$\mathcal{M}$olecules $\mathcal{O}$f $\mathcal{K}$nowledge: Self-Organisation in Knowledge-Intensive Environments

Stefano Mariani, Andrea Omicini
{s.mariani, andrea.omicini}@unibo.it

Dipartimento di Informatica: Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna

Laboratory of Systems and Applications 2012/2013
Master Degree in Computer Science Engineering

Outline

Where It All Started I

From my own Master Degree Thesis introduction (roughly):

> *"Information specialists are facing new and critical challenges in their knowledge production process: the increasing amount of information to mine, the pace at which it's made available and the heterogeneity of its structure are just a few to mention.*
> *[...]*
> *By developing models, technologies, and tools to explore the new landscape of information, computer scientists can help people to discover, manage, and publish information at lower costs."*

What can we learn?

Where It All Started II

A lot of software systems, nowadays, are in some way socio-technical knowledge-intensive environments, that is systems in which:

- a *huge* amount of information has to be handled
- (human) users' *behaviour* deeply affects the system own behaviour

The challenge

As software engineers, we should build systems capable of dealing with such issues, which could mean:

- able to self-organise information such as the most useful survives and "grows" whereas the useless one disappears
- able to self-organise information such as it *spontaneously* flows toward more interested users whereas running away from less interested ones

What I Observed I

Again, quotes from myself:

> *"People reach the knowledge they need from different sources of information. Such sources could be either external or internal to their working sw system.*
> *[. . .]*
> *Whichever is the nature of a source of knowledge, hopefully either (i) it is already structured or (ii) there exists a proper sw entity able to do so."*

Ok. . . so?

What I Observed II

Unfortunately, knowledge people needs to achieve their goals is often *messed up* with information they don't care much about. Worse, information can be represented according to *different formats*—e.g. XML, JSON, OWL, HTML, ...

How to solve?

We really need many many things...

- efficient *natural language processing* techniques to structure otherwise unstructured information
- *text mining* to summarize and classify information
- *reasoning engines* to correlate knowledge for which some semantical relationship exists
- *user profiling* methods to distinguish useful information from useless one
- last but not least, *something* to put all this stuff together!

What I Envisioned I

"These sources could be reified within the sw system as "seeds", continuously and autonomously producing "atoms of knowledge", which have to be seen as autonomous "living" pieces of knowledge.
[...]
Such injection should not be a "one-shot" operation, rather it should be continuous in time, according to an injection rate which should be dynamic, thus changing according to the system's state and users' expected behaviour.
[...]
Furthermore, every single injection should not produce a single atom, but a (dynamically varying) number of identical copies of an atom, let's say, its "concentration"—as in chemistry."

For God's sake, why??

What I Envisioned II

Adaptive and self-organising systems seem the only possible answer when

- the *scale* of the problem is too big—too much data available
- *unpredictability* too high—"what the hell the user is doing??"
- global *control* unrealistic—anyone has its own smartphone
- *deterministic* solutions simply won't work [Omicini and Viroli, 2011]—look at natural systems...

$\mathcal{M}o\mathcal{K}$ purpose

The goal of the $\mathcal{M}$olecules $\mathcal{O}$f $\mathcal{K}$nowledge model is that of building a self-* software system able to deal with all the above issues by drawing inspiration from the many natural systems studied so far—e.g. physical systems, social systems, biochemical systems, ...

Outline

Outline

What's $\mathcal{M}o\mathcal{K}$?

$\mathcal{M}$olecules $\mathcal{O}$f $\mathcal{K}$nowledge ($\mathcal{M}o\mathcal{K}$)

$\mathcal{M}o\mathcal{K}$ is a biochemically-inspired *coordination model* promoting self-organisation of knowledge toward the idea of self-organising workspaces [Omicini, 2011]:

- knowledge sources produce atoms of knowledge in biochemical compartments, which then *may* diffuse and/or aggregate in molecules by means of biochemical reactions, acting locally within and between such spaces
- knowledge prosumers *workspaces* are mapped into such compartments, which *reify* information-oriented user actions to *drive* atoms and molecules aggregation and diffusion

Wait...what??

Coordination — Our goal is to autonomously compose information in the attempt to obtain meaningful knowledge

→ we need a coordination model to *govern* interactions between independent data chunks

Sources — No way to know a priori which information is useful and which not, *at any time*, *for any user*

→ we need techniques to keep sources within the system, even if they are temporarily unused

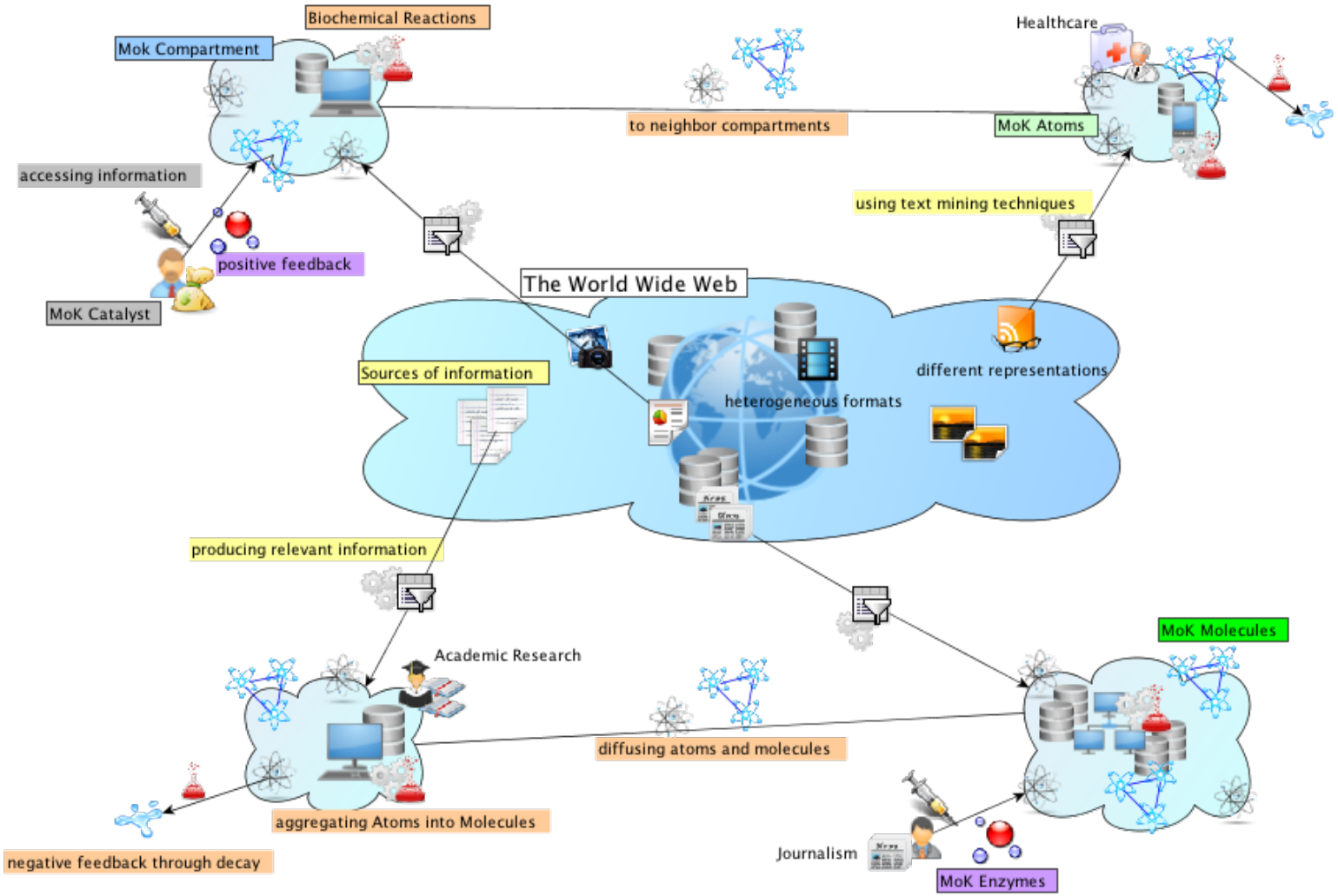
Reactions — As much work as possible should be done by the system, *transparently* to users

→ we need a *proactive* computational model able to spontaneously start computations

Reification — Being our target a socio-techinal scenario, users will have a central role to play in the system

→ hence the system must be able to properly model users and their actions in order to promptly react to their stimuli

Envisioning $\mathcal{M}o\mathcal{K}$ Systems



$\mathcal{M}o\mathcal{K}$ Abstractions I

atoms the smallest *unit of knowledge* in $\mathcal{M}o\mathcal{K}$, contain information from a source and belong to a compartment—thus being subject to its "laws of nature"

molecules the $\mathcal{M}o\mathcal{K}$ units for knowledge aggregation, *bond* together *"somehow-related"* atoms

enzymes emitted by $\mathcal{M}o\mathcal{K}$ catalysts, represent *prosumer's actions* and participate $\mathcal{M}o\mathcal{K}$ reactions to *affect* the way in which atoms and molecules evolve

reactions working at a given rate, they drive the evolution of each $\mathcal{M}o\mathcal{K}$ compartment, by ruling the way in which molecules *aggregate*, are reinforced, *diffuse*, and decay

$\mathcal{M}o\mathcal{K}$ Abstractions II

compartments the spatial abstraction of $\mathcal{M}o\mathcal{K}$, compartments represent the conceptual loci for all $\mathcal{M}o\mathcal{K}$ entities as well as for $\mathcal{M}o\mathcal{K}$ biochemical processes, also providing $\mathcal{M}o\mathcal{K}$ with the notions of locality and neighbourhood

sources each one associated to a compartment, $\mathcal{M}o\mathcal{K}$ sources are the *origins of knowledge*, which is continuously injected at a certain *rate* in the form of $\mathcal{M}o\mathcal{K}$ atoms

catalysts the abstraction for *knowledge prosumers*, catalysts emit enzymes in order to attract to him/her relevant knowledge items

Wrap-up

Sources

$\mathcal{M}o\mathcal{K}$ sources, atoms and molecules, represent the information within the system *at a certain point in time and space*.

Reactions

$\mathcal{M}o\mathcal{K}$ biochemical reactions and compartments are the proactive part of a $\mathcal{M}o\mathcal{K}$ system, meant to support, ease, even replace users work.

Reification

$\mathcal{M}o\mathcal{K}$ enzymes, catalysts and compartments together bring users actions as well as their effects inside a $\mathcal{M}o\mathcal{K}$ system, to be exploited within reactions in order to drive $\mathcal{M}o\mathcal{K}$ behaviour.

Outline

The $\mathcal{M}o\mathcal{K}$ Model I

Atoms

Belonging to a *source*, atoms carry a piece of *data*, possibly some *metadata* and keep track of their concentration in the local compartment

$$\texttt{atom}(src,\ val,\ attr)^c$$

Molecules

Molecules are unordered collections of *somehow related atoms*, again equipped with a concentration value

$$\texttt{molecule}(Atoms)^c$$

Enzymes

Enzymes are *strictly coupled* to the atom/molecule being accessed—and, as usual, have a concentration attached

$$\texttt{enzyme}(Atoms)^c$$

The $\mathcal{M}o\mathcal{K}$ Model II

Aggregation

In the *aggregation reaction* two *somehow related* atoms/molecules[a] are joined to spring a new molecule

$$\texttt{molecule}(Atoms_1) + \texttt{molecule}(Atoms_2) \longmapsto^{r_{agg}} \texttt{molecule}(Atoms_1 \biguplus Atoms_2) + Residual(Atoms_1 \bigcup Atoms_2)$$

[a] From now on we'll se atoms as "singleton" molecules.

Reinforcement

Reinforcement reaction increases the concentration of molecules by consuming *compliant* enzymes from the local compartment

$$\texttt{enzyme}(Molecule_1) + Molecule_1^c \longmapsto^{r_{reinf}} Molecule_1^{c+1}$$

The $\mathcal{M}o\mathcal{K}$ Model III

Decay

Enforcing time situatedness, molecules should fade away as time passes

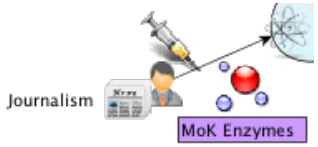
$$Molecule^{c} \longmapsto^{r_{decay}} Molecule^{c-1}$$

Diffusion

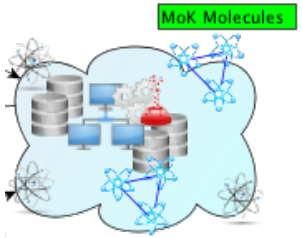
Analogously, space situatedness is based upon *diffusion*, being inspired by biology

$$\{Molecule_1 \bigcup Molecules_1\}_{\sigma_i} + \{Molecules_2\}_{\sigma_{ii}} \longmapsto^{r_{diffusion}} \{Molecules_1\}_{\sigma_i} + \{Molecules_2 \bigcup Molecule_1\}_{\sigma_{ii}}$$

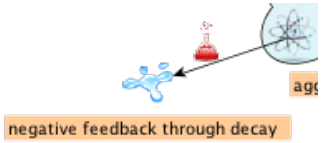
Imagine...



...as users are busy doing their work, e.g. searching for news,...

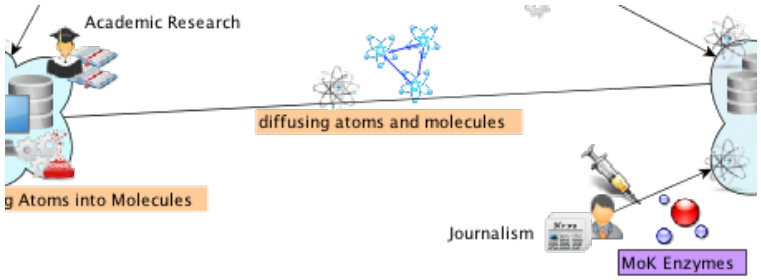


...they release enzymes within their compartment, which are then used to increment the concentration of the molecules being accessed...



...meanwhile, non-accessed molecules decay...

...and any molecule may migrate to neighbours compartment, giving them a chance to be reinforced by other users



$\mathcal{M}o\mathcal{K}$ Model IV

What is "somehow related/compliant"?

$\mathcal{M}o\mathcal{K}$ reactions should check reactants correlation to apply, so, in order to define a $\mathcal{M}o\mathcal{K}$ system, one should first of all define a $\mathcal{F}_{\mathcal{M}o\mathcal{K}}$ function $\mathcal{F}_{\mathcal{M}o\mathcal{K}}$: *molecule* $\times$ *molecule* $\mapsto D$, which takes two molecules and determines if (and *how much*) they are related.

The $\mathcal{F}_{\mathcal{M}o\mathcal{K}}$ function

The exact definition of $\mathcal{F}_{\mathcal{M}o\mathcal{K}}$ – that is the mathematical description of domain D – depends on the application at hand, however will likely depend on the fields `val` and `attr` inside $\mathcal{M}o\mathcal{K}$ atoms.

N.B. The $\mathcal{F}_{\mathcal{M}o\mathcal{K}}$ function could range from the simple LINDA syntactical matching – hence $D = \{\texttt{true}, \texttt{false}\}$ – to more complex semantical fuzzy matching mechanisms—for which typically $D \in [0, 1]$.

On the Table

Semantics — Matching between actual molecules wandering in a compartment and molecules templates within reactions *must be semantic* to have a meaningful aggregation of knowledge

Stigmergy — Enzymes are crucial in the process of driving a compartment behaviour toward its user needs, 'cause they are the *traces of its own behaviour*

Chemistry — $\mathcal{M}o\mathcal{K}$ reactions execution is driven by rates, which in turn are influenced by molecules concentrations, thus *the computational model is that of a chemical reaction*

Situatedness — Information relevance may change depending on *when* it is available and *where* (also *to whom*, of course), hence *spatio-temporal situatedness is a mandatory feature*

Outline

Tuple-based Coordination Models I

Coordination models

Tuple-based coordination models and languages have already shown their effectiveness in the engineering of *complex software systems*, like knowledge-intensive, pervasive and *self-organising* ones [Omicini and Viroli, 2011].

Tuple-based Coordination Models II

Biochemical tuple spaces

The biochemical tuple space abstraction brings self-organisation into tuple-based coordination, by exploiting the *(bio)chemical metaphor* enhanced with *topology* aspects [Viroli and Casadei, 2009].

- → tuples are seen as chemical reactants, thus equipped with an *activity/pertinency* value—resembling chemical concentration[1]
- → chemical reactions evolve tuples and possibly diffuse them to neighboring chemical compartments
- → tuple spaces act as *chemical solutions simulators*, that is update concentrations following the *Gillespie algorithm* [Gillespie, 1977], host and execute chemical reactions and manage the topology-related features

[1] Their relative quantity w.r.t. the others.

Outline

TuCSoN & ReSpecT I

TuCSoN [Omicini and Zambonelli, 1999]

TuCSoN is a LINDA-inspired coordination model & infrastructure providing developers with a distributed, tuple-based middleware exploiting *programmable tuple spaces* called tuple centres.

ReSpecT [Omicini, 2006]

The behaviour of such tuple centres can be programmed through the ReSpecT logic language so to encapsulate any coordination laws directly into the coordination abstraction.

TuCSoN & ReSpecT II

TuCSoN provides the basic ingredients to enable biochemical coordination:

topology multiple tuple centres can be deployed in different nodes of a network and/or can coexist in a single node, promoting the notions of locality and neighbourhood

programmability chemically-inspired evolution of tuples is enabled by implementing the Gillespie algorithm [Gillespie, 1977] as a ReSpecT program

matching general purpose $\mathcal{M}o\mathcal{K}$ reactions can be applied to actual reactants by using ReSpecT logic tuples and templates to represent atoms, molecules and enzymes

Outline

TuCSoN for $\mathcal{M}o\mathcal{K}$ I

$\mathcal{M}o\mathcal{K}$ {atoms, molecules, enzymes} $\mapsto$ ReSpecT logic tuples

Being generated, accessed, moved and consumed both by users and by the $\mathcal{M}o\mathcal{K}$ system itself – through reactions –, atoms, molecules and enzymes could all be implemented as ReSpecT tuples so to be effectively managed by the TuCSoN middleware.

$\mathcal{M}o\mathcal{K}$ compartments $\mapsto$ TuCSoN tuple centres

By definition, compartments are the *locality* abstraction in $\mathcal{M}o\mathcal{K}$, thus the mapping with TuCSoN tuple centres is straightforward since they host both:

- the ordinary tuples, that is data chunks—thus atoms, molecules and enzymes
- the specification tuples, that is ReSpecT programs statements—hence $\mathcal{M}o\mathcal{K}$ reactions

TuCSoN for $\mathcal{MoK}$ II

$\mathcal{MoK}$ reactions $\longmapsto^*$ ReSpecT programs

$\mathcal{MoK}$ reactions are simply declarative statements specifying how existing knowledge should combine, fade away, replicate or move, hence they need to be *interpreted* and *executed*. Furthermore, being chemically-inspired, this should be done according to Gillespie's chemical simulation algorithm.

$\mathcal{MoK}$ reactions $\mapsto$ ReSpecT logic tuples $\mapsto$ ReSpecT programs

So, mapping to TuCSoN could be "two-layered":

- $\mathcal{MoK}$ reactions are encapsulated into ReSpecT tuples of the kind `law([Inputs], Rate, [Outputs]);`
- such tuples basically constitute the *raw data* consumed by the ReSpecT implementation of Gillespie algorithm—which continuously, in a chemical-like fashion, schedules and executes them.

Outline

Why News

News management systems are a prominent example of knowledge-intensive, socio-technical systems, due to:

heterogeneity News sources can be virtually anything, from handwritten notes to printed official documents through web published articles

ubiquity Netbooks, tablets and smartphones pushed information production, sharing and consumption to be *pervasive* as never before

unpredictability News producers are no longer graduated journalists solely, they include bloggers and whoever has access to the web though

$\mathcal{M}o\mathcal{K}$-News Model

Formally

A generic $\mathcal{M}o\mathcal{K}$ atom of the form $\texttt{atom(src, val, attr)}_c$ becomes a specialised $\mathcal{M}o\mathcal{K}$-News [Mariani and Omicini, 2012] atom of the form

$$\texttt{atom(src, val, sem(tag, catalog))}_c$$

where

```
src ::= news source uri
val ::= news content
attr ::= sem(tag, catalog)
    tag ::= NewsML tag | NITF tag
    catalog ::= NewsCode uri | ontology uri
```

Envisioning $\mathcal{M}o\mathcal{K}$-News Systems I

A $\mathcal{M}o\mathcal{K}$-News system should hence be seen as a self-organising news repository in which

- ! news pieces – "tag-content" pairs – are injected either automatically (e.g. using XML parsers) or manually (by journalists) in the form of $\mathcal{M}o\mathcal{K}$-News atoms
- ! enzymes are released by catalysts (journalists) as manifestations of their actions over knowledge
- ! biochemical reactions

  aggregate: together *semantically related* atoms — based upon catalog information

  diffuse: atoms/molecules in neighborhood compartments

  reinforce: them by using enzymes

  decay: non-relevant information

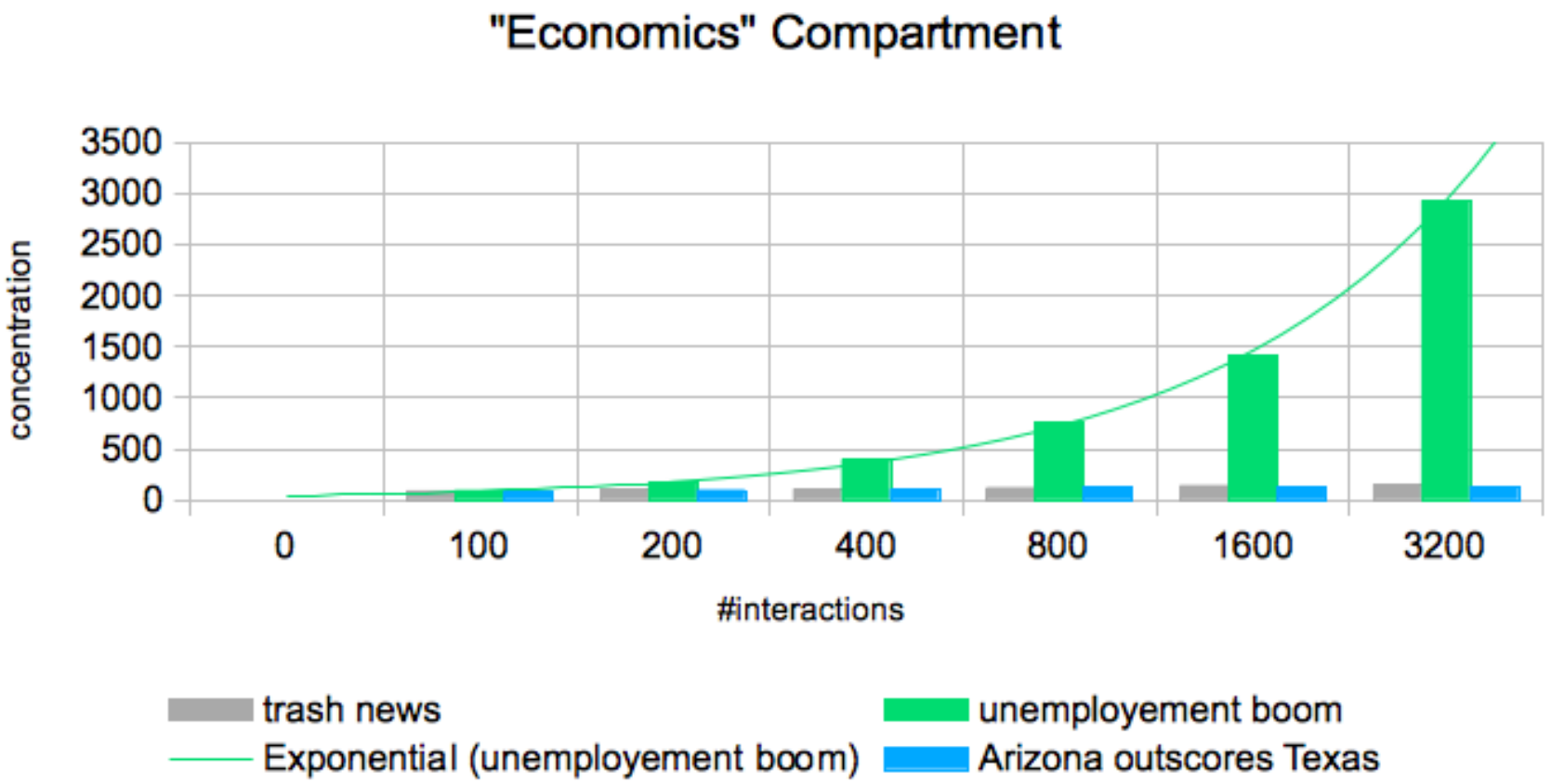
Envisioning $\mathcal{M}o\mathcal{K}$-News Systems II

"Smart" diffusion

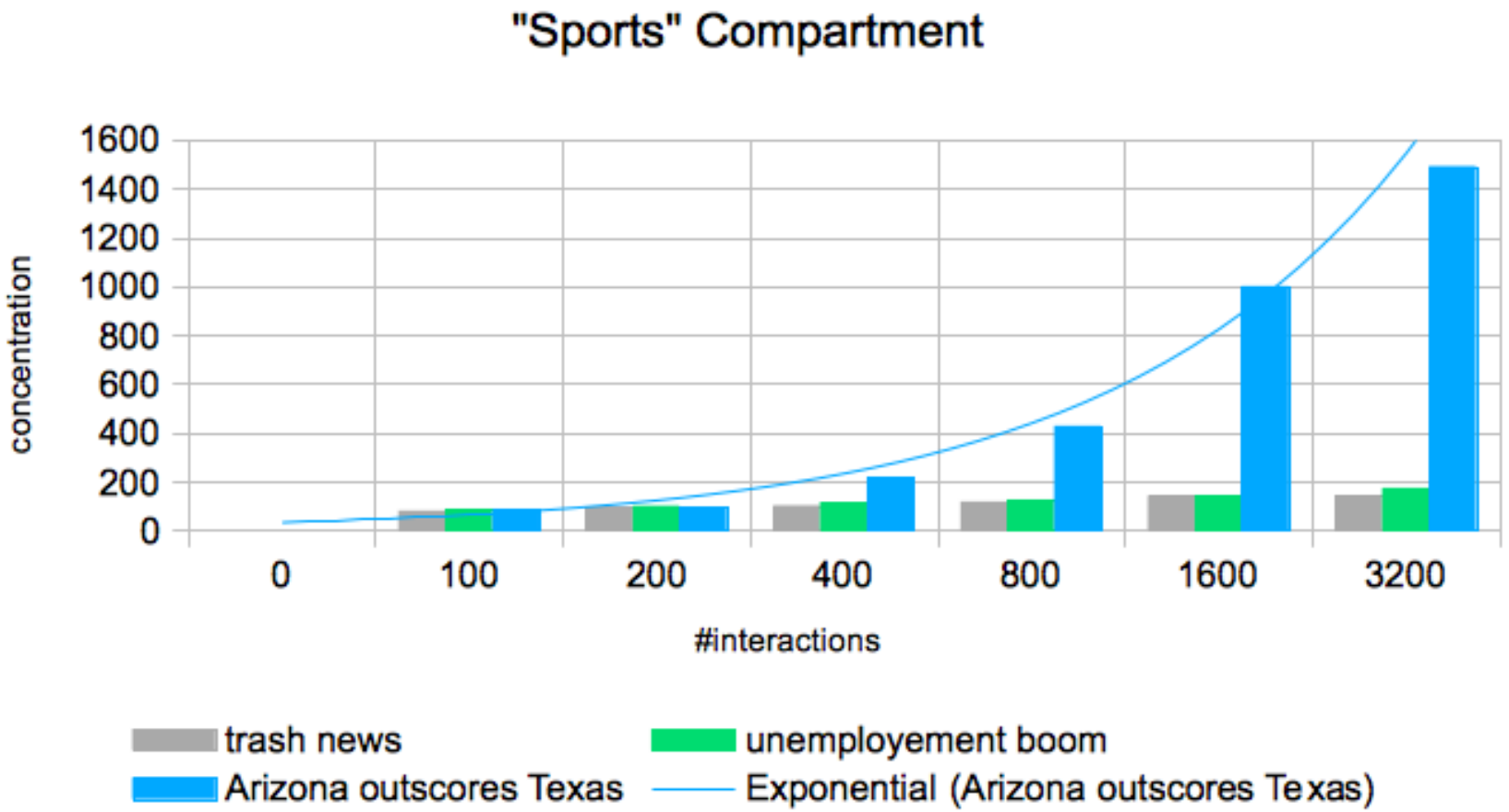
It is achieved as a self-organising process caused by the cooperation among diffusion, reinforcement – of relevant knowledge, that is more frequently accessed – and decay—of useless information, ignored by catalysts.

E.g., a journalist interested in sports news is *more likely* to search, read, annotate – generally, *access* – sport-related atoms. In the process, she releases enzymes which reinforce accessed molecules concentration. In the very end, her compartment will *mainly store* sports-related knowledge.

Envisioning $\mathcal{M}o\mathcal{K}$-News Systems III



Envisioning $\mathcal{M}o\mathcal{K}$-News Systems IV



A stochastic equilibrium between diffusion, reinforcement and decay laws, makes a "smart migration" pattern appear by emergence.

Outline

Final Remark

$\mathcal{M}$olecules $\mathcal{O}$f $\mathcal{K}$nowledge

The $\mathcal{M}o\mathcal{K}$ model

- → provides knowledge workers in general with a novel approach both in thinking and managing knowledge
- → supports their work through self-organising shared workspaces able to autonomously cluster and spread information

Further Developments

How to...

? push the $\mathcal{M}o\mathcal{K}$ model toward the idea of *self-organising workspace* [Omicini, 2011], fully supporting adaptiveness of compartments rather than information solely?

? effectively implement efficient semantic matching mechanisms [Nardini et al., 2013] to lift LINDA purely syntactical one currently exploited in TuCSoN?

Bibliography I

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

Mariani, S. and Omicini, A. (2012).
Self-organising news management: The *Molecules of Knowledge* approach.
In Fernandez-Marquez, J. L., Montagna, S., Omicini, A., and Zambonelli, F., editors, *1st International Workshop on Adaptive Service Ecosystems: Natural and Socially Inspired Solutions (ASENSIS 2012)*, pages 11–16, SASO 2012, Lyon, France.
Pre-proceedings.

Nardini, E., Omicini, A., and Viroli, M. (2013).
Semantic tuple centres.
Science of Computer Programming, 78(5):569–582.
Special section: Self-Organizing Coordination.

Omicini, A. (2006).
Formal ReSpecT in the A&A perspective.
In Canal, C. and Viroli, M., editors, *5th International Workshop on Foundations of Coordination Languages and Software Architectures (FOCLASA'06)*, pages 93–115, CONCUR 2006, Bonn, Germany. University of Málaga, Spain.
Proceedings.

Bibliography II

Omicini, A. (2011).
Self-organising knowledge-intensive workspaces.
In Ferscha, A., editor, *Pervasive Adaptation. The Next Generation Pervasive Computing Research Agenda*, chapter VII: Human-Centric Adaptation, pages 71–72. Institute for Pervasive Computing, Johannes Kepler University Linz, Austria.

Omicini, A. and Viroli, M. (2011).
Coordination models and languages: From parallel computing to self-organisation.
The Knowledge Engineering Review, 26(1).
Special Issue for the 25th Years of the Knowledge Engineering Review.

Omicini, A. and Zambonelli, F. (1999).
Coordination for Internet application development.
Autonomous Agents and Multi-Agent Systems, 2(3):251–269.
Special Issue: Coordination Mechanisms for Web Agents.

Viroli, M. and Casadei, M. (2009).
Biochemical tuple spaces for self-organising coordination.
In Field, J. and Vasconcelos, V. T., editors, *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 143–162. Springer, Lisbon, Portugal.

$\mathcal{M}$olecules $\mathcal{O}$f $\mathcal{K}$nowledge: Self-Organisation in Knowledge-Intensive Environments

Stefano Mariani, Andrea Omicini
{s.mariani, andrea.omicini}@unibo.it

Dipartimento di Informatica: Scienza e Ingegneria (DISI)
Alma Mater Studiorum—Università di Bologna

Laboratory of Systems and Applications 2012/2013
Master Degree in Computer Science Engineering