WP1 - Model & Methodology:

People, Activities, Hints to other WPs

Mirko Viroli
mirko.viroli@unibo.it

Alma Mater Studiorum—Università di Bologna, Cesena

SAPERE kick-off meeting, 6/10/2010

Group presentations

aliCE – Agents, Languages and Infrastructures for Complexity Engineering

- Faculty members: Antonio Natali, Andrea Omicini, Enrico Denti, Mirko Viroli, Alessandro Ricci
- Cesena site of Università di Bologna

Research Activities

- Coordination models and languages
- Object-Oriented Languages
- Multi-agent systems (meta-models, methodologies, infrastructures)
- Self-organisation (in coordination)

Systems

- TuCSoN coordination infrastructure
- tuProlog engine

People, their expertise and role

- Mirko Viroli, Researcher
  - Expertise: Computational Models and Languages, Self-org
  - Role: UNIBO leader, WP1 leader, Tasks T1.1, T*.*
- Andrea Omicini, Associate Professor
  - Expertise: Coordination Models and Languages, MAS
  - Role: WP6 leader, Tasks T1.2, T1.3
- Matteo Casadei, Post-doc (full-time)
  - Expertise/Role: Coordination & Self-org, Tasks T2.*, T4.*
- Ambra Molesini, Post-doc (half-time)
  - Expertise/Role: Agent-Oriented Methodologies, Task T1.3
- Elena Nardini, Phd Student (soon Post-doc, full-time)
  - Expertise/Role: Semantic Coordination, Tasks T1.2, T4.1
- Sara Montagna, Phd Student (soon Post-doc, half-time)
  - Expertise/Role: Bio-ICT convergence, Task T1.1

This is the initial staff, changes and new entries are possible

UNIBO Tasks and efforts overview

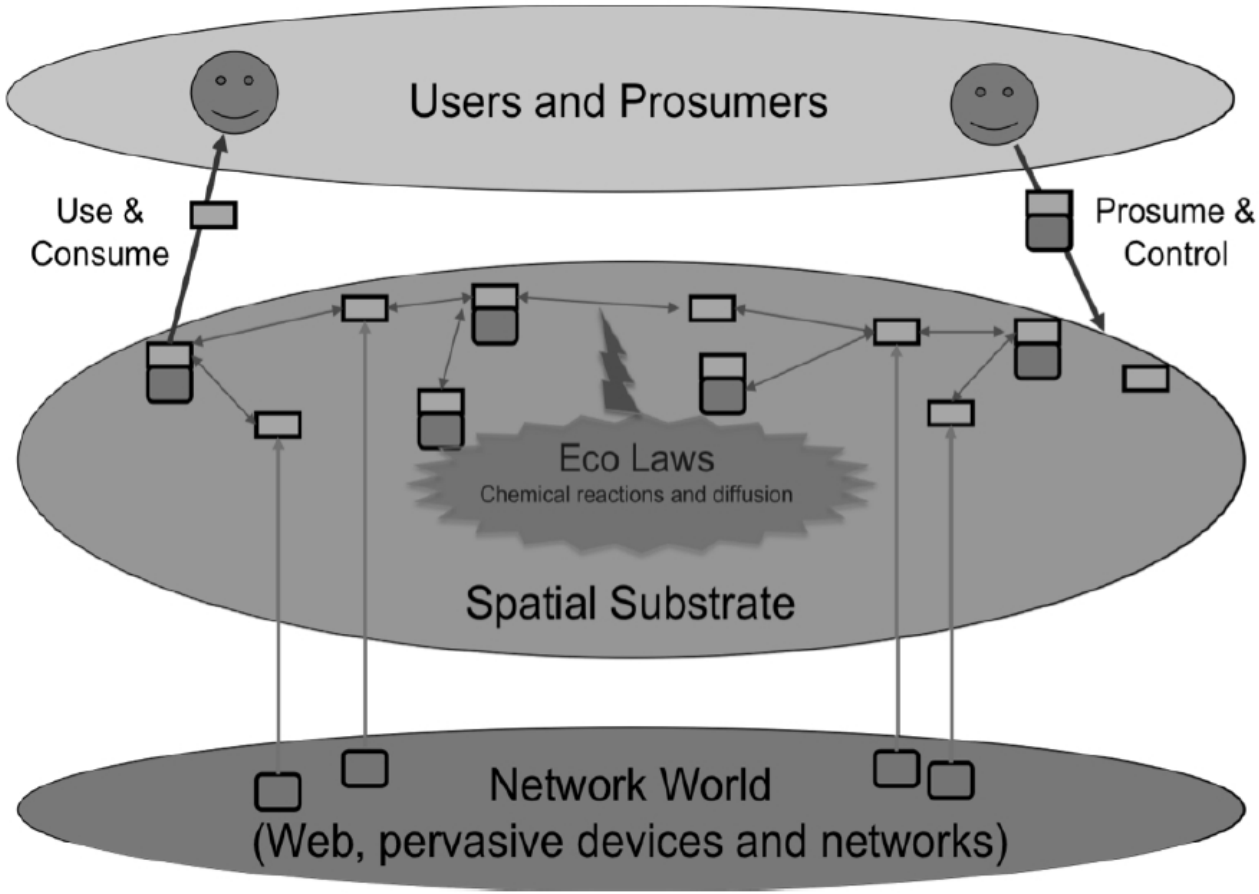
WP	Task	Period	UNIBO effort	People
WP1 – Model & Methodology	T1.1 – The component and Interaction model	1-24	**15/28**	Mirko Viroli, Sara Montagna
WP leader: Viroli	T1.2 – Semantic Representation	1-24	9/28	Elena Nardini, Andrea Omicini
	T1.3 – Methodology (& Tools)	12-36	**14/24**	Ambra Molesini
WP2 – Structures and Space	T2.1 – Self-aggregation	1-24	7/28	Matteo Casadei, Mirko Viroli
	T2.2 – Self-composition	1-32	7/29	Matteo Casadei
	T2.3 – Self-management and Control	13-36	6/23	??
WP4 – Infrastructure	T4.1 – Basic Middleware and API	5-32	10/33	Matteo Casadei, Elena Nardini
	T4.2 – Integeration and Libraries	21-36	6/28	??
	T4.3 – Security Issues	25-36	2/11	??
WP6 – Dissemination etc.	T6.1 – Dissemination	1-36	**2/5.5**	Andrea Omicini
WP leader: Omicini	T6.2 – Collaboration	1-36	**3/7.5**	Andrea Omicini
	T6.3 – Exploitation	1-36	1/4.0	Andrea Omicini

WP1: Model & Methodology

Quoting the DoW

"The goal of this WP is to frame the foundational basis of the SAPERE framework, and correspondingly identify the model that will ground the development of the ecosystems and of its applications and services, develop the methodological guidelines (and identify/integrate associated tools) to support the development process. The model will take into account aspects such as (*i*) the chemically-inspired laws of behaviour (i.e., the eco-laws), (*ii*) the model of system components and interactions, (*iii*) the LSAs model, and (*iv*) the modalities of creation, destruction, access and exploitation of services, data, and devices. Implications to the development of systems and services will be studied as well. Overall, the model and its associated methodological tools will have strong influence on all the other WPs of the project, serving as a backbone foundation"

Considering this architectural view



WP1: Tasks

T1.1 – The Components and Interaction Models

- Abstract (representation independent) model of eco-laws
- Services structure and interactions
- Studying analysis tools for behaviour verification

T1.2 – Semantic representation

- Shape of LSAs
- Studying analysis tools for logic reasoning

T1.3 – Methodology

- Finding a SE methodology
- Conceiving tools for development/analysis

Basic timing facts (relevant to the kick-off meeting)

Deliverable by Month 12: D1.1 Early operational model

- First version of abstract computational model (T1.1)
- First version of LSA framework (T1.2)
- Early demo of the model at work (cooperation with W2,3,4)

Next stages

- Complete model (M20), after "Interact" stage
- Final model (M24), after "Integrate" stage
- From M12 to M36 developing task T1.3

