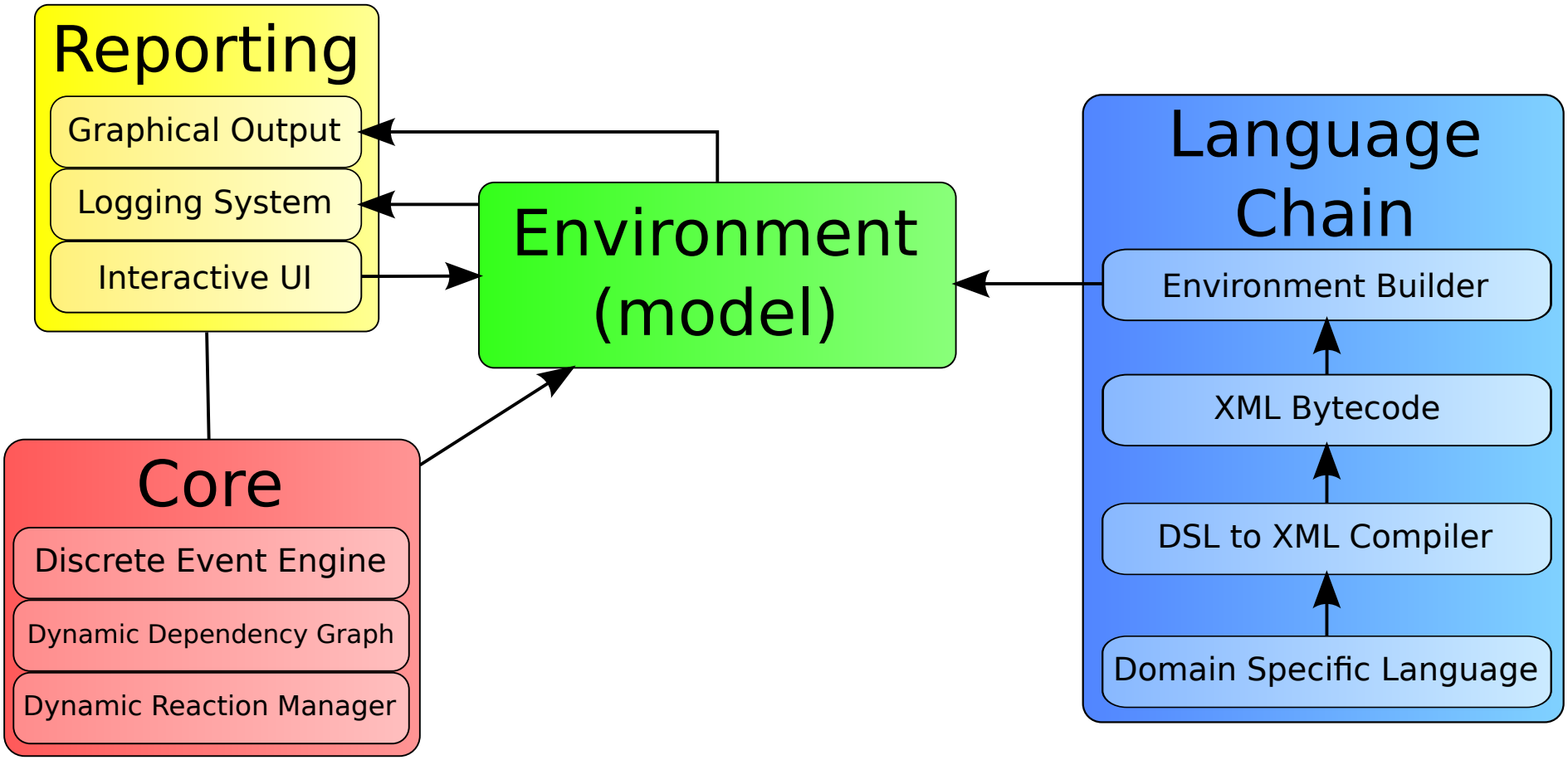
- The dependency graph is key for the high performance of SSAs [STP08]
- In general, it is very hard to build a dependency graph in an open environment composed of multiple entities
- We extended the original concept of (static) dependency graph with:
  - Events can be added and removed at runtime, the 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

# Alchemist

## Chemical-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`)

# Architecture



# Crowd-sensitive user steering

## Steering against GPS traces taken at Vienna City Marathon 2013

# 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 enviromemnt, 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


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.

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.

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


# References II

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.