Anticipatory Coordination in Socio-Technical Knowledge-Intensive Environments: Behavioural Implicit Communication in $\mathcal{MoK}$

Stefano Mariani(✉) and Andrea Omicini

DISI, Alma Mater Studiorum–Università di Bologna, Bologna, Italy
{s.mariani,andrea.omicini}@unibo.it

Abstract. Some of the most peculiar traits of socio-technical KIE (knowledge-intensive environments) – such as unpredictability of agents' behaviour, ever-growing amount of information to manage, fast-paced production/consumption – tangle coordination of information, by affecting, e.g., reachability by knowledge prosumers and manageability by the IT infrastructure. Here, we propose a novel approach to coordination in KIE, by extending the $\mathcal{MoK}$ model for knowledge self-organisation with key concepts from the cognitive theory of BIC (behavioural implicit communication).

1 Introduction

Socio-technical systems (STS) arise when cognitive and social interaction are mediated by information technology, rather than by the natural world alone [18]: in other words, any system in which the infrastructure enabling and constraining interaction is technological, but the evolution of the system is driven by social and cognitive interactions, is a STS. By definition, STS are heavily interaction-centred, so they need proper *coordination* mechanisms at the infrastructure level to harness the intricacies of run-time dependencies between the agents (either software or human) participating the system [8]. However, designing effective coordination is made complex by, at least, two aspects of STS:

unpredictability — By definition, STS have "humans-in-the-loop", and, whereas software behaviour is programmable and predictable, human's one is not. Accordingly, the coordination infrastructure may only draw the boundaries within which user behaviour can stretch, by defining the set of admissible actions and interactions at users' disposal.

scale — STS are typically physically-distributed open systems, often large-scale ones, connecting an ever-increasing number of people, devices, data. Hence, the coordination infrastructure of STS should exploit decentralised coordination mechanisms to be able to scale in/out upon need.

In addition, STS are often deployed within *knowledge-intensive environments* (KIE), that is, workplaces in which sustainability of the organisation long-term goals is influenced by (if not even dependant on) the evolution of knowledge

M. Gavanelli et al. (Eds.): AI*IA 2015, LNAI 9336, pp. 102–115, 2015.
DOI: 10.1007/978-3-319-24309-2_8

embodied within the organisation itself [1]. The fact that knowledge is an *organised* combination of data, procedures, and operations, continuously interacting and evolving driven by human users' practice and (learnt) experience [1], motivates why, usually, KIE are computationally supported by STS. Therefore, KIE, too, call for suitable coordination mechanisms, whose development is far from trivial, mostly due to the following key aspects of KIE:

size — KIE store a massive amount of raw data (knowledge-intensive in space), aggregated information, reification of procedures and best-practices, and the like. The coordination infrastructure should then minimise the overhead of additional information needed for coordination-related functional and non-functional requirements, by relying as much as possible on the information already in the KIE.

pace — Likewise, data within KIE is produced and consumed at a fast pace (knowledge-intensive in time): when the system features a huge number of users, an ever-increasing computational load is inevitably charged on the underlying coordination infrastructure. Hence, coordination mechanisms adopted to organise information should be as simple and efficient as possible.

In order to tackle the issues above, coordination models and technologies draw inspiration from *distributed collective intelligence* phenomena in natural systems, looking for self-organising and adaptive coordination mechanisms—as witnessed, e.g., by [9,11,15,16,19]. Similarly, in this paper we focus on the "social layer" of STS, looking for novel coordination approaches inspired by the latest cognitive and social sciences research results. In particular, we take as a reference the $\mathcal{M}$olecules of $\mathcal{K}$nowledge ($\mathcal{MoK}$) coordination model for knowledge self-organisation in KIE [11], and extend it toward the notion of *anticipatory coordination* – as an efficient form of *collective intelligence* arising by emergence from a number of distributed non-intelligent agents –, according to the theory of *behavioural implicit communication* (BIC) [3].

Accordingly, the remainder of the paper is structured as follows: Section 2 summarises BIC and recaps the key features of $\mathcal{MoK}$; Section 3 presents the main contribution of the paper, that is, the BIC-oriented extension of $\mathcal{MoK}$ supporting anticipatory coordination; Section 4 reports on an early validation of the model; finally, Section 5 provides for concluding remarks and further works.

2 Background

2.1 Behavioural Implicit Communication

Behavioural implicit communication (BIC) is a form of implicit interaction with no specialised signal conveying the message, since the message is the practical behaviour itself [3]. This presupposes advanced *observation* capabilities: participants should be able to observe others' actions, as well as to *mind-read* the intentions behind them. Mind-reading enables the process of *signification*, that is, the ability to ascribe goals and intentions to actions and their effects (*traces*), or, in other words, meanings to signs. In turn, signification enables *anticipatory coordination*, that is, the ability to foresee possible interferences/opportunities so as to adapt accordingly, or, at least, to plan suitable coordinated actions [2].