Remainder this talk

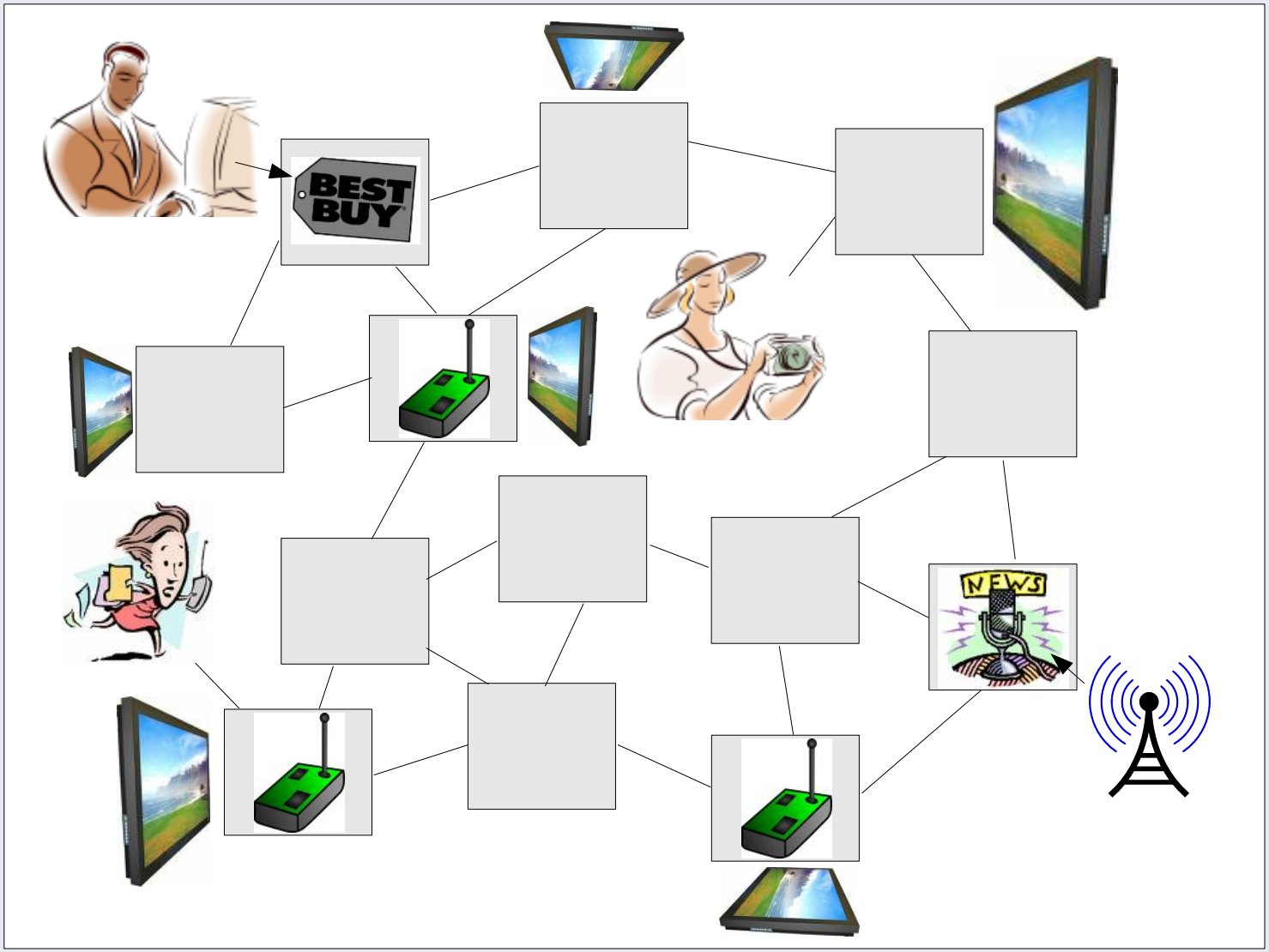
According to the above facts..

- Will stick to the SAPERE computational model (T1.1)
- Only frame the role of semantics (T1.2)
- Entirely neglect methodological aspects (T1.3)

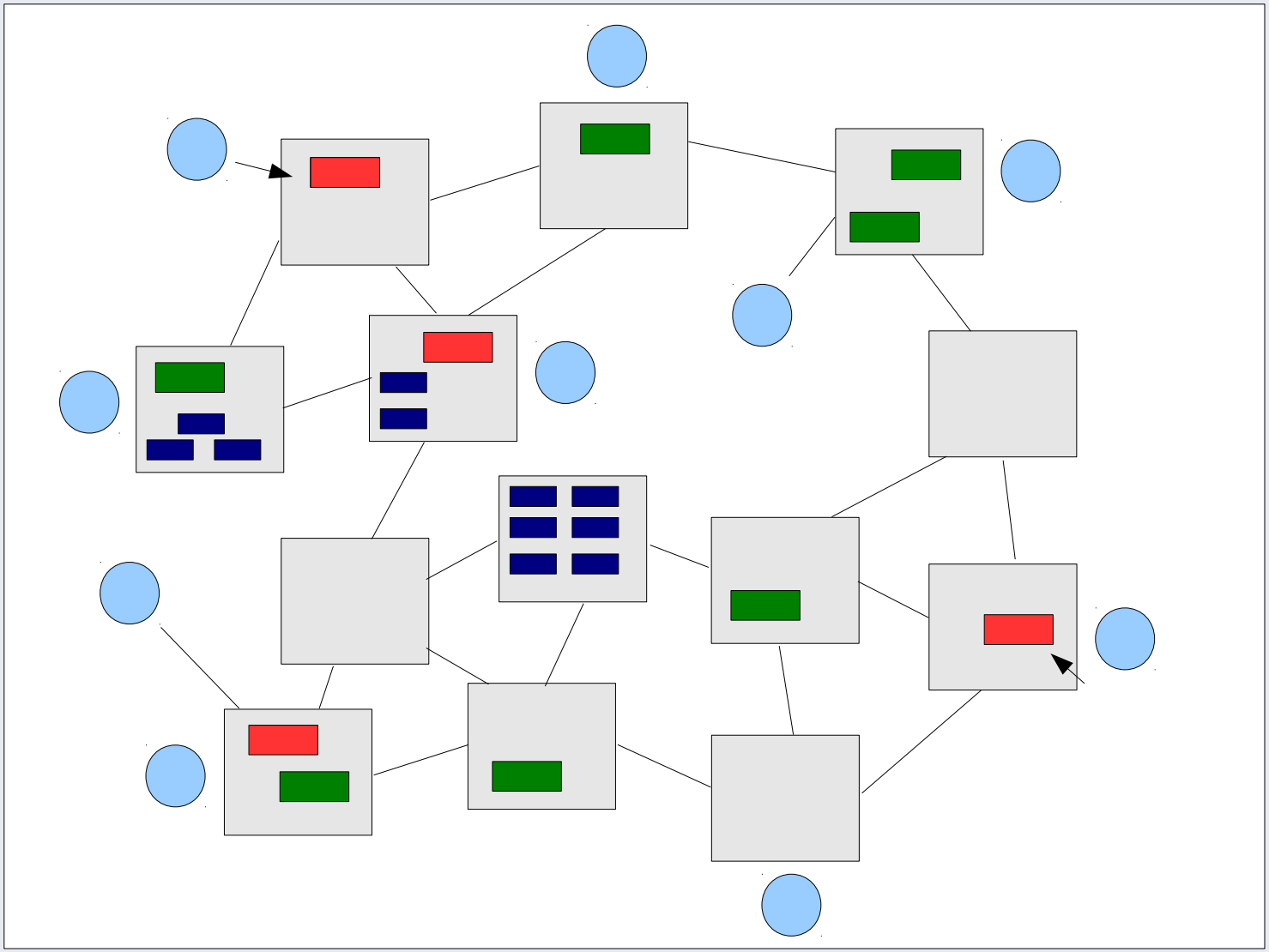
Outline

1. Modelling the Space level
2. LSAs and LSA-based Interaction
3. Eco-laws
4. Local vs. Global
5. Taking the chemical metaphor really seriously

From a Pervasive Computing Scenario . . .



. . . to a Network of LSA-spaces



The Space as a pure coordination abstraction

Globally

- Each node of the network hosts an LSA-space
- The sytem is a dynamically evolving graph of spaces
- Proximity means capability of direct interaction

Locally

- Each space hosts a set of LSAs
- SW services and devices are agents interacting only through spaces
- LSAs represent occurrence/state of agents, data, events
- LSAs are injected/retrieved by agents
- LSAs are evolved/produced/removed/diffused by spaces

On Live Semantic Annotations

Ingredients of a LSA

- Has a unique identifier
- Has a semantic description of its "content", uniformly modelling either:
  - a service description
  - a user profile
  - a device features
  - some piece of knowledge
  - ...
- May have additional information:
  - strength, age, source agent, target agent, related LSAs, dangling bonds (for aggregation/composition),...
  - explicitly coded, or included in the semantic description
- May implicitly/explicitly refer to a (domain) ontology

Example interaction through LSAs

Showing an ad due to user proximity to a pervasive display

Consider a room hosting one node/location. It will store LSAs for:

- Each display in the room – including display properties, and featuring 1 dangling bond (for an ad)
- Each available ad – including description of content, URL of the clip, display requirements
- Each person in the room – including her profile

How interaction takes place?

- 1 display-LSA, 1 ad-LSA, and 1 user-LSA match altogether
- The display-LSA binds the ad-LSA (namely, display-LSA gets somehow transformed)
- The display agent is observing its LSA, detects the change, and accordingly fires the visualisation process

Schematic eco-law: `DISPLAY+AD+USER` $\rightarrow$ `DISPLAY[AD]+AD+USER`

The role of eco-laws

General Ideas

- They are coordination laws embedded in all spaces
- They are sort of chemical reactions semantically applied to LSAs
- They match few LSAs, combine them, and obtain some LSA product (a new service/data, or establishing/destroying some bond)

Proper eco-laws should enact the following behaviour:

- Matching services/users
- Extinguishing services no longer used
- Promoting ecological-like competition for the best services
- Creating/destroying bonds
- Contextualising LSAs
- Structure & Space: generating local/distributed structures of LSAs
- Knowledge & Time: reacting/proacting to situations

An abstract view over an eco-law

Quoting the DoW again..

For the "eco-laws" that will drive the dynamics of the ecosystem, we envision them to define the basic policies to rule sorts of virtual "chemical reactions" among the LSAs of the various individuals of the ecology.

Ingredients of the "language" of eco-law

1. A chemical template, e.g.: $X + Y \xrightarrow{s} X + Z$
2. A function yielding the likelyhood (e.g. rank, factor) of LSAs lx matching X (and ly matching Y)
3. A function ranking the "correspondence" of lx and ly given that X may be correlated to Y
4. A function taking lx and ly and producing lz (based on X, Y, Z)
5. A function taking lx, ly and yielding the "speed" of the reaction

On the generality of eco-laws: the role of semantics

Three stances:

1. Maximum Opennes:
   - Eco-laws: application-independent (a small universal set)
   - Matching and LSAs: depend only on application ontology
2. Medium Opennes:
   - Eco-laws: application-specific
   - Matching and LSAs: depend only on application ontology
3. Minimum Opennes:
   - Eco-laws: application-specific
   - Matching and LSAs: application-specific

Some thought

- SAPERE aims at approach 1
- There is an acknowledged risk we can achieve just approach 2
- By orthogonality, approach 3 will often be assumed by WPs 2,3

Which semantic approach in SAPERE?

Task T1.2

- Several approaches can be used, no need to choose now
- Semantics is very orthogonal to WP2,3, but is key in WP4,5
- Semantics is key to the general goals of SAPERE

Early work in UNIBO

- We favour Web-oriented expressive frameworks:
  e.g. built over OWL and fuzzy Description Logic
- But still have no idea about performance issues, nor know if it is expressive enough to SAPERE
- We studied semantic tuple space approaches [2]
- We are studying a framework of semantic chemical coordination [4]

How to create distributed LSA structures?

