Composing gradients for a context-aware navigation of users in a smart-city

Sara Montagna Mirko Viroli

Email: {sara.montagna,mirko.viroli}@unibo.it

ALMA MATER STUDIORUM—Università di Bologna, Cesena

Spatial Computing Workshop at AAMAS 2013 (SCW 2013)

Saint Paul, Minnesota, USA, 6th May 2013

Outline

The SAPERE project – AWARE Coordination action[1]

Self-aware Pervasive Service Ecosystems (www.sapere-project.eu)

- An EU Project under the "Autonomic Computing" umbrella
- Partners (F.Zambonelli, G.d.Marzo, S.Dobson, A.Ferscha)

Main ideas of the project

- A new execution/deployment model for pervasive computing services
- Ecosystem view: open injection of services, self-awareness
- Relying on an engine of "semantic chemical reactions"
- Reference application scenario: pervasive display ecosystems

[1] Franco Zambonelli et al. "Self-aware Pervasive Service Ecosystems". In: *Procedia CS* 7 (2011), pages 197–199.

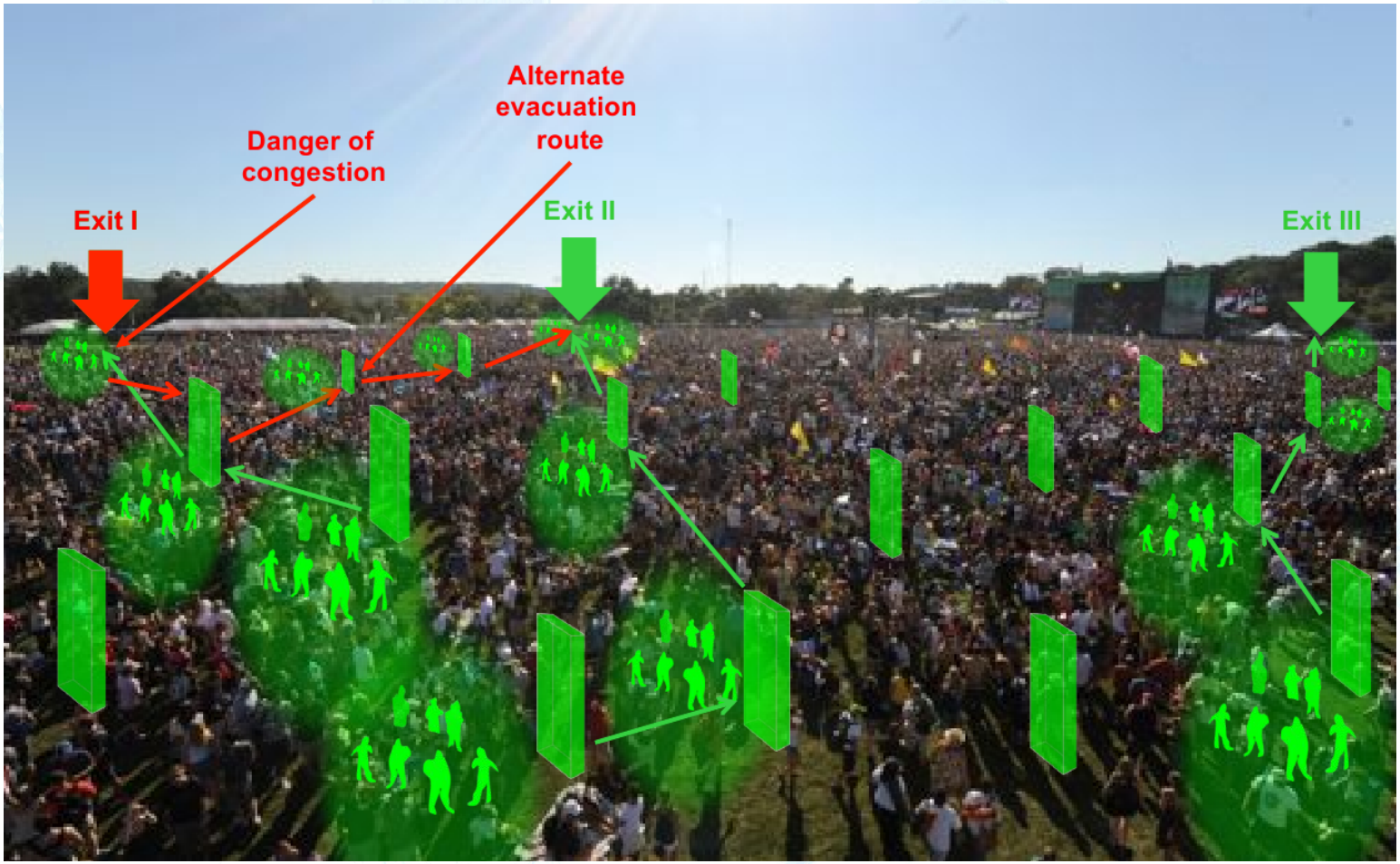
Steering people in pervasive computing

Unsupervised guide of people in complex environemnts

- Consider an articulated environment (e.g. a city, a museum)
- Devices with sensing/interaction abilities fill the environment
- POIs appear, and are advertised by gradients
- People are steered to the POI by public/private displays

(Pict. by A. Ferscha, JKU)

Towards Crowd Engineering



(Pict. by A. Ferscha, JKU)

Context is key!

Contextual services

- provide information about the environmental context