The crucial point of BIC is that it applies to human beings, to both cognitive and non-cognitive agents, and to computational environments as well [17]. This paves the way towards the notion of *smart environments*, that is, pro-active, intelligent working environments able to autonomously and spontaneously adapt their behaviour according to users' interactions [3]—which is, not by chance, the very notion of anticipatory coordination. Also, smart environments enable BIC based on the observation of traces of actions, too. *Trace-based communication* is related to the notion of *stigmergy*, introduced in the biological study of social insects [6] to explain the coordination of termites building their nest without exchanging messages—another form of distributed collective intelligence. Adopting the perspective taken in [3], stigmergy is communication via environment modifications which are not specialised signals: so, stigmergy can be interpreted as a special form of BIC, where the addressee does not directly perceive the behaviour, but just other post-hoc traces and outcomes of it.

In [14], an abstract model for smart environments, supporting BIC in the context of multi-agent systems (MAS), defines two types of environment:

c-env — A *common environment*, where agents can observe only the state of the environment, not the actions of their peers. A trace is modelled as a part of the environment, instead of as a product of other agents. *c-env* enables agents to modify environment state while keeping track of such changes.

s-env — A *shared environment*, as an enhanced *c-env* enabling different forms of observability of actions, and awareness of this observability—by the agents, and by the environment itself as well.

Accordingly, three fundamental features are required for a computational environment to fully support BIC-based coordination, closely related to observation, mind-reading, and signification abilities [14]: *(i)* observability of (human / software) agent actions, and of their traces as well, should be an intrinsic property of the environment; *(ii)* agents and the environment should be able to understand actions and their traces, possibly inferring intentions and goals motivating them—regardless of whether they are intelligent enough to perform true reasoning, or merely programmed to react properly; *(iii)* agents and the environment should also be able to understand the effects of their activity on other agents, so as to exploit the opportunity to obtain a desired reaction.

Section 3 describes how such requirements can be met in the specific case of a *MoK*-coordinated socio-technical KIE, and how *MoK compartments* [11] can be extended to support the notions of *c-env* and *s-env*.

2.2 The *M*olecules *o*f *K*nowledge Model

*M*olecules *o*f *K*nowledge (*MoK*) is a coordination model promoting *self-organisation* of information [11]. Drawing inspiration from *biochemical tuple* *spaces* [15] and *stigmergic coordination* [12], *MoK* pursues two main goals: *(i)* self-aggregation of information into more complex heaps, possibly reifying useful knowledge previously hidden; *(ii)* diffusion of information toward the interested agents, that is, those agents needing it to achieve their goals. The *MoK* model is built around the following abstractions:

seeds — The sources of information. Seeds continuously and spontaneously inject atoms (data chunks) into compartments (tuple-based repositories).

compartments — The repositories of information. Compartments are the computational and topological abstraction of *MoK*, *(i)* defining the notions of *locality* and *neighborhood*, *(ii)* responsible for storing atoms, molecules and enzymes, and *(iii)* in charge of reactions scheduling and execution.

catalysts — The information *prosumers* (consumer + producer). Catalysts are the agents willing to exploit information living within the *MoK* system for their own purposes. As a side effect of their activity, catalysts influence the way in which information spontaneously aggregate and diffuse within compartments – in one word, evolves – driven by *MoK* reactions.

atoms — The atomic unit of information. Continuously injected into compartments by seeds, atoms are subject to *MoK* reactions and agents actions.

molecules — The composite unit of information. Molecules are the reification of similarities between atoms, spontaneously tied together by *MoK* reactions.

enzymes — The reification of catalysts' actions. Enzymes are automatically produced by the compartment within which the action is being done, then exploited by *MoK* reactions to influence information evolution.

reactions — The "laws of nature" driving information evolution. Reactions are the *coordination laws* dictating how information evolves, and how catalysts may influence such process. *MoK* features five reactions[1]:
  - *injection* extracts atoms from seeds and puts them into compartments
  - *aggregation* ties together semantically related atoms into molecules, or molecules into other molecules
  - *diffusion* moves atoms and molecules between neighboring compartments
  - *decay* destroys atoms and molecules as time passes by
  - *reinforcement* consumes enzymes to increase *concentration* of atoms and molecules (relevance w.r.t. others in the same compartment)

A *MoK*-coordinated system is thus a network of *MoK* compartments (tuple-space-like information repositories), in which *MoK* seeds (sources of information) continuously and spontaneously inject *MoK* atoms (information pieces). *MoK* atoms may then aggregate (into molecules, more complex information chunks), diffuse, being reinforced, decay. Such autonomous and decentralised processes are driven by *MoK* reactions (coordination laws) and influenced by *MoK* enzymes (reification of user actions), transparently released by *MoK* catalysts (users, either human or software agents) while performing their activities. *MoK* reactions are scheduled by *MoK* compartments according to Gillespie's *chemical dynamics simulation* algorithm [5], so as to promote chemical-inspired self-organisation based on *locality*, *situatedness*, and *stochasticity* [11].