Important bio-inspired phenomena

- A computational-field to retrieve (mobile) resources
- Stigmergic mechanisms (creating paths, clustering data, . . .)
- Competition, extinction, diversity via formation of ecological niches

How eco-laws can deal with spatiality?

Serveral possible approaches:

- An eco-law produces a whole "field"
- An eco-law creates an LSA into one neighbour (or all neighbours)
- An eco-law can match an LSAs into one neighbour
- An eco-law can match all LSAs in the neighbour

Taking the chemical metaphor really seriously

Towards chemical-inspired tuple spaces

- The discussion so far was necessarily abstract
- At some point we must choose a concrete model of LSAs and eco-laws
- Recently we started developing a promising framework of chemical-inspired tuple spaces (CTS): It *might* be a direction for SAPERE, but feedback is rather urgently needed
- In a CTS, a chemical law evolves the population of tuples *exactly* as in chemistry (see Gillespie's simulation algorithm [1])

Chemical Tuple Spaces

Motivations for chemistry

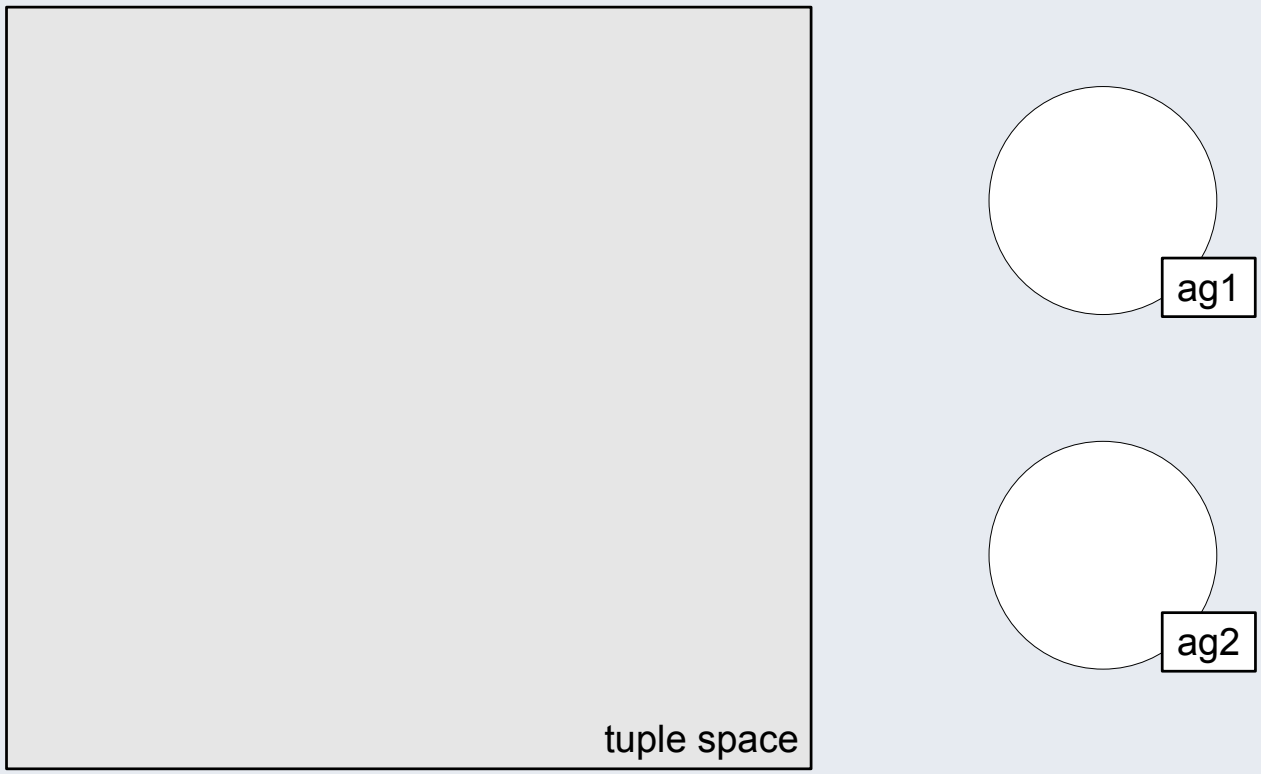
1. We might want to simulate some known chemical behaviour (auto-catalytic reactions)
2. We might want to get inspiration from population dynamics, often modelled as chemical systems
   - Lotka-Volterra equations (a.k.a. prey-predator)
   - Turing's Reaction-diffusion mechanisms

Main idea

- Tuple spaces + chemical reactions as coordination laws
- Tuples have a concentration (a.k.a. weight, or activity value)
- Concentration is evolved "exactly" as in chemistry [1]
- Some reactions can even fire a tuple from one space to another