  - congestion, pollution, street slopes, environment (pedestrian area, industrial zone, residential district...), presence of other POIs (shops, parks, bars).
- can be opportunistically used by the steering service
- are dynamically injected, with no a-priori knowledge of
  - position, perception (positive / negative), data structure

Context-aware navigation

Problem: find mechanisms to make contextual services affect opportunistically the standard gradient-based navigation service

Outline

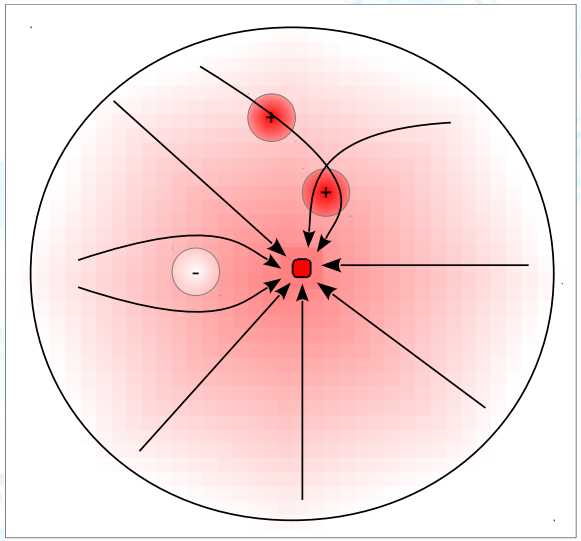
Context-aware navigation by gradient compositions

Key idea

- Achieve context-aware navigation compositionally
- Compose contextual services and the basic navigation service
- Obtain new gradient-like fields and steering criteria

$\Rightarrow$ Dynamically provide a new set of "tailored" navigation services

$\Rightarrow$ Let such a composition be multi-level and user-centric



Gradient (Self-)Composition

Aspects of composition

- Service Composition in SOA, Evolutionary techniques,. . .
- Competition-based approaches[ab] – prey-predator dynamics

[a] Mirko Viroli. "On competitive self-composition in pervasive services". In: *Science of Computer Programming* 78.5 (2013), pages 556 –568.

[b] Mirko Viroli, Matteo Casadei, Sara Montagna, and Franco Zambonelli. "Spatial Coordination of Pervasive Services through Chemical-inspired Tuple Spaces". In: *ACM Transactions on Autonomous and Adaptive Systems* 6.2 (2011), 14:1–14:24.

