Molecules of Knowledge: Self-organisation in Knowledge-Intensive Environments

Stefano Mariani and Andrea Omicini

Abstract. We propose a novel self-organising knowledge-oriented model based on *biochemical tuple spaces*, called *Molecules of Knowledge* (MoK). We introduce MoK basic entities, define its computational model, and discuss its mapping on the TuCSoN coordination model for its implementation.

1 Introduction

The issues of knowledge management are becoming critical in most of the complex computational systems of today—in particular, in socio-technical systems. There, people working in knowledge-intensive environments typically face the challenges of a huge and ever-growing collection of knowledge sources, producing possibly relevant information at a fast pace, in many heterogeneous formats – so, essentially unmanageable –, which they are anyway supposed to handle and govern in order to perform their tasks and achieve their own goals. For physicians, journalists, researchers, lawyers, even politicians – *knowledge workers*, in general – today ICT systems provide at the same time new opportunities and new obstacles: the ability to find all the relevant information needed in the short time being a issue that even the most advanced general-purpose research engines are not able to face today.

Adaptive and self-organising systems seems 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. Nature-inspired models provide clear examples of systems where order out of chaos is achieved by means of simple and pervasive mechanisms, based on local interactions. Also, coordination models – in particular, tuple-based ones – have already proven their effectiveness in the engineering of complex software systems, like knowledge-intensive, pervasive and self-organising ones—see [3] for a survey.

Stefano Mariani · Andrea Omicini
DISI, ALMA MATER STUDIORUM – Università di Bologna, Italy
e-mail: {s.mariani,andrea.omicini}@unibo.it

G. Fortino et al. (Eds.): Intelligent Distributed Computing VI, SCI 446, pp. 17–22.
springerlink.com

In this paper we introduce a self-organising knowledge-oriented model called *Molecules of Knowledge*. There, knowledge sources produce *atoms* of knowledge in *biochemical compartments*, which then diffuse and aggregate in *molecules* by means of *biochemical reactions*, acting locally within and between such spaces. Knowledge consumer's workspaces are mapped into such compartments, which reify information-oriented user actions in terms of their state – hence atoms and molecules in the compartments –, influencing atoms aggregation and molecule diffusion—for instance, making potentially relevant knowledge atoms and molecules autonomously move toward the interested users.

2 The *Molecules of Knowledge* Coordination Model

Molecules of Knowledge (MoK) is a biochemically-inspired model promoting self-organisation of knowledge. The main ideas behind MoK are: *(i)* knowledge should autonomously aggregate and diffuse to reach knowledge consumers, *(ii)* the biochemical metaphor fits knowledge self-organisation in distributed knowledge-intensive environments. The main abstractions of MoK are:

atoms the smallest unit of knowledge in MoK, an *atom* contains information from a *source*, and belongs to a *compartment* where it "floats";
molecules the MoK units for knowledge aggregation, *molecules* bond together related atoms;
enzymes emitted by MoK *catalysts*, *enzymes* influence MoK reactions, thus affecting the dynamics of knowledge evolution within MoK compartments;
reactions working at a given *rate*, *reactions* are the biochemical laws regulating the evolution of each MoK compartment, by governing knowledge aggregation, diffusion, and decay within MoK compartments.

Other relevant MoK abstractions are instead in charge of important aspects like topology, knowledge production, and knowledge consumption:

compartments the spatial abstraction of MoK, *compartments* provide MoK with the notions of *locality* and *neighbourhood*;
sources associated with compartments, MoK *sources* originate knowledge, which is injected in the form of MoK atoms at a certain *rate* within the compartment;
catalysts the abstraction for knowledge *prosumers*, *catalysts* emit enzymes which represent prosumer's actions, affecting knowledge dynamics within the prosumer's compartment.

2.1 Atoms

Atoms are the most elementary pieces of knowledge within the model. A MoK atom is produced by a knowledge source, and conveys a piece of information spawning from the source itself. Hence, along with the content they store – the piece of raw information –, atoms should also store some contextual information to refer to the content's origin, and to preserve its original meaning.

A MoK atom is essentially a triple of the form $atom(\ src,\ val,\ attr\ )_c$. There, *src* identifies the source of knowledge; *val* is the actual piece of knowledge carried by the atom—some content, possibly along with an associated ontology; *attr* is essentially the content's attribute, that is, the additional information that helps understanding the content—again, possibly expressed according to some well-defined ontology; c is the current *concentration* of the atom, set at injection time, then evolving according to the system evolution.

2.2 Molecules

A MoK system can be seen as a collection of atoms generated at a certain rate, wandering through compartments, and possibly colliding within the compartment they belong to. The result of collisions are the *molecules of knowledge*, that is, spontaneous, stochastic, "environment-driven" aggregations of atoms, which in principle are meant to reify some semantic relationship between atoms, thus possibly adding new knowledge to the system. Each molecule is simply a set of atoms—an unordered collection of semantically-related atoms.