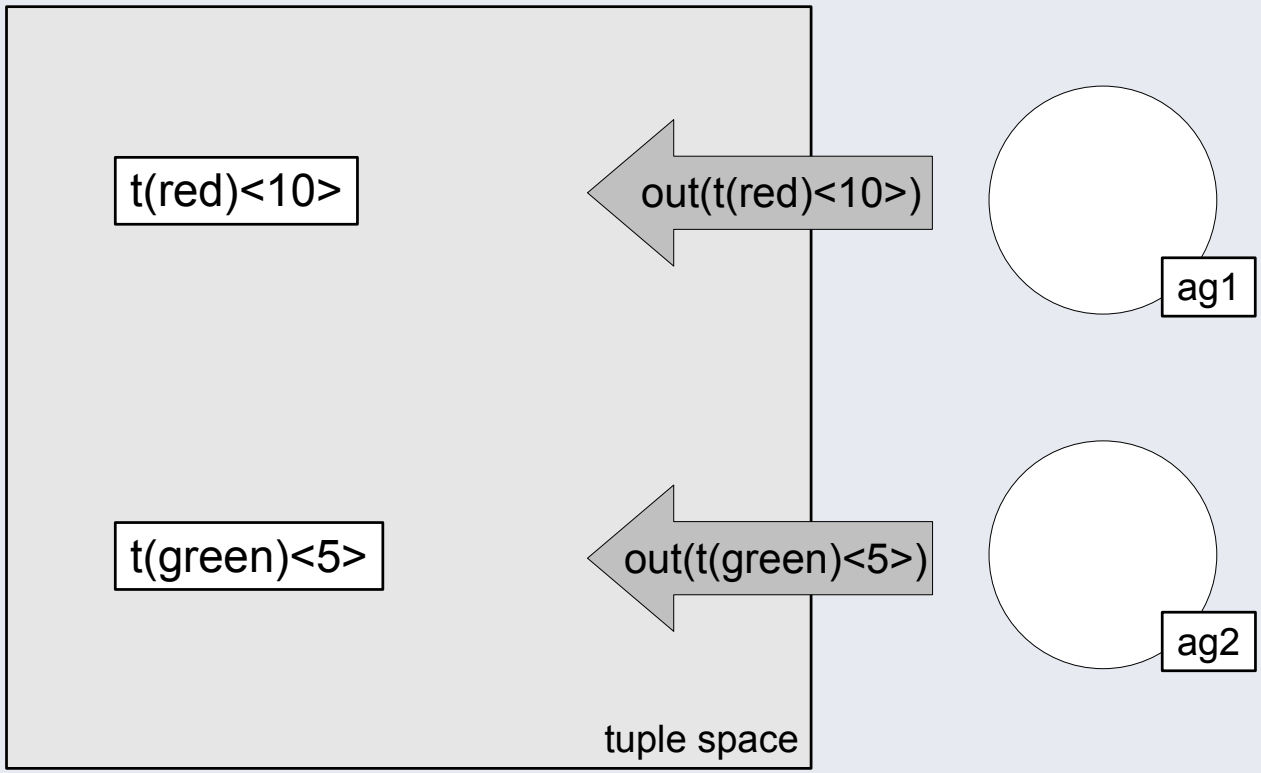
First settings

One tuple space, two agents



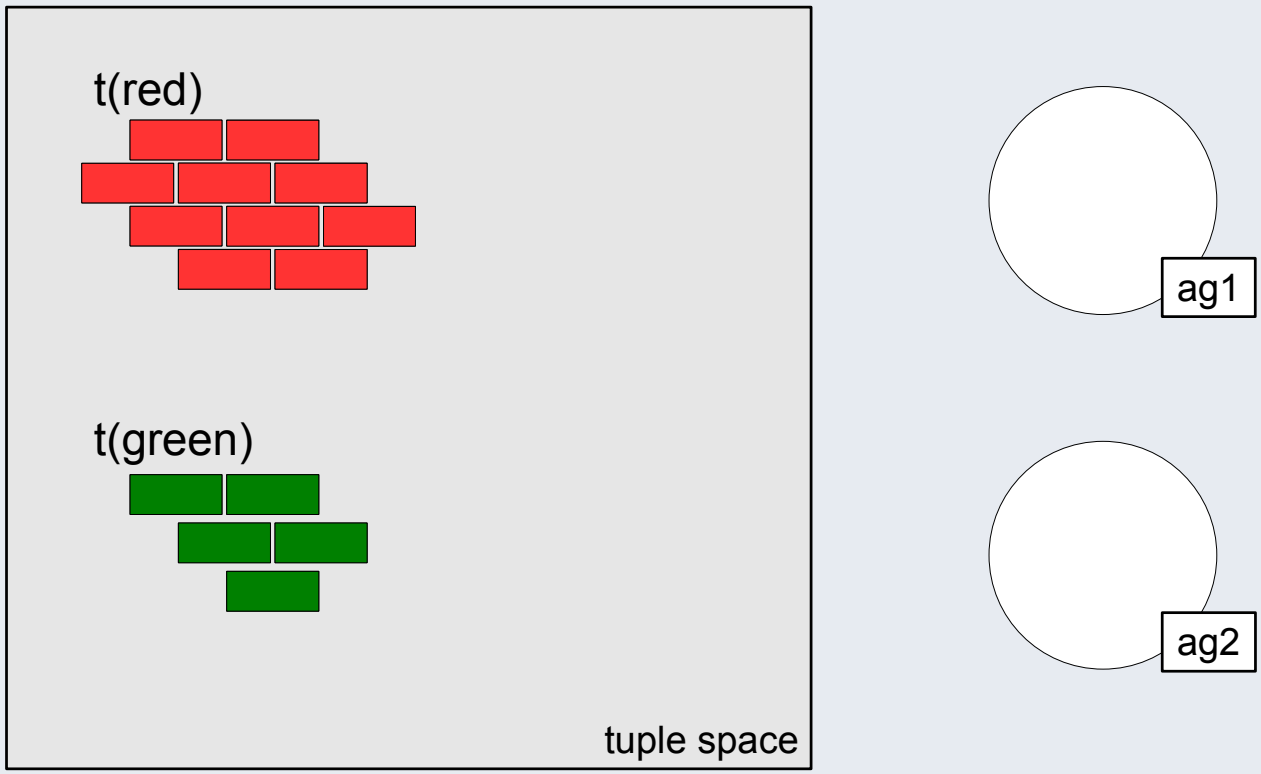
Inserting tuples

Primitive out: default concentration is 1



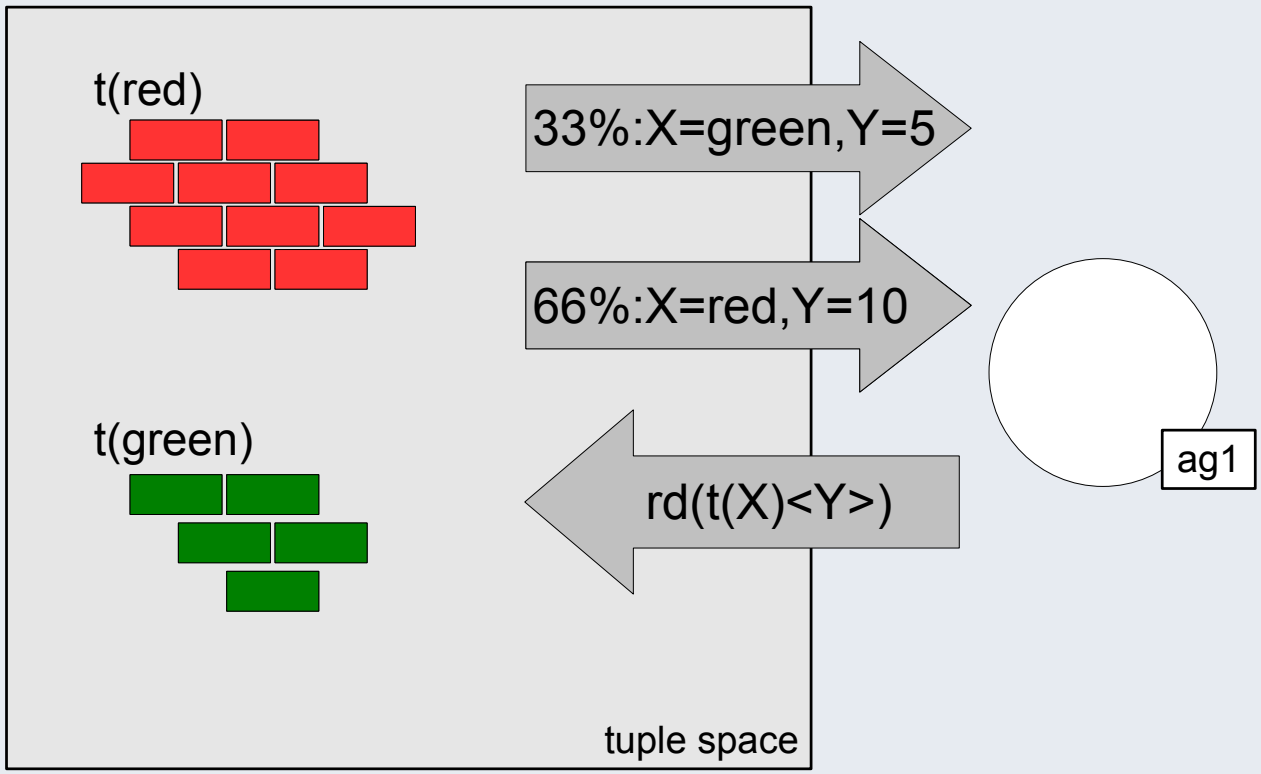
A pictorial representation

A tuple as substance of uniform molecules – but still a single tuple



Retrieving Tuples

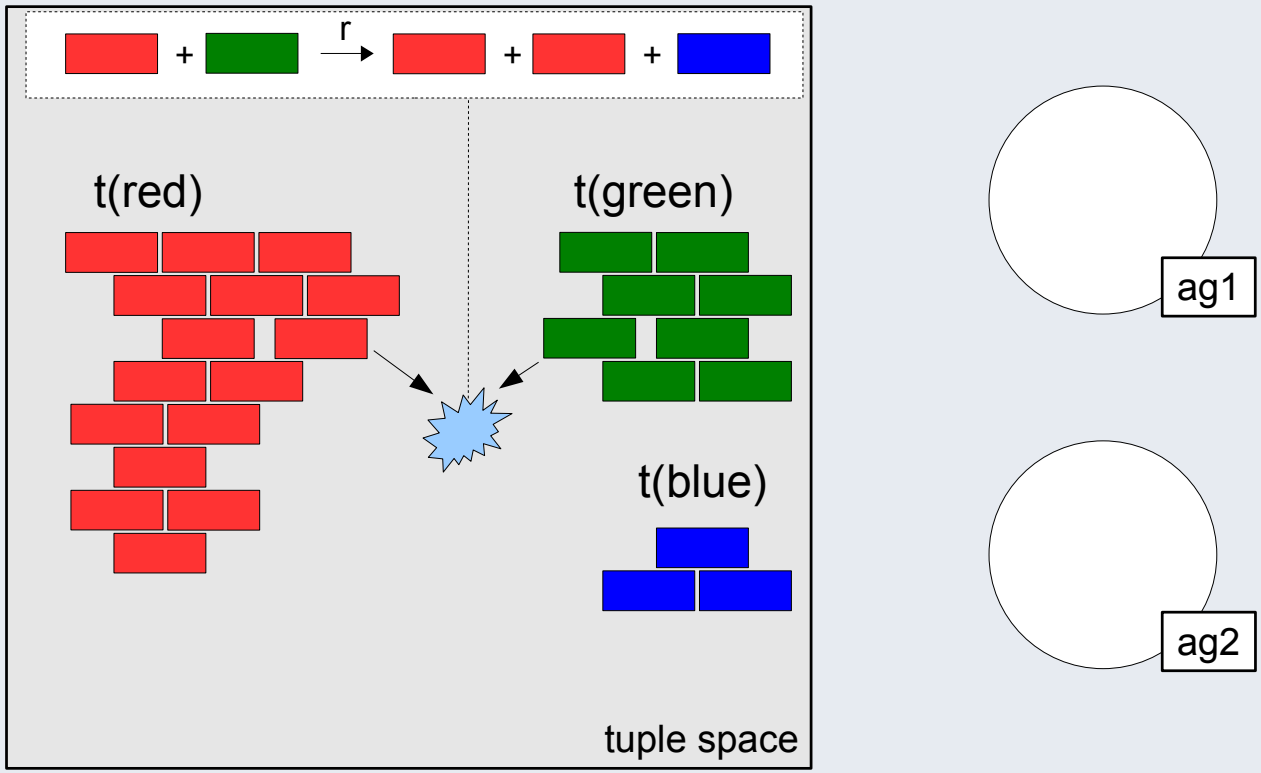
Primitive `rd`: concentration as probability, i.e., relevance



Installing Chemical Reactions

A chemical reaction, with tuples in place of molecules

$$\mathtt{t(red)} + \mathtt{t(green)} \xrightarrow{r} \mathtt{t(red)} + \mathtt{t(red)} + \mathtt{t(blue)}$$

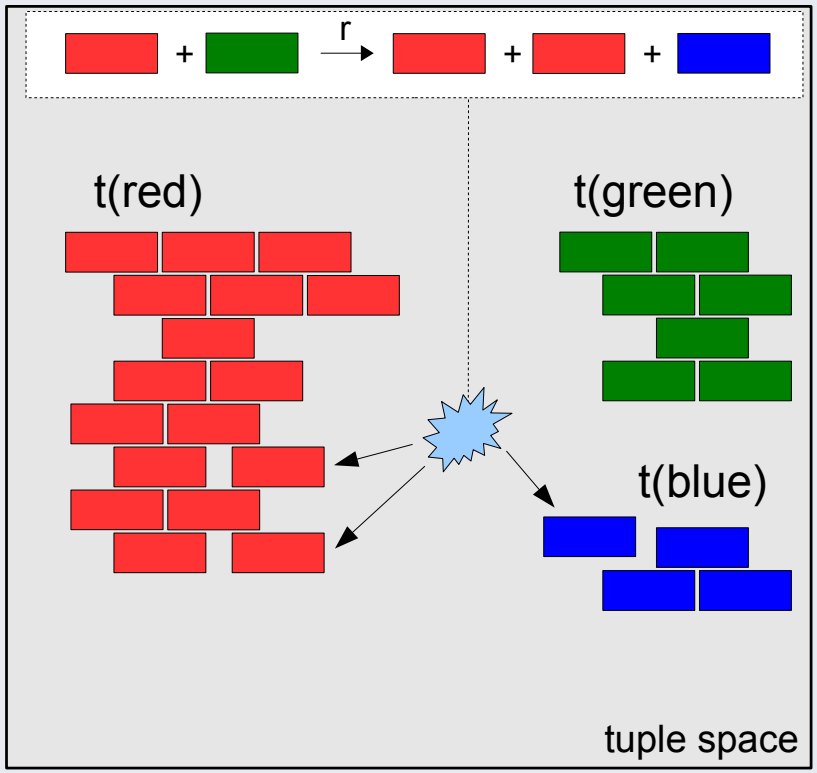


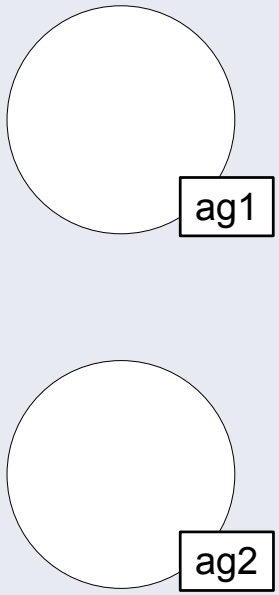
Firing Chemical Reactions

Reactions are executed over time according to [1]

$$\mathtt{t(red)} + \mathtt{t(green)} \xrightarrow{r} \mathtt{t(red)} + \mathtt{t(red)} + \mathtt{t(blue)}$$