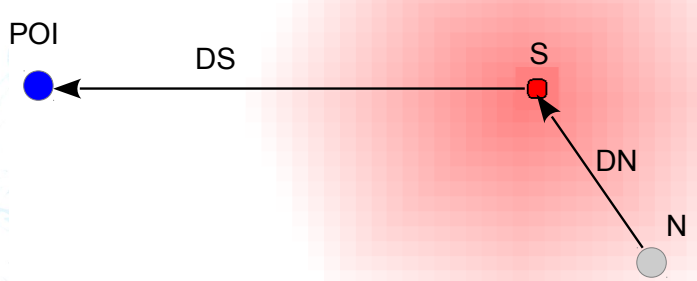
An ecology-inspired model (SAPERE approach)

1. Spatial composition of gradients
2. Steering service exploits (semantic) matching with user profile
3. Selection by competition of meaningful (composed) services
   - Taming the huge number of possible compositions

Model: part 1

Composition of gradients

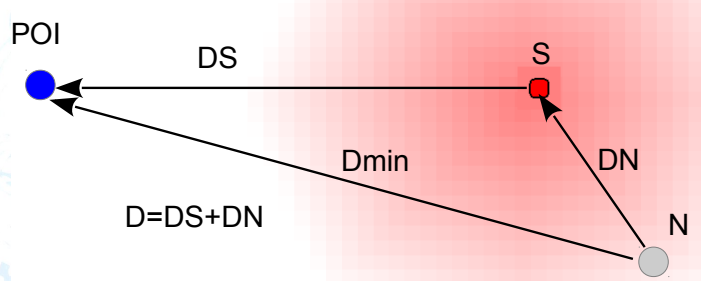
- POIs are sources of a **gradient** (service) *g*
- A new sources *S* (at distance *DS*) is created as *g* meets a contextual service
- *S* represents a compositions (identified by a list of service tags)
- *S* creates a gradient (starting at distance *DS*)



Model: part 2

Steering Service

- Distance value D gets decreased of a ΔD depending on the preference: $\Delta D = \sum_{i=1}^{N} \frac{k_i}{(D - D_{min})}$
  - N number of contextual services composed
  - D_{min} value of distance of the shortest path
  - k_i (semantically) estimates how much the user values context i
- The **crowd steering service** picks shortest route as usual



Model: part 3

Competition

- Each service has a numerical value
- It increases of a quantity proportional to k_i when transiting to service i
- It decreases as time passes
- When reaches 0, the service (source) disappears

Expected result

- Services with worse performance than an existing one will vanish
- They are periodically reconsidered, though, to support opennes

$\Rightarrow$ expecting a higher user satisfaction with smaller number of services

