

Review Article

Nature-Inspired Coordination Models: Current Status and Future Trends

Andrea Omicini

Dipartimento di Informatica–Scienza e Ingegneria (DISI), Alma Mater Studiorum–Università di Bologna, 40126 Bologna, Italy

Correspondence should be addressed to Andrea Omicini; andrea.omicini@unibo.it

Received 15 October 2012; Accepted 20 November 2012

Academic Editors: X. Liu, Y. Malaiya, M. Meringer, and Z. Shen

Copyright © 2013 Andrea Omicini. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Coordination models and languages are meant to provide abstractions and mechanisms to harness the space of *interaction* as one of the foremost sources of complexity in computational systems. Nature-inspired computing aims at understanding the mechanisms and patterns of complex natural systems in order to bring their most desirable features to computational systems. Thus, the promise of *nature-inspired coordination models* is to prove themselves fundamental in the design of complex computational systems—such as intelligent, knowledge-intensive, pervasive, adaptive, and self-organising ones. In this paper, we survey the most relevant nature-inspired coordination models in the literature, focussing in particular on tuple-based models, and foresee the most interesting research trends in the field.

1. Motivation

1.1. Why Nature-Inspired Models? Many of the most characteristic human artefacts are *not* nature inspired: neither wheels nor computers, for instance, to quote some of the tools that mostly influenced the evolution of humankind, could be easily mapped upon any preexisting natural item. So, the first question to be answered here is why should we care at all about *nature-inspired models* when building artificial systems, and in particular *computational systems*, where natural inspiration seems so far and remote?

Nature-inspired computing provides a simple yet convincing answer: natural systems are good in dealing with *complexity* [1]. We are aware of many sorts of complex natural systems—such as physical, chemical, biochemical, biological, and ethological systems—featuring many properties that we would really like to bring to our computational systems—such as robustness, fault tolerance, self-repair, and the like. These are the features of complex natural system that we would like to understand, capture, then bring to computational systems—and, in the very end, what makes computer scientists and engineers look for natural inspiration. For instance, a quick look to research activities in the field of nature-inspired computing reveals the two main properties that nature-inspired computational systems

should feature: *autonomy*, at the component level, and *self-organisation*, at the system level [2].

In short, the main goals of nature-inspired computing could be roughly summarised as follows:

- (i) to devise out the sources of complexity in natural systems;
- (ii) to understand the mechanisms and patterns that natural systems exploit in order to successfully deal with issues such as openness, distribution, large scale, and unpredictability of environment;
- (iii) to map such mechanisms and patterns upon suitable computational equivalents;
- (iv) to make them work within computational systems, so as to provide them with the desirable features of natural systems.

1.2. Why Coordination Models? Suitably expressive models, technologies, and methodologies are required to allow programmers and designers to dominate the complexity of today computational systems—such as intelligent, knowledge-intensive, pervasive, and self-organising systems. Systems of those sorts could be typically seen as (dynamic) ensembles of a large (possibly huge) number of distributed components,

heterogeneous in nature, structure, and behaviour, which are put together *somehow* so as to build up a coherent system behaviour. Roughly speaking, that “somehow” is typically one of the key issues in the engineering of complex systems—as well as the main concern of research on *coordination models and languages* [3–8].

In fact, most of the complexity in computational systems comes from *interaction* [9, 10]—along with an essential part of their expressive power [11]. Given that *coordination* [6] has been defined as the science of *managing the space of interaction* [11], *coordination models and languages* [12] are meant to provide the basic abstractions and technologies for dealing with the intricacies of system interaction [13]. This is why, in spite of their origin in the context of closed and parallel systems [14, 15], coordination models and languages should be expected to play a key role in the engineering of complex systems [16].

This holds in particular for *tuple-based coordination models* [17]—as discussed later in Section 3. Their intrinsic properties—as derived from their common ancestor LINDA [18]—along with the most recent developments and extensions, apparently make them suitable to work as the sources for the abstractions and mechanisms around which complex systems could be designed and built.

1.3. Why Nature-Inspired Coordination? Ranging from physics to biology [19] and from economics to sociology and organisation sciences [20], modelling the dynamics of interaction is essential to understand the complexity of systems of any sort [10]—not just of computational ones. This might be seen as a possible reason why the issue of coordination did not emerge first in computational systems. For instance, in [21], Grassé noted that in termite societies

The coordination of tasks and the regulation of constructions are not directly dependent from the workers, but from constructions themselves.

