Parameter Engineering vs. Parameter Tuning: the Case of Biochemical Coordination in $\mathcal{MoK}$

Stefano Mariani

s.mariani@unibo.it

DISI

Alma Mater Studiorum—Università di Bologna

WOA 2013

Torino, Italy - 2-3 December 2013

Outline

Nature-inspired Engineering I

Coordination models and languages have historically drawn inspiration from *self-organising coordination* in natural phenomena to cope with nowadays MAS increasing complexity [Omicini and Contucci, 2013].

Nature-inspired Coordination

Among the many, e.g. pheromone-based [Grassé, 1959] and chemical [Banâtre et al., 2001] coordination, biochemical coordination has been shown to be particularly effective [Viroli and Casadei, 2009], [Zambonelli et al., 2011].

Being a self-organising process, biochemical coordination effectiveness likely depends on a correct parameter tuning stage, where parameters are usually *inherited* from the metaphor adopted.

Nature-inspired Engineering II

"Imitation" is not "Engineering"

- A given natural system may rely on a given set of parameters, each of which with a given set of functional dependencies with others
  - ! *this doesn't mean the same set of parameters and functional dependencies (nature "as it is") will work for an artificial system!*

→ to identify and engineer the relevant parameters along with their functional dependencies, a *disciplined approach* is needed

Prior to parameter tuning we need to ask ourselves: *are the natural system's own parameters (as well as their dependencies) good also for the artificial MAS?*

Nature-inspired Engineering III

In the case of biochemical coordination, such parameters are, e.g., the rate of application of a chemical reaction, the concentration of the participating chemicals and their stoichiometry—the "extent" to which they participate.

Parameter Engineering

Accordingly, we show that:

- the law of mass action (chemistry "as it is") may be not enough to effectively engineer a biochemical coordination middleware
- designing *arbitrary* rate expressions demands for a disciplined and principled approach, that we call parameter engineering

We do so by taking the $\mathcal{M}$olecules of $\mathcal{K}$nowledge model and the BioPEPA tool as our *subject* and *means* of investigation, respectively.

Outline

Outline

The Biochemical Metaphor I

- The key idea is to coordinate any MAS entity (agents as well as information) as "molecules" floating in a *network of chemical compartments*
- Each compartment resembles a chemical "solution", whose evolution is driven by chemical "reactions" consuming and producing molecules—possibly from/to neighbouring compartments
- As many reactions can occur concurrently, system evolution is driven by *race conditions* among reactions rates, thus certain reactions are *stochastically* selected over others—according to the law of mass action, as in chemistry actually is [Gillespie, 1977]

The Biochemical Metaphor II

The law of mass action[1] is a mathematical model that *explains* and *predicts* the behaviour of solutions in dynamic equilibrium.

Chemistry "as it is"

The law states that the rate of a reaction (r_f) is *proportional* (k_f) *to the product of the concentrations* of the participating molecules (R^1, R^2):

$$r_f = k_f[R^1][R^2]$$

k_f is called *rate constant* and, in chemistry, is a function of participating molecules affinity.

[1] From `http://en.wikipedia.org/wiki/Law_of_mass_action`.

The $\mathcal{MoK}$ Model

$\mathcal{M}$olecules *of* $\mathcal{K}$nowledge [Mariani and Omicini, 2013] ($\mathcal{MoK}$ for short) is a model for *knowledge self-organisation in MAS* whose main goals are:

- to let information chunks *autonomously aggregate* into heaps of knowledge
- to let knowledge *autonomously flow* toward the interested agents—rather than be searched

To do so, $\mathcal{MoK}$ exploits the biochemical metaphor:

- information is represented as molecules[2]
- agents' actions transparently inject enzymes in the system to influence $\mathcal{MoK}$ reactions behaviour:

Injection	generating molecules from "sources" of information
Reinforcement	increasing molecules "relevance" according to agents interactions
Decay	decreasing molecules relevance as time passes by
Diffusion	moving molecules where they are "needed"

[2] The distinction between molecules and *atoms* in $\mathcal{MoK}$ is not considered in this presentation. For details, check out the full paper.

Testing our Goals I

- Is it sufficient to stick with the law of mass action to achieve $\mathcal{MoK}$ goals, or should we build our "custom" functional dependencies between parameters (rate expressions)?
- If so, which parameters and which kind of dependencies (direct, inverse, etc.) are to be used in each rate expression?
- And how can MAS designers make such decisions?