Transition (Markovian) rate: $r * \#\mathtt{t(red)} * \#\mathtt{t(green)}$

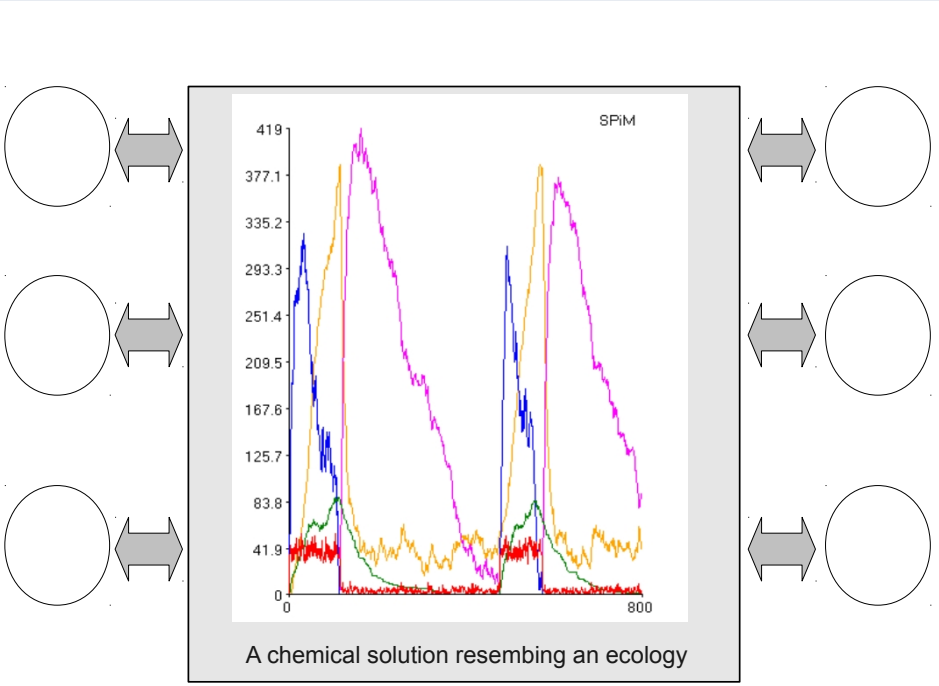




A tuple space as a chemical/ecological solution

Consider the ecological perspective

- The tuple spaces run an "ecology"
- Each tuple resembles a species, concentration is abundancy
- Agents observe, insert and remove species
- Tuple concentration drives the selection of chemical-like reactions

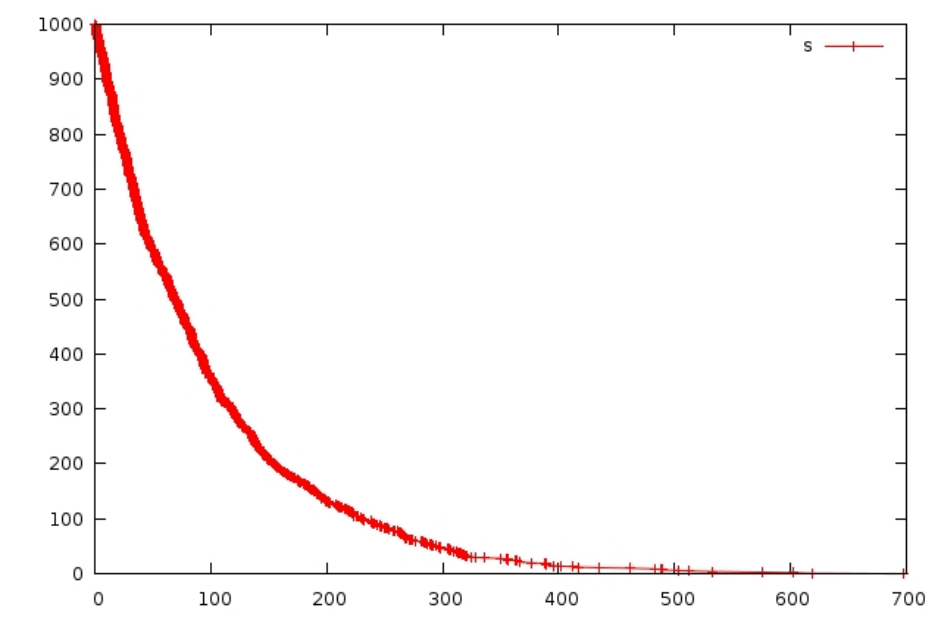


A chemical solution resembing an ecology

Decay example

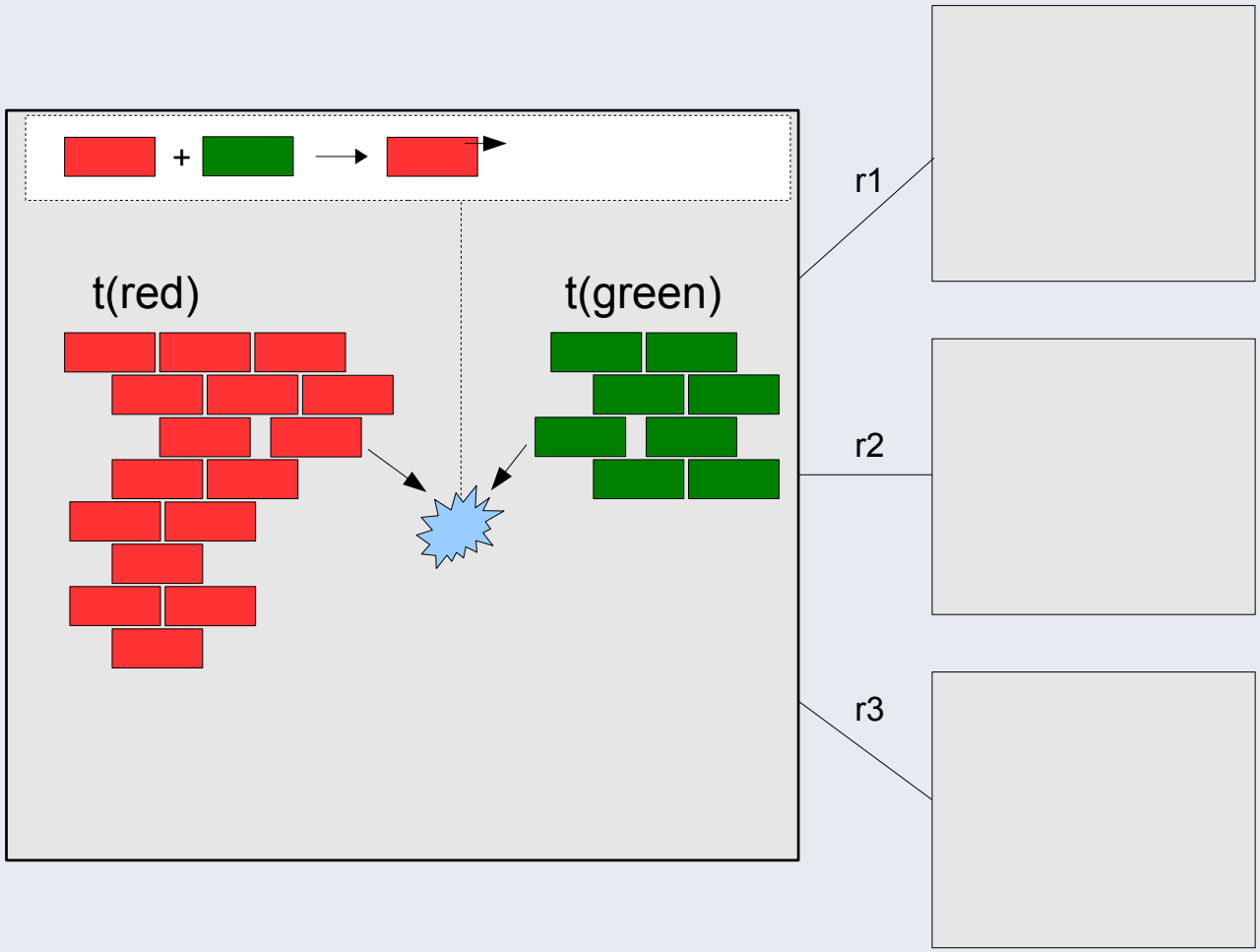
After installing reaction $\mathtt{t(X)} \xrightarrow{0.01} 0$

- We let tuples decade (evaporate like pheromones)
- This is useful to enact time-pertinency
- An agent perceives that the tuple is fading until disappearing
- E.g. `t(s)` represents the temporaneous publication of a service