From the very beginning, then, the study of complex natural systems has been intertwined with the study of coordination. Further works on other social insects—such as ants [22]—show that coordination is a key issue in complex natural systems [10]. Many well-known examples of natural systems—and, more generally, of complex systems—seemingly rely on simple yet powerful *coordination mechanisms* for their key features—such as adaptiveness and self-organisation [23]. For instance, termites and ants construct very complex nests, whose building criteria are anything but trivial—such as inner temperature, humidity, and oxygen concentration, through a coordinated, robust, adaptive, and self-organised social activity.

Altogether, it definitely makes sense to focus on *nature-inspired coordination models* as they have the potential to work as the core of complex nature-inspired computational systems. Accordingly, in this paper, we first survey the most interesting nature-inspired coordination language and models, focussing in particular on tuple-based models as the most prominent class of coordination models. Then, we attempt to point out the hottest research trends in the field and discuss their possible goals and future evolutions.

2. Remarkable Examples and Main Issues

As suggested previously, nature-inspired coordination models have a history behind them, yet. So, before discussing their potential and perspectives, it is worthwhile to shortly discuss the most remarkable examples in the field and to point out the basic issues of nature-inspired coordination.

2.1. Early Models of Nature-Inspired Coordination

2.1.1. Stigmergy. Nature-inspired models of coordination are grounded on studies on the behaviour of social insects, like ants or termites. In [21], Grassé introduced the notion of *stigmergy* as the fundamental coordination mechanism in termite societies, pointing out the role of environmental artefacts in their individual behaviour and overall organisation. The most widely known and studied example of stigmergic coordination in insect societies is probably that of ant colonies [24]. In short, pheromones are released in the environment by the ants that find food on their way back to the nest, thus building pheromone trails towards food that other ants are then stimulated to follow. There, pheromones could be seen as acting as environment markers for specific social activities, driving both the *individual* and the *social* behaviours of ants.

Nowadays, the term “stigmergy” generally refers to a set of nature-inspired coordination mechanisms mediated by the *environment* [25, 26]. Once a notion of *computational environment* is defined [27–29], the reification of the actions of coordinated entities within the hosting environment works as the basis for stigmergy in software systems. So, *digital pheromones* [30] and other *signs* left and sensed in a shared environment can be exploited for the engineering of adaptive and self-organising computational systems based on stigmergy [31].

2.1.2. Chemical Coordination. Besides insect societies, another early source of inspiration for coordination model was provided by *chemistry*. Chemical reactions, in fact, could be seen as (relatively) simple laws that regulate the evolution of quite complex physical phenomena, coordinating the behaviours of a huge amount of components, as well as the global evolution of complex systems of many sorts—such as biological organisms and meteorological systems, for instance.

Gamma [32] was the first and—for more than a decade—the most prominent example of a *chemistry-inspired coordination* model. There, coordination is conceived as the evolution of a space governed by chemical-like rules, globally working as a rewriting system [33]. A step further was taken by the CHAM (chemical abstract machine) model [34], whose states are interpreted as chemical solutions where floating molecules representing coordinated entities can interact according to some reaction rules, and where a notion of *membrane* was introduced in order to model locality of reaction execution.

However, despite their promises, neither Gamma nor CHAM was ever able to demonstrate their potential in the coordination of complex software systems. More recent

examples of nature-inspired models proved to be more effective under many aspects—as discussed in the next subsection.

2.2. Recent Models of Nature-Inspired Coordination

2.2.1. Field-Based Coordination. More recently, new models of coordination were proposed, which are inspired by the way masses and particles in our universe move and self-organise according to environmental properties represented by gravitational and electromagnetic fields.

In the so-called *field-based* coordination models [35], computational force fields—generated either by the coordinated components or by the coordination middleware—propagate across the environment and drive the actions and motion of the component themselves. For instance, Co-fields [36] exploit computational fields to coordinate collective intelligent behaviours, such as the many form of *swarm intelligence* [37].

In turn, TOTA [38, 39] provides an interesting blend of stigmergy and field-based coordination, being first of all a stigmergic coordination model, whose middleware is adopted for supporting and implementing the Co-fields model [35]—for instance, local coordination abstractions provided by the TOTA middleware are used to build local Co-fields computational fields.

2.2.2. Biochemical Coordination. The “chemical nature” of Gamma coordination laws is indeed quite limited: while their structure reminds of chemical reaction, their behaviour is essentially that of a mere rewriting system, indeed far from resembling real chemical laws.

Instead, *chemical tuple spaces* [40] exploit the chemical metaphor up to its full extent, by capturing most of the essential properties of chemical processes. There, data, devices, and software agents are uniformly represented in the form of chemical reactants, and system behaviour is expressed by means of full-fledged chemical-like laws—which are actually *time dependent* and *stochastic*—embedded within the coordination media. Chemical tuple spaces could be then exploited to implement probabilistic coordination laws, thus to promote stochastic behaviour in coordinated systems.

