Coordination Issues in Complex Socio-technical Systems: Self-organisation of Knowledge in $\mathcal{M}o\mathcal{K}$

Candidate: Stefano Mariani
Supervisor: Prof. Andrea Omicini

DISI
Università di Bologna

PhD Presentation
Bologna, Italia
February 24th, 2016

Outline

Outline

Context, Motivation, and Goal I

Why "Coordination Issues"?

- Complex ICT systems gone beyond Turing Machine like computation, as Turing himself pointed out [Tur39], and Wegner later highlighted [Weg97] as *computation = algorithm + interaction*
- Question "how to manage interactions?" answered by research in coordination models and languages, meant to manage dependencies arising from interactions [MC94]
- *Open*, *distributed*, highly *dynamic*, and mostly *unpredictable* systems, such as self-organising ones [OV11], present novel challenges demanding for novel coordination approaches

We approach coordination issues exhibited by self-organising and pervasive systems by leveraging chemical-inspired coordination and situatedness

Context, Motivation, and Goal II

Why "Complex Socio-technical Systems"?

- Socio-technical systems and knowledge-intensive environments are systems combining business processes, technologies and people's skills to store, handle, and make accessible large repositories of information
- Managing their interaction space is of paramount importance for both functional and non-functional properties
- Engineering coordination mechanisms and policies is far from trivial, mostly due to peculiar characteristics such as unpredictability of agents' behaviour, and pace of interactions

We combine a cognitive theory of interaction, behavioural implicit communication, with the aforementioned chemical-inspired coordination model, to promote behaviour-driven coordination

Context, Motivation, and Goal III

Why "Self-organisation of Knowledge in *MoK*"?

- The data-driven approach to coordination, such as tuple space based models, aims at coordinating interacting agents by properly managing access to information
- One possible research line departs from question: why do we stick to view data as passive, "dead" things to run algorithms upon in the traditional I/O paradigm?

Molecules of Knowledge tries to answer with a coordination model for self-organising knowledge management in which data is alive

Walkthrough

The main contribution of the thesis is thus the conception and development of the $\mathcal{M}$olecules $\mathcal{o}$f $\mathcal{K}$nowledge model for *self-organisation of knowledge* in knowledge-intensive socio-technical systems

To get there, three ingredients are needed, corresponding to three other major contributions — complementing each other in view of $\mathcal{MoK}$:

- a chemical-inspired coordination model, to leverage *self-organisation* in our target scenario
- a language and infrastructure for situated coordination, to embrace *situatedness* of our target scenario
- an interaction model, to tame the *social component* included in our target scenario

Outline

Outline

From LINDA to Biochemical Tuple Spaces I

- The archetypal LINDA model [Gel85] gave us a simple yet expressive model for coordinating *distributed* processes in *open* systems [Cia96] in a *fully uncoupled* way w.r.t. space, time, and reference, even with *partial information*
- Socio-technical systems enjoy all these features, but have others which cannot be dealt with effectively by LINDA as it is
  - uncertainty
  - unpredictability
  - adaptiveness

From LINDA to Biochemical Tuple Spaces II

- A natural solution to account for *unpredictability* and *uncertainty* is to embrace stochasticity, by tolerating *probabilistic* rather than deterministic computations and decision making
  - digital pheromones [PBS02], probabilistic pi-calculus [HP01], stochastic KLAIM [DNLKM06], SwarmLinda [TM04]
- A natural solution to account for *adaptiveness* is to support programmability of the coordination machinery, by allowing interacting agents, or the coordination medium itself, to *change the coordination laws*
  - TOTA [MZ09], MARS [CLZ00], LGI [MU00], ReSpecT [Omi07]

From LINDA to Biochemical Tuple Spaces III

- It appears natural then, to combine the afore mentioned approaches into a *single, comprehensive and coherent approach* for self-organising coordination under uncertainty
  - biochemical tuple spaces [VC09] and SAPERE [ZCF+11]

We build exactly upon the ground layered by the aforementioned approaches

- theoretically, cherry-picking those features most appealing for our target scenario
- practically, using ReSpecT to prototype a biochemical tuple space like coordination medium

Outline

Motivation

- A foremost issue while engineering *self-organising systems* is the local-vs-global issue
    - how to link the *local* mechanisms, through which the components of the system interact and coordinate, to the *emergent, global* behaviour, exhibited by the system as a whole [BB06]
- A possible approach to alleviate the issue, is to rely on bio-inspired design patterns [FMMSM+12]
    - interpreted as *coordination laws* and implemented as *artificial chemical reactions*, then properly tuned so as to obtain the desired emergent coordination

We engineer different bio-inspired patterns as *artificial chemical reactions*, then experiment different custom kinetic rate expressions, so as to showcase viability and flexibility of the approach

Patterns Overview I

- Survey of the literature regarding such design patterns and others basic mechanisms for self-organisation
  - [Nag04], [DWH07], [FMSM12], [FMDMSA11], [TRDMS11] and [VCMZ11]
- Set of *atomic patterns* identified:

decay (aka evaporation, cleaning) destroys information as time flows

feed (aka reinforcement) increases information relevance — usually, its quantity

activation/inhibition changes information status or attributes depending on external stimuli — the same holds for the dual inhibition pattern

aggregation fuses information together — e.g. filtering, merging, composing, transforming, etc.

Patterns Overview II

diffusion (aka spreading, propagation) moves information within a topology, either migrating or replicating it

repulsion/attraction drifts apart / approaches information, usually fairly

- Each modelled as an *artificial chemical reaction*
- Temporal evolution of the "information population"[1] involved in the coordination process has been simulated, according to different custom kinetic rates, so as to investigate the different *emergent behaviours* achievable
  - the simulation tool exploited is BioPEPA[2] [CH09]

[1] The term information is used in reference to $\mathcal{MoK}$, but can be replaced with process, component, namely anything which can be subject of self-organisation.

[2] Eclipse plugin at `http://groups.inf.ed.ac.uk/pepa/update/` update site.

Patterns Overview III

In the following slides, we report on a few of them, while others are omitted for the lack of space, but are available within the thesis, and in [Mar13a][a] and [Mar14a][b]

[a] Stefano Mariani. Parameter engineering vs. parameter tuning: the case of biochemical coordination in MoK.
In *From Objects to Agents*, volume 1099 of *CEUR Workshop Proceedings*, 2013

[b] Stefano Mariani. On the "local-to-global" issue in self-organisation: Chemical reactions with custom kinetic rates.
In *Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014*, 2014.
Best student paper award

Decay I

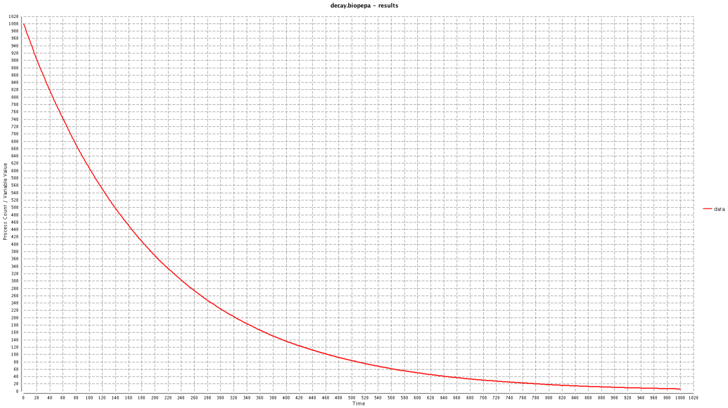


Figure : Decay chemical reaction with usual law of mass action rate[3].

[3] Mathematical model explaining the behaviour of solutions in dynamic equilibrium [Car08].

Decay II

Figure : Decay chemical reaction with custom kinetic rate.

Feed I

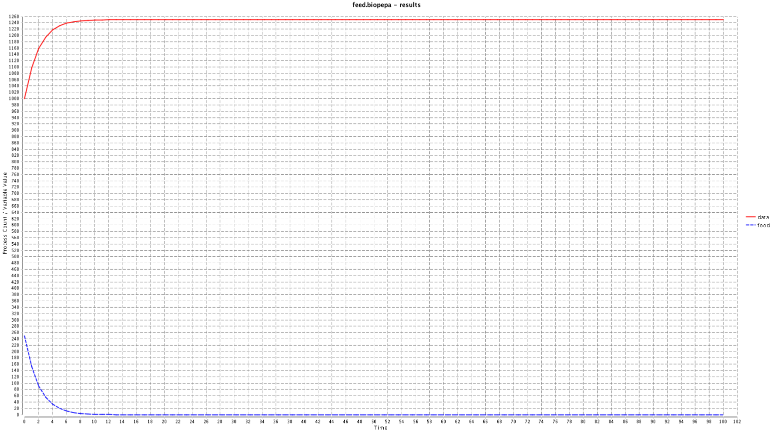


Figure : Feed chemical reaction with usual law of mass action rate.

Feed II

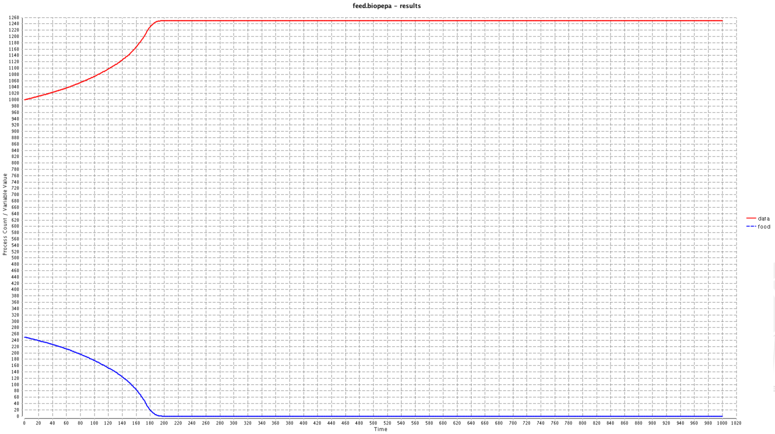


Figure : Feed chemical reaction with custom kinetic rate.