Outline

Complexity Calls for Simulation

The Role of Simulation

Simulation has been widely recognized as a fundamental development stage in the process of designing and implementing both *MAS* as well as *biochemical processes* [Merelli et al., 2007], [Gardelli et al., 2006]

- This is mostly due to the high number of system parameters needed as well as of local interactions between components, but also to the influence of randomness and probability on system evolution
- We use simulations to investigate $\mathcal{MoK}$ reactions behaviour when engineered according to different rate expressions

BioPEPA

BioPEPA [Ciocchetta and Hillston, 2009] is a language (behind a simulation tool) to model biochemical processes whose main features are:

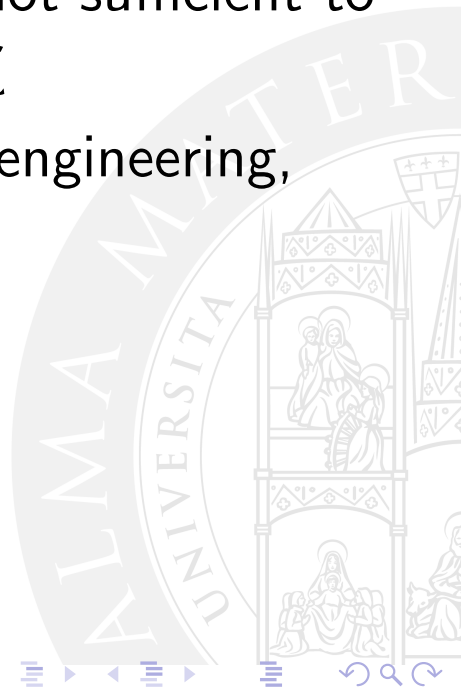
- custom kinetic laws (rate expressions) represented as *functional rates*
- definition of *stoichiometry* and role played by the species (reactant, product, enzyme, etc.) in a given reaction
- theoretical roots in *CTMC* semantics

In BioPEPA, rate expressions are mathematical equations involving:

- reactants' *concentrations*—denoted with the reactant name and dynamically computed at run-time
- mathematical operators
- built-in kinetic laws—e.g. the *law of mass action* (keyword `fMA`)
- *time dependency*—through the variable `time`, changing value dynamically according to the simulation time step

Testing our Goals II

- We use BioPEPA to simulate $\mathcal{MoK}$ reactions effect on system evolution using different rate expressions
- In particular, we show that the law of mass action is not sufficient to achieve the self-organising behaviours desired in $\mathcal{MoK}$
- Also, such simulations are used to support parameter engineering, thus engineering of custom rate expressions



Outline

Beyond Imitation

As regards nature-inspired MAS, simulation tools are usually exploited to fine-tune *those parameters inherited from the natural metaphor*.

The "Overlooked" Question

But, what about the question of *whether the natural system's parameters are well suited also for the artificial MAS?*

For each of the following experiments, we

1. identify which are the desiderata for the MAS run-time behaviour
2. *engineer* rate expressions by designing functional dependencies which are likely to pursue the chosen goal
3. include a pure parameter tuning stage (if needed)

We do so one rate expression at a time, incrementally. This approach is what we call parameter engineering.

Outline

Injection Rate I

- Molecules should be *perpetually* injected into the MAS, since there is no way to know a-priori *when* some information will be needed
- But, we would likely avoid *flooding* the system without any control on how many molecules are in play
- Thus, three options are viable:
  1. make injection rate decrease as time passes
  2. enforce some kind of "saturation" to stop injection
  3. a combination of the two
  4. stick with the law of mass action—for comparison