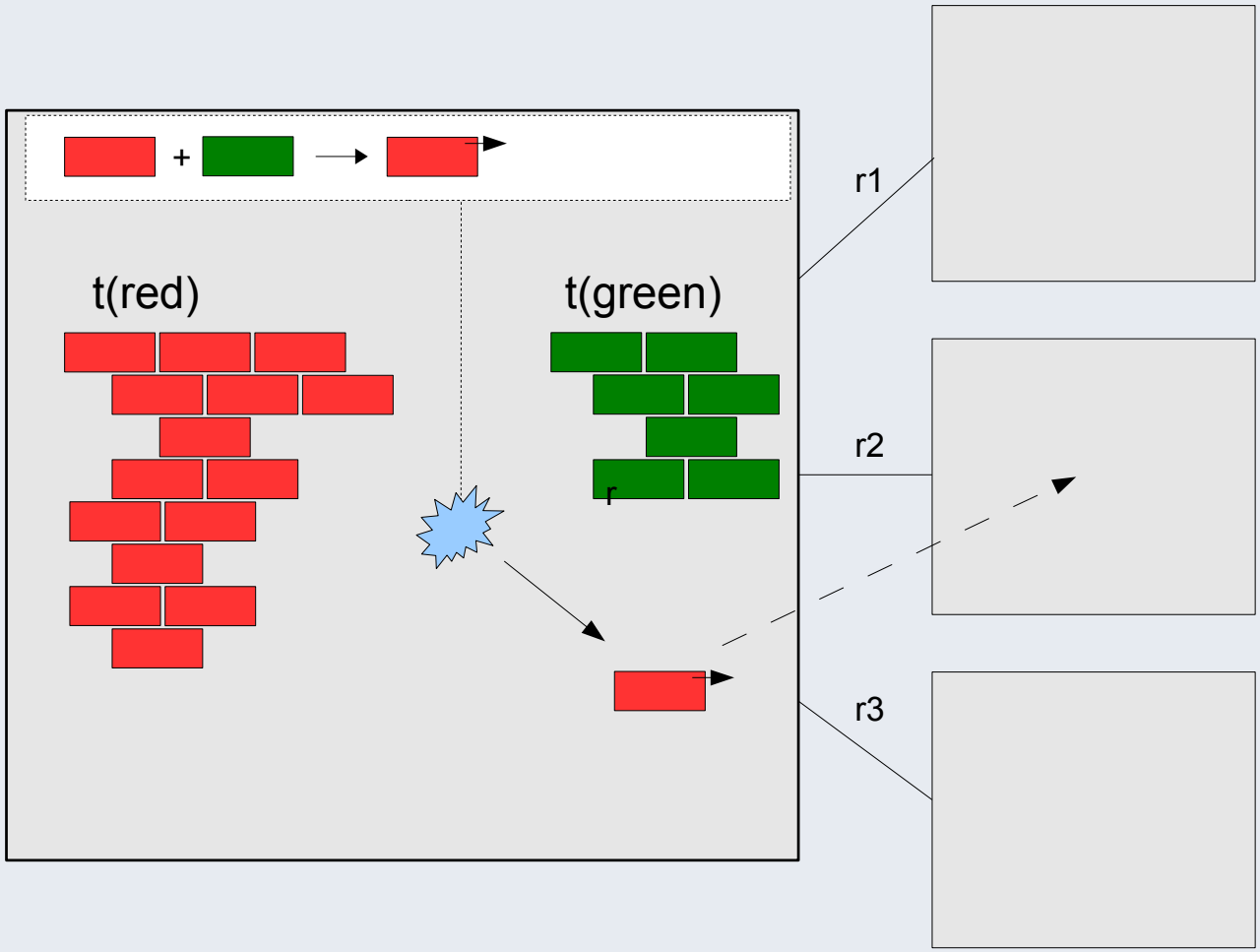
Tuple Transfer

Right-hand side of a reaction can have a firing tuple



From one node to a full biochemical network

Firing tuples are sent to any neighbour, probabilistically



On matching and rates

Overcoming discrete matching

- We assume an application-dependent match function $\mu(t, t')$
  - $0 =$ no match, $1 =$ perfect match, otherwise partial match
  - Chemical reactions are applied "modulo match ranking"
  - E.g. with $\mu = 0.5$, actual chemical rate is divided by 2
- A typical scenario of match-making with preferences

Example of general decay rule: $\texttt{DECAY} \xrightarrow{r\_dec} 0$

- Tuple t decays with chemical rate $\mu(\texttt{DECAY}, t) * r\_dec$
- E.g., t is service publication
- μ tunes service life-time: e.g. influenced by past successful usages

Some implementation fact

Gillespie "direct" simulation algorithm [1]

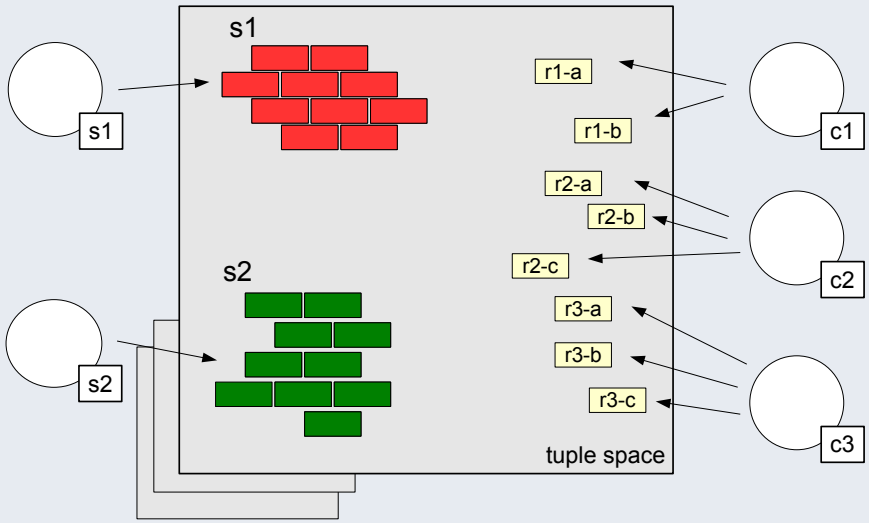
1. Compute the markovian rate $r_1, \ldots, r_n$ of reactions, let R be the sum
2. Choose one of them probabilistically, and execute its transition
3. Proceed again after $\frac{1}{R} * ln\frac{1}{\tau}$ seconds, with $\tau = random(0, 1)$

Tuple Space implementation

- Prototyped a "toy" on top of TuCSoN
- Exact chemistry might be too demanding!

The scenario of service ecosystems

Services and requests as tuples



Clients and services as "individuals of an ecology"

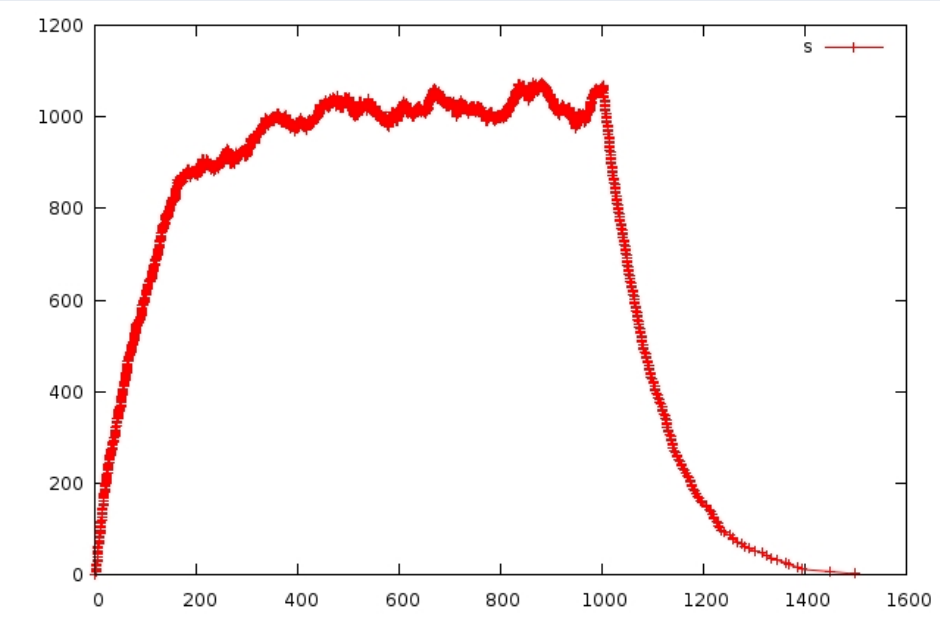
- Tuples as LSAs
- Unsuccessful services fade until completely disappearing
- Concentration of a service increases with (successful) usage
- Similar services compete for survival

Positive-Negative feedback

Service tuples decay, but can be sustained by a feedback token

- Decay rule: $\texttt{DECAY} \xrightarrow{r\_dec} 0$
- Feed rule: $\texttt{publish(SER)} \xrightarrow{r\_feed} \texttt{publish(SER)} + \texttt{SER}$

Example simulation: $r\_dec = 0.01, r\_feed = 10$



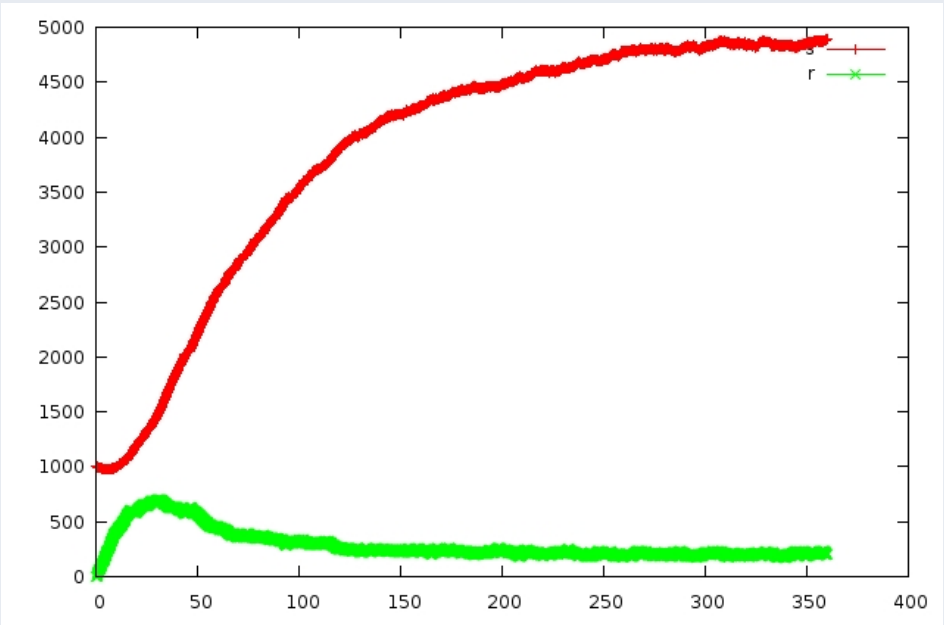
- time 0: Catalyst Token `publish(S)` is inserted
- time 400: Service `S` reaches an equilibrium
- time 1000: The token is removed (or decays)
- time 1600: Service `S` vanishes