Activation/Inhibition I

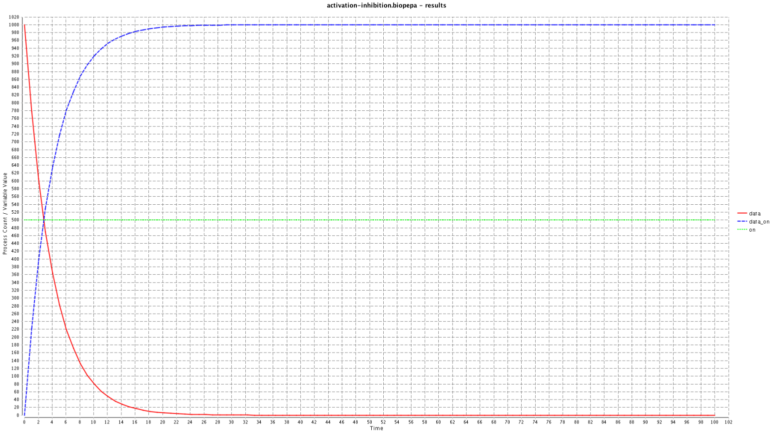


Figure : Activation chemical reaction with usual law of mass action rate.

Activation/Inhibition II

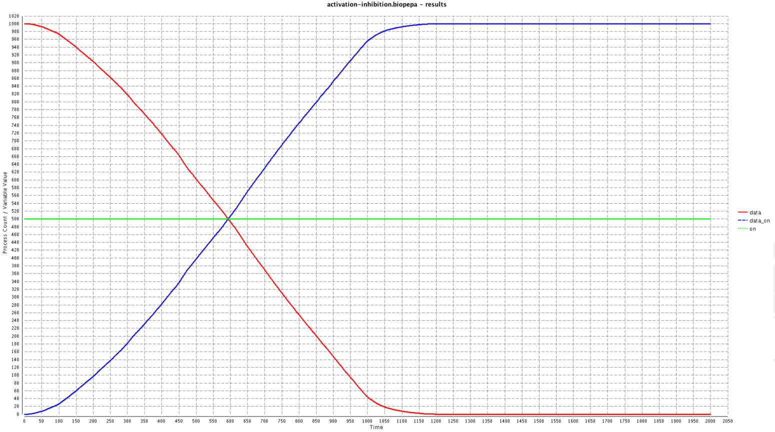


Figure : Activation chemical reaction with activated-dependant kinetic rate.

Activation/Inhibition III

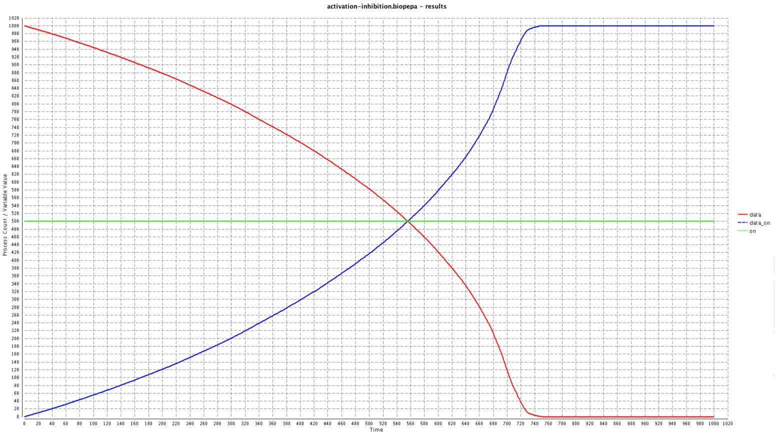


Figure : Activation chemical reaction with latent-dependant kinetic rate.

Implications

- When adopting the *chemical metaphor* for self-organising coordination, imitating nature *as is* may be not the optimal approach
- Engineering custom kinetic rates allows for great flexibility in adapting the emergent population evolution to different coordination needs
- Factors included in custom kinetic rate expressions have a well-defined, *controllable* effect on the global behaviour achieved, which helps alleviating the *local-vs-global* issue

Outline

Motivation

LINDA and many of its derivatives feature don't know *non-determinism* handled with a don't care approach: which tuple among the matching ones is retrieved can be *neither specified nor predicted* (don't know); nonetheless, the system is designed so as to keep on working *whichever is the tuple returned* (don't care)

- which specific tuple is actually retrieved should not be relevant for a process
- this is not the case in many modern systems where processes implement *stochastic behaviours* [OV11]

We discuss uniform coordination primitives[a] (e.g., `uin`, `urd`) as the *specialisation* of LINDA getter primitives featuring *probabilistic non-determinism*

[a] First mentioned in [GVCO07].

Definition I

- If a LINDA getter primitive requires a tuple with template T, and m tuples $t_1, \ldots, t_m$ matching T are in the tuple space when the request is served, any tuple $t_{i \in 1 \ldots m}$ could be retrieved: no other assertion is possible about the result of the getter operation
- If a uniform getter primitive requires a tuple with a template T, and m tuples $t_1, \ldots, t_m$ matching T are available in the tuple space when the request is served, one assertion is possible about the result: each of the m matching tuples $t_1, \ldots, t_m$ has exactly the *same probability* $\frac{1}{m}$ to be returned

Definition II

- Whereas LINDA getter primitives exhibit point-wise properties, uniform primitives feature global properties

  space LINDA returns a tuple *independently* of others, uniform primitives return tuples based on their *relative multiplicity*

  time sequences of LINDA operations feature no meaningful properties, sequences of uniform operations exhibit a *uniform distribution*

The formal definition is omitted for the lack of space, but available within the thesis and in [MO13c][a]

[a] Stefano Mariani and Andrea Omicini. Probabilistic embedding: Experiments with tuple-based probabilistic languages.
In *28th ACM Symposium on Applied Computing (SAC 2013)*, 2013

Expressiveness I

Transparent Load Balancing I

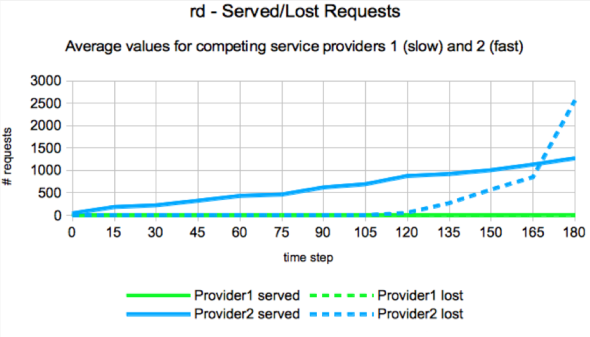


Figure : LINDA environment, no transparent load balancing: only one service is used, due to clients `rd` matching *always* the same advertised service.

Expressiveness II

Transparent Load Balancing II

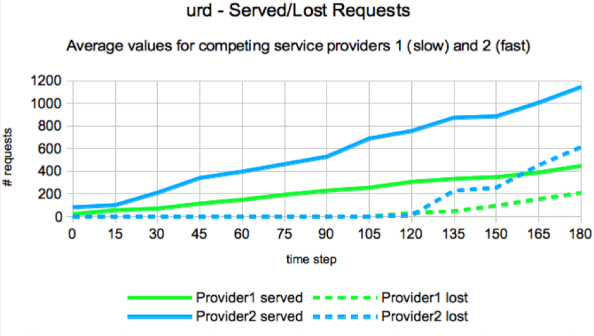


Figure : Uniform primitives, transparent load balancing: both services are used, due to clients `urd` *likely* matching the most advertised one.

Expressiveness III

Transparent Load Balancing III

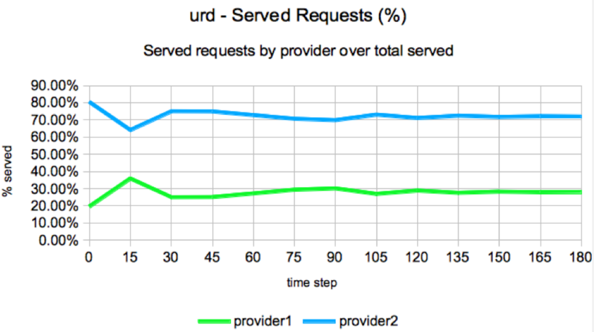


Figure : Uniform primitives, transparent load balancing: the *likelihood* of matching a precise service is driven by its history of successfully served requests, reified in the form of advertisement tuples.

Expressiveness IV

Pheromone-based Coordination I

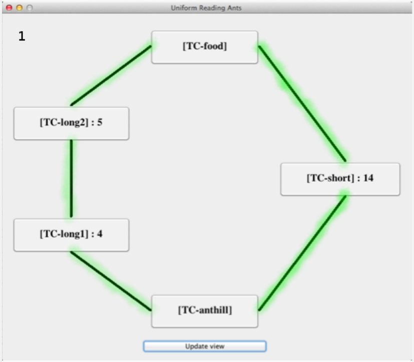


Figure : Digital ants search food ("TC-food" box) wandering randomly from their anthill ("TC-anthill" box).

Expressiveness V

Pheromone-based Coordination II

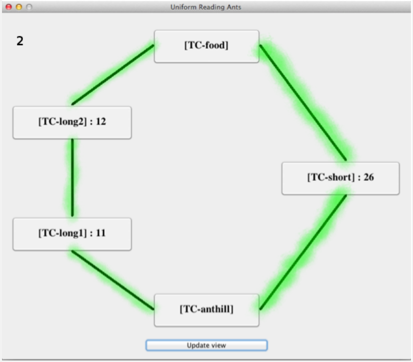


Figure : By `urd`-ing digital pheromones left while carrying food. . .