[1] In [11] reactions were four; injection was added in [10].

3 Towards Anticipatory Coordination

In this section, the *M*olecules *o*f *K*nowledge model is extended toward anticipatory coordination, by borrowing BIC concepts [3]. In particular, Subsection 3.1 extends the notion of compartment according to the definition of smart environments provided in [14], while Subsection 3.2 extends the definition of enzymes and introduces the trace abstraction into the *MoK* model. In addition, Subsection 3.3 proposes a set of actions that catalysts may use to interact with a *MoK* system, along with their impact on enzymes and traces generation.

3.1 *MoK* Compartments as Shared Smart Environments

MoK compartments may play the role of smart environments, since they model locality in *MoK*, and also constitute *MoK* computational environment. *Neighbourhoods* are another topological abstraction, defined in *MoK* as the set of compartments connected by some infrastructural relationship—in the simplest case, physical spatial proximity or direct ("1-hop") network reachability. Since neighbourhoods, too, define a notion of locality and computational environment – being *MoK* diffusion reaction explicitly bound to neighbouring compartments – they can be regarded as smart environments, too. Recursively, the characterisation of smart environment can be extended to the network of compartments—therefore, to the whole *MoK*-coordinated system.

According to [11], the only sort of smart environment enabled in *MoK* is *c-env*, mapped upon a compartment, because: *(i)* a $n : 1$ relationship is assumed between compartments and catalysts—no sharing of working environments is supported; *(ii)* enzymes are visible only to *MoK* reactions; *(iii)* enzymes cannot diffuse, thus neighbourhoods cannot perceive them [11]. So, *MoK* does not support *s-env* since there is no observable action reification in shared environments. Also, support to *c-env* is limited to compartments – not neighbourhoods – since enzymes cannot diffuse. Hence, an extension of the notions of compartment and enzyme is needed to enable *s-env* and improve support to *c-env*:

- each compartment no longer belongs to a single catalyst
- enzymes are: *(i) diversified* to resemble the epistemic nature of the action they reify; *(ii)* made observable to users sharing the compartment they live in; *(iii)* no longer consumed by reinforcement reaction, but now subject to decay; *(iv)* now generating *traces* through a *deposit* reaction
- traces are introduced as the *MoK* abstraction resembling (side) effects of actions; as such, traces are: *(i)* different in kind, according to their father enzyme; *(ii)* observable only by *MoK* reactions; *(iii)* subject to diffusion, decay and to an enzyme-dependant *perturbation reaction*—novel in *MoK*

This enables full support to the notions of *s-env* and *c-env* in *MoK*, and makes it possible to match the three requirements for anticipatory coordination mentioned in Subsection 2.1.

Now, *compartments* represent *s-env*, as the shared working environment where catalysts' actions are made observable to others, and to the environment itself. Also, neighbourhoods represent *c-env*, where action traces may diffuse, becoming part of the environment as they participate $\mathcal{MoK}$ reactions. Observability is now an intrinsic environment property, since compartments enable observability by design. Also, actions may be observed either directly or via their traces (their effects), making it easier to infer goals, as well as to understand how actions affect peers—in particular, when epistemic actions are concerned.

3.2 Enzymes and Traces as BIC Enablers

In [11], traces, along with both perturbation and deposit reactions, are missing, whereas enzymes and reinforcement reaction are formalised as follows:

$$\texttt{enzyme}(mol)_c \qquad \texttt{enzyme}(mol') + mol_c \xrightarrow{r_{reinf}} mol_{c+1}$$

where subscript c denotes concentration, and mol, mol' are supposed to match according to some matching criteria—e.g., Linda matching [4] or OWL subsumption [7]. Traces, perturbation reaction, and deposit reaction are defined below, while enzymes and reinforcement are re-defined accordingly:

$$\begin{gathered}
\texttt{enzyme}(species,\ s,\ mol)_c \\
\texttt{enzyme}(species,\ s,\ mol') + mol_c \xrightarrow{r_{reinf}} \texttt{enzyme}(species,\ s,\ mol') + mol_{c+s} \\
\texttt{trace}(enzyme,\ p,\ mol)_c \\
\texttt{trace}(enzyme,\ p,\ mol') + mol_c \xrightarrow{r_{pert}} \texttt{.}\textit{\texttt{exec}}(p,\ trace,\ mol) \\
enzyme \xrightarrow{r_{dep}} enzyme + \texttt{trace}(enzyme,\ p[species],\ mol)
\end{gathered}$$

where *species* defines the epistemic nature of the action, s is the strength of reinforcement, p is the perturbation the trace wants to perform, and `.exec` starts execution of perturbation p—notice, p is implicitly defined by *species*, as highlighted by notation $p[species]$. Also, decay reaction is extended to enzymes and traces, whereas diffusion to traces solely. Thus, in the extended version of $\mathcal{MoK}$:

- enzymes belong to a certain *species*, reflecting the epistemic nature of actions, and determine the perturbation action performed by generated traces; enzymes also provide a bounded feedback (strength s)
- reinforcement reaction no longer consumes enzymes, which now decay
- traces belong to enzymes—defining (through species) perturbation action p
- perturbation reaction consumes a trace and the related molecule, then triggers execution of the perturbation action
- deposit reaction generates traces from enzymes, without consuming them

The role played by enzymes and traces in anticipatory coordination is then fundamental: they are the abstractions supporting observation of catalysts' actions by both other users and by the environment. In addition, reinforcement and perturbation reactions are the mechanisms enabling mind-reading and signification on the environment side. Reinforcement is meant to influence relevance of the information users manipulate during their workflows, according to the nature and frequency of their actions, so as to better support them in pursuing their goals. Enzymes cannot participate in diffusion reaction because the actions they reify are *situated*, that is, happen at a precise time as well as in a precise space (the compartment). Mind-reading and signification are supported by assuming that users manipulating a given corpus of information are interested in that information more than other. Perturbation is meant to influence relevance, location, content, namely any domain-specific trait of information, in response to users' actions and according to their nature (enzymes' species), with the goal of easing and optimising users' workflows.

Thus, traces are free to wander in the network of $\mathcal{MoK}$ compartments looking for a chance to apply their perturbation action, actually enabling the environment not only to perceive users' action traces, but also to exploit them for the profit of the coordination process—promoting the distributed collective intelligence leading to anticipatory coordination. Mind-reading and signification are supported by assuming that every user action may be interpreted by the environment without the need to directly estimate users' intentions and goals, but inferring them from the characteristics of the business domain within which the $\mathcal{MoK}$-coordinated socio-technical KIE is deployed.

3.3 Tacit Messages to Steer Anticipatory Coordination

Based on a survey of heterogeneous socio-technical KIE – such as Facebook, Twitter, Mendeley and Storify – we devised the most common actions provided to users: here we discuss the BIC *tacit messages* they could convey, and the kind of *perturbation actions* that could be designed accordingly.

Tacit messages are proposed in [3] to describe the kind of messages a practical action (and its traces) may implicitly send to the observers:

1. *presence* — "Agent X is here". Since an action (trace) is observable in shared compartments (neighbourhoods), any agent therein becomes aware of X existence and location—likewise for the environment.
2. *intention* — "X plans to do action b". If the agents' workflow determines that action b follows action a, peers (as well as the environment) observing X doing a may assume X next intention to be "do b". Accordingly, the environment may decide to undertake anticipatory coordination actions easing/hindering action b—e.g. because action b is computationally expensive.
3. *ability* — "X is able to do $a_{i=1,\ldots,n}$". Assuming actions $a_{i=1,\ldots,n} \in A$ have similar pre-conditions, agents (and the environment) observing X doing a_i may infer X is also able to do $a_{j \neq i}$. Accordingly, the environment may further (no longer) support such pre-conditions, enabling (prohibiting) actions $\in A$.
4. *opportunities* — "$[e_1, \ldots, e_n]$ is the set of pre-conditions for doing a". Agents observing X doing a may infer that $[e_1, \ldots, e_n]$ hold, thus, they may take the opportunity to do a immediately. The environment in turn, making similar observations, may act as seen for tacit message 3.
5. *accomplishment* — "X achieved S". If S is the state of affairs reachable as a consequence of doing action a, agents observing X doing a may infer that X is now in state S. Since the environment too can make a similar inference, it may anticipate X next intentions from, e.g., its estimated state S.