Feedback by using (a.k.a. prey-predator)

Idea: Matching Service-Request sustains the service

- Use rule: $\texttt{SER} + \texttt{REQ} \xrightarrow{r\_use} \texttt{SER} + \texttt{SER} + \texttt{toserve}(\texttt{SER}, \texttt{REQ})$

Sim: $r\_dec = 0.01, r\_use = 0.00005, request\_arrival\_rate = 50$



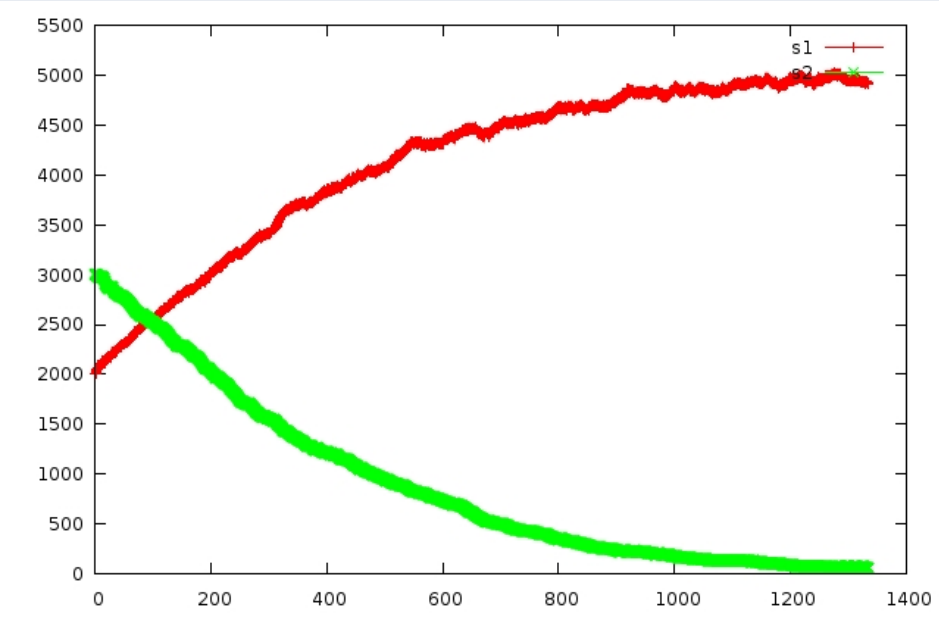
- time 0: Injection of requests raises service level
- time 30: Requests are tamed
- time 350: Unserved requests and service stabilise

Competition

What if more services can handle the same requests?

- higher concentration means higher match frequency
- some service may match better the request, being more proper

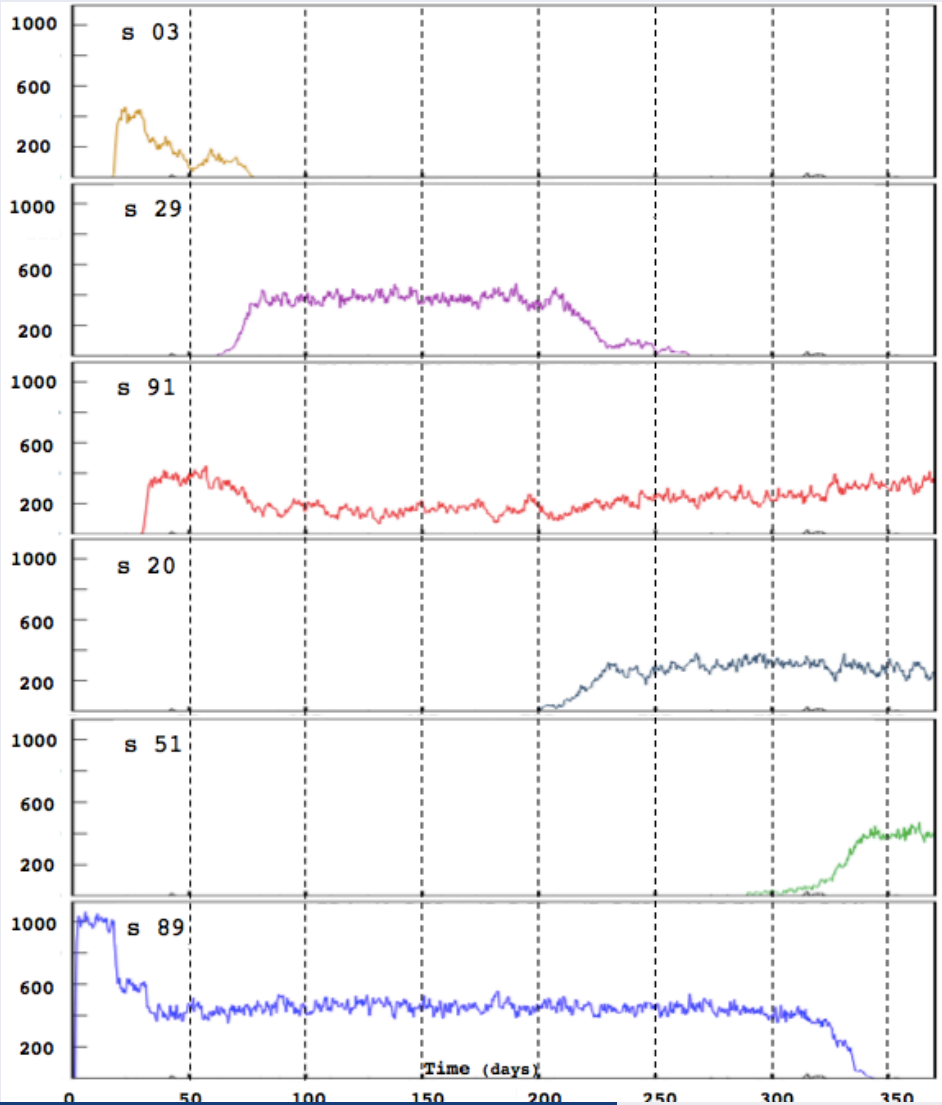
Sim: $r\_use_1 = 0.06, r\_use_2 = 0.04$



- time 0: The two services are in competition for the same requests
- time 100: The one with better use rate (better match) is prevailing
- time 1300: Service $s2$ lost competition and fades

Long-Term Competition

one year period: one new service each day



A simulated airport scenario of pervasive displays:

- services (ads) for 5 (possibly overlapping) marketing targets
- requests are injected about 1 per minute
- each request is a user's profile specifying one target

Results:

- few services are active at a given time
- new services may cause extinction of previous ones

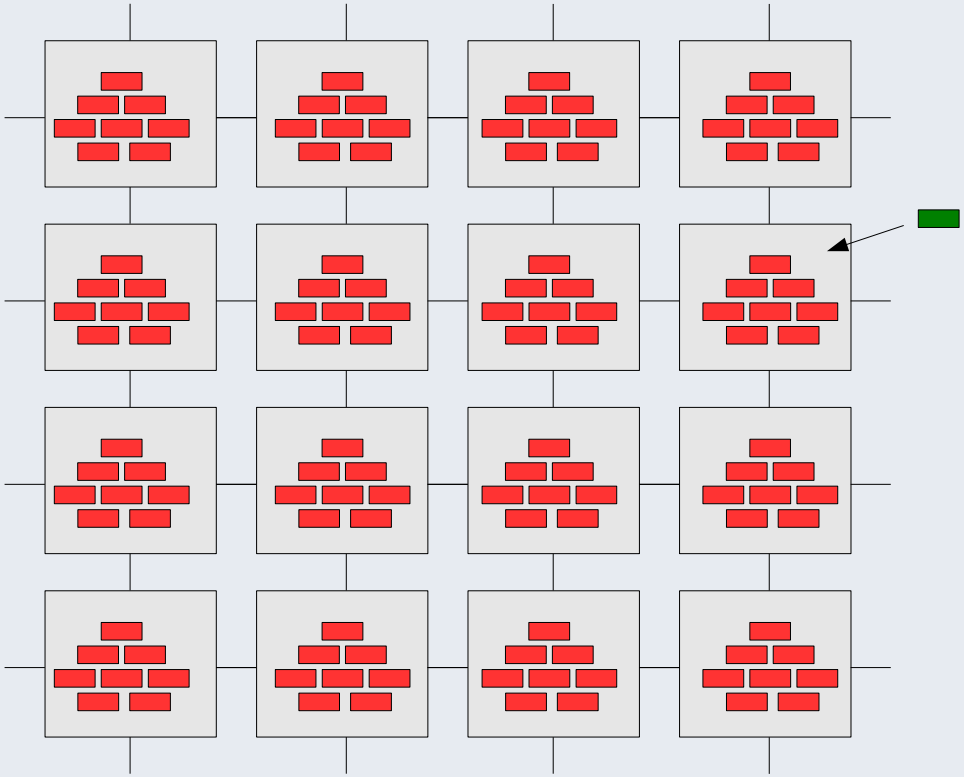
Spatial Diffusion and Competition

One service monopolises a network and its requests

Services continuously diffuse around, by rule:

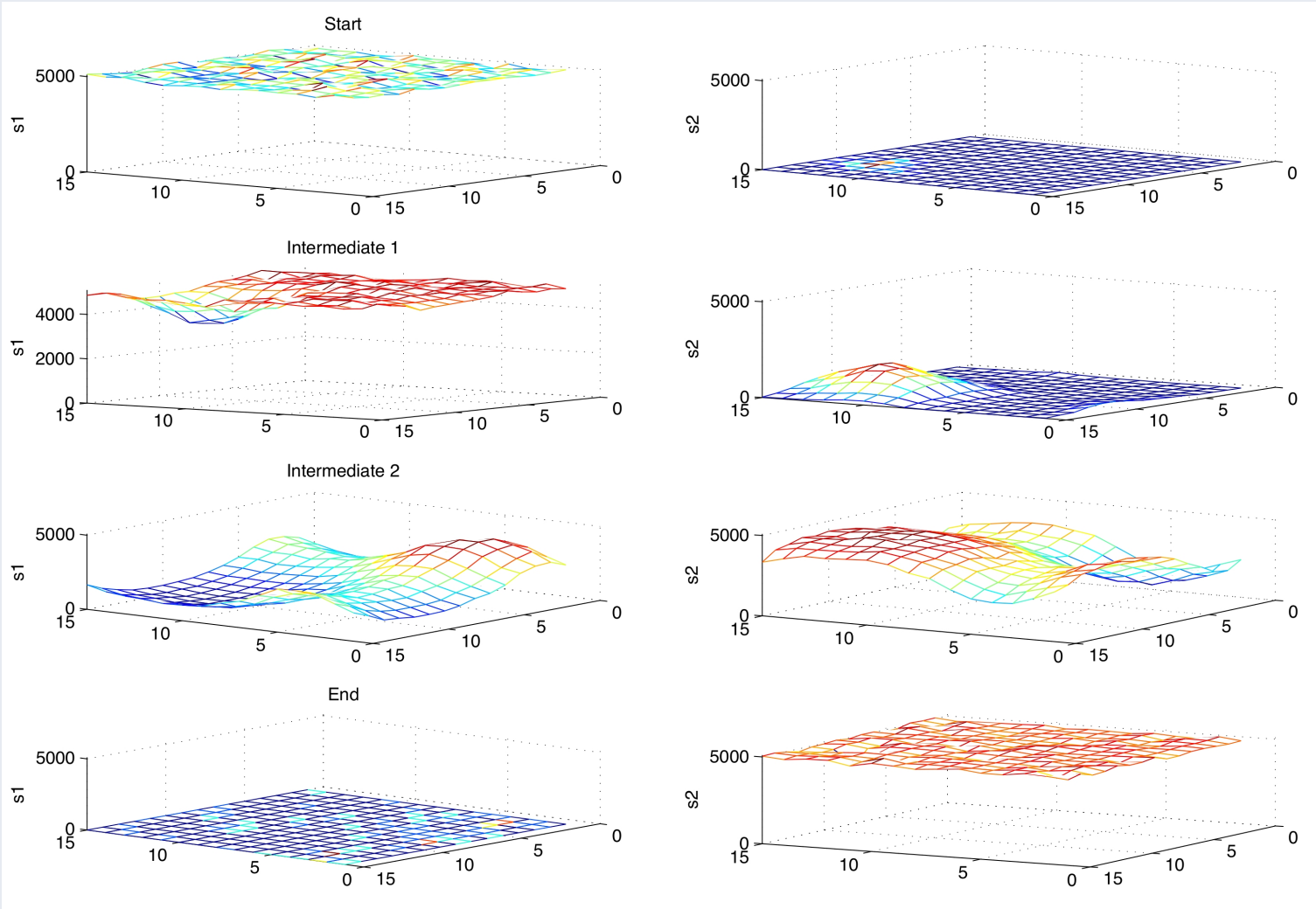
- Diffuse rule: $\mathtt{SER} \xrightarrow{r\_diff} \mathtt{SER}^{\rightsquigarrow}$

Scenario: a better service is injected in a node



Resembling a distributed habitat scenario

Example Simulation: $r\_use_1 = 0.05, r\_use_2 = 0.1$

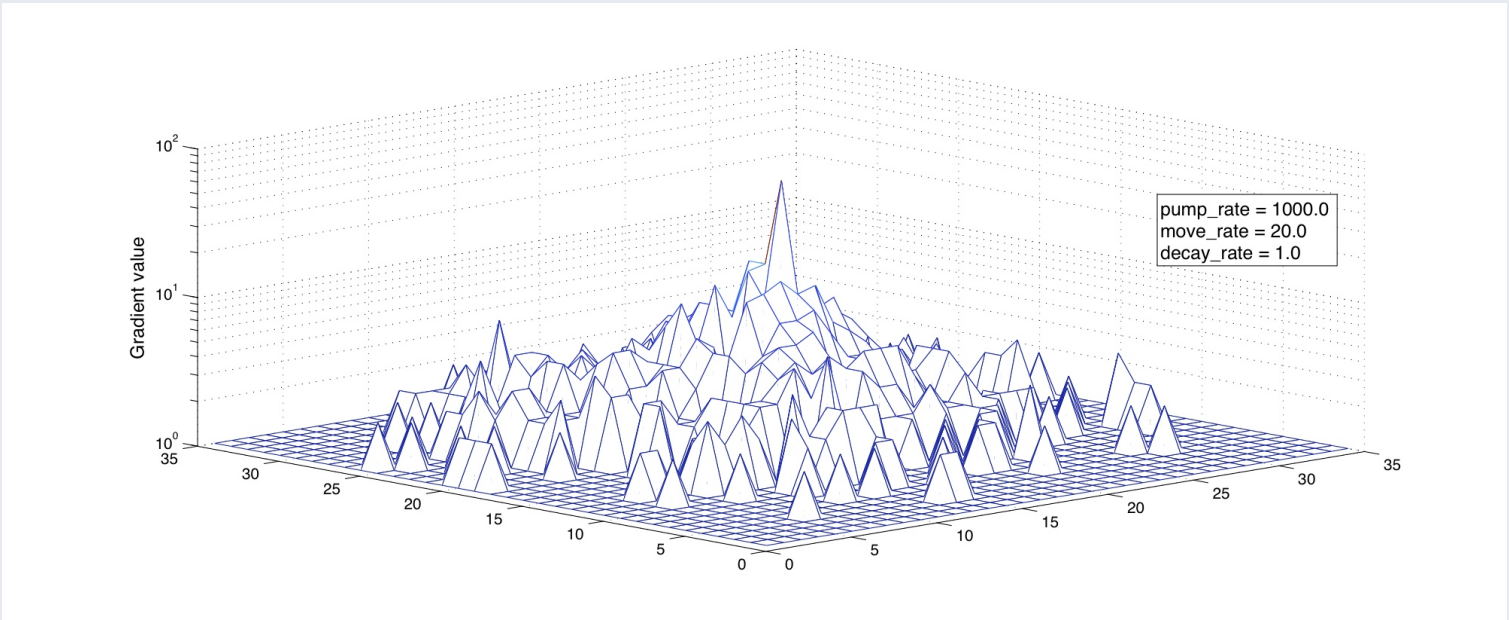


Computational field (a.k.a. gradient)

Chemical laws: a single PUMP token generates a GRADIENT

$$
\begin{array}{lll}
\text{(PUMP)} & \text{PUMP} \xrightarrow{p} & \text{PUMP} + \text{GRAD} \\
\text{(DECAY)} & \text{GRAD} \xrightarrow{d} & 0 \\
\text{(MOVE)} & \text{GRAD} \xrightarrow{m} & \text{GRAD}^{\rightsquigarrow}(\text{GRAD}^{-})
\end{array}
$$

Chart



Discussion

Open issues

- Prey-predator dynamics is interesting, but is there more?
  Should check patterns based on other models of population dynamics
- Is chemical dynamics (and the concentration concept) sufficient, or more expressiveness is needed? is it too resource-demanding?
- Can we flexibly obtain various fields, e.g., *à la* Tota?
- What about self-composition/aggregation of services? (see [3])
- How to fit expressive semantic representation in this model?

A suggested schedule

Milestones

- M4: WP2/WP3 should think at known/new algorithms in terms of chemical rules "like" the above ones – without making hard assumptions on what chemical-like means
- M8: With those hints, an early operational model for eco-laws will be available
- M12: First version of the operational model, with tentative semantic framework
- M20: Complete operational model (eco-laws + LSAs + Components template)

Daniel T. Gillespie.
Exact stochastic simulation of coupled chemical reactions.
The Journal of Physical Chemistry, 81(25):2340–2361, 1977.

Elena Nardini, Mirko Viroli, and Emanuele Panzavolta.
Coordination in open and dynamic environments with tucson semantic tuple centres.
In *25th Annual ACM Symposium on Applied Computing (SAC 2010)*, volume III, pages 2037–2044, Sierre, Switzerland, 22–26 March 2010. ACM.

Mirko Viroli and Matteo Casadei.
Chemical-inspired self-composition of competing services.
In Sung Y. Shin, Sascha Ossowski, Michael Schumacher, Mathew Palakal, Chih-Cheng Hung, and Dongwan Shin, editors, *25th Annual ACM Symposium on Applied Computing (SAC 2010)*, volume III, pages 2029–2036, Sierre, Switzerland, 22–26 March 2010. ACM.

Mirko Viroli, Matteo Casadei, Elena Nardini, and Andrea Omicini.
Towards a chemical-inspired infrastructure for self-* pervasive applications.
In Danny Weyns, Sam Malek, Rogério de Lemos, and Jesper Andersson, editors, *Self-Organizing Architectures*, volume 6090 of *LNCS*, chapter 8, pages 152–176. Springer, July 2010.
1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 September 2009, Revised Selected and Invited Papers.