Expressiveness VI

Pheromone-based Coordination III

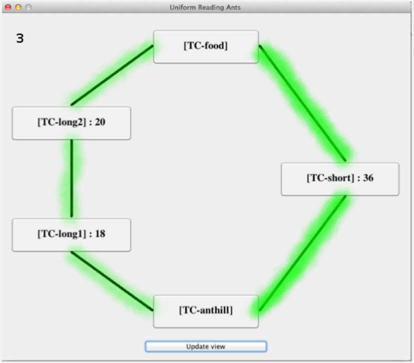


Figure : . . . digital ants stochastically find the optimal path toward the food source.

Expressiveness VII

Pheromone-based Coordination IV

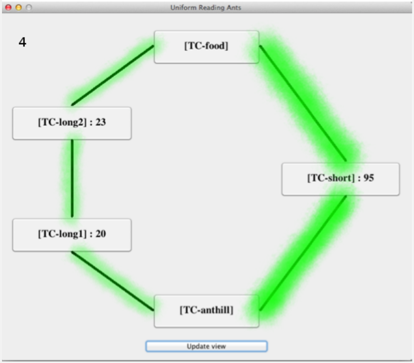


Figure : Numbers next to tuple centre names denote the pheromone strength.

Expressiveness VIII

The formal investigation of uniform primitive expressiveness, in the style of language embedding [Sha91], is omitted for the lack of space, but available within the thesis and in [MO13c][a]. Consider reading [MO13e][b] for a thorough and improved definition of the embedding used.

---

[a] Stefano Mariani and Andrea Omicini. Probabilistic embedding: Experiments with tuple-based probabilistic languages.
In *28th ACM Symposium on Applied Computing (SAC 2013)*, 2013

[b] Stefano Mariani and Andrea Omicini. Probabilistic modular embedding for stochastic coordinated systems.
In *Coordination Models and Languages*, LNCS. Springer, 2013

Outline

What to Save for Later?

- From Underpinnings

The LINDA model as the reference *conceptual framework* upon which to build our own coordination model for self-organisation of knowledge — that is, $\mathcal{MoK}$

- From Chemical reactions as Coordination Laws

The chemical metaphor for (programmable) self-organising coordination, adopted by engineering coordination laws as artificial chemical reactions with *custom kinetic rates*, and tuple spaces as *chemical solution simulators*

- From Uniform Primitives as Coordination Primitives

The basic mechanisms to tame *uncertainty*, given by uniform primitives, exploited to prototype the mentioned chemical metaphor upon a tuple space based setting

Outline

Outline

Premises

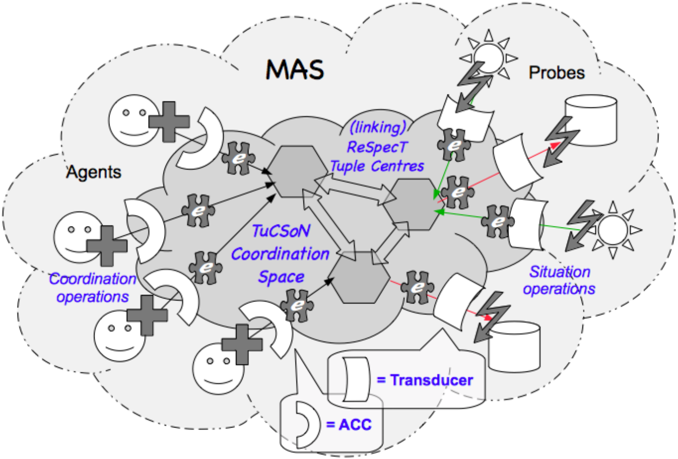
- Things happen in a multi-agent system (MAS) because of either *agent activity* or *environment change*, the only sources of events in MAS
- Complexity arises from both activity-activity and activity-change *dependencies*, that is, from both social (agent-agent) and situated (agent-environment) interaction
- *Coordination*, *managing dependencies* [MC94], can be used to deal with both forms of dependency, exploiting coordination artefacts to handle social and situated interaction [OM13]

We propose an agent-oriented, event-driven architecture for situated pervasive systems exploiting coordination artefacts to handle both social and situated interaction [MO15a][a]

---

[a] Stefano Mariani and Andrea Omicini. Coordinating activities and change: An event-driven architecture for situated MAS.
Engineering Applications of Artificial Intelligence, 2015

TuCSoN Architecture I



TuCSoN Architecture II

agents any *autonomous* computational entity experiencing *social* and *situated interactions*, either with other agents or probes within the MAS

probes environmental resources dealt with either as sources of perceptions (like *sensors*) or targets of actions (like *actuators*) — or both

ACC *boundary artefacts* devoted to agents – *Agent Coordination Context* [Omi02] –, enabling and constraining agents interaction capabilities while preserving agent autonomy, and mapping coordination operations to *external events*

transducers boundary artefacts geared toward probes [CO09], *specialised* to handle events from a given probe, and mapping changes of properties to external events

TuCSoN Architecture III

events according to ReSpecT situated event model (depicted in next slide), they represent in a *uniform way* both *external* events from agents (ACC) and probes (transducers), as well as *internal* events originating from tuple centres

tuple centres the *coordination artefacts*, decoupling in *control, reference, space* and *time* dependencies between both social and situated interactions

TuCSoN Architecture IV

$$
\begin{array}{rcl}
\langle \mathit{Event} \rangle & ::= & \langle \mathit{StartCause} \rangle \texttt{ , } \langle \mathit{Cause} \rangle \texttt{ , } \langle \mathit{Evaluation} \rangle \\
\langle \mathit{StartCause} \rangle \texttt{ , } \langle \mathit{Cause} \rangle & ::= & (\langle \mathit{Activity} \rangle \mid \langle \mathit{Change} \rangle) \texttt{ , } \langle \mathit{Source} \rangle \texttt{ , } \langle \mathit{Target} \rangle \texttt{ , } \langle \mathit{Time} \rangle \texttt{ , } \langle \mathit{Space{:}Place} \rangle \\
\langle \mathit{Source} \rangle \texttt{ , } \langle \mathit{Target} \rangle & ::= & \langle \mathit{AgentId} \rangle \mid \langle \mathit{CoordArtefactId} \rangle \mid \langle \mathit{EnvResId} \rangle \mid \perp \\
\langle \mathit{Evaluation} \rangle & ::= & \perp \mid \{\langle \mathit{Result} \rangle\}
\end{array}
$$

$$
\begin{array}{rcl}
\langle \mathit{Activity} \rangle & ::= & \langle \mathit{Operation} \rangle \mid \langle \mathit{Situation} \rangle \\
\langle \mathit{Operation} \rangle & ::= & \texttt{out(} \langle \mathit{Tuple} \rangle \texttt{)} \mid (\texttt{in} \mid \texttt{rd} \mid \texttt{no} \mid \texttt{inp} \mid \texttt{rdp} \mid \texttt{nop}) \texttt{(} \langle \mathit{Template} \rangle [\texttt{,} \langle \mathit{Term} \rangle] \texttt{)} \\
\langle \mathit{Situation} \rangle & ::= & \texttt{env(} \langle \mathit{Key} \rangle \texttt{ , } \langle \mathit{Value} \rangle \texttt{)} \\
\langle \mathit{Change} \rangle & ::= & \texttt{env(} \langle \mathit{Key} \rangle \texttt{ , } \langle \mathit{Value} \rangle \texttt{)} \mid \texttt{time(} \langle \mathit{Time} \rangle \texttt{)} \mid \texttt{from(} \langle \mathit{Space} \rangle \texttt{ , } \langle \mathit{Place} \rangle \texttt{)} \mid \\
 & & \texttt{to(} \langle \mathit{Space} \rangle \texttt{ , } \langle \mathit{Place} \rangle \texttt{)}
\end{array}
$$

Figure : ReSpecT event model.

Implications

- Situatedness of interactions is of paramount importance for the effectiveness of the coordination mechanisms exploited
- Having a *situated event model*, as well as a *situated event-based architecture*, dealing with both social and situated interactions in a uniform way, makes it easier to implement situated coordination mechanisms
- *Context-awareness* w.r.t. both the environmental and the spatial dimensions[4] of computation is embedded within the coordination medium abstraction, and immediately usable for coordination purposes

[4] The temporal dimension being already tackled in [ORV05].