More generally, *biochemical tuple spaces* [41] enhance chemical tuple spaces by adding the topological dimension to coordination. Along with *stochastic behaviour* of coordination primitives, in fact, they introduce *compartments* and *diffusion* as first-class notions for coordination, along with a notion of *neighbourhood*. On the one hand, compartments make it possible to structure the coordination topology, by making local spaces available for the execution of chemical reactions. On the other hand, diffusion represents a basic mechanism for computational mobility within a structured environment, allowing for motion coordination.

2.3. Basic Issues of Nature-Inspired Coordination. The aforementioned examples, even though they possibly do not cover the full variegated spectrum of nature-inspired coordination models [42], they are indeed representative enough to

allow us to point out a couple of basic issues. Such issues, which implicitly define a minimal set of essential properties that a coordination model aimed at addressing complexity should anyway feature, are (i) the role of the *environment* in coordination and (ii) the *stochastic behaviour* of coordinated systems.

2.3.1. Environment. In stigmergic coordination, environment plays a fundamental role: in termite societies, through nest constructions, and in ant colonies, by collecting pheromones, making their scent diffuse, and letting them decay and evaporate in the end. The role of the environment was well emphasised by Resnick in [43]:

The hills are alive.

The environment is an active process that impacts the behaviour of the system, not just a passive communication channel between agents.

Correspondingly, environment is essential in nature-inspired coordination and features some fundamental properties [44]:

- (i) it works as a *mediator* for component interaction—the components of a distributed system can communicate and coordinate *indirectly* through the environment;
- (ii) it is *active*—the environment features autonomous *dynamics* and affecting component coordination;
- (iii) it has a *structure*—the environment provides a notion of *locality*, of *neighbourhood*, and allows components of any sort to *move* through a *topology* of some sort.

Accordingly, any nature-inspired coordination model should provide some suitably expressive abstractions to model and build the environment and its dynamics.

2.3.2. Stochastic Behaviour. Complex systems typically exhibit stochastic behaviours. Ants follow pheromone scent according to some (high) probability values; chemical reactions occur according to some probabilistic rates. The main point here is that such stochastic behaviours are *essential* to the overall system evolution. Thus, any model aimed at modelling and engineering complex computational systems should feature some *probabilistic* mechanisms—see for instance [45–47]—based on which stochastic behaviours could be designed and implemented [48].

While some coordination models intrinsically feature some *don't know/don't care nondeterministic* mechanisms, it should be noted that they are not expressive enough to capture all the properties of complex systems, such as biochemical and social systems [49]: for instance, if ants would choose whether to follow the pheromone trail or not just nondeterministically, with no probabilistic distribution, ant colonies would rapidly cease to exist as we know them. Correspondingly, full-fledged probabilistic mechanisms are required to really capture the dynamics of coordination in nature-inspired systems: coordination models should feature (possibly simple yet) expressive mechanisms to provide coordinated systems with stochastic behaviours [48].

3. Tuple-Based Nature-Inspired Models

3.1. Interaction and Coordination. While a number of different notions of coordination have been introduced in the literature [50, 51], *harnessing the complexity of interaction* [11] is surely a common goal of most of the approaches. As such, coordination is thus often defined as an independent dimension with respect to computation [11, 18], dealing with interaction as an *independent design dimension* [52], which could be exploited as the source of collective *social intelligence* [37, 53].

Also, many different definitions of coordination models have been proposed, such as:

A coordination model is the glue that binds separate activities into an ensemble [18]

A coordination model provides a framework in which the interaction of active and independent entities called agents can be expressed

A coordination model should cover the issues of creation and destruction of agents, communication among agents, and spatial distribution of agents, as well as synchronization and distribution of their actions over time [12].

According to the simple metamodel proposed in [12], a coordination model defines:

- (i) the *coordination entities*—whose mutual interaction is ruled by the model, also called the *coordinables*;
- (ii) the *coordination media*—that is, the abstractions enabling and ruling agent interactions;
- (iii) the *coordination laws*—that is, the rules that govern the space of interaction—ruling the observable behaviour of coordinables and the computational behaviour of coordination media as well.