6. *goal* — "X has goal g". By observing X doing action a, peers of X may infer X's goal to be g, e.g. because action A is part of a workflow aimed at achieving g—likewise for the environment. Accordingly, the environment may act similarly to what seen for tacit message 2.
7. *result* — "Result r is available". If peer agents know that action a brings result r, whenever agent X does a they can expect result r to be soon available—in case action a completes successfully. The environment in turn, may start planning coordination actions involving result r, e.g., synchronisation of parallel activities for agents waiting for r.

Since agents can undertake the above described inferences, $\mathcal{MoK}$ compartments actually act as BIC-based enablers of distributed collective intelligence phenomena—e.g., anticipatory coordination emerging due to agent interaction.

The above categorisation is general enough to suit several different application domains and practical actions. In the case of socio-technical KIE, we identified a set of fairly-common actions, in spite of the diversity in scope of the software platforms—e.g. Facebook vs. Mendeley:

- *quote/share* — re-publishing or mentioning someone else's information can convey, e.g., tacit messages $1, 3, 5$. If X shares Y's information through action a, every other agent observing a becomes aware of existence and location of both X and Y (1). The fact that X is sharing information I from source S lets X's peers infer X can manipulate S (3). If X shared I with Z, Z may infer, e.g., that X expects Z to somehow use it (5).
- *like/favourite* — marking as relevant a piece of information can convey, e.g., tacit messages $1, 4$. If the socio-technical platform lets X be aware of Y marking information I as relevant, X may infer that Y exists (1). If Y marks as relevant I belonging to X, X may infer that Y is interested in her work, perhaps seeking for collaborations (4).
- *follow* — subscribing for updates regarding some piece of information or some user can convey tacit messages $2, 4$. Since X manifested interest in Y's work through subscription, Y may infer X intention to use it somehow (2). Accordingly, Y may infer the opportunity for, e.g., collaboration (4).
- *search* — performing a search query to retrieve information can convey, e.g., tacit messages $1, 2, 4$—notice however, which assumptions to make about a search action heavily depends on which search criteria are supported. If X search query is observable by peer agents, they can infer X existence and location (1). Also, they can infer X goal to acquire knowledge related to its search query (2). Finally, along the same line, they can take the chance to provide matching information (4).

Accordingly, perturbation actions may range from sending discovery messages informing agents about the presence and location of another (1), to establishing privileged communication channels so as to ease collaborations (4); from undertaking coordination actions enabling/forbidding some interaction protocol (2, 3, 6), to proactively notifying users about availability of novel information (4, 7).

4 Experiment

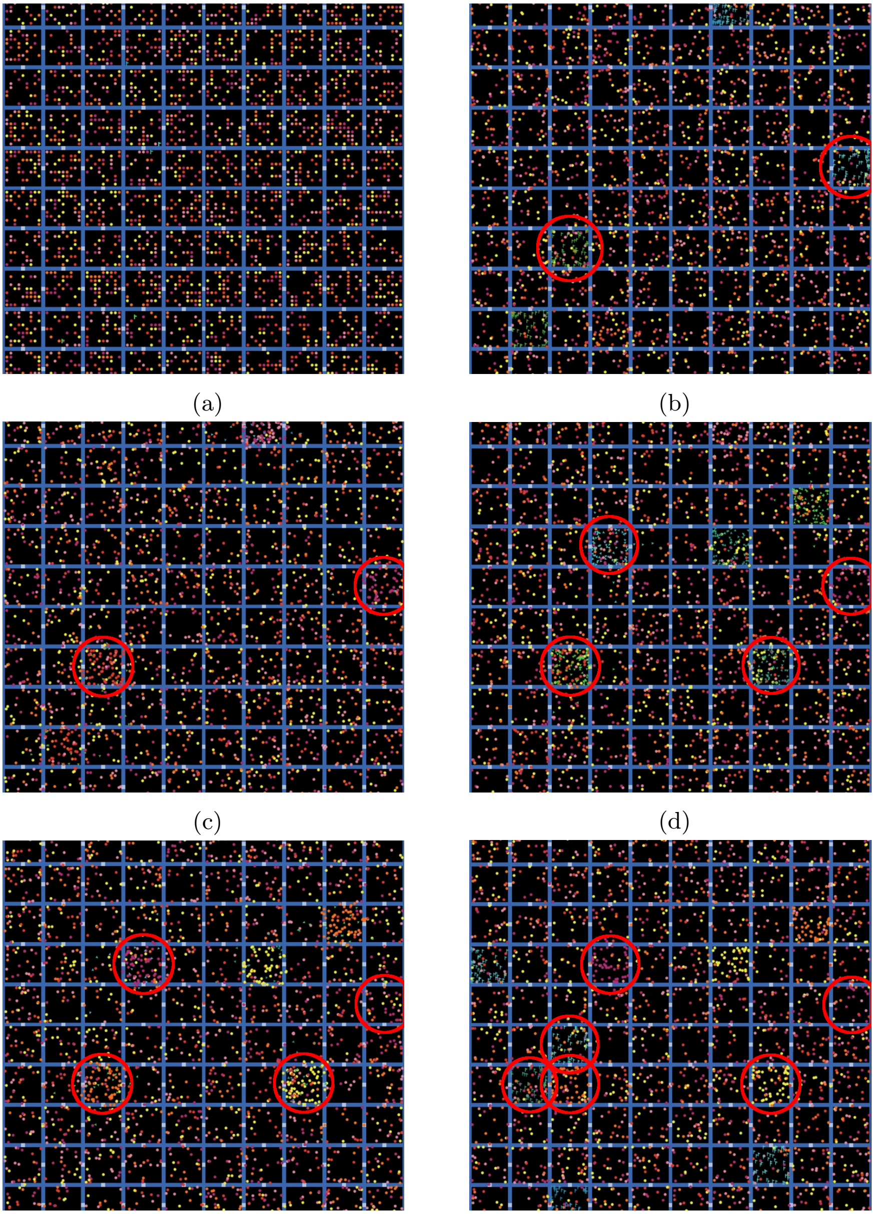
In the following we simulate a citizen journalism scenario, where users share a $\mathcal{MoK}$-coordinated IT platform for retrieving and publishing news stories. Users have personal/shared devices (smartphones, tablets, pcs, workstations) running the $\mathcal{MoK}$ middleware, which they use to search the IT platform for relevant information. Searches can spread up to a logical neighbourhood of the searched compartment – for a number of reasons: limiting bandwidth consumption, boosting security, optimising information location, etc. –, including those of colleagues interested in stories belonging to similar topics. User searches leave traces that the $\mathcal{MoK}$ middleware exploits to attract similar information, actually enacting anticipatory coordination.