Outline

Motivation

- Multi-agent systems deployed in *pervasive computing* scenarios are stressing more and more the requirements for coordination middleware [ZCF+11]
- *Spatial issues* are fundamental in many sorts of complex software systems, including multi-agent, pervasive, adaptive, and self-organising ones [BMS11]
- In most of the application scenarios where *situatedness* plays an essential role, coordination is required to be space aware

We propose the notion of *space-aware coordination medium*, by extending ReSpecT tuple centre abstraction and language so as to support space-aware coordination [MO13h][a]

---

[a] Stefano Mariani and Andrea Omicini. Space-aware coordination in ReSpecT.
In *From Objects to Agents*, CEUR Workshop Proceedings, 2013

Space-aware Tuple Centre

- The location of a ReSpecT tuple centre is obtained through the notion of *current place* — e.g., the absolute position of the computational device where it is running, the domain name of the TuCSoN node hosting it, or any other notion of "place"
  - motion is represented by moving from a starting place, and stopping at an arrival place — whatever the notion of space is
- Whenever motion occurs a spatial event is generated
  - it is possible to specify reactions triggered by spatial events: the so-called spatial reactions

A spatial tuple centre can be *programmed* to *react* to motion events either in physical or in virtual space, so as to enforce space-aware coordination policies

Space-aware Coordination Language I

- ReSpecT tuple centres are based on first-order logic (FOL), adopted both for the *communication language* (logic tuples), and for the *behaviour specification language* (ReSpecT)
- Reactions in ReSpecT are defined as Prolog-like facts (*reaction specification* tuples) of the form

  `reaction(`*Activity* `,` *Guards* `,` *Goals*`)`

- A reaction specification tuple specifies the list of the operations (*Goals*) to be executed when a given event occurs (called *triggering event*, caused by an *Activity*), and some conditions about the event hold (*Guards* evaluate to true)

Space-aware Coordination Language II

The ReSpecT language is extended

- spatial predicates get information about the *spatial properties* of both the tuple centre and the triggering event
- it is possible to specify reactions to the occurrence of spatial events

Spatial Observation Predicates

`current_place(@S,?P)` succeeds if P unifies with the position of the node which the tuple centre belongs to

`event_place(@S,?P)` succeeds if P unifies with the position of the node where the triggering event was originated

`start_place(@S,?P)` succeeds if P unifies with the position of the node where the event chain that led to the triggering event was originated

Space-aware Coordination Language III

- position P can be specified according to a notion of space S encompassing: physical position (S=`ph`), IP number (S=`ip`), domain name (S=`dns`), geographical location (S=`map`), or organisational position (S=`org`) — application-specific

Spatial Guard Predicates

`at(`$@S$`,`$@P$`)` succeeds when the tuple centre is currently executing at the position P

`near(`$@S$`,`$@P$`,`$@R$`)` succeeds when the tuple centre is currently executing at the position included in the spatial region with centre P and radius R

Space-aware Coordination Language IV

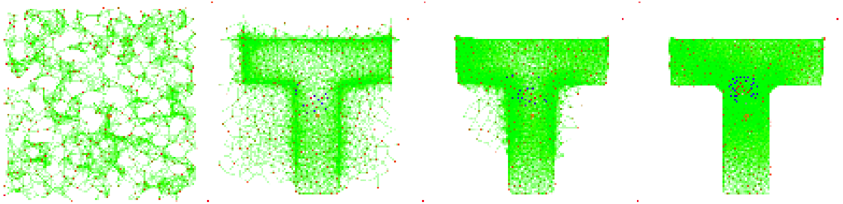
Spatial Event Descriptors

`from(?S,?P)` matches a spatial event raised when the device hosting the tuple centre starts moving from position P

`to(?S,?P)` matches a spatial event raised when the device hosting the tuple centre stops moving and reaches position P

Expressiveness

- This spatial extension to ReSpecT meets the requirements for the "T-Program" benchmark proposed in [BDU+12]



(a) Initial distribution showing network links. (b) $t = 65s$. (c) $t = 135s$. (d) Stabilized center-of-gravity.

- Explanation is omitted for the lack of space, however, it is available in the thesis as well as in [MO13g][5]

[5]Stefano Mariani and Andrea Omicini. Promoting space-aware coordination: ReSpecT as a spatial-computing virtual machine.
In *Spatial Computing Workshop (SCW 2013)*, AAMAS 2013

Outline

What to Save for Later?

- From the infrastructural point of view

The TuCSoN architecture as the reference architecture for the $\mathcal{M}o\mathcal{K}$ middleware, and the TuCSoN infrastructure as the ground upon which to design and implement the $\mathcal{M}o\mathcal{K}$ prototype

- From the linguistic point of view

The ReSpecT language as the language for programming $\mathcal{M}o\mathcal{K}$ artificial chemical reactions and the chemical metaphor machinery in such prototype

Outline

Premises

- Social-technical systems (STS) arise when *cognitive* and *social interaction* is mediated by information technology rather than by the natural world [Whi06]
    - STS include non-technical elements such as people, processes, regulations, etc., which are *inherent parts* of the system
- In a dynamic business environment, where an organisation faces unexpected and novel problems, a computational platform can, at best, be used as an *enabler* to turn data into information, then knowledge, through people interpretation
    - we refer to knowledge management (KM) as a socio-technical process encompassing knowledge creation, validation, distribution, presentation, and application [Bha01], and call knowledge-intensive environments (KIE) those workplaces where KM is necessary

Outline

Challenges in STS I

- STS have emergent properties, which cannot be attributed to individual parts of the system
- STS are often non-deterministic, because system behaviour depends on human operators, and people do not always react in the same way (output) to the same situation (input)
- People may use the STS in ways completely unpredictable from the designers standpoint
- Awareness, that is, knowing who is present, and *peripheral awareness*, namely low-level monitoring of others' activity, are fundamental in STS, because visibility of information flow enables learning and greater efficiency [Ack00]
- People not only adapt to their systems, they adapt their systems to their needs (*co-evolution*) [Ack00]

Challenges in STS II

All the mentioned challenges can be dealt with from a *coordination perspective*, in particular, by exploiting coordination techniques supporting

- programmable — to deal with unpredictability and adaptation
- self-organising — to account for emergence and non-determinism
- situated — supporting awareness

coordination

Challenges in KIE I

- Knowledge creation is an emergent process in which experimentation and pure *chance* play an important role, thus, providing the means to spot opportunities is a great challenge [Bha01]
- Knowledge validation is a painstaking process of continually assessing relevance of the knowledge base, and designing computational techniques seamlessly integrating with users workflows, while transparently assisting them in doing so, is far from trivial
- *Interactions* between technologies, techniques, and people can have *direct bearing* on knowledge distribution, which demands for adaptation techniques supporting continuous and *autonomous* knowledge re-distribution;

Challenges in KIE II

Again, a *coordination perspective* may help alleviate such issues, by applying coordination policies and mechanisms not solely to the agents (either software or human) participating KM, but also to the raw data subject of the process

- data becomes organised information autonomously, through coordination mechanisms seamlessly integrated with knowledge workers' own workflows

Outline

From A&A to Computational Smart Environments I

- Activity Theory (AT) is a social psychological theory for conceptualising human activities
  - the A&A meta-model [ORV08] as a *reference framework* for designing the computational part of a STS for knowledge management
- Cognitive stigmergy [ROV$^+$07] is a first generalisation of *stigmergy* where traces are amenable of a *symbolic interpretation*
  - cognitive stigmergy directly supports both awareness and peripheral awareness in socio-technical systems
- Behavioural Implicit Communication (BIC) is a cognitive theory of communication [Cas06], where tacit messages describe the kind of messages a practical action (and its traces) may *implicitly* send to its observers [CPT10]
  - BIC provides a sound cognitive and social model of action and interaction w.r.t. to both human agents and computational agents

From A&A to Computational Smart Environments II

- There is a gap in current approaches to STS engineering, which can be closed by dealing with [SS00]

  mutual awareness as the basis for *opportunistic*, ad hoc alignment and improvisation, which ensure *flexibility*

  coordinative artefacts *encapsulating* those portions of the coordination responsibilities that is better to *automatise*

BIC seem to provide mutual awareness, while *coordination artefacts* the required coordinative capabilities, paving the way toward computational smart environments [TCR$^+$05]

Outline

Survey

- Looking at real-world STS we devise *practical (virtual) actions* and *tacit messages*
  - Facebook, Twitter, and Google$^+$, the business-oriented LinkedIN, academic social networks such as Mendeley, Academia.edu, and ResearchGate, as well as the Storify[6] citizen journalism publishing platform
- Many actions, therefore messages, can be devised
  - posting to Facebook, sharing tweets, connecting to LinkedIN profiles, publishing papers, following people or papers, asking questions on ResearchGate, and so on
  - "I'm interested in your work", "This is relevant to me", "I want you to read this", "I got this done", ...

We report solely on a set of common actions, factorised from all the above, for the lack of space. However, the full list is available within the thesis.

[6] `http://storify.com`

Factorisation of Actions

share
: encompasses posting information, sharing or citing someone else's, and so on — namely, any action whose effect is *adding information* to the system

mark
: liking a post, voting a question/answer, bookmarking a publication, and so on — any action marking a given piece of information as *relevant or not*

comment
: besides comments to existing posts, replies to comments, answers to questions, and so on — any action *attaching information* to existing information

connect
: adding friends, following people or sources of information, and so on — any action *expanding the network* of relationships between a users and information

harvest
: all kinds of search actions, whatever their target is — any action *acquiring knowledge* about (sources of) information

Sorts of Tacit Messages I

share can convey, e.g., tacit messages *presence*, *ability*, *accomplishment*

- if agent X shares Y's information through action a, agents observing a become aware of existence and location of both X and Y (presence)
- X sharing information i from source S lets X's peers infer X can manipulate S (ability)
- if X shared i with Z, Z may infer, e.g., that X expects Z to exploit it (accomplishment)

mark can convey, e.g., tacit messages *presence*, *opportunity*

- if X is aware of Y marking information i as relevant, X may infer that Y exists (presence)
- then, if i belongs to X, X may infer Y is seeking for collaborations (opportunity)

comment can convey, e.g., tacit messages *opportunity*, *goal*

Sorts of Tacit Messages II

- if X commented Y's information, both may infer collaboration opportunities (opportunity)
- also, by interpreting X's comment, Y may infer X's goal (goal)

connect can convey, e.g., tacit messages *intention*, *opportunity*

- since X manifested interest in Y's work, Y may infer X intention to exploit it (intention)
- accordingly, Y may infer the opportunity for, e.g., collaboration (opportunity)

harvest can convey, e.g., tacit messages *intention*, *opportunity* — however, depends on search criteria

- also, they can infer X goal to acquire knowledge related to its query (intention)
- along the same line, they can take the chance to provide matching information (opportunity)

Outline

What to Save for Later?