3.2. Why LINDA and Tuple-Based Models? LINDA [14] is the ancestor of all *tuple-based coordination models* [17], which represent the most prominent and widespread coordination models nowadays, with dozens of different implementations from both academia and industry—such as T Spaces [54, 55], JavaSpaces [56], TuCSon [57, 58], and GigaSpaces [59]. There, *coordinables* synchronise, cooperate, and compete:

- (i) based on *tuples*, representing information chunks,
- (ii) available in the *tuple spaces*, working as the *coordination media*,
- (iii) by *associatively* accessing, consuming, and producing tuples.

The same holds for any tuple-based coordination model: all the many extensions, variations, and revisions of LINDA differ under many aspects, but could all be described as featuring the previous abstractions and mechanisms at their core.

The point is that LINDA is definitely *not* a nature-inspired model [60]. So, the main issue to be clarified here, before we

proceed, is why should tuple-based models work for nature-inspired coordination?

An interesting survey of the technologies and platforms for tuple-based coordination can be found in [61], where the authors analyse the ability of tuple-based systems to adapt to the predictable and unpredictable changes in distributed environments and focus on several aspects related to adaptiveness. In the following, instead, we take into account the intrinsic features of tuple-based models that make them suitable, in principle, for nature-inspired coordination.

3.2.1. Expressiveness of the Model. The first motivation lays in the *expressiveness* of tuple-based models. Since its inception [14], LINDA was conceived as a *core* coordination model, where most of the typical problems of concurrent systems could be (quite) easily faced and solved. Also, a remarkable amount of the fascination of the LINDA coordination language—mostly responsible of its early and widespread acknowledgement—derives from its apparent ability to provide *few, simple* primitives, based on which many *complex* coordination problems can be expressed and solved.

So, tuple-based languages are highly-expressive according to most of the possible acceptations of the term “expressiveness” [62–64]. Testing their expressiveness to model and engineer nature-inspired systems looks in principle mostly reasonable, if not obvious.

3.2.2. Extensibility of the Language. Whatever its expressiveness, LINDA was initially conceived as a coordination model for closed parallel systems [15]. Almost inevitably, some of the relevant issues of today open concurrent systems cannot be easily addressed with LINDA for either theoretical or practical reasons—see for instance [65, 66]. As a result, since its early days, LINDA has been awarded of an almost continuous flow of revisions and extensions—from new implementations [67] to new models [17], which, besides generating a plethora of tuple-based coordination models and languages, witness the *extensibility* of the basic tuple-based coordination model.

Extending tuple-based models toward nature-inspired coordination is apparently quite a sensible approach, which already produced an interesting stream of research results [16]. Among the many other, it is worth noticing that many of the examples already cited in Section 2 are in fact tuple based—such as digital pheromones [30] and the like [31], Co-fields [36], TOTA [38, 39], and chemical [40] and biochemical tuple spaces [41].

3.2.3. Environment-Based Coordination. One of the most essential features of LINDA—as well as one of its main original motivations—is the *generative communication*. Generative communication simply means that tuples—the object of communication as well as the means for coordination—are *generated* by a coordinable—in the tuple space, by means of a coordination primitive—to have a life cycle which is independent of the life of the generating coordinable itself [14]. Besides temporal uncoupling properties, which are essential in a distributed setting, generative communication mandates for *persistent* coordination abstractions,

independent of the coordinated entities. So, tuple spaces in tuple-based coordination models are provided as persistent *coordination services* [68] by the coordination middleware, shaping the computational *environment* around coordination entities [69]. As such, they can be interpreted as *coordination artefacts* [70], that is, environmental abstractions embodying coordinative behaviours which can be exploited to support *environment-based coordination* [71].

Given the essential role that environment-based coordination plays in the context of nature-inspired models (Section 2.3), this is another reason why tuple-based models look fit to support nature-inspired coordination.

3.3. Beyond Nature-Inspired Tuple-Based Models. So, while LINDA is *not* a nature-inspired model, many of its extensions actually *are*. Besides the already mentioned approaches—such as stigmergy [31], field-based [38], and chemical [40] and biochemical [41] coordination—, some further examples of nature-inspired tuple-based models are worth mentioning here—as examples of coordination models extending their reach beyond the mere scope of natural inspiration.

3.3.1. From Distributed Cognition to Cognitive Stigmergy. The realisation that humans regularly use environmentally mediated signals to coordinate their activities [31] is among the main issues of distributed cognition [72]. In particular, the observation that work environments are typically rich of information associated to resources—such as labels, signs, visual alerts, and the like—which humans exploit in order to organise and coordinate their individual and social activities within a shared environment, is what led to the recognition of environment-mediated, stigmergic coordination as a typical phenomenon in human societies and organisations [31, 44]. There, typically, modifications to the environment are often amenable of an interpretation in the context of a shared conventional system of signs, and the interacting entities feature cognitive abilities that could be used in the stigmergy-based interaction and possibly affect its outcome.