Fig. 1, 2a-2b demonstrate how the emergent collective intelligence phenomena enabling anticipatory coordination is effectively supported by suitable BIC-inspired abstractions and mechanisms. The coordination infrastructure does not know in advance the effectiveness of its coordination activities in supporting users' workflows: it can only try to react to users' activities at its best, according to its own interpretation of users' goals. This is exactly what anticipatory coordination is: the infrastructure tries to foresee the user coordination needs even before users do, with the aim of satisfying them at best.

Fig. 1a shows the initial configuration: information molecules (coloured dots) are randomly scattered throughout the grid (black squares)—light-blue little squares represent links between compartments, allowing diffusion. Fig. 1b highlights two compartments in which enzymes (coloured flags) have just been released, thus traces begin to spawn and diffuse (coloured arrows): green enzymes in the bottom-left one, cyan enzymes in the top-right one[2]. Fig. 1c demonstrates that the expected clusters appear: red molecules (brought by green traces' perturbation action) have the highest concentration in the bottom-left (highlighted) compartment, likewise magenta molecules (brought by cyan traces) in the top-right one. Fig. 1d-1f demonstrate that clusters are transient: they last as long as users' action effects (enzymes and traces) last. In fact, besides new clusters appearing (magenta molecules, top-left and yellow molecules, bottom-right), the previous ones either disappear (magenta cluster, top-right) or are replaced (orange cluster, bottom-left). This *adaptiveness* feature is confirmed by Fig. 2a-2b, plotting the oscillatory trend of clustered ("still") molecules and traces. Also, Fig. 1d-1f highlight other desirable features of $\mathcal{MoK}$, stemming from its biochemical inspiration and BIC, respectively: *locality* and *situatedness* (of both computations and interactions). In fact, as neighbouring compartments can influence each other through diffusion, they can also act independently by, e.g., aggregating different molecules.

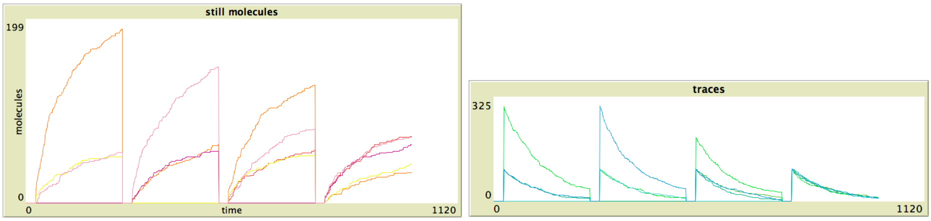
As a last note, we remark how the extended $\mathcal{MoK}$ model deals with the typical issues of socio-technical KIE highlighted in Section 1. In terms of unpredictability,

[2] Colours represent semantic differences for different matches: red molecules match green enzymes/traces, orange molecules match lime enzymes/traces, yellow molecules match turquoise enzymes/traces, magenta molecules match cyan enzymes/traces, pink molecules match sky blue enzymes/traces.