Injection Rate II

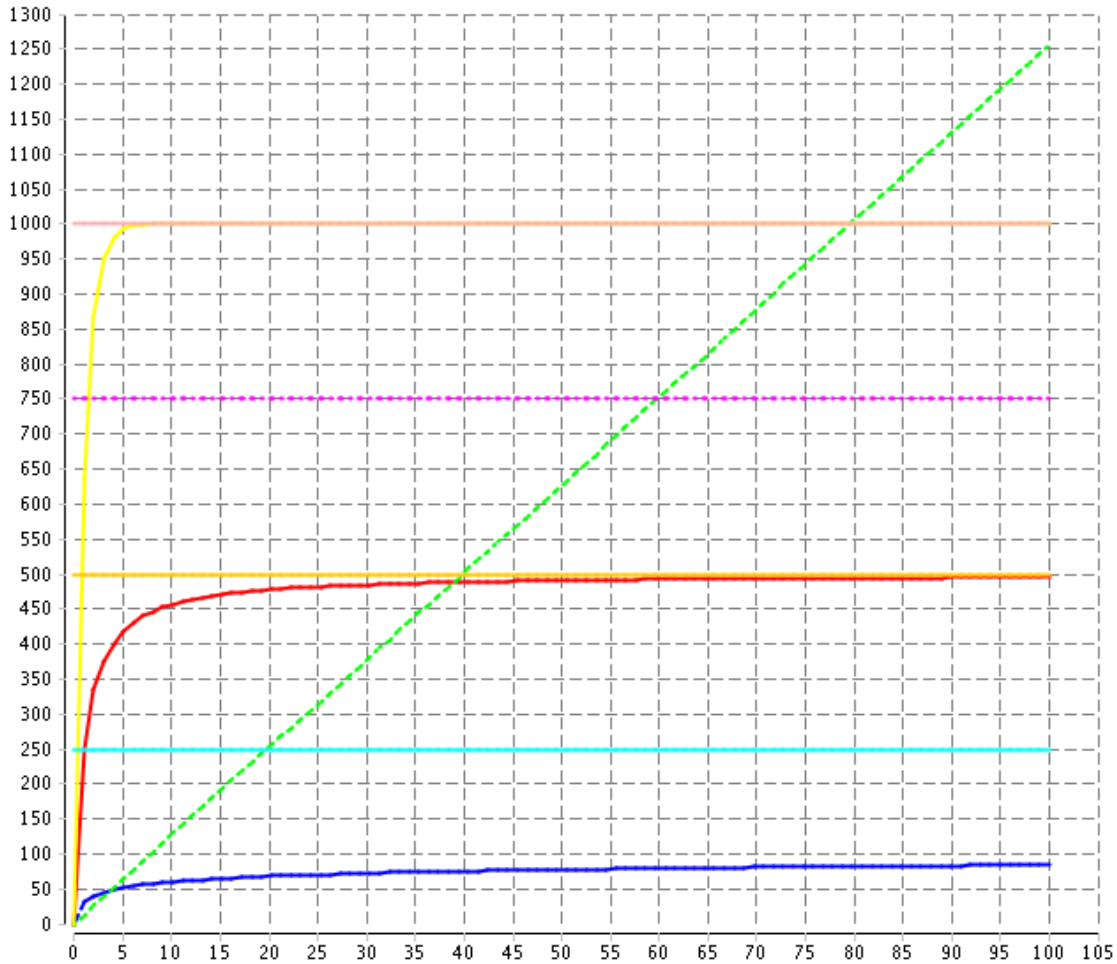


Figure : Comparison of functional rates for molecules injection. Option (1) is in blue, option (2) in yellow, option (3) in red and option (4) in green (dashed).

Injection Rate III

- Clearly, relying on the law of mass action is out of question: its behaviour follows none of $\mathcal{MoK}$ injection reaction desiderata
- Once discarded also option (1), whose trend is clearly too slow in reaching saturation, options (2) and (3) may seem almost identical, but they are not:
  - option (2) is *"saturation-driven" only*, thus, no matter how long sources are within the system, molecules will strive to reach saturation-level as fast as possible
  - option (3) instead, makes the saturation process *time-dependant*, hence, the longer sources are within the system, the slower saturation will be

  Choosing among the two obviously depends on the application-specific context in which the $\mathcal{MoK}$ model is used.

Decay Rate I

- $\mathcal{MoK}$ decay reaction is an effective way to resemble the relationship between information relevance and time flow
- Furthermore, decay enforces the *negative feedback* which, together with the *positive feedback* provided by $\mathcal{MoK}$ enzymes, enables the *feedback loop* peculiar of natural systems
- Four options are worth to be considered:
  1. linear time dependency + $\frac{\sharp sources}{\sharp molecules}$ dependency
  2. logarithmic time dependency + $\frac{\sharp sources}{\sharp molecules}$ dependency
  3. linear time dependency + law of mass action
  4. law of mass action alone—for comparison

