

The simulation alchemy

Danilo Pianini – danilo.pianini@unibo.it

ALMA MATER STUDIORUM—Università di Bologna

Cesena, May 8th, 2012



1 Simulation

- Definitions
- Models
- When it's useful?

2 Alchemist

- SAPERE background
- Computational model
- Engine and architecture
- Case studies
- Performance

3 Development

- Development model
- Contributors
- Future activities



Simulation



Scientific Method

Traditional science workflow [Parisi, 2001]

- Traditional scientific method
 - identification
 - direct observation
 - theories / hypothesis
 - empirical observation
 - quantitative analysis
 - validation / invalidation



Definition of Simulation

- A new way for describing scientific theories

[Parisi, 2001]

- Simulation is the process with which we can study the dynamic evolution of a model system, usually through computational tools

[Banks, 1999]

- Simulation is the imitation of the operation of a real-world process or system over time



Simulation Requires a Model

M. Minsky – Models, Minds, Machines

A model (M) for a system (S), and an experiment (E) is anything to which E can be applied in order to answer questions about S .

- Representation / abstraction
- Formalisation
- Aggregation, Simplification, Omission

Building a model...

- How complex should be the model?
- Which assumptions should be done?



From Model to Simulation...

Computer simulation

- Models are designed runnable processes
- Simulation creates a virtual laboratory
 - controlled conditions
 - it's easy to modify the experiment (variables, parameters, etc.)
- Simulations imitate the operations of the modelled process
 - generation of an artificial evolution of the system

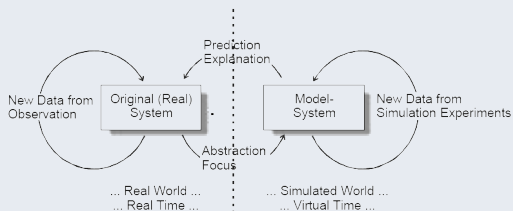


... and Back

- Deductions on the real system represented
- Evaluation of theories about the model

Model validation [Klugl and Norling, 2006]

- if the data is reliable;
- if prediction doesn't match the observed behaviour do not match
- $\Rightarrow$ the model must be revised



Why do we Need Simulations?

[Parisi, 2001, Klugl and Norling, 2006]

- The real system cannot actually be observed
- The time scale of the real system is too small or too large for observation
- The original system doesn't yet exist (or not anymore)
- The system is too complex



Alchemist



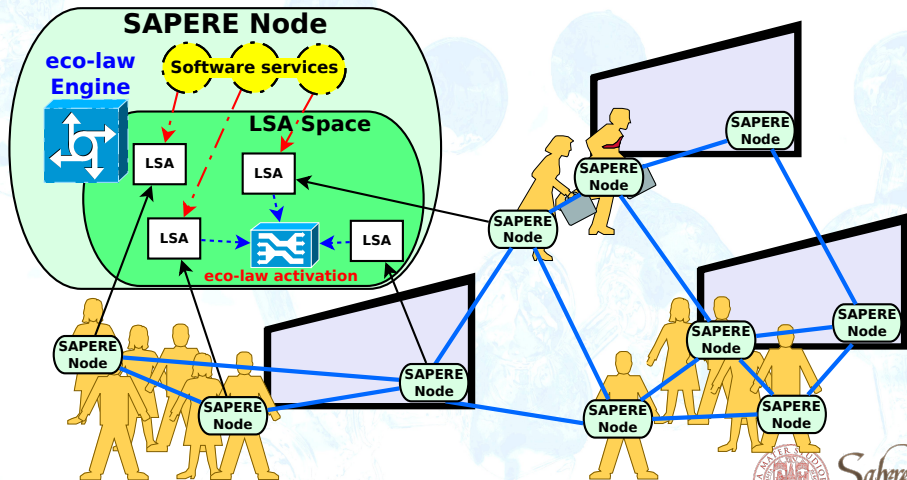
The SAPERE Project

<http://www.sapere-project.eu/>

- SAPERE (Self-aware Pervasive Service Ecosystems) is an EU STREP project under the FP7 FET Proactive Initiative: Self-Awareness in Autonomic Systems (AWARENESS)
- The objective of SAPERE is the development of a highly-innovative theoretical and practical framework for the decentralized deployment and execution of self-aware and adaptive services for future and emerging pervasive network scenarios



SAPERE World



Simulation of a SAPERE environment

The role of simulation

- Emergence cannot be fully designed
- It's crazy to deploy a whole ecosystem without any test

Fist class abstractions

- Dynamic environment
- Different, mobile, communicating nodes
- Programmability through a set of chemical-like laws
- Continuous Time Markov Chain (CTMC) model
- Autonomous agents



Two approaches

Classic ABM modelling

- High flexibility
- Topology as first-class abstraction
- Dynamics explicitly modelled
- No native support for CTMC model

Chemical inspired modelling

- Natively CTMC
- Very fast and reliable algorithms exist in literature
- Very limited topology: multicompartment at best
- Only classic reactions can change the world status: limited flexibility



Alchemist simulation approach

Base idea

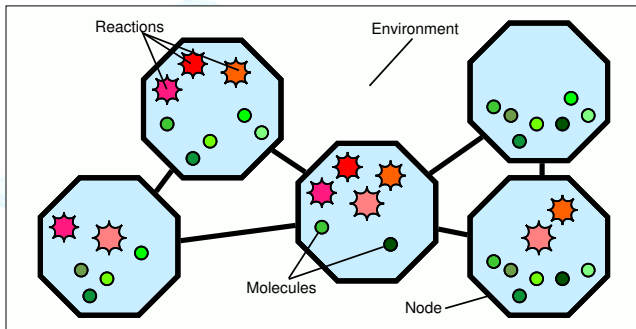
- Start from the existing work with stochastic chemical systems simulation
- Extend it as needed to reach desired flexibility

Goals

- Full support for Continuous Time Markov Chains (CTMC)
- Support for differently distributed events (e.g. Triggers)
- Rich environments, with obstacles, mobile nodes, etc.
- More expressive reactions
- Fast and flexible SSA engine



Enriching the environment description

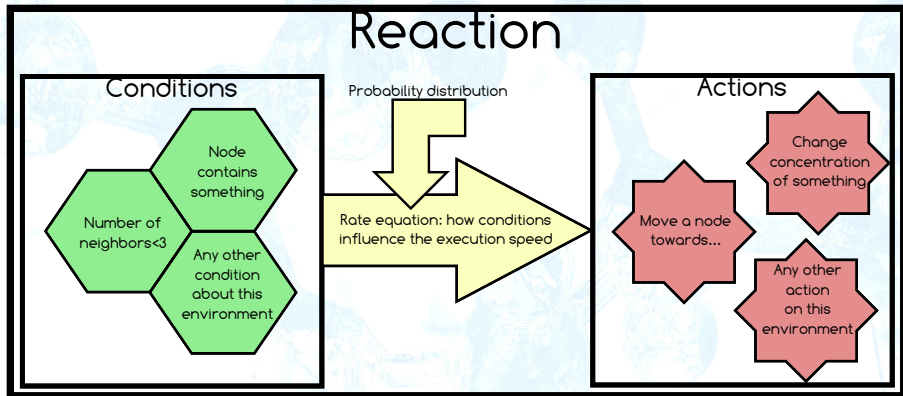


Alchemist world

- The Environment contains and links together Nodes
- Each Node is programmed with a set of Reactions
- Nodes contain Molecules
- Each Molecule in a node is described with a Concentration

Extending the concept of reaction

From a set of reactants that combine themselves in a set of products to:



In Alchemist, every event is an occurrence of a Reaction



Making efficient SSA Algorithms more flexible

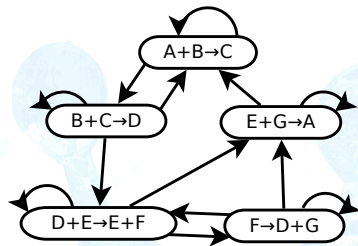
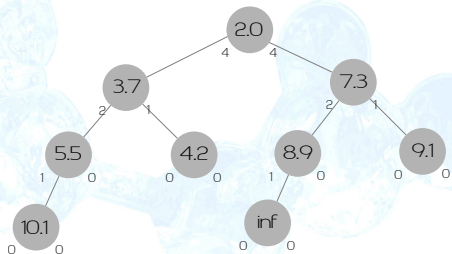
Existing SSA algorithms

- Several versions, but same base schema [Gillespie, 1977]:
 - 1 Select next reaction using markovian rates
 - 2 Execute it
 - 3 Update the rates which may have changed
- Very efficient versions exist: logarithmic [Gibson and Bruck, 2000] and constant [Slepoy et al., 2008]

What they miss is what we added

- Reactions can be added and removed during the simulation
- Support for non-exponential time distributed events (e.g. triggers)
- Dependencies among reactions are evaluated considering their “context”, speeding up the update phase

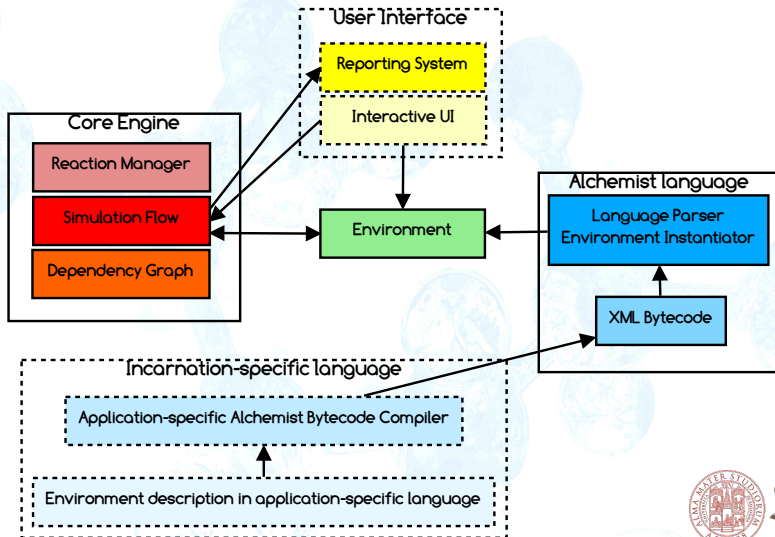
Smart data structures $\Rightarrow$ bleeding edge performances



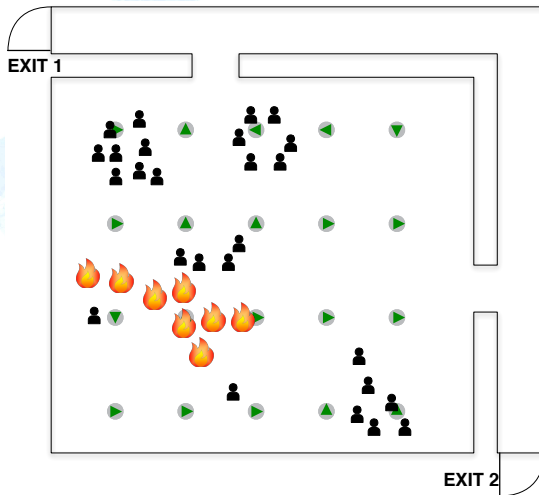
Next Reaction efficient structures made dynamic

- Dynamic Indexed Priority Queue
 - Allow to access the next reaction to execute in $O(1)$ time
 - Worst case update in $\log_2(N)$ (average case a lot better)
 - Extended to ensure balancing with insertion and removal
- Dynamic Dependency Graph
 - Allows to smartly update only a subset of all the reaction
 - Extended with the concept of input and output context