(a) (b) (c) (d) (e) (f)

Fig. 1. *Self-organising, adaptive anticipatory coordination.* Whereas data is initially randomly scattered across workspaces (a), as soon as users interact (b) clusters appear by emergence thanks to *BIC-driven self-organisation* (c, d). Whenever new actions are performed by catalysts, the $\mathcal{M}o\mathcal{K}$ infrastructure *adaptively* re-organises the spatial configuration of molecules (e) so as to better tackle the new coordination needs (f).



(a) (b)

Fig. 2. On the left, concentration of still molecules over time. Still molecules represent molecules currently in the right compartment—the one storing matching enzymes. The oscillatory trend is due to periodic injection of enzymes (thus traces) which clears the "still" state of molecules. The different colours correspond to the different molecules. On the right, concentration of traces over time. Traces move molecules to the right compartment—the one storing matching enzymes. The oscillatory trend is due to decay of traces over time. The different colours correspond to the different traces.

$\mathcal{M}o\mathcal{K}$ anticipates user coordination needs, not based on future behaviour prediction, but rather on present actions and its mind-reading and signification abilities. In terms of scale, $\mathcal{M}o\mathcal{K}$ reactions act only locally, thus self-organisation exploits local information only. In terms of size, $\mathcal{M}o\mathcal{K}$ decay helps mitigating the issue by destroying information and meta-data as time passes; furthermore, the overhead brought by $\mathcal{M}o\mathcal{K}$ BIC-based extension is minimal, since it exploits only information already in the system. In terms of pace, whereas reactions execution and BIC-related mechanisms are rather efficient – mostly due to their local nature – there is a fundamental issue still to be addressed in $\mathcal{M}o\mathcal{K}$: the semantic similarity measure ($\mathcal{F}_{\mathcal{M}o\mathcal{K}}$ in [11]). On the one hand, an accurate measure likely leads to more meaningful clusters; on the other hand, it often requires more expensive computations. Thus, a tradeoff is needed—our efforts for further developments of $\mathcal{M}o\mathcal{K}$ are also devoted to investigate this issue.

Technical details of the experiment are as follows[3]: 100 $\mathcal{M}o\mathcal{K}$ compartments are networked in a grid (4 neighbours per compartment, except border)—see Fig. 1; 2500 molecules, split in 5 non-overlapping semantic categories (representing matching with different enzymes), are uniformly sampled then randomly scattered in the grid—statistically, 500 molecules per category; 250 enzymes, split in the same categories, are generated in 5 random compartments; enzymes' categories are uniformly sampled in batches consisting of 50 enzymes each, so that generated enzymes of a given category are always multiple of 50; enzymes are generated *periodically* (every 250 time steps) and subject to decay; 2 traces

[3] Simulation tool used is NetLogo 5.0.5, available from http://ccl.northwestern.edu/netlogo/. Videos of the simulations are available on YouTube (https://youtu.be/8ibkXdukTfk). Source code of the simulations are to be released as a NetLogo model, available from http://ccl.northwestern.edu/netlogo/models/community/.

per enzyme are generated, coherently with enzymes' category and according to the same time interval; traces too are subject to decay, although at a lower rate w.r.t. enzymes—due to their different purpose: representing *long-term effects* of actions for the former, reifying *situated* actions for the latter.

The simulations proceed as follows: molecules randomly diffuse among neighbouring compartments; enzymes reify a search action which successfully collects a set of molecules from the local compartment; enzymes stand still in the compartment where the action took place until decay, generating traces; traces, representing tacit message 2, randomly diffuse among neighbouring compartments until either (i) decay or (ii) find a matching molecule to apply their perturbation action to; the perturbation action makes the involved molecule diffuse toward the compartment where the trace's father enzyme belong.

5 Conclusion and Further Work

In this paper we propose a novel approach to coordination in socio-technical KIE. In particular, we extend the Molecules of Knowledge model [11] to support the notion of *anticipatory coordination* [3]. To this end, concepts from the cognitive theory of BIC are brought within the MoK model both by extending existing abstractions – compartments and enzymes – and by introducing new abstractions and mechanisms—traces, deposit reaction and perturbation action. To evaluate our proposal, we simulate how to obtain intelligent spatial distribution of information with MoK, based solely on user interaction, as an example of distributed collective intelligence—in particular, anticipatory coordination.