Decay Rate II

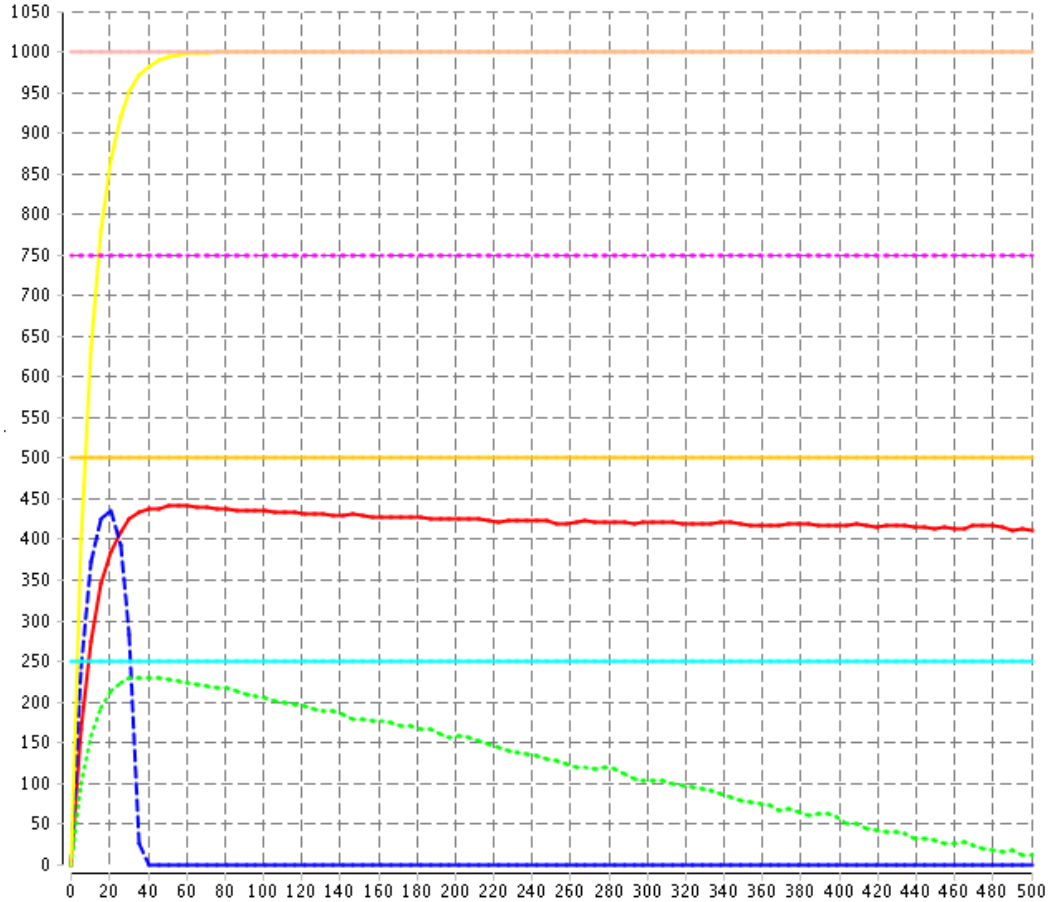


Figure : Comparison of functional rates for molecules decay. Option (1) is in blue (dashed), option (2) in red, option (3) in green (dashed) and option (4) in yellow.

Decay Rate III

- Again, the law of mass action is unsatisfactory, as well as option (1)
- Options (2) and (3) are both viable solutions instead[3]. The choice is mostly driven by *how fast* information should lose relevance in the scenario in which $\mathcal{MoK}$ has to be deployed

[3] Nevertheless, notice option (3) has an additional parameter w.r.t. option (2): the law of mass action "rate constant". Having less parameters is usually favourable.

Reinforcement Rate I

- Enzymes are meant to represent a situated interest manifested by an agent w.r.t. a piece of knowledge—exploited to reinforce such knowledge "relevance" within the MAS
- Thus, $\mathcal{MoK}$ reinforcement reaction should:
  - *be prompt*, that is rapidly increase molecules concentration—despite decay
  - *limited* both in time and space, to resemble relevance relationship with situatedness of (inter-)actions
- Three options have been tried:
  1. $\frac{\sharp molecules}{\sharp sources}$ dependency
  2. law of mass action with dynamic rate constant ($= \frac{\sharp molecules}{\sharp sources}$)
  3. law of mass action with static rate constant

Reinforcement Rate II

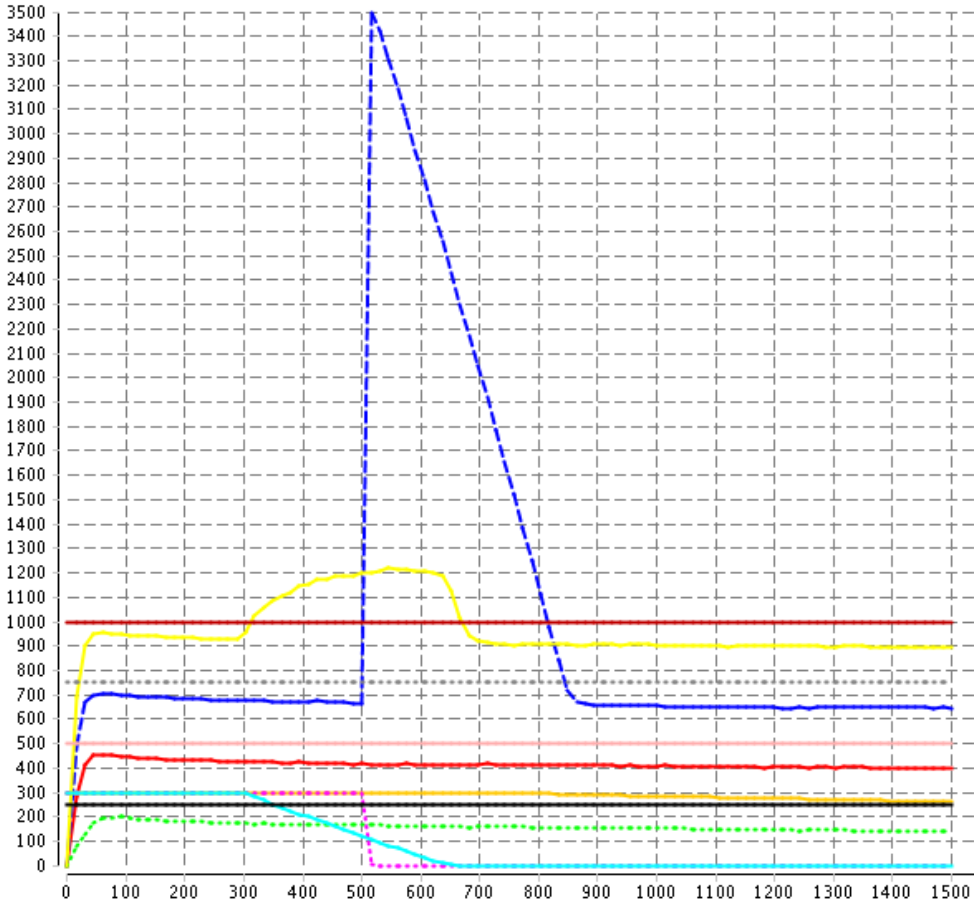


Figure : Comparison of functional rates for molecules reinforcement. Lines worth to be considered are: the yellow one, plotting option (1), the dashed blue one, plotting option (2), the red one, plotting option (3).