- From Activity Theory

Artefacts as a fundamental abstraction in engineering multi-agent systems

- From stigmergy and cognitive stigmergy

The central role played by the notion of trace in supporting *awareness* and *peripheral awareness* in STS

- From Behavioural Implicit Communication

A model of action providing *mutual awareness* through the notion of tacit messages, attached to both actions and traces of actions

Outline

Outline

Overview I

Molecules of Knowledge (MoK) is a coordination model for *self-organisation of knowledge* in STS [MO13a][a]

[a] Stefano Mariani and Andrea Omicini. Molecules of Knowledge: Self-organisation in knowledge-intensive environments. In *Intelligent Distributed Computing VI*. Springer, 2013

- MoK promotes the idea that *data is alive*, spontaneously interacting with other data and its prosumers
- MoK pursues two main goals
  - self-aggregation of information into more complex heaps, possibly reifying useful knowledge previously hidden
  - autonomous diffusion of information toward the interested agents, that is, those needing it to achieve their own goals

Overview II

- A *MoK*-coordinated system is a network of information containers (*compartments*), in which sources of information (*seeds*) continuously and autonomously inject atomic information pieces (*atoms*)
- *MoK* atoms may then aggregate into composite information chunks (*molecules*), diffuse to neighbouring compartments, lose relevance as time flows, gain relevance, or have their properties modified, as a consequence of agents' interactions
- Such autonomous and decentralised processes are the coordination laws dictating how the system evolves (*reactions*), and are influenced by user agents' actions (reified as *enzymes*) and their side effects (reified as *traces*)
- Such traces are transparently, and possibly unintentionally, left within the working environment by either human or software agents (*catalysts*) while performing their activities

Core Abstractions I

Seeds *sources of information*, continuously generating information at a *rate* depending on contextual information, with no need for external intervention

$$\texttt{seed}(\mathit{Src},\mathit{Atoms})_c$$

Atoms any *atomic piece of information*, susceptible to $\mathcal{MoK}$ reactions

$$\texttt{atom}(\mathit{src},\mathit{Content},\mathit{Meta\text{-}info})_c$$

Molecules *collections of semantically related atoms*, spontaneously generated by $\mathcal{MoK}$ reactions

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

Compartments *computational abstraction* in $\mathcal{MoK}$, responsible for handling information storage, lifecycle, evolution, and scheduling of reactions

Core Abstractions II

$$\sum_j (\asymp^j) [\![\mathit{Seeds} \cup \mathit{Atoms} \cup \mathit{Molecules} \ldots \cup \mathit{Reactions}]\!]$$

Membranes *communication* and *topological* abstraction in $\mathcal{M}o\mathcal{K}$, enabling 1 : 1 exchange of information, defining the notions of *locality* and *neighbourhood*

$$[\![\ldots]\!]_i \asymp [\![\ldots]\!]_j$$

Catalysts agents undertaking *(epistemic) actions*, which *influence* both the way in which reactions apply, and information properties

$$\alpha \approx [\![\mathtt{share(\ldots)} \mid \mathtt{mark(\ldots)} \mid \ldots \mid \mathtt{harvest(\ldots)}]\!]$$

Enzymes *transparently* produced by catalysts, reify the *epistemic nature* of actions, enabling adaptiveness, situatedness, and awareness

$$\mathtt{enzyme}(\mathit{Species}, s, \mathit{Reactant}, \mathit{Context})_c$$

Traces *transparently* produced by enzymes, reify (side) effects of actions in the form of *tacit messages*

Core Abstractions III

$$\texttt{trace}(\mathit{Msg}, P_{species}, \mathit{Reactant})_c$$

Perturbations *computational processes* applying (application-specific) side effects of agents' interactions, which may affect both reactions functioning, and information and system properties

$$\texttt{perturbation}(P_{species}, \mathit{Reactant}, \mathit{Msg}, \mathit{Context})_c$$

Reactions the *autonomous* and *decentralised* processes supporting information evolution as well as knowledge inference, discovery and sharing, driven by (application-specific) *semantic similarity* between involved information pieces — details in following slides

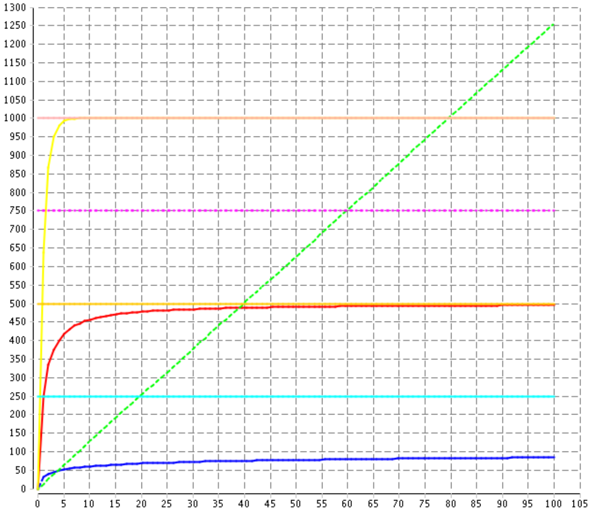
Focus on Reactions I

Injection generates atoms from seeds

$$\texttt{seed}(src\,,Atoms\,) \xrightarrow{r_{inj}} \texttt{seed}(src\,,Atoms\,) + atom^{i}_{c+c_i},$$
$$r_{inj} = \frac{1}{1+\texttt{time}} * \texttt{diff}(\sharp seed, \sharp atom)$$

r_{inj} is a trade-off: on one hand, atoms should be *perpetually* produced, since there is no way to know a-priori *when* some information will be useful; on the other hand, it is desirable to avoid *flooding*

Focus on Reactions II



Focus on Reactions III

Aggregation ties together *semantically related* atoms into molecules, or molecules into other molecules

$$\mathtt{Reactant}' + \mathtt{Reactant}'' \xrightarrow{r_{agg}}$$
$$(\mathtt{Reactant}' \oplus \mathtt{Reactant}'') + (\mathtt{Reactant}' \ominus \mathtt{Reactant}'')$$

$$r_{agg} = \frac{\mathtt{time}}{\sharp \mathtt{Reactant}^{lhs}}$$

r_{agg} enforces direct proportionality between the "size" of a molecule and the "speed" of aggregation[7]

[7] Superscript *lhs* denotes the left-hand-side of the reaction

Focus on Reactions IV

Diffusion moves atoms, molecules, and traces among *neighbouring* compartments

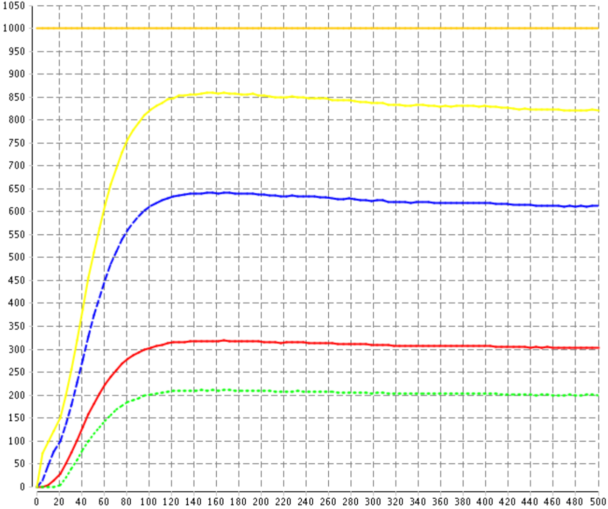
$$\llbracket \mathit{Reactants}' \cup \mathit{Reactant} \rrbracket_i \asymp \llbracket \mathit{Reactants}'' \rrbracket_j \xrightarrow{r_{diff}} \llbracket \mathit{Reactants}' \rrbracket_i \asymp \llbracket \mathit{Reactants}'' \cup \mathit{Reactant} \rrbracket_j$$

$$r_{diff} = \mathcal{d} * \mathtt{diff}(\llbracket \sharp \mathit{Reactant} \rrbracket_i, \llbracket \sharp \mathit{Reactant} \rrbracket_j)$$

r_{diff} is meant to avoiding unbounded proliferation of "foreign" information pieces in compartments which are not their "origin"[8]

[8] Symbol $\asymp$ denotes compartments connected by a membrane. $\mathcal{d}$ is an arbitrary weight factor tuning diffusion "strength".

Focus on Reactions V



Focus on Reactions VI

Decay decreases the concentration of atoms, molecules, enzymes and traces as *time flows*

$$\mathit{Reactant}_c \xrightarrow{r_{dec}} \mathit{Reactant}_{c-1}, \quad c \in \Re^+$$

$$r_{dec} = \begin{cases} \mathtt{fMA}(a_R) * \mathtt{log(1+time)}, & \mathit{Reactant} \setminus \mathit{enzyme} \\ \mathtt{diff}(\sharp\mathit{Reactant}_{\triangleleft}, \sharp\mathit{Reactant}) * \mathtt{log(1+time)} \end{cases}$$

r_{dec} has been designed to asymptotically tend to *homogenise* information relevance[9]

[9] fMA is the *law of mass action* [Car08], a mathematical model explaining the behaviour of solutions in dynamic equilibrium. Symbol $\triangleleft$ denotes where a MoK abstraction comes from.