The notion of *cognitive stigmergy* mostly derives from the previous considerations: when signals (e.g., pheromones) are read as *signs* and given a symbolic interpretation by intelligent agents being either human or software agents, stigmergic coordination becomes a multiple-level coordination between heterogeneous components. There, in fact, ordinary components perceive environment markers as mere signals and react accordingly, whereas intelligent components can read them as signs and behave according to their symbolic interpretation. So, cognitive stigmergy allows both reactive and intelligent components to fruitfully coexist within emergently coordinated activities, even though with different levels of understanding of the coordinating environment [73].

In terms of tuple-based coordination, cognitive stigmergy could be simply achieved by using tuple spaces as environmental abstractions, adopting a tuple language, providing the suitable symbolic level, and exploiting tuples as environment markers amenable of a two-level interpretation—as signals and as signs. In [44], a possible architecture based on the tuple-based TuCSon middleware [57, 58] is presented,

where tuples are *logic tuples* [74, 75] that reactive agents just perceive through standard LINDA primitives, whereas intelligent agents can also interpret them at symbolic level and act accordingly.

3.3.2. Pervasive Ecosystems for Adaptive Services. In the last years, the emergence of an ever-growing number of pervasive computing technologies—mostly caused by the easy accessibility of distributed network infrastructures—led to the definition of the notion of *pervasive ecosystem* [76]. A pervasive ecosystem is a distributed computational structure intertwined with an everyday environment and modelled after a real-life natural ecosystem—representing its natural inspiration. In the same way as in natural ecosystems, computational ecosystems have components and actors that abide by the laws of the system and act autonomously to achieve some goals.

The SAPERE model [77, 78] deals with coordination within pervasive ecosystems [79, 80]. There, heterogeneous components of any sorts are ruled by *ecolaws*, inspired by biochemical reaction laws. Each entity, whatever its structure, is uniformly represented in terms of *Live Semantic Annotations* (LSA), representing an up-to-date state of the entity itself, related to the specific context in which the information is produced and used. LSAs are sort of tuples, which are stored locally in every node of the infrastructure representing a virtual counterpart of the ecosystem. The abstraction where LSAs are reified is called *LSA-space*—a sort of tuple space, representing the context where entities live and act. To promote context awareness, the SAPERE model provides *LSA-bonding*, a mechanism allowing an LSA to link to other LSA in the same space, and SAPERE entities to inspect—via bonds—the state of their peers, of the overall context, and to act accordingly.

Coordination rules—the *ecolaws*—are instead in charge of managing the global behaviour of the whole system, by manipulating—deleting, updating, moving, and bonding—LSA in the system, in the way of chemical reactions. Like chemical reactions, *ecolaws* act following specific stochastic rates. Using the chemical approach, *ecolaws* can put into effect processes such as LSA diffusion, interaction, composition, and disposal.

3.4. Toward Self-Organising Coordination. According to [2], the essential system behaviour in nature-inspired computational systems is represented by *self-organisation*. In short, self-organisation generally refers to the internal process leading to an increasing level of organisation—where *organisation* stands for relations between parts in term of structure and interactions, whereas *self* means that the driving force must be internal and distributed among components. For instance, [23] proposes the following definition of self-organisation.

Self-organisation is a process in which pattern at the global level of a system emerges solely from numerous interactions among the lower-level components of the system. Moreover, the rules specifying interactions among the system's

components are executed using only local information, without reference to the global pattern.

Also, the best-known models for nature-inspired coordination—such as stigmergy and field-based ones—are self-organising ones. Correspondingly, research on *self-organising coordination* [19, 20, 47, 48] is meant at devising the fundamental features that a coordination model should feature in order to support self-organising system behaviours. In [48], self-organising coordination is defined as follows.

Self-organising coordination is the management of system interactions featuring self-organising properties, namely, where interactions are local, and global desired effects of coordination appear by emergence.

Most of the coordination models originated in the field of distributed systems—and in particular tuple-based ones—feature abstractions enacting coordination laws that are typically *reactive*, (mostly) *deterministic*, and *global* as well. Instead, complex systems featuring self-* properties typically exhibit coordination patterns appearing at the global level by *emergence*, from *probabilistic*, *time-dependent* coordination laws based on *local* criteria [48]. This is why a number of coordination models choose instead to either implicitly or explicitly recognise that full expressiveness requires addressing the issues of *time dependency* and *stochastic behaviour*.

Time dependency means that the laws of coordination should possibly be time dependent [81]—or, more generally, *time aware*. For instance, