

Alchemist simulator

Danilo Pianini – danilo.pianini@unibo.it

ALMA MATER STUDIORUM—Università di Bologna

Frankfurt, February 27th, 2012



- 1 Main goal
- 2 What's already there
- 3 What will be there soon
- 4 Performance comparison with RePast
- 5 Conclusion



Allow users to simulate SAPERE systems by writing directly eco-laws, LSAs and agents behaviour. No gap between what would be expressed in the real system and what is simulated.

- Faster deployment of simulations
- With other systems, you may end up simulating something which is not exactly the final system
- Concrete risk to spend hours developing simulations without knowing if (and under which conditions) fit the SAPERE one



Current status



Environments

Although multiple environments are available, those specifically meant for SAPERE are:

- Continuous
- Allow for node mobility
- Auto link nodes within a range
- Support the definition of obstacles (walls, doors...)



Eco-laws

Eco-laws can be written as String inside the Java code:

```
LsaExpTimeReaction.build(  
    "<An,Lsa>,<Another,Lsa, [with, a, list, of Elements]>",  
    rate,  
    "<Lsa>,*<Broadcast>,+<Unicast>");
```

Differences from D1.1:

- position of items is relevant
- reading on neighbourhood is allowed
- broadcast is supported

Our experience is that once that eco-laws are expressed in D1.1 language, translation is pretty easy and fast.



Agents

Agents behaviour is outside the model, however it must be defined in order to simulate

- written in Java
- Can read and write LSAs
- Can move the node within the environment
- The same node can host multiple agents and eco-laws

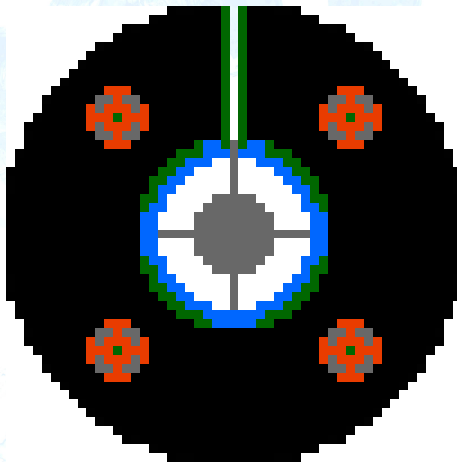


Next future



Environments

We plan to support the instantiation of environments by “parsing” 2D bitmap images. This should guarantee very fast deployment of even complex environments



Higher level language

There is ongoing work to support a full environment definition by an high level language:

```
Environment 10.0 10.0 from "/home/user/image.png"  
lsa pattern pump <source,Nmax>  
lsa pattern field <field,X>  
place 3 nodes in Rect (0.0,0.0,5.0,5.0)  
define lsaSpace IN ALL pump; field;  
with reaction  
[pump] -->[pump, +field] rate "100"  
AT TIME "10" [field] -->[*<"field,X-#D">] rate "X"
```



Documentation

The tool will be soon documented, around a month is needed for a good user guide.



Performance



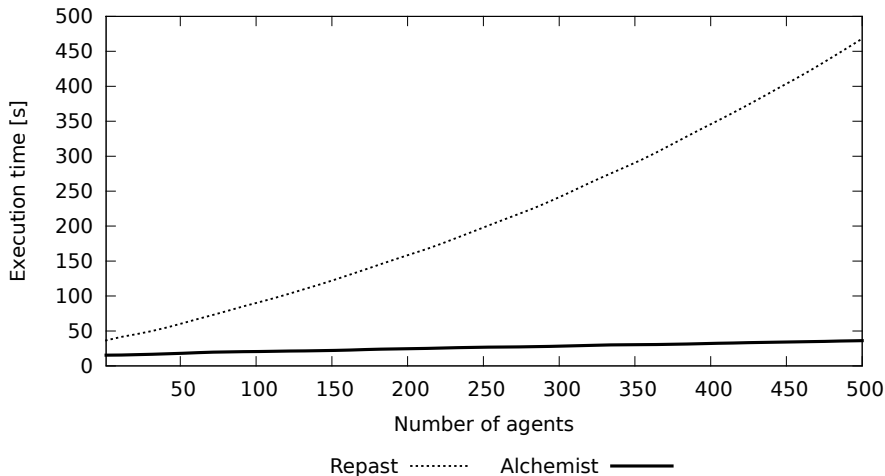
Testbed

- 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



Conclusion

- Alchemist is part of the WP1 (Metodology and analysis suite)
- It will be completely usable in few weeks
- We are available for a tutorial, to who is interested in using it
- We propose Alchemist as a reference simulation tool for SAPERE



Alchemist simulator

Danilo Pianini – danilo.pianini@unibo.it

ALMA MATER STUDIORUM—Università di Bologna

Frankfurt, February 27th, 2012