Focus on Reactions VII

Focus on Reactions VIII

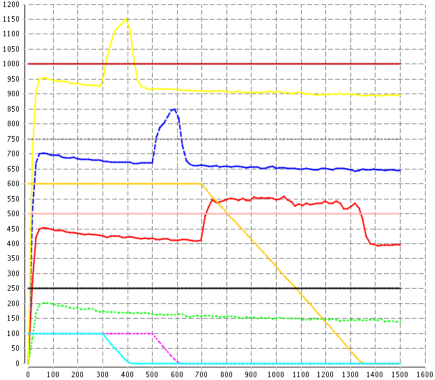
Reinforcement increases the concentration of atoms and molecules according to catalysts' *interactions*

$$\mathtt{enzyme}(\mathit{species}, s, \mathit{Reactant}, \mathit{Context}) + \mathit{Reactant}'_c \xrightarrow{r_{reinf}} \mathtt{enzyme}(\mathit{species}, s, \mathit{Reactant}, \mathit{Context}) + \mathit{Reactant}'_{c+s}$$

$$r_{reinf} = \mathtt{diff}(\sharp \mathit{Reactant}_{\triangleleft}, \sharp \mathit{Reactant})$$

r_{reinf} is meant to enforce *situatedness* of the feedback catalysts implicitly provide to $\mathcal{MoK}$ through epistemic actions

Focus on Reactions IX



Focus on Reactions X

Deposit generates traces from enzymes

$$\texttt{enzyme}(species\,,s\,,Reactant\,,Context) \xrightarrow{r_{dep}}$$
$$\texttt{enzyme}(species\,,s\,,Reactant\,,Context) +$$
$$\texttt{trace}(Msg\,,p_{\,species}\,,Reactant)_{c+s}$$

$$r_{dep} = \frac{1}{1+\texttt{time}} * \texttt{diff}(\sharp enzyme\,, \sharp trace)$$

r_{dep} represents the same trade-off between *flooding* and *availability* seen for injection reaction

Perturbation carries out the *(side) effects* of (interaction) activities undertaken by catalysts

$$\texttt{trace}(Msg\,,p_{\,species}\,,Reactant) + Reactant' \xrightarrow{r_{pert}}$$
$$\dagger(p_{\,species}, Reactant')$$

$$r_{pert} \propto (p_{\,species}, Reactant', trace)$$

r_{pert} depends on the specific perturbation action the trace applies, in relation to its species

Interactions in a Nutshell

- Any action α carried out by any agent transparently produces enzymes within the compartment where the action took place
  - each action produces only one $\mathcal{Species}$ of enzyme, so as to reify its *epistemic nature*
- Any enzyme spontaneously and temporarily deposits traces within its compartment solely, for situatedness sake
  - each enzyme may produce different traces, with different $\mathcal{Msg}$ and different $p_{species}$, so as to reify the different sorts of tacit message and perturbation actions traces may convey
- Any trace is then free to diffuse among neighbouring compartments, looking for a chance to apply their perturbation action
  - e.g., matching reactants (atoms, molecules, . . .)
- Each perturbation action may have different effects on the $\mathcal{MoK}$ system, based on $\mathcal{Context}$ and $\mathcal{Msg}$
  - e.g., different tacit messages ($\mathcal{Msg}$) and a different system state ($\mathcal{Context}$) when the action was undertaken may lead to different behaviours on $\mathcal{MoK}$ side

Semantic Matchmaking in a Nutshell

- $\mathcal{MoK}$ needs a function for measuring semantic similarity between information items
  - so as to promote *semantically-driven* aggregation, reinforcement, diffusion, and perturbation
- The $\mathcal{F}_{\mathcal{MoK}}$ function is devoted to this, as a *fuzzy matchmaking function* considering different *text-mining related* techniques for measuring semantic similarity between documents, and, more generally, text
  - e.g., cosine similarity, euclidean distance, average quadratic difference, ...
- The $\mathcal{F}_{\mathcal{MoK}}$ function depends on the way in which documents, or excerpts of documents, are represented within $\mathcal{MoK}$
  - experimented techniques include key-phrases extraction, concept mining, vector-spaces, ...

Early Evaluation: $\mathcal{M}o\mathcal{K}$ for Anticipatory coordination

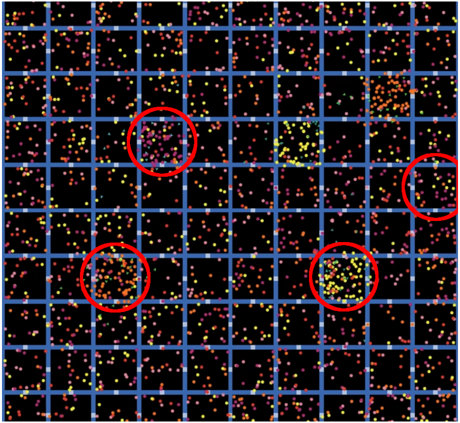


Figure : Whereas atoms and molecules are initially randomly scattered across compartments, as soon as catalysts interact clusters appear by emergence, thanks to *BIC-driven self-organisation*. Whenever new actions are performed by catalysts, $\mathcal{M}o\mathcal{K}$ adaptively re-organises the spatial configuration of information so as to better tackle the new coordination needs.

Outline

Premises

- The TuCSoN infrastructure is an ideal candidate for the implementation of a MoK-like middleware
  - its situated architecture enables definition of a *topology* of *situated computational loci* while promoting *awareness*
  - its distributed nature provides for a notion of *locality* and tackles scalability
  - programmability of its ReSpecT tuple centres supports implementation of MoK chemical-inspired coordination model and BIC-based interaction model

Core Abstractions Mapping I

MoK reactants – atoms, molecules, ... – map to TuCSoN first-order logic tuples of the form

`reactant(`*`Reactant`*`, `*`Concentration`*`)`

MoK reactions are implemented as a combination of TuCSoN tuples and ReSpecT specifications

- *stochastic*, *temporal*, and *instantaneous* chemical law tuples represent MoK reactions as tuples of the form

  `law(`*`[Reactants]`*`, `*`Rate`*`, `*`[Products]`*`)`

- ReSpecT specifications implement different *functional* and *non-functional* aspects of MoK reactions — e.g. sending reactants to remote compartments, adaptation of rates, ...

Core Abstractions Mapping II

MoK compartments map to ReSpecT tuple centres

- suitable ReSpecT specifications transform tuple centres in *chemical engines*[a]

[a] A ReSpecT program implements a variation of Gillespie algorithm for stochastic simulation of a chemical solution in dynamic equilibrium [Gil77].

MoK catalysts are simulated by TuCSoN agents

- ReSpecT allows to define *new coordination primitives* — MoK actions
- as well as how the tuple centre must *react* to their invocation — enzymes release, traces deposit, . . .

Core Abstractions Mapping III

MoK perturbation actions map to TuCSoN spawned activities

- the `spawn` primitive is a TuCSoN implementation of LINDA `eval` primitive, allowing to spawn *independent computational processes* running within the coordination medium

The notion of *neighbourhood* is implemented through application-specific links between TuCSoN nodes

Semantic matchmaking is achieved thanks to tuProlog [DOR01], the Prolog engine which interprets TuCSoN first-order logic tuples, that allows to re-define LINDA matching function

The Chemical Engine Logic

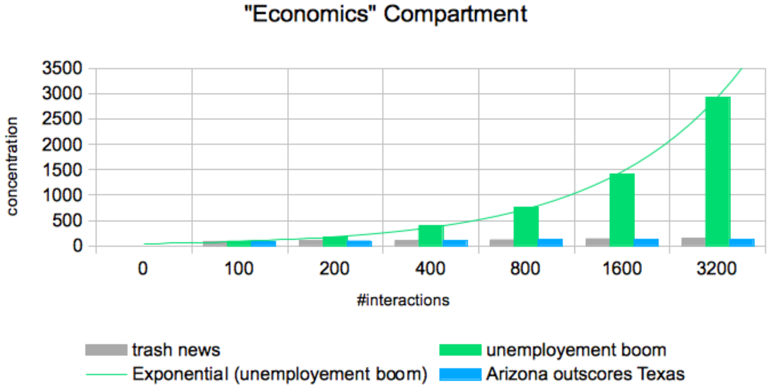
1. select the chemical law to schedule for execution
   1. match reactant *templates* against available reactants, to collect *triggerable* laws
   2. compute *effective* rates for all the triggerable laws
   3. randomly select a triggerable law, *stochastically* chosen according to the computed effective rates and following *Gillespie algorithm* for chemical solution simulation
2. execute the selected chemical law
   1. instantiate products
   2. update reactants and products quantity in the space
   3. enqueue diffusing reactants
   4. update the state of the system—e.g., Gillespie exponential decay

Thorough description of the ReSpecT specifications implementing this is omitted for the lack of space, but available within the thesis

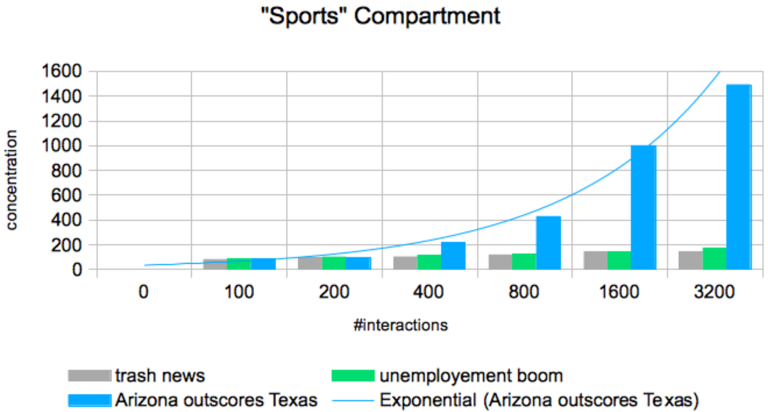
Early Evaluation: $\mathcal{MoK}$-`News` I

- Citizen journalism and news management provide a prominent example of a knowledge-intensive environment, thus we choose it as the first case study for $\mathcal{MoK}$
  1. we studied the NewsML[10] and NITF[11] standards for knowledge representation in the news management domain
  2. we *specialised* $\mathcal{MoK}$ seeds to represent sources of information, $\mathcal{MoK}$ atoms pieces of news stories, and so on
     - most notably, concentration now resembles news items' relevance, and $\mathcal{F}_{\mathcal{MoK}}$ is based on the *NewsCodes*[12] ontology
  3. the resulting $\mathcal{MoK}$ system, which we call $\mathcal{MoK}$-`News`, has been then evaluated on a "smart knowledge diffusion" scenario
     - different $\mathcal{MoK}$ compartments are deployed, used as workspaces by journalists interested in different news topics
     - despite $\mathcal{MoK}$ diffusion being *equiprobable* w.r.t. to each compartment, the interplay with decay and reinforcement reactions make news spread among compartments according to journalists' interests — as desired, and predicted

Early Evaluation: MoK-News II



Early Evaluation: MoK-News III



[10] https://iptc.org/standards/newsml-g2/
[11] https://iptc.org/standards/nitf/
[12] https://iptc.org/standards/newscodes/

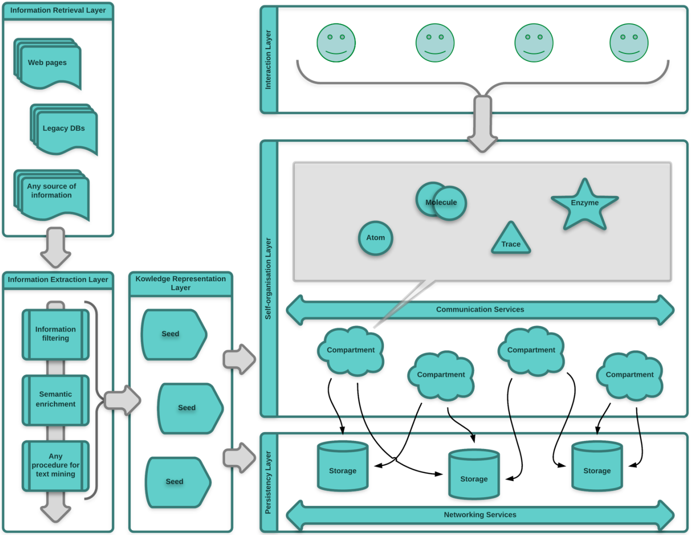
Outline

Premises

Besides MoK prototype implemented on TuCSoN, and ad-hoc implementations of use cases, a comprehensive MoK ecosystem is currently under development

- automatic *information retrieval* and extraction
- automatic *semantic enrichment* of unstructured text
- graph and document oriented storage layer
- networking and communication facilities such as automatic discovery of compartments, dynamic topology re-configuration, gossiping, adaptive routing
- automatic *knowledge inference and discovery*, based on semantic similarity measures
- interaction layer supporting behavioural implicit communication mechanisms to *assist and drive* automatic knowledge inference and discovery

MoK Ecosystem Architecture



Development Overview

Information harvesting

Google-based crawlers mining wikipedia pages for semi-structured information

Networking

Asynchronous, channel-based services supporting automatic discovery of compartments, dynamic re-configuration of the network topology upon (dis)connections, point-to-point and multicast communication based on message passing

Communication

Gossiping algorithm based on probabilistic recursive multicast, adaptive routing for targeted communications between compartments, tolerant to dynamic network re-configurations

Outline

Conclusion

- Engineering effective coordination for large-scale, knowledge-intensive STS is a difficult task
- Nature-inspired approaches proven successful in mitigating the issue, by leveraging self-organisation and adaptiveness
- We may further improve by shifting attention toward the social side of STS, transparently exploiting the epistemic nature of users' (inter-) actions for coordination purposes

The tools in our hands

BIC, (cognitive) stigmergy, and biochemical coordination give us the right models and approaches to do so

Outlook I

- The world needs *efficient* and *smart* ways of preserving, managing, and analysing the astonishing amount of knowledge it produces and consumes every day
- Big data approaches are more or less the standard now, mostly because they are good in finding *patterns* of knowledge, but:
  - they mostly neglect "humans-in-the-loop", relying on algorithms and measures (e.g. of similarity) which are completely *user-neutral* and *goal-independent*
  - they won't scale forever
  - they are not suitable for pervasive and privacy-demanding scenarios

Outlook II

- We are in the perfect spot to start a paradigm shift toward self-organising knowledge, where:
  - user-centric adaptiveness of knowledge discovery processes is the foremost goal
  - measures and algorithms exploited for knowledge discovery, inference, management and analysis natively account for users' goals
  - seamlessly scale up/down/out/in naturally, being operating on the assumption that only local-information is available consistently
- As witnessed by the latest H2020 calls, increasingly demanding *user-inclusive* policy making, governance *participation*, *user-centric* knowledge sharing platforms, etc.
  - H2020-SC6-CO-CREATION-2016-2017
  - H2020-EINFRA-2016-2017
  - H2020-FETPROACT-2016-2017

Thanks for your attention

(Friendly) Questions are welcome

URLs

Slides

http://apice.unibo.it/xwiki/bin/view/Talks/MokPhD2016

MoK

http://mok.apice.unibo.it/

References I

Mark S Ackerman.
The intellectual challenge of cscw: the gap between social requirements and technical feasibility.
Human–Computer Interaction, 15(2-3):179–203, 2000.

Jacob Beal and Jonathan Bachrach.
Infrastructure for engineered emergence on sensor/actuator networks.
Intelligent Systems, IEEE, 21(2):10–19, 2006.

Jacob Beal, Stefan Dulman, Kyle Usbeck, Mirko Viroli, and Nikolaus Correll.
Organizing the aggregate: Languages for spatial computing.
CoRR, abs/1202.5509, 2012.

Ganesh D. Bhatt.
Knowledge management in organizations: Examining the interaction between technologies, techniques, and people.
Journal of Knowledge Management, 5(1):68–75, 2001.

Jacob Beal, Olivier Michel, and Ulrik Pagh Schultz.
Spatial computing: Distributed systems that take advantage of our geometric world.
ACM Trans. Auton. Adapt. Syst., 6(2):11:1–11:3, June 2011.

References II

Luca Cardelli.
On process rate semantics.
Theoretical computer science, 391(3):190–215, 2008.

C Castlefranchi.
From conversation to interaction via behavioral communication: For a semiotic design of objects, environments, and behaviors.
Theories and practice in interaction design, pages 157–79, 2006.

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

Paolo Ciancarini.
Coordination models and languages as software integrators.
ACM Computing Surveys, 28(2):300–302, June 1996.

Giacomo Cabri, Letizia Leonardi, and Franco Zambonelli.
MARS: A programmable coordination architecture for mobile agents.
IEEE Internet Computing, 4(4):26–35, July/August 2000.

References III

Matteo Casadei and Andrea Omicini.
Situated tuple centres in ReSpecT.
In Sung Y. Shin, Sascha Ossowski, Ronaldo Menezes, and Mirko Viroli, editors, *24th Annual ACM Symposium on Applied Computing (SAC 2009)*, volume III, pages 1361–1368, Honolulu, Hawai'i, USA, 8–12 March 2009. ACM.

Cristiano Castelfranchi, Giovanni Pezzullo, and Luca Tummolini.
Behavioral implicit communication (BIC): Communicating with smart environments via our practical behavior and its traces.
International Journal of Ambient Computing and Intelligence, 2(1):1–12, January–March 2010.

Rocco De Nicola, Diego Latella, Joost-Pieter Katoen, and Mieke Massink.
StoKlaim: A stochastic extension of Klaim.
Technical Report 2006-TR-01, Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" (ISTI), 2006.

References IV

Enrico Denti, Andrea Omicini, and Alessandro Ricci.
tuProlog: A light-weight Prolog for Internet applications and infrastructures.
In I.V. Ramakrishnan, editor, *Practical Aspects of Declarative Languages*, volume 1990 of *LNCS*, pages 184–198. Springer, 2001.
3rd International Symposium (PADL 2001), Las Vegas, NV, USA, 11–12 March 2001. Proceedings.

Tom De Wolf and Tom Holvoet.
Design patterns for decentralised coordination in self-organising emergent systems.
In *Engineering Self-Organising Systems*, pages 28–49. Springer, 2007.

Jose Luis Fernandez-Marquez, Giovanna Di Marzo Serugendo, and Josep Lluis Arcos.
Infrastructureless spatial storage algorithms.
ACM Transactions on Autonomous and Adaptive Systems (TAAS), 6(2):15, 2011.

JoseLuis Fernandez-Marquez, Giovanna Marzo Serugendo, Sara Montagna, Mirko Viroli, and JosepLluis Arcos.
Description and composition of bio-inspired design patterns: a complete overview.
Natural Computing, pages 1–25, 2012.

References V

Jose Luis Fernandez-Marquez, Giovanna Di Marzo Serugendo, and Sara Montagna.
Bio-core: Bio-inspired self-organising mechanisms core.
In *Bio-Inspired Models of Networks, Information, and Computing Systems*, volume 103 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 59–72. Springer Berlin Heidelberg, 2012.

David Gelernter.
Generative communication in Linda.
ACM Transactions on Programming Languages and Systems, 7(1):80–112, January 1985.

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

Luca Gardelli, Mirko Viroli, Matteo Casadei, and Andrea Omicini.
Designing self-organising MAS environments: The collective sort case.
In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNAI*, pages 254–271. Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

References VI

Oltea Mihaela Herescu and Catuscia Palamidessi.
Probabilistic asynchronous pi-calculus.
CoRR, cs.PL/0109002, 2001.

Stefano Mariani.
Parameter engineering vs. parameter tuning: the case of biochemical coordination in MoK.
In Matteo Baldoni, Cristina Baroglio, Federico Bergenti, and Alfredo Garro, editors, *From Objects to Agents*, volume 1099 of *CEUR Workshop Proceedings*, pages 16–23, Turin, Italy, 2–3 December 2013. Sun SITE Central Europe, RWTH Aachen University.

Stefano Mariani.
Parameter engineering vs. parameter tuning: the case of biochemical coordination in MoK.
In *From Objects to Agents*, volume 1099 of *CEUR Workshop Proceedings*, 2013.

References VII

Stefano Mariani.
On the "local-to-global" issue in self-organisation: Chemical reactions with custom kinetic rates.
In *Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014*, Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014, pages 61 – 67, London, UK, September 2014. IEEE.
Best student paper award.

Stefano Mariani.
On the "local-to-global" issue in self-organisation: Chemical reactions with custom kinetic rates.
In *Eighth IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2014*, 2014.
Best student paper award.

Thomas W. Malone and Kevin Crowston.
The interdisciplinary study of coordination.
ACM Computing Surveys, 26(1):87–119, 1994.

References VIII

Stefano Mariani and Andrea Omicini.
Promoting space-aware coordination: ReSpecT as a spatial-computing virtual machine.
In *Spatial Computing Workshop (SCW 2013)*, AAMAS 2013.

Stefano Mariani and Andrea Omicini.
Molecules of Knowledge: Self-organisation in knowledge-intensive environments.
In Giancarlo Fortino, Costin Bădică, Michele Malgeri, and Rainer Unland, editors, *Intelligent Distributed Computing VI*, volume 446 of *Studies in Computational Intelligence*, pages 17–22. Springer, 2013.

Stefano Mariani and Andrea Omicini.
Molecules of Knowledge: Self-organisation in knowledge-intensive environments.
In *Intelligent Distributed Computing VI*. Springer, 2013.

Stefano Mariani and Andrea Omicini.
Probabilistic embedding: Experiments with tuple-based probabilistic languages.
In *28th ACM Symposium on Applied Computing (SAC 2013)*, pages 1380–1382, Coimbra, Portugal, 18–22 March 2013.
Poster Paper.

References IX

Stefano Mariani and Andrea Omicini.
Probabilistic embedding: Experiments with tuple-based probabilistic languages.
In *28th ACM Symposium on Applied Computing (SAC 2013)*, 2013.

Stefano Mariani and Andrea Omicini.
Probabilistic modular embedding for stochastic coordinated systems.
In Christine Julien and Rocco De Nicola, editors, *Coordination Models and Languages*, volume 7890 of *LNCS*, pages 151–165. Springer, 2013.
15th International Conference (COORDINATION 2013), Florence, Italy, 3–6 June 2013. Proceedings.

Stefano Mariani and Andrea Omicini.
Probabilistic modular embedding for stochastic coordinated systems.
In *Coordination Models and Languages*, LNCS. Springer, 2013.

Stefano Mariani and Andrea Omicini.
Promoting space-aware coordination: ReSpecT as a spatial-computing virtual machine.
In *Spatial Computing Workshop (SCW 2013)*, AAMAS 2013, Saint Paul, Minnesota, USA, May 2013.

References X

Stefano Mariani and Andrea Omicini.
Space-aware coordination in ReSpecT.
In Matteo Baldoni, Cristina Baroglio, Federico Bergenti, and Alfredo Garro, editors, *From Objects to Agents*, volume 1099 of *CEUR Workshop Proceedings*, pages 1–7, Turin, Italy, 2–3 December 2013. Sun SITE Central Europe, RWTH Aachen University.
XIV Workshop (WOA 2013). Workshop Notes.

Stefano Mariani and Andrea Omicini.
Space-aware coordination in ReSpecT.
In *From Objects to Agents*, CEUR Workshop Proceedings, 2013.

Stefano Mariani and Andrea Omicini.
Coordinating activities and change: An event-driven architecture for situated MAS.
Engineering Applications of Artificial Intelligence, 41:298–309, May 2015.
Special Section on Agent-oriented Methods for Engineering Complex Distributed Systems.

Stefano Mariani and Andrea Omicini.
Coordinating activities and change: An event-driven architecture for situated MAS.
Engineering Applications of Artificial Intelligence, 2015.

References XI

Naftaly H. Minsky and Victoria Ungureanu.
Law-Governed interaction: A coordination and control mechanism for heterogeneous distributed systems.
ACM Transactions on Software Engineering and Methodology (TOSEM), 9(3):273–305, 2000.

Marco Mamei and Franco Zambonelli.
Programming pervasive and mobile computing applications: The TOTA approach.
ACM Transactions on Software Engineering and Methodology (TOSEM), 18(4):15:1–15:56, July 2009.

Radhika Nagpal.
A catalog of biologically-inspired primitives for engineering self-organization.
In *Engineering Self-Organising Systems*, pages 53–62. Springer, 2004.

Andrea Omicini and Stefano Mariani.
Coordination for situated MAS: Towards an event-driven architecture.
In Daniel Moldt and Heiko Rölke, editors, *International Workshop on Petri Nets and Software Engineering (PNSE'13)*, volume 989 of *CEUR Workshop Proceedings*, pages 17–22. Sun SITE Central Europe, RWTH Aachen University, 6 August 2013.

References XII

Andrea Omicini.
Towards a notion of agent coordination context.
In Dan C. Marinescu and Craig Lee, editors, *Process Coordination and Ubiquitous Computing*, chapter 12, pages 187–200. CRC Press, Boca Raton, FL, USA, October 2002.

Andrea Omicini.
Formal ReSpecT in the A&A perspective.
Electronic Notes in Theoretical Computer Science, 175(2):97–117, June 2007.
5th International Workshop on Foundations of Coordination Languages and Software Architectures (FOCLASA'06), CONCUR'06, Bonn, Germany, 31 August 2006. Post-proceedings.

Andrea Omicini, Alessandro Ricci, and Mirko Viroli.
Time-aware coordination in ReSpecT.
In Jean-Marie Jacquet and Gian Pietro Picco, editors, *Coordination Models and Languages*, volume 3454 of *LNCS*, pages 268–282. Springer-Verlag, April 2005.
7th International Conference (COORDINATION 2005), Namur, Belgium, 20–23 April 2005. Proceedings.

References XIII

Andrea Omicini, Alessandro Ricci, and Mirko Viroli.
Artifacts in the A&A meta-model for multi-agent systems.
Autonomous Agents and Multi-Agent Systems, 17(3):432–456, December 2008.
Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems.

Andrea Omicini and Mirko Viroli.
Coordination models and languages: From parallel computing to self-organisation.
The Knowledge Engineering Review, 26(1):53–59, March 2011.
Special Issue 01 (25th Anniversary Issue).

H. Van Dyke Parunak, Sven Brueckner, and John Sauter.
Digital pheromone mechanisms for coordination of unmanned vehicles.
In Cristiano Castelfranchi and W. Lewis Johnson, editors, *1st International Joint Conference on Autonomous Agents and Multiagent systems*, volume 1, pages 449–450, New York, NY, USA, 15–19 July 2002. ACM.

References XIV

Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva.
Cognitive stigmergy: Towards a framework based on agents and artifacts.
In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for MultiAgent Systems III*, volume 4389 of *LNCS*, pages 124–140. Springer, May 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

Ehud Shapiro.
Separating concurrent languages with categories of language embeddings.
In *23rd Annual ACM Symposium on Theory of Computing (STOC'91)*, pages 198–208, New York, NY, USA, 1991. ACM.

C Simone and K Schmidt.
Mind the gap! towards a unified view of cscw.
In *Fourth International Conference on Design of Cooperative Systems (COOP2000), Sophia-Antipolis (Fr)*, 2000.

References XV

Luca Tummolini, Cristiano Castelfranchi, Alessandro Ricci, Mirko Viroli, and Andrea Omicini.
"Exhibitionists" and "voyeurs" do it better: A shared environment approach for flexible coordination with tacit messages.
In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for Multi-Agent Systems*, volume 3374 of *Lecture Notes in Artificial Intelligence*, pages 215–231. Springer, February 2005.

Robert Tolksdorf and Ronaldo Menezes.
Using Swarm Intelligence in Linda Systems.
In Andrea Omicini, Paolo Petta, and Jeremy Pitt, editors, *Engineering Societies in the Agents World IV*, volume 3071 of *LNCS*, pages 49–65. Springer, June 2004.
4th International Workshops (ESAW 2003), London, UK, 29-31 October 2003. Revised Selected and Invited Papers.

A.-E. Tchao, M. Risoldi, and G. Di Marzo Serugendo.
Modeling self-* systems using chemically-inspired composable patterns.
In *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, pages 109 –118, oct. 2011.

References XVI

Alan Mathison Turing.
Systems of logic based on ordinals.
Proceedings of the London Mathematical Society, 2(1):161–228, 1939.

Mirko Viroli and Matteo Casadei.
Biochemical tuple spaces for self-organising coordination.
In John Field and Vasco T. Vasconcelos, editors, *Coordination Languages and Models*, volume 5521 of *Lecture Notes in Computer Science*, pages 143–162. Springer, Lisbon, Portugal, June 2009.

Mirko Viroli, Matteo Casadei, Sara Montagna, and Franco Zambonelli.
Spatial coordination of pervasive services through chemical-inspired tuple spaces.
ACM Transactions on Autonomous and Adaptive Systems, 6(2):14:1–14:24, June 2011.

Peter Wegner.
Why interaction is more powerful than algorithms.
Communications of the ACM, 40(5):80–91, May 1997.

Brian Whitworth.
Socio-technical systems.
Encyclopedia of human computer interaction, pages 533–541, 2006.

References XVII

Franco Zambonelli, Gabriella Castelli, Laura Ferrari, Marco Mamei, Alberto Rosi, Giovanna Di Marzo, Matteo Risoldi, Akla-Esso Tchao, Simon Dobson, Graeme Stevenson, Yuan Ye, Elena Nardini, Andrea Omicini, Sara Montagna, Mirko Viroli, Alois Ferscha, Sascha Maschek, and Bernhard Wally.
Self-aware pervasive service ecosystems.
Procedia Computer Science, 7:197–199, December 2011.

Coordination Issues in Complex Socio-technical Systems: Self-organisation of Knowledge in $\mathcal{M}o\mathcal{K}$

Candidate: Stefano Mariani
Supervisor: Prof. Andrea Omicini

DISI
Università di Bologna

PhD Presentation
Bologna, Italia
February 24th, 2016