Outline

The Computational Model of Pervasive Ecosystems

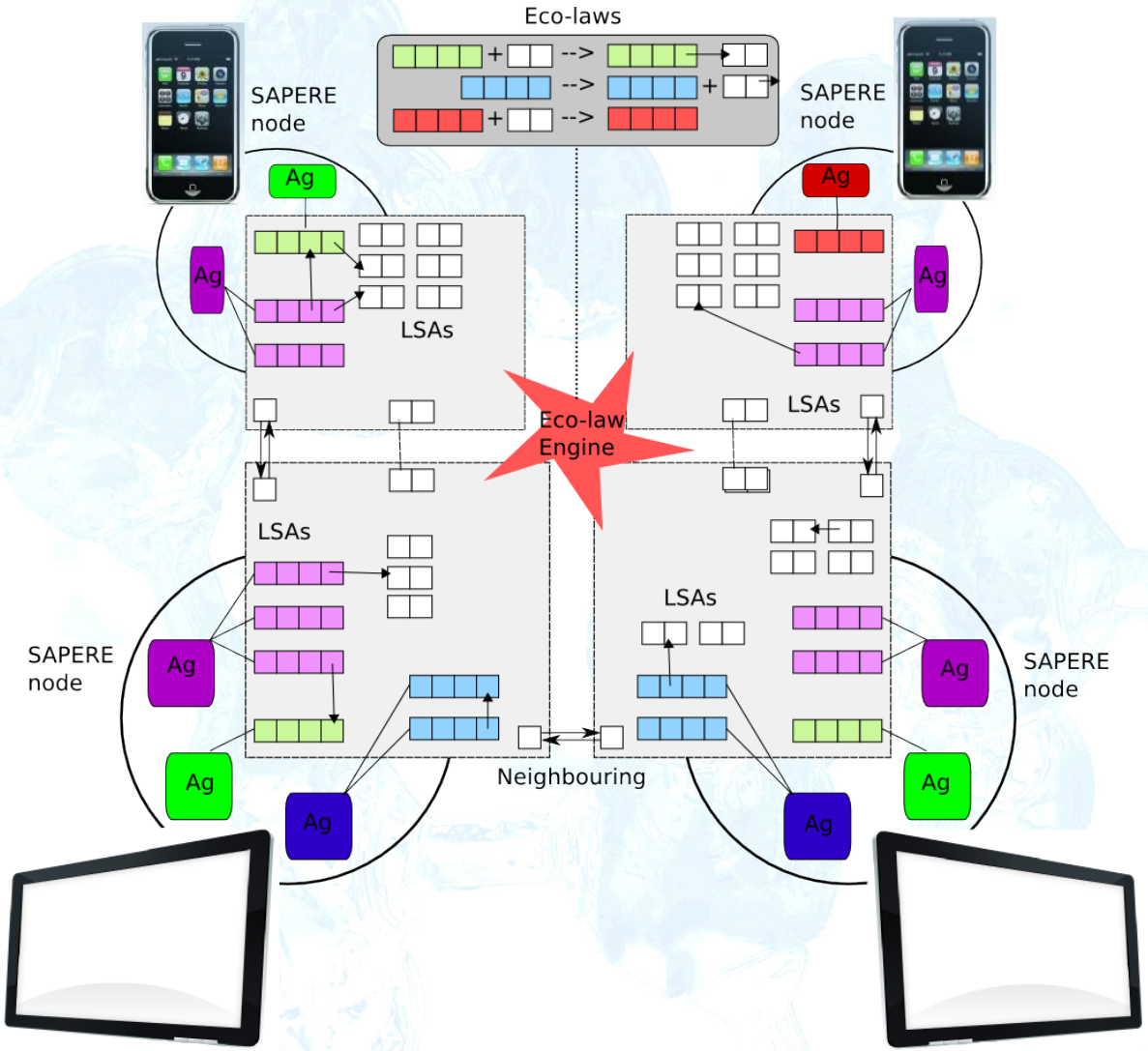
Main choices

- Software agents are represented by LSAs (Live Semantic Annotation)
  - $\Rightarrow$ injected and evolved "in the space"
- Using chemistry as a metaphor to manipulate LSAs
  - $\Rightarrow$ "eco-laws" as chemical-like reactions
  - $\Rightarrow$ "bonds" as interaction channels

Main ingredients

Situation	Agents are situated in SAPERE nodes
Action	Agents manifest through local LSAs
Bonding	LSAs can bond to each other
Observation	An agent observes the world via bonds from his LSAs
Reaction	Eco-laws manipulate LSAs semantically

Architectural view: LSA and eco-laws in evidence



Incarnation in SAPERE

LSAs

Sensor services :	$\langle \texttt{sensor}, \mathit{Desc}, \mathit{Value}, \mathit{Id} \rangle$	$\langle \texttt{service}, \mathit{List}, \mathit{SV} \rangle$
Gradient services :	$\langle \texttt{source}, \mathit{Type}, \mathit{Id}, \mathit{Tag}, \mathit{Distance}, \mathit{SV} \rangle$	$\langle \texttt{service}, \mathit{grad}, \mathit{Type}, \mathit{Id}, \mathit{Tag}, \mathit{Distance}, \mathit{SV} \rangle$
Users :	$\langle \texttt{preference}, \mathit{Desc}, V, K \rangle$	$\langle \texttt{feedback}, \mathit{Id} \rangle$

Key eco-laws

$$\langle \texttt{sensor}, \mathit{Desc}, \mathit{Value}, \mathit{Id} \rangle \mapsto \langle \texttt{sensor}, \mathit{Desc}, \mathit{Value}, \mathit{Id} \rangle, \langle \texttt{service}, [\mathit{Id};], \mathit{initSV} \rangle \quad (1)$$

$$\langle \texttt{service}, \mathit{grad}, T, \mathit{Id}, \mathit{Tag} \textbf{ has not } [L], D, \mathit{SV}2 \rangle, \langle \texttt{service}, L, \mathit{SV}1 \rangle \mapsto \langle \texttt{service}, L, \mathit{SV}1 \rangle, \langle \texttt{service}, \mathit{grad}, T, \mathit{Id}, \mathit{Tag}, D, \mathit{SV}2 \rangle, \langle \texttt{source}, T, \mathit{Tag} \textbf{ add } [L], D, \mathit{initSV} \rangle \quad (3)$$

$$\langle \texttt{source}, \mathit{Type}, \mathit{ID}, \mathit{Tag}, \mathit{Distance}, \mathit{SV} \rangle \mapsto \langle \texttt{source}, \mathit{Type}, \mathit{ID}, \mathit{Tag}, \mathit{Distance}, \mathit{SV} - \Delta \mathit{SV} \rangle \quad (4)$$

$$\langle \texttt{source}, \mathit{Type}, \mathit{ID}, \mathit{Tag}, D, \mathit{SV} <= 0 > \rangle \mapsto \epsilon \quad (5)$$

$$+\langle \texttt{sensor}, \mathit{Desc}, \mathit{Value}, \mathit{Id}1 \rangle, +\langle \texttt{source}, \mathit{Type}, \mathit{Id}, \mathit{Tag} \textbf{ has } [\mathit{Id}1], \mathit{Distance}, \mathit{SV} \rangle, \langle \texttt{preference}, \mathit{Desc}, V, K \rangle, \langle \texttt{diffLength}, \Delta D \rangle, \langle \texttt{feedback}, \mathit{Id} \rangle \mapsto +\langle \texttt{sensor}, \mathit{Desc}, \mathit{Value}, \mathit{Id}1 \rangle, +\langle \texttt{source}, \mathit{Type}, \mathit{Id}, \mathit{Tag}, \mathit{Distance}, \mathit{SV} + K/(\Delta D + 1) \rangle \quad (6)$$

Steering agent

Take care of giving direction, and producing LSAs `preference`, `feedback` and `difflength`

Outline

The Alchemist sub-project[2]

Alchemist Simulator: `http://alchemist.apice.unibo.it`

- Fast — An extension of standard Gillespie's simulation approach
- Flexible — Supporting large topologies, and structured "data"
- Smart — Including statistical modelchecking

Applications

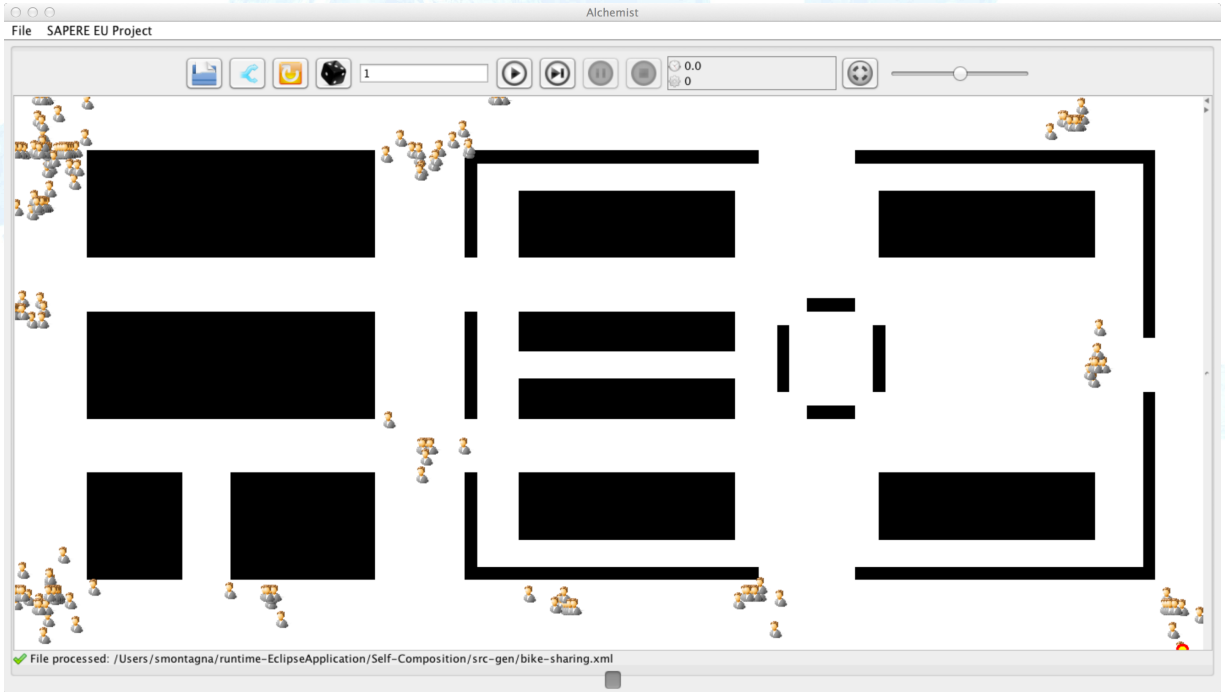
- Simulating pervasive computing applications (SAPERE DSL)
- (also used in simulation of embryogenesis processes)

[2] Danilo Pianini, Sara Montagna, and Mirko Viroli. "Chemical-oriented simulation of computational systems with Alchemist". In: *Journal of Simulation* (2013). Advance online publication. ISSN: 1747-7786.

ALCHEMIST Simulation of Gradient Compositions

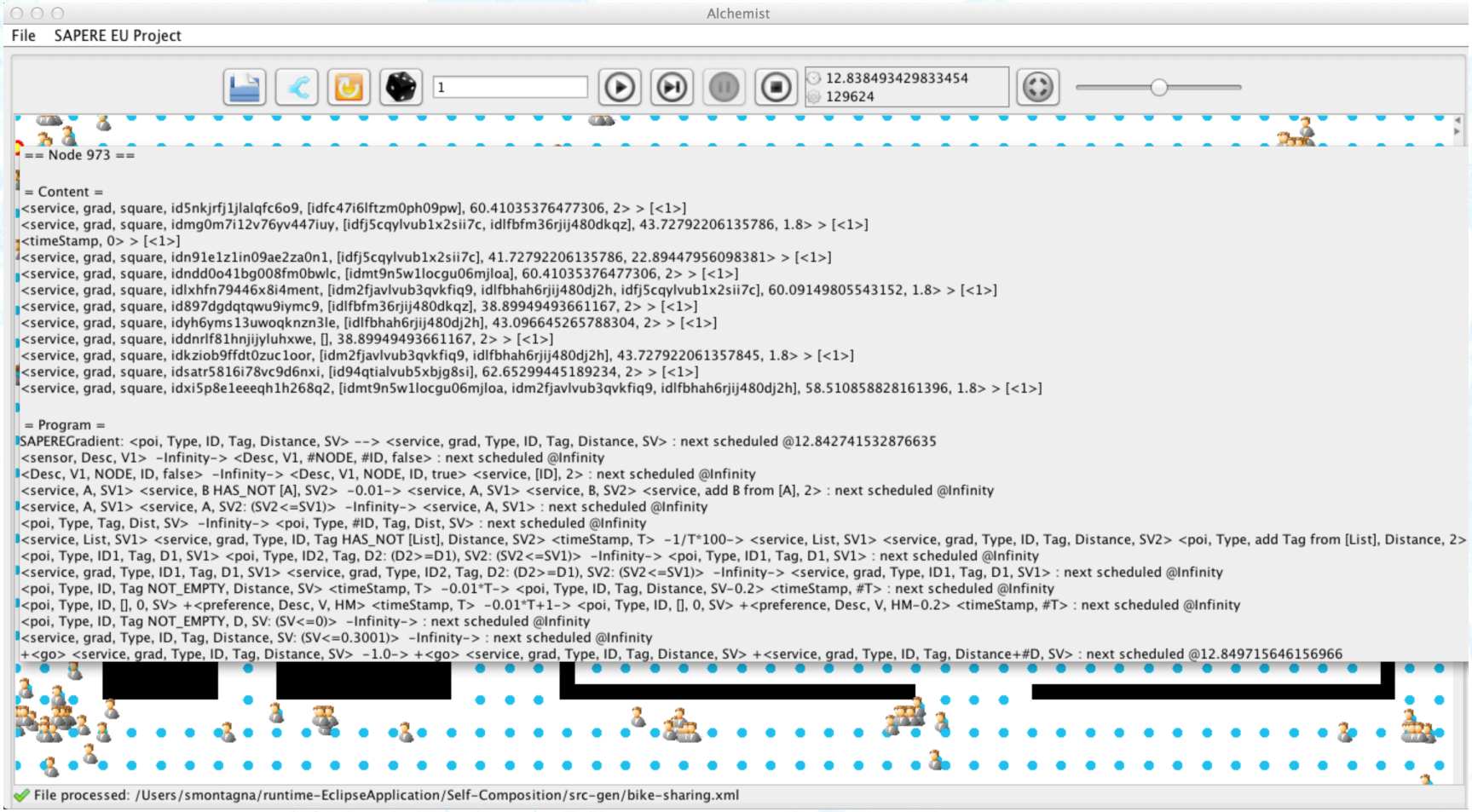
Early simulations, as proof of concepts

- Experiments on a toy city map, no real-life data so far



ALCHEMIST Instantiation

ALCHEMIST Instantiation



Main Elements

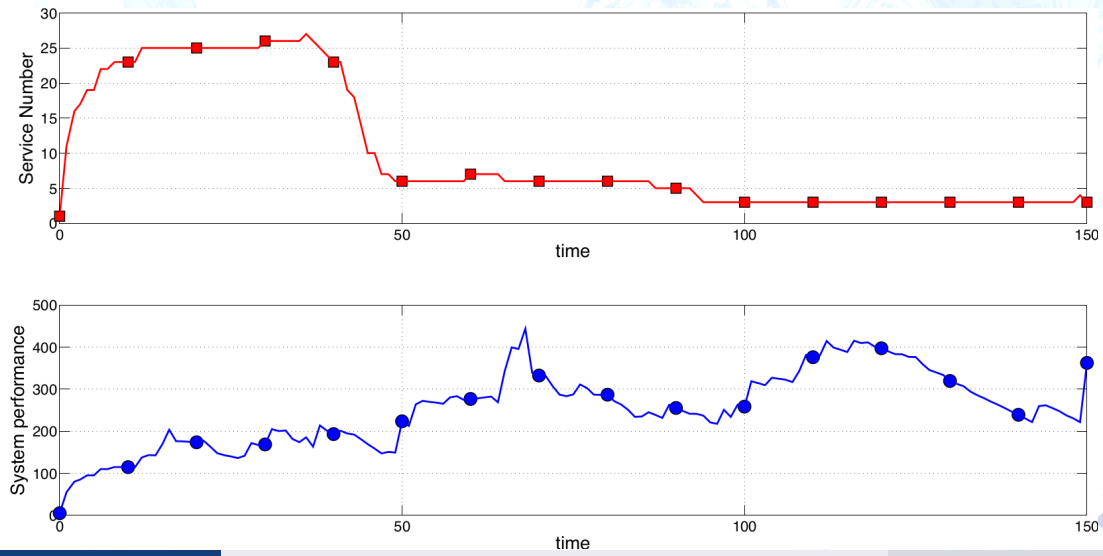
Simulation data

- 1 POI + 6 contextual services
- 80 groups (profiles) of 5 people each, arriving in different moments

General results, after a transient phase

⇒ Increasing overall satisfaction (values of gradient services)

⇒ Decreasing number of winning compositions



Conclusion and Future Work

Lessons Learned

- Generally difficult to tune system parameters properly
- Difficult also to control the training time
- Probably need some sort of (distributed) self-management
  - rates depend on actual current activity
  - accelerating creation of new compositions

Future Work

- Consider realistic scenarios – from SAPERE deployments
- Adopt also advanced composition recommenders
- Evaluate different forms of gradient composition
- Experiment simplified implementations on existing deployments

Composing gradients for a context-aware navigation of users in a smart-city

Sara Montagna Mirko Viroli

Email: {sara.montagna,mirko.viroli}@unibo.it

ALMA MATER STUDIORUM—Università di Bologna, Cesena

Spatial Computing Workshop at AAMAS 2013 (SCW 2013)

Saint Paul, Minnesota, USA, 6th May 2013