Although our experiment focusses on one specific pattern of anticipatory coordination, we believe that the results achieved are more than encouraging, thus deserve further investigation. In particular, simulations of other MoK behaviours – e.g., re-arrange the network of compartments so as to reflect the current collaborations among catalysts – are actually in progress, and will help further validating both the extended MoK model, and the practice of applying BIC theory to coordination in socio-technical KIE.

Furthermore, our efforts are currently devoted to fully implement and run a MoK coordinated system on a large-scale scenario—e.g. the one here simulated. In fact, although a prototype implementation of MoK exists, such a large-scale deployment has not been achieved, yet. As far as implementation is concerned, special care will be paid to the semantic similarity measure. In our experience, ontology-based semantic matching is rather unfeasible, except for basic relationships only, e.g., subsumption alone. On the contrary, purely syntactical matching has too low expressiveness. Viable tradeoffs may be usage of wildcards, e.g. as in Java *regular expressions*[4], or of synonymy relationships only (hyperonymy for "is-a" relationships, meronymy fort "part-of", etc.), e.g., as done in [13] using WordNet[5].

[4] http://docs.oracle.com/javase/tutorial/essential/regex/
[5] http://wordnet.princeton.edu

References

1. Bhatt, G.D.: Knowledge management in organizations: Examining the interaction between technologies, techniques, and people. Journal of Knowledge Management **5**(1), 68–75 (2001)
2. Castelfranchi, C.: Modelling social action for AI agents. Artificial Intelligence **103**(1–2), 157–182 (1998)
3. Castelfranchi, C., Pezzullo, G., Tummolini, L.: 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 (2010)
4. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems **7**(1), 80–112 (1985)
5. Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry **81**(25), 2340–2361 (1977)
6. Grassé, P.P.: 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 (1959)
7. Horrocks, I.: OWL: a description logic based ontology language. In: Gabbrielli, M., Gupta, G. (eds.) ICLP 2005. LNCS, vol. 3668, pp. 1–4. Springer, Heidelberg (2005)
8. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Computing Surveys **26**(1), 87–119 (1994)
9. Mamei, M., Zambonelli, F.: Programming pervasive and mobile computing applications: The TOTA approach. ACM Transactions on Software Engineering and Methodology (TOSEM) **18**(4), July 2009
10. Mariani, S.: Parameter engineering vs. parameter tuning: the case of biochemical coordination in MoK. In: Baldoni, M., Baroglio, C., Bergenti, F., Garro, A. (eds.) CEUR Workshop Proceedings of the From Objects to Agents, vol. 1099, pp. 16–23. Sun SITE Central Europe, RWTH Aachen University, Turin, December 2–3, 2013
11. Mariani, S., Omicini, A.: Molecules of knowledge: self-organisation in knowledge-intensive environments. In: Fortino, G., Badica, C., Malgeri, M., Unland, R. (eds.) IDC 2012. SCI, vol. 446, pp. 17–22. Springer, Heidelberg (2012)
12. Van Dyke Parunak, H.: A survey of environments and mechanisms for human-human stigmergy. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2005. LNCS (LNAI), vol. 3830, pp. 163–186. Springer, Heidelberg (2006)
13. Pianini, D., Virruso, S., Menezes, R., Omicini, A., Viroli, M.: Self organization in coordination systems using a WordNet-based ontology. In: Gupta, I., Hassas, S., Jerome, R. (eds.) 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2010), pp. 114–123. IEEE CS, Budapest (2010)
14. Tummolini, L., Castelfranchi, C., Ricci, A., Viroli, M., Omicini, A.: "Exhibitionists" and "Voyeurs" do it better: a shared environment for flexible coordination with tacit messages. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2004. LNCS (LNAI), vol. 3374, pp. 215–231. Springer, Heidelberg (2005)
15. Viroli, M., Casadei, M.: Biochemical tuple spaces for self-organising coordination. In: Field, J., Vasconcelos, V.T. (eds.) COORDINATION 2009. LNCS, vol. 5521, pp. 143–162. Springer, Heidelberg (2009)

16. Viroli, M., Pianini, D., Beal, J.: Linda in space-time: an adaptive coordination model for mobile ad-hoc environments. In: Sirjani, M. (ed.) COORDINATION 2012. LNCS, vol. 7274, pp. 212–229. Springer, Heidelberg (2012)
17. Weyns, D., Omicini, A., Odell, J.J.: Environment as a first-class abstraction in multi-agent systems. Autonomous Agents and Multi-Agent Systems **14**(1), 5–30 (2007)
18. Whitworth, B.: Socio-technical systems. Encyclopedia of human computer interaction, 533–541 (2006)
19. 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., Wally, B.: Self-aware pervasive service ecosystems. Procedia Computer Science **7**, 197–199 (2011)