A MoK molecule has then the simple structure $molecule(\ Atoms\ )_c$. There, c is the current concentration of the molecule, and $Atoms$ is the collection of all the atoms currently bonded together in the molecule—which essentially represent the "pool" of related pieces of knowledge that a chain of reactions has aggregated during the natural system evolution.

In the same way as atoms, molecules are in turn involved in (biochemical) reactions, described below: they can aggregate and generate more complex molecules based on semantical relations between some atoms of them. So, whenever it may help simplifying the view of a MoK systems, one could also see atoms as molecules with a single atom, and viceversa, so that for instance, given an atom a and a molecule $m = molecule(a)$, we can consider $a = m$ whenever useful in the model specification. Along the same line, a MoK system can also be seen as a *collection of molecules of knowledge* wandering through compartments, and reacting so as to make knowledge autonomously aggregate and diffuse from sources to prosumers.

2.3 Enzymes

One of the key features of MoK is that the system interprets prosumer's knowledge-related actions as *positive feedbacks* that increase their concentration of related atoms and molecules within the prosumer's compartment. To this end, the MoK model includes *catalysts* (representing prosumers, typically knowledge workers) injecting *enzymes* whenever they access atoms and molecules in their own *compartment*. Enzymes are then used by MoK reactions to increase involved atoms/molecules' concentration, producing the positive feedback required to enable self-organisation of knowledge.

A MoK enzyme has the structure $enzyme(\ Atoms\ )_c$. There, an enzyme – with its own concentration c – explicitly refers to a collection of $Atoms$ that the catalyst's

actions have in any way pointed out as of interest—either explicitly, by an epistemic action, or implicitly, by a generic knowledge-related action.

2.4 Reactions

The behaviour of a MoK system is actually determined by the last fundamental abstraction of the model, the one actually defining the behaviour of a MoK system: the *(biochemical) reaction*. Biochemical reactions drive atoms and molecules aggregations, as well as atoms and molecules reinforcement, decay, and diffusion.

As a knowledge-oriented model, the main issue of MoK is determining the semantic correlation between MoK atoms. So, in order to define a MoK system, one should first of all define the basic *mok* function, taking two atoms — as in $mok(\ atom_1,\ atom_2\ )$ – and returning true if $atom_1$ and $atom_2$ are semantically related. The precise definition of *mok* depends on the specific application domain: different notions of semantic correlation could be used depending on the sort of the knowledge-intensive environment—such as news, scientific research, health care. On the other hand, it requires the analysis of the possible correlations between the ingredients of the atoms, so that the value of $mok(atom_1, atom_2)$ typically depends on the *val* and $attr$ elements of $atom_1$ and $atom_2$.

The *aggregation reaction* bounds molecules based on semantic correlation:

$$molecule(Atoms_1)\ +\ molecule(Atoms_2) \longmapsto^{r\_agg} molecule(Atoms_1 \bigcup Atoms_2)\ +\ Residual(Atoms_1, Atoms_2)$$

There, *r_agg* is the reaction rate; $mok(atom_1, atom_2)$ holds for some $atom_1 \in Atoms_1$, $atom_2 \in Atoms_2$; and $Residual(Atoms_1, Atoms_2)$ is the multiset of atoms obtained as the difference $(Atoms_1 \uplus Atoms_2) \setminus (Atoms_1 \bigcup Atoms_2)$—in short, all the atoms of $Atoms_1$ and $Atoms_2$ that do not belong to the resulting molecule, thus preserving the total number of atoms. In short, more complex molecules are formed by aggregation whenever some atoms in the reacting molecules (or in reacting atoms, handled here as one atom molecules) are semantically correlated (via the *mok* function).

Positive feedback is obtained by the *reinforcement reaction*, consuming an enzyme injected by a catalyst to produce a single unit of the relevant atom/molecule:

$$enzyme(Atoms_1)\ +\ molecule(Atoms_2)_c \longmapsto^{r\_reinf}\ molecule(Atoms_2)_{c+1}$$

There, *r_reinf* is the reaction rate; $enzyme(Atoms_1)$, $molecule(Atoms_2)_c$ exist in the compartment, with $c \neq 0$; and $mok(atom_1, atom_2)$ holds for some $atom_1 \in Atoms_1$, $atom_2 \in Atoms_2$.

