Molecules of Knowledge: 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

IDC 2012
Intelligent Distributed Computing
Calabria, Italy – 25th of September 2012

1. Motivations
2. The Molecules of Knowledge Model
   - Informal MoK
   - Formal MoK
3. A MoK Infrastructure
   - The TuCSoN Coordination Infrastructure
   - Mapping MoK over TuCSoN
4. A Case Study: MoK-`News`
5. Conclusions & Further Works

Outline

1. Motivations
2. The Molecules of Knowledge Model
   - Informal MoK
   - Formal MoK
3. A MoK Infrastructure
   - The TuCSoN Coordination Infrastructure
   - Mapping MoK over TuCSoN
4. A Case Study: MoK-News
5. Conclusions & Further Works

The Challenge

Knowledge-intensive environments (KIE)

KIE present new critical challenges in the *knowledge management process*: the ever-increasing amount of information to handle, its heterogeneity in structure, and the pace at which it is made available are just a few to mention.

Knowledge workers

For journalists, researchers, lawyers and the like, today ICT systems provide both new opportunities and new obstacles: finding *all and only* the relevant information they need is a issue that even the most advanced general-purpose research engines are not able to face today.

A Tuple-based Answer I

Adaptive and self-organising systems. . .

. . . seem the only possible answer when the scale of the problem is too huge, *unpredictability* too high, global control unrealistic, and *deterministic* solutions simply do not work [Omicini and Viroli, 2011].

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].

A Tuple-based Answer II

Biochemical tuple spaces

They self-organisation features into tuple-based coordination, by exploiting the *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 [a]

→ 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 the chemical reactions and manage the topology-related features

[a] their relative quantity w.r.t. the others

Goals

On one hand...

...to bring the biochemical tuple space abstraction and its self-organising features to their full realization into *knowledge intensive environments*, so to harness their complexity.

On the other hand...

...to help knowledge workers, by providing them a model in which knowledge autonomously aggregate in a meaningful and useful way and eventually diffuse to autonomously reach knowledge consumers — rather than be "searched".

Outline

1. Motivations

2. The Molecules of Knowledge Model
   - Informal MoK
   - Formal MoK

3. A MoK Infrastructure
   - The TuCSoN Coordination Infrastructure
   - Mapping MoK over TuCSoN

4. A Case Study: MoK-News

5. Conclusions & Further Works

MoK Vision

Molecules of Knowledge (MoK)

MoK 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 consumers *workspaces* are mapped into such compartments, which *reify* information-oriented user actions to *drive* atoms and molecules aggregation and diffusion.

MoK Abstractions I

MoK main abstractions

atoms the smallest *unit of knowledge* in MoK, contain information from a source and belong to a compartment — thus being subject to its "laws of nature"

molecules the MoK units for knowledge aggregation, *bond* together "somehow-related" atoms

enzymes emitted by MoK catalysts, represent *prosumer's actions* and participate MoK reactions to *affect* the way in which atoms and molecules evolve

reactions working at a given rate [a], they regulate the evolution of each MoK compartment, by ruling the way in which molecules *aggregate*, are reinforced, *diffuse*, and decay

[a] affected by atoms/molecules concentrations

MoK Abstractions II

MoK other abstractions

compartments the spatial abstraction of MoK, compartments represent the conceptual loci for all MoK entities as well as for MoK biochemical processes, also providing MoK with the notions of locality and neighbourhood

sources each one associated to a compartment, MoK sources are the *origins of knowledge*, which is continuously injected at a certain *rate* in the form of MoK atoms