Reinforcement Rate III

- Once again, sticking with the law of mass action alone is out of question:
  - option (2) exhibits an exceedingly high and "fast-to-decay" peak
  - option (3) almost completely ignores the feedback—enzymes are too slowly consumed (orange line, plotting enzymes concentration)
- Option (1) instead, suits well $\mathcal{MOK}$ reinforcement reaction desiderata

Diffusion Rate I

- As the reference topology, four $\mathcal{MoK}$ "compartments" are imagined to be connected one to each other, allowing in principle any molecule to move anywhere
- Similarly to $\mathcal{MoK}$ injection reaction, on one hand we would like to *perpetually spread information* around, because agents working in other compartments may be interested in it; on the other hand we would also like to keep *some degree of control* on "how much" information is moved around
- Hence, we can reuse the concept of "saturation". In particular, it seems reasonable to allow only a fraction of molecules to move away from their "origin" compartment

Diffusion Rate II

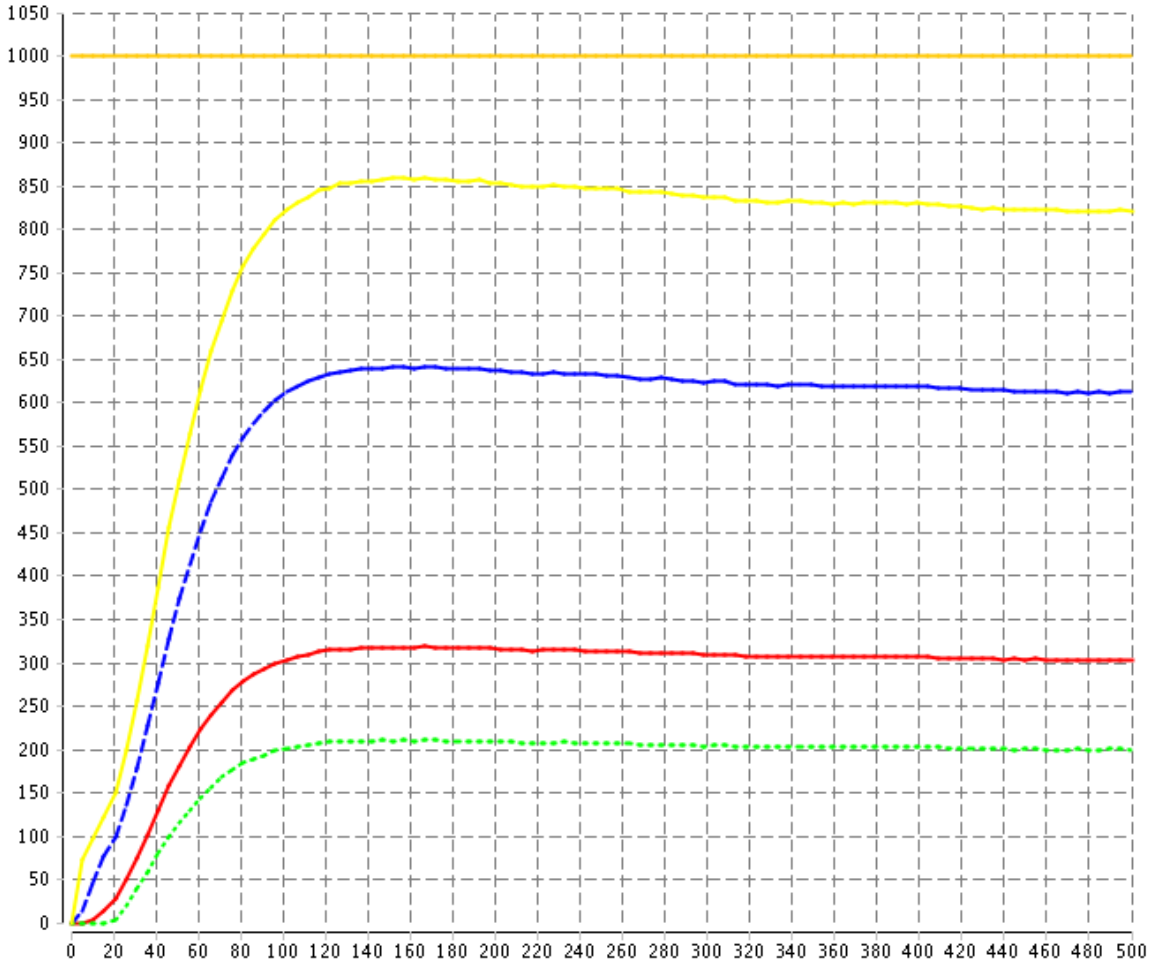
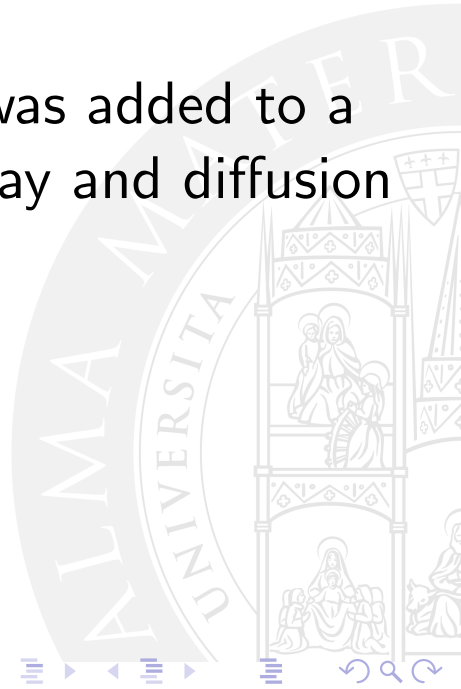


Figure : $\mathcal{MoK}$ diffusion reaction trend. The yellow line plots the concentration level of the molecules in their "origin" compartment (the orange horizontal line represents their sources).