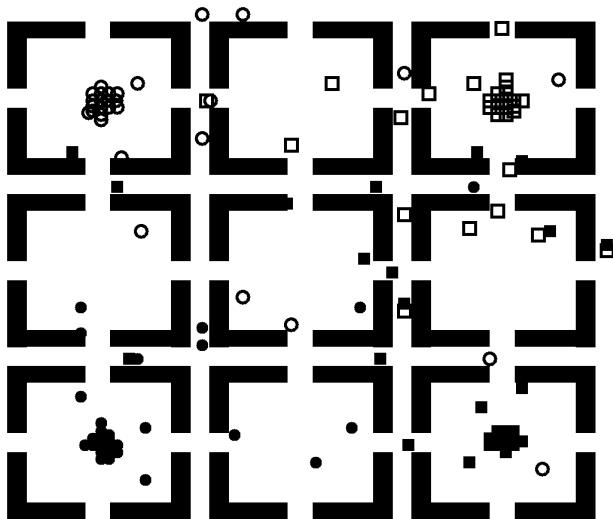
Alchemist is fully modular



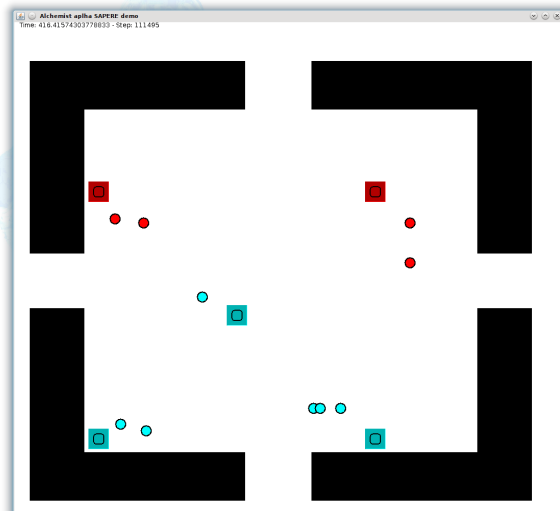
Crowd evacuation



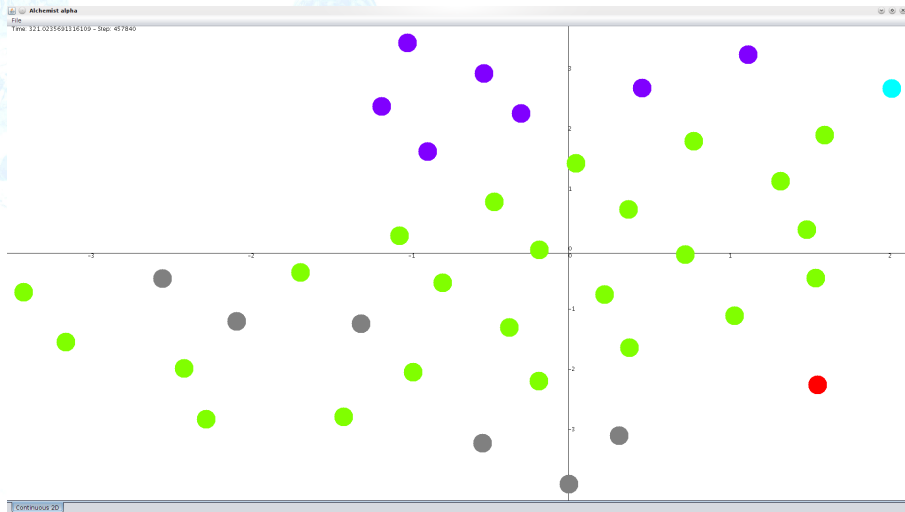
Crowd steering



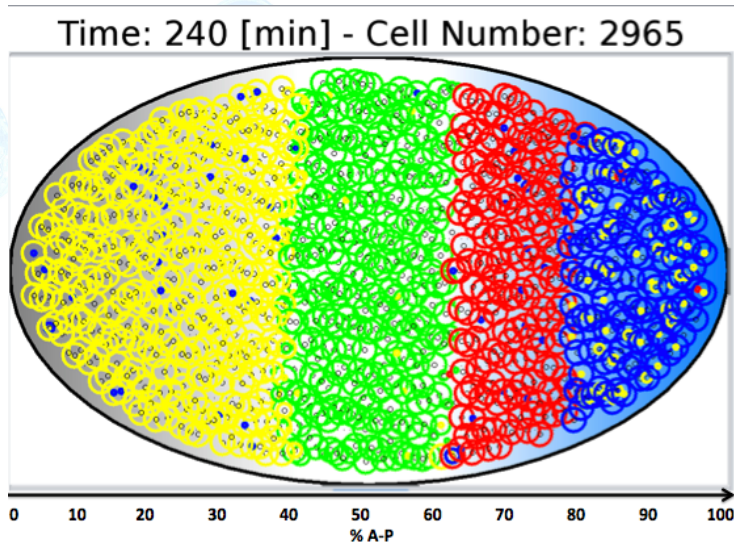
Adaptive Displays



Simple Morphogenesis proof of concept



Morphogenesis of a *Drosophila Melanogaster*



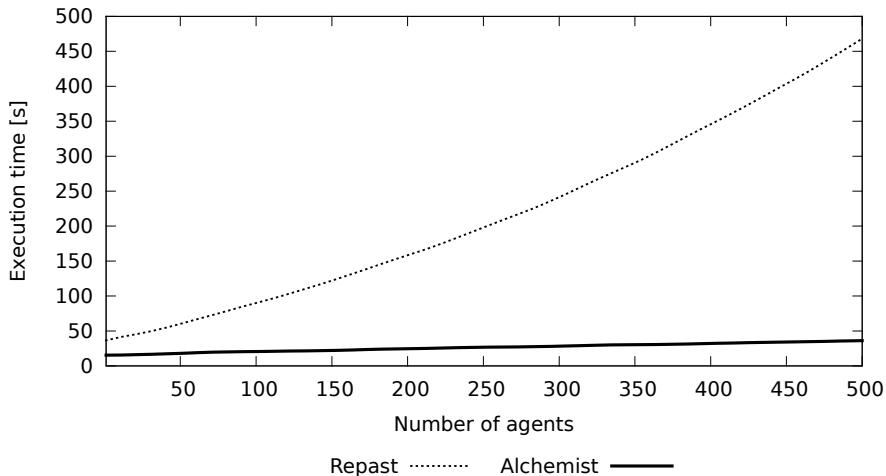
Testbed

- We realised the same crowd-steering application for both Alchemist and RePast, in order to evaluate the performance gap (if any)
- Source code for RePast and standalone Alchemist application available at
<http://www.apice.unibo.it/xwiki/bin/view/Alchemist/JOS>
- We choose a simplified use case which allows simulation in RePast without any change in its engine
 - This actually cripples out the most important Alchemist optimization for complex environment (the dependency graph)



Results

Performance comparison with Repast



Development



Distributed development

Alchemist is a training ground for some good team development practices

- Linux-like flat development model
- Java
- XText
- Mercurial DCVS
- Bitbucket web-based code hosting

More awesomeness will come!

- OSGi
- Maven
- JUnit
- Source will soon be released under GPL



Some students are already involved

- **Francesca Cioffi** (Ongoing)
 - Further experiments with Alchemist-SAPERE
- **Enrico Galassi** (Ongoing)
 - Alchemist-SAPERE high level language
- **Alessandro Montalti** (Done)
 - MS-BioNet to AlchemistXML translator
- **Giacomo Pronti** (Done)