Some MoK biochemical reactions are meant to make knowledge evolve according to the space and time patterns that typically characterise self-organising scenarios. So, in order to provide the required *negative feedback*, molecules should fade as time passes, lowering their own concentration according to some well-defined decay law. The temporal *decay reaction* is hence defined as follows:

$$molecule(Atoms)_c \longmapsto^{r\_decay}\ molecule(Atoms)_{c-1}$$

where one molecule instance disappears, at a rate *r_decay*, hence its concentration is decreased by one.

Analogously, a distributed self-organisation model should provide some kind of spatial evolution pattern. According to its natural inspiration, MoK adopts *diffusion* as its knowledge migration mechanism. Here, diffusion is simply the migration of an atom or molecule from one chemical compartment to another, thus providing the basic mechanism for knowledge sharing and autonomous motion in a distributed system. According to the biochemical metaphor, atoms and molecules can only diffuse between *neighbour* compartments, resembling membrane crossing among cells. Assuming that σ identifies a biochemical compartment, and $\|\|_\sigma$ brackets molecules in a compartment σ, the *diffusion reaction* is modelled as follows:

$$\begin{array}{c} \| \ Molecules_1 \bigcup molecule_1 \ \|_{\sigma^i} + \| \ Molecules_2 \ \|_{\sigma^{ii}} \longmapsto^{r\_diffusion} \\ \| \ Molecules_1 \ \|_{\sigma^i} + \| \ Molecules_2 \bigcup molecule_1 \ \|_{\sigma^{ii}} \end{array}$$

There, σ^i and σ^{ii} are neighbour compartments, *r_diffusion* is the diffusion rate, and $molecule_1$ moves from σ^i to σ^{ii}. Along with reinforcement and decay, diffusion ensures that relevant knowledge is brought toward interested prosumers: roughly speaking, diffusion scatters knowledge around, reinforcement increases the concentration of relevant knowledge, and decay abates non-relevant one.

3 Mapping MoK over TuCSoN

TuCSoN [4] is a coordination model & infrastructure exploiting programmable tuple spaces, called *tuple centres*. The behaviour of a tuple centre can be defined through the ReSpecT logic language [2] to encapsulate any coordination law into the coordination abstraction—so, in principle, biochemical reactions, too.

Since logic-based ReSpecT tuple centres are well-suited to handle knowledge representation in general, our first implementation of MoK is built upon TuCSoN. In particular, TuCSoN easily provides the two basic ingredients for biochemical coordination: *(i)* chemical-inspired stochastic evolution of tuples – achieved by implementing the well-known Gillespie's exact simulation algorithm [1] as a ReSpecT specification –, and *(ii)* matching of biochemical laws against tuples in the space—obtained by interpreting tuples as individuals in the biochemical system, and using templates to denote reactants in biochemical reactions.

According to the MoK biochemical metaphor discussed in Section 2, the first natural mapping of MoK concepts over TuCSoN abstractions is the following:

Individuals → *Atoms / Molecules / Enzymes* → *(Logic) Tuples*

Viewing computational systems as eco-systems, individuals are the inhabitants of the environment, ruled by its laws, and at the same pro-actively affecting the environment itself. MoK individuals are all the knowledge chunks floating in a biochemical compartment—the subjects and enablers of its biochemical reactions. In TuCSoN all the abovementioned abstractions translate into (logic) tuples.

The representation of the environmental substrate in terms of TuCSoN tuple centres is plain: the environment stores both the individuals and the laws of nature in the same way as tuple centres store tuples and coordination laws:

Environmental Substrate → *Compartments / Catalysts* → TuCSoN *Tuple Centres*

Since catalysts are users, and each user has a compartment as its working place, both catalysts and compartments are in principle mapped upon tuple centres.

Finally, in a biochemical compartment, the role of the laws of nature is played by biochemical reactions, built as ReSpecT reactions:

Laws of Nature → *Biochemical Reactions* → ReSpecT *Reactions*

Each TuCSoN tuple centre representing a MoK compartment is programmed to work as a *biochemical virtual machine* implementing the Gillespie's chemical simulation algorithm, enacting MoK biochemical reactions as ReSpecT reactions.

Based on the above mapping, the prototype implementation of MoK has been already developed, then tested on a first set of experiments—not reported here, however, for the lack of space.

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

References

1. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry 81(25), 2340–2361 (1977), doi:10.1021/j100540a008
2. Omicini, A., Denti, E.: From tuple spaces to tuple centres. Science of Computer Programming 41(3), 277–294 (2001), doi:10.1016/S0167-6423(01)00011-9
3. Omicini, A., Viroli, M.: Coordination models and languages: From parallel computing to self-organisation. The Knowledge Engineering Review 26(1), 53–59 (2011), doi:10.1017/S026988891000041X
4. Omicini, A., Zambonelli, F.: Coordination for Internet application development. Autonomous Agents and Multi-Agent Systems 2(3), 251–269 (1999), doi:10.1023/A:1010060322135