Outline

Unexpected Interactions I

- As already said, all the experiments were conducted *incrementally*, that is, each $\mathcal{MoK}$ reaction was added to the BioPEPA specification one at a time
- Figure 5 depicts what happened when reinforcement was added to a BioPEPA specification already including injection, decay and diffusion



Unexpected Interactions II

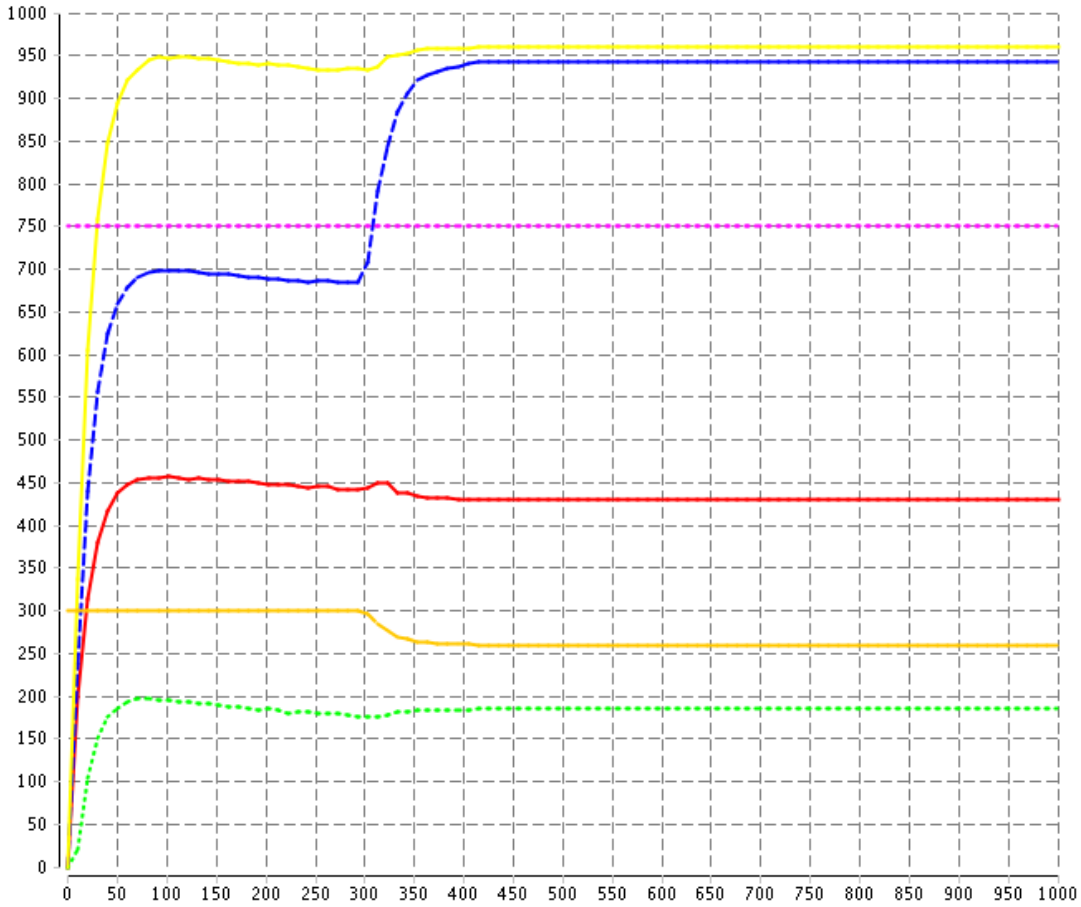


Figure : $\mathcal{MoK}$ reinforcement reaction addition to injection, decay and diffusion. Not only enzymes are not fully depleted, but also undesirable and unexpected interferences with other reactions are clearly highlighted.