 - SAPERE incarnation main author



There is a lot of work to do!

- Parallel “revolver”
- Stochastic model checking tools
- MS-Bionet compatibility layer
- Image-to-environment tool
- OpenStreet Map importer and Google Map importer
- (Interactive) Reporting System
- Blender Integration
- User guide
- CellML / SBML to AlchemistXML
- Advanced diffusion mechanisms

These are examples, if you have something in mind, be proactive!



Bibliography I



Banks, J. (1999).

Introduction to simulation.

In Farrington, P., Nembhard, H. B., Sturrock, D. T., and Evans, G. W., editors, *Proceedings of the 1999 Winter Simulation Conference*, pages 7–13.



Gibson, M. A. and Bruck, J. (2000).

Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels.

The Journal of Physical Chemistry A, 104(9):1876–1889.



Gillespie, D. T. (1977).

Exact stochastic simulation of coupled chemical reactions.

Journal of Physical Chemistry, 81(25):2340–2361.



Bibliography II



Klugl, F. and Norling, E. (2006).

Agent-based simulation: Social science simulation and beyond.
Technical report, The Eighth European Agent Systems Summer School (EASSS 2006), Annecy.



Parisi, D. (2001).

Simulazioni - La realtà rifatta al computer.
Società editrice il Mulino.



Slepoy, A., Thompson, A. P., and Plimpton, S. J. (2008).

A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks.

The Journal of Chemical Physics, 128(20):205101+.



The simulation alchemy

Danilo Pianini – danilo.pianini@unibo.it

ALMA MATER STUDIORUM—Università di Bologna

Cesena, May 8th, 2012