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

Envisioning MoK Systems

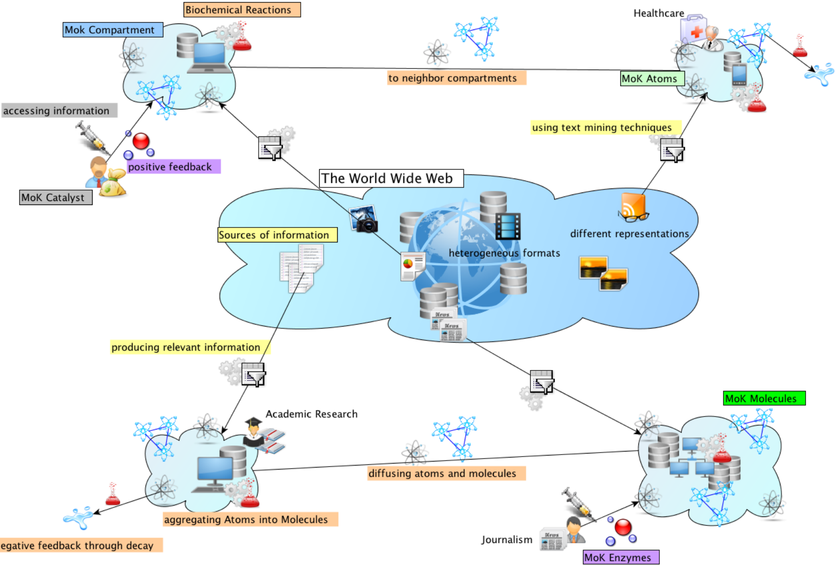
MoK systems

MoK systems should be seen as a *network of distributed, shared information spaces* in which some source entities continuously inject *information pieces*. These may then aggregate to shape more complex knowledge chunks and diffuse between different, networked shared spaces.

MoK users

Users can interact with the system through information-oriented actions over knowledge, which are *reified in terms of enzymes* by their associated workspace and exploited to influence *knowledge lifecycle*.

A MoK System


Biochemical Reactions
Mok Compartment
Healthcare
to neighbor compartments
MoK Atoms
accessing information
using text mining techniques
positive feedback
The World Wide Web
MoK Catalyst
Sources of information
different representations
heterogeneous formats
producing relevant information
MoK Molecules
Academic Research
diffusing atoms and molecules
aggregating Atoms into Molecules
negative feedback through decay
Journalism
MoK Enzymes

MoK Formal Model I

MoK main abstractions syntax

It is straightforwardly derived from abstractions descriptions:

atoms belong to a source, carry a piece of data, possibly some metadata and are equipped with their concentration in the space

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

molecules are unordered collections of somehow related atoms, again with a concentration

$$\texttt{molecule(Atoms)}_c$$

enzymes are strictly coupled to the atom/molecule being accessed, they too equipped with their concentration

$$\texttt{enzyme(Atoms)}_c$$

MoK Formal Model II

Reactions semantics

It should comply with the following rewriting rules

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

$$\texttt{molecule(Atoms}_1\texttt{)} + \texttt{molecule(Atoms}_2\texttt{)} \xmapsto{\texttt{r\_agg}}$$

$$\texttt{molecule(Atoms}_1 \bigcup \texttt{Atoms}_2\texttt{)} + \texttt{Residual(Atoms}_1\texttt{, Atoms}_2\texttt{)}$$

reinforcement atoms/molecules are reinforced by consuming compliant enzymes from the space

$$\texttt{enzyme(Atoms}_1\texttt{)} + \texttt{molecule(Atoms}_2\texttt{)}_c \xmapsto{\texttt{r\_reinf}}$$

$$\texttt{molecule(Atoms}_2\texttt{)}_{c+1}$$

[a] atoms can be seen as molecules with a single atom inside

MoK Formal Model III

Reactions semantics [continue]

decay molecules naturally fade away during time

$$\texttt{molecule(Atoms)}_c \xrightarrow{\texttt{r\_decay}} \texttt{molecule(Atoms)}_{c-1}$$

diffusion molecules randomly diffuse to somehow defined neighbouring compartments

$$\begin{gathered} \| \ \texttt{Molecules}_1 \bigcup \texttt{molecule}_1 \ \|_{\sigma^i} + \| \ \texttt{Molecules}_2 \ \|_{\sigma^{ii}} \\ \xrightarrow{\texttt{r\_diffusion}} \\ \| \ \texttt{Molecules}_1 \ \|_{\sigma^i} + \| \ \texttt{Molecules}_2 \bigcup \texttt{molecule}_1 \ \|_{\sigma^{ii}} \end{gathered}$$

N.B.: MoK other abstractions and MoK reactions syntax is left unspecified: it can be tailored to the application domain or legacy infrastructure language at hand.

MoK Formal Model IV

What is “somehow related/compliant”?

MoK reactions should check reactants correlation to apply, so in order to define a MoK system one should first of all define a *mok* function *mok*: *molecule* × *molecule* ↦ *D*, which takes two atoms/molecules and verifies if they are related.

The *mok* function

The exact definition of *mok* – 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 MoK atoms.

N.B. The *mok* 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 $D \in [0, 1]$.

Outline

1. Motivations
2. The Molecules of Knowledge Model
   - Informal MoK
   - Formal MoK
3. A MoK Infrastructure
   - The TuCSoN Coordination Infrastructure
   - Mapping MoK over TuCSoN
4. A Case Study: MoK-News
5. Conclusions & Further Works

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

MoK requirements

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 MoK reactions can be applied to actual reactants by using ReSpecT logic tuples and templates to represent atoms, molecules and enzymes

TuCSoN for MoK I

MoK {atoms, molecules, enzymes} ↦ ReSpecT logic tuples

Being generated, accessed, moved and consumed both by users and by the MoK system itself – through reactions –, atoms, molecules and enzymes should all be implemented as ReSpecT tuples so to be effectively managed by the TuCSoN middleware.

MoK compartments ↦ TuCSoN tuple centres

By definition, compartments are the *locality* abstraction in MoK, 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 MoK reactions

TuCSoN for MoK II

MoK reactions $\longmapsto^*$ ReSpecT programs

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.

MoK reactions $\mapsto$ ReSpecT logic tuples $\mapsto$ ReSpecT programs

So actual mapping to TuCSoN is “two-layered”:

- 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

1 Motivations

2 The Molecules of Knowledge Model
- Informal MoK
- Formal MoK

3 A MoK Infrastructure
- The TuCSoN Coordination Infrastructure
- Mapping MoK over TuCSoN

4 A Case Study: MoK-News

5 Conclusions & Further Works

Why News

News management systems

They are a prominent example of

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

MoK-News Model

Formally

A generic MoK atom of the form `atom(src, val, attr)`$_c$ becomes a specialised MoK-News [Mariani and Omicini, 2012] atom of the form

`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 MoK-News Systems I

A MoK-News systems...

...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 MoK-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 MoK-News Systems II

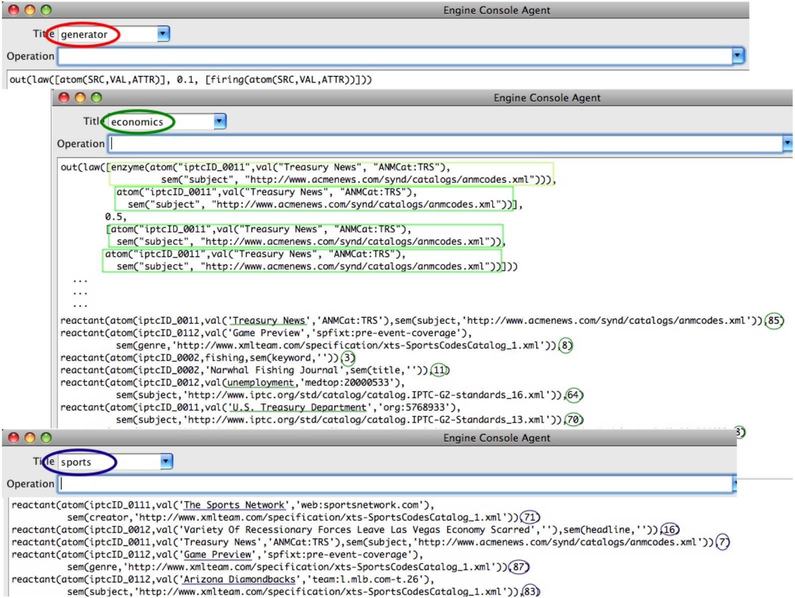
"Smart" diffusion

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, he/she releases enzymes which reinforce accessed atoms/molecules concentration. In the very end, his/her compartment will *mainly store* sports-related knowledge.

Envisioning MoK-News Systems III



Title generator

```
out(law([atom(SRC,VAL,ATTR)], 0.1, [firing(atom(SRC,VAL,ATTR))]))
```

Title economics

```
out(law([enzyme(atom("iptcID_0011",val("Treasury News", "ANMCat:TRS"),
              sem("subject", "http://www.acmenews.com/synd/catalogs/anmcodes.xml"))),
          atom("iptcID_0011",val("Treasury News", "ANMCat:TRS"),
            sem("subject", "http://www.acmenews.com/synd/catalogs/anmcodes.xml"))],
        0.5,
        [atom("iptcID_0011",val("Treasury News", "ANMCat:TRS"),
          sem("subject", "http://www.acmenews.com/synd/catalogs/anmcodes.xml")),
        atom("iptcID_0011",val("Treasury News", "ANMCat:TRS"),
          sem("subject", "http://www.acmenews.com/synd/catalogs/anmcodes.xml"))]))
  ...
  ...
  ...

reactant(atom(iptcID_0011,val('Treasury News','ANMCat:TRS'),sem(subject,'http://www.acmenews.com/synd/catalogs/anmcodes.xml')),85)
reactant(atom(iptcID_0112,val('Game Preview','spfixt:pre-event-coverage'),
         sem(genre,'http://www.xmlteam.com/specification/xts-SportsCodesCatalog_1.xml')),8)
reactant(atom(iptcID_0002,fishing,sem(keyword,'')),3)
reactant(atom(iptcID_0002,'Narwhal Fishing Journal',sem(title,'')),11)
reactant(atom(iptcID_0012,val(unemployment,'medtop:20000533'),
         sem(subject,'http://www.iptc.org/std/catalog/catalog.IPTC-G2-standards_16.xml')),64)
reactant(atom(iptcID_0011,val('U.S. Treasury Department','org:5768933'),
         sem(subject,'http://www.iptc.org/std/catalog/catalog.IPTC-G2-Standards_13.xml')),70)
```

Title sports

```
reactant(atom(iptcID_0111,val('The Sports Network','web:sportsnetwork.com'),
         sem(creator,'http://www.xmlteam.com/specification/xts-SportsCodesCatalog_1.xml')),71)
reactant(atom(iptcID_0012,val('Variety Of Recessionary Forces Leave Las Vegas Economy Scarred',''),sem(headline,'')),16)
reactant(atom(iptcID_0011,val('Treasury News','ANMCat:TRS'),sem(subject,'http://www.acmenews.com/synd/catalogs/anmcodes.xml')),7)
reactant(atom(iptcID_0112,val('Game Preview','spfixt:pre-event-coverage'),
         sem(genre,'http://www.xmlteam.com/specification/xts-SportsCodesCatalog_1.xml')),87)
reactant(atom(iptcID_0112,val('Arizona Diamondbacks','team:l.mlb.com-t.26'),
         sem(subject,'http://www.xmlteam.com/specification/xts-SportsCodesCatalog_1.xml')),83)
```

Outline

1. Motivations
2. The Molecules of Knowledge Model
   - Informal MoK
   - Formal MoK
3. A MoK Infrastructure
   - The TuCSoN Coordination Infrastructure
   - Mapping MoK over TuCSoN
4. A Case Study: MoK-`News`
5. Conclusions & Further Works

Final Remarks

Molecules of Knowledge

The MoK model

→ provides knowledge workers in general with a novel approach both in thinking and managing knowledge

→ supports their work with self-organising shared workspaces able to autonomously cluster and spread information

TuCSoN for MoK

The MoK implementation over TuCSoN is a first step toward a full implementation of the MoK model, featuring topology-related aspects, self-aggregation of information and smart diffusion toward interested users.

Open Issues & Further Developments

How to...

- ? push the MoK 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., 2012] to lift Linda purely syntactical one currently exploited in TuCSoN?

Further works

- ! explore the literature to address first issue
- ! improve the current prototypal implementation of the MoK model upon TuCSoN coordination infrastructure
- ! test MoK against other application domains — e.g. research publications, sensor networks, social media, healthcare...

Thanks to. . .

- . . . everybody here for listening
- . . . the SAPERE team for supporting our work [1]

[1] This work has been supported by the EU-FP7-FET Proactive project SAPERE *Self-aware Pervasive Service Ecosystems*, under contract no.256873.

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. (2012).
Semantic tuple centres.
Science of Computer Programming.
Special Issue on 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.

Molecules of Knowledge: 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

IDC 2012
Intelligent Distributed Computing
Calabria, Italy – 25th of September 2012