Unexpected Interactions III

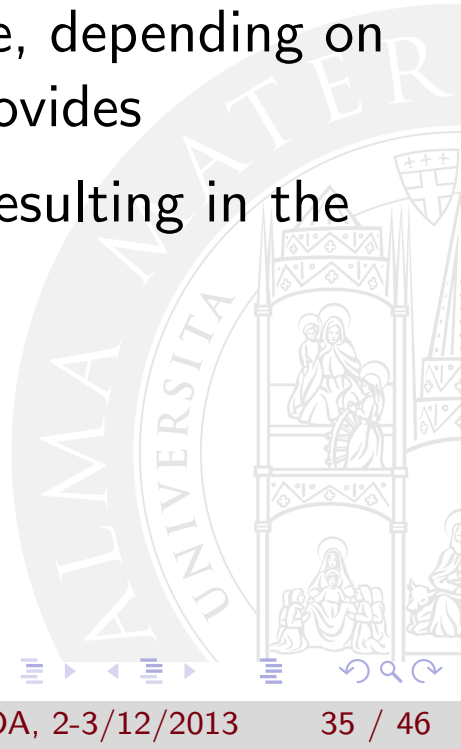
- First of all, our desiderata for $\mathcal{MoK}$ reinforcement reaction are not met (dashed blue line): in particular, it seems *molecules cannot go beyond their original compartment concentration level* (yellow line)
- Second, *enzymes are not fully depleted* (orange line)

Reactions Interference

Last but not least, other molecules are affected by a successful application of $\mathcal{MoK}$ reinforcement reaction (yellow, red and green lines): in particular, in the time interval during which enzymes are consumed *all other trends experiment some fluctuations*

Unexpected Interactions IV

- The reason at the root of all these issues is still unknown: being chemical-like reactions scheduling based on race conditions between the correspondent functional rates – evaluated at a given point in time –, understanding what exactly happens within the system at a given time step is not trivial at all—or even impossible, depending on the debugging services the simulation tool adopted provides
- Nevertheless, the satisfactory BioPEPA specification resulting in the plots shown in Figure 6 has been found



Unexpected Interactions V

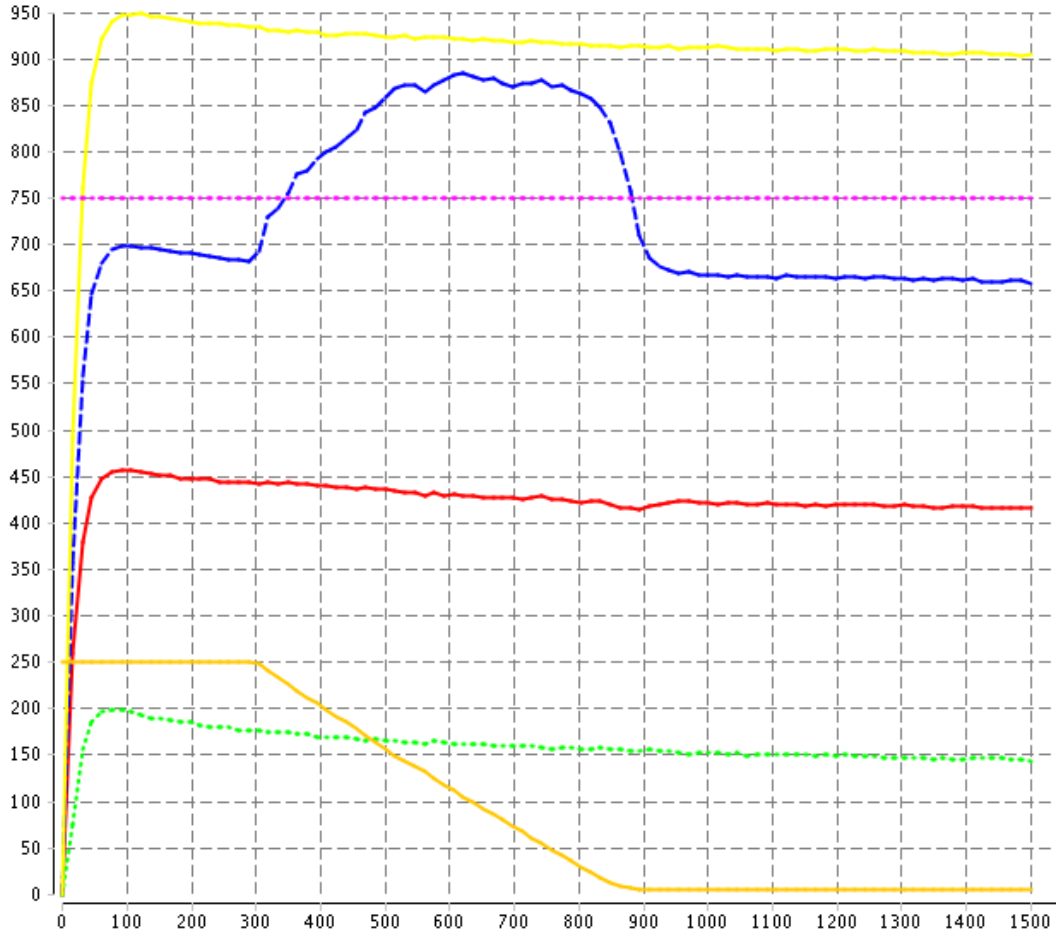


Figure : Adjusted $\mathcal{MOK}$ reinforcement reaction: enzymes are now completely depleted and other reactions no longer affected.

Unexpected Interactions VI

The Need for Engineering

This clearly demonstrates the intricacies behind rates design in biochemical coordination, therefore motivating the principled and disciplined – namely, engineered – approach to parameter tuning we call parameter engineering.

Outline

Engineering Natural Metaphors I

Imitation is not Enough

Simply imitating nature "as it is" may not be the optimal approach while engineering nature-inspired MAS.

1. Once a suitable natural metaphor has been found, MAS designers should ask themselves if the natural system's parameters are the "optimal" ones also for the artificial system they aim to build
2. If not, they should
   1. state which goals their MAS is pursuing
   2. detect which parameters are more likely to help achieving them
   3. design those functional dependencies between such parameters that better drive the MAS toward its goals

Engineering Natural Metaphors II

The Case of Biochemical Coordination

In particular, by focussing on the $\mathcal{M}$olecules of $\mathcal{K}$nowledge model, we showed that

- sticking with the law of mass action for rate expressions is not enough to model $\mathcal{M}o\mathcal{K}$ desired self-organising behaviours
- designing custom rate expressions demands for a principled approach which goes beyond parameter tuning, which we call parameter engineering, likely to be supported by incremental *simulation* of each single basic "law of nature" in play

Engineering Natural Metaphors III

- Our work can be seen as to be *complementary* to that in [Fernandez-Marquez et al., 2012] about self-organising design patterns as well as to [Gardelli et al., 2006] about simulation's role in engineering
- In fact, once a design pattern has been recognized as a potential solution to a given problem, a simulation stage is out of doubts useful to fine-tune the parameters in play; therefore, a parameter engineering phase turns out to be necessary

Roadmap Toward a Methodology

We hope an integration of our work with those cited above could lead to a novel, more conscious and effective approach to the engineering of *nature-inspired* self-organising MAS.

References I

Banâtre, J.-P., Fradet, P., and Le Métayer, D. (2001).
Gamma and the chemical reaction model: Fifteen years after.
In Calude, C. S., Păun, G., Rozenberg, G., and Salomaa, A., editors, *Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View*, volume 2235 of *LNCS*, pages 17–44. Springer.

Ciocchetta, F. and Hillston, J. (2009).
Bio-PEPA: A framework for the modelling and analysis of biological systems.
Theoretical Computer Science, 410(33–34):3065 – 3084.
Concurrent Systems Biology: To Nadia Busi (1968–2007).

Fernandez-Marquez, J., Marzo Serugendo, G., Montagna, S., Viroli, M., and Arcos, J. (2012).
Description and composition of bio-inspired design patterns: a complete overview.
Natural Computing, pages 1–25.

References II

Gardelli, L., Viroli, M., and Omicini, A. (2006).
On the role of simulations in engineering self-organising MAS: The case of an intrusion detection system in TuCSoN.
In Brueckner, S. A., Di Marzo Serugendo, G., Hales, D., and Zambonelli, F., editors, *Engineering Self-Organising Systems*, volume 3910 of *LNAI*, pages 153–168. Springer. 3rd International Workshop (ESOA 2005), Utrecht, The Netherlands, 26 July 2005. Revised Selected Papers.

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

Grassé, P.-P. (1959).
La reconstruction du nid et les coordinations interindividuelles chez Bellicositermes natalensis et Cubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs.
Insectes Sociaux, 6(1):41–80.

References III

Mariani, S. and Omicini, A. (2013).
Molecules of Knowledge: Self-organisation in knowledge-intensive environments.
In Fortino, G., Bădică, C., Malgeri, M., and Unland, R., editors, *Intelligent Distributed Computing VI*, volume 446 of *Studies in Computational Intelligence*, pages 17–22. Springer.
6th International Symposium on Intelligent Distributed Computing (IDC 2012), Calabria, Italy, 24-26 September 2012. Proceedings.

Merelli, E., Armano, G., Cannata, N., Corradini, F., d'Inverno, M., Doms, A., Lord, P., Martin, A., Milanesi, L., Möller, S., Schroeder, M., and Luck, M. (2007).
Agents in bioinformatics, computational and systems biology.
Briefings in Bioinformatics, 8(1):45–59.

Omicini, A. and Contucci, P. (2013).
Complexity & interaction: Blurring borders between physical, computational, and social systems. Preliminary notes.
In Bădică, C., Nguyen, N. T., and Brezovan, M., editors, *Computational Collective Intelligence. Technologies and Applications*, volume 8083 of *LNCS*, pages 1–10. Springer Berlin Heidelberg.
5th International Conference (ICCCI 2013). Craiova, Romania, 11–13 September 2013, Proceedings. Invited Paper.

References IV


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.
11th International Conference (COORDINATION 2009), Lisbon, Portugal, June 2009. Proceedings.


Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Di Marzo, G., Risoldi, M., Tchao, A.-E., Dobson, S., Stevenson, G., Ye, Y., Nardini, E., Omicini, A., Montagna, S., Viroli, M., Ferscha, A., Maschek, S., and Wally, B. (2011).
Self-aware pervasive service ecosystems.
Procedia Computer Science, 7:197–199.
Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11).

Parameter Engineering vs. Parameter Tuning: the Case of Biochemical Coordination in $\mathcal{MoK}$

Stefano Mariani

s.mariani@unibo.it

DISI

Alma Mater Studiorum—Università di Bologna

WOA 2013

Torino, Italy - 2-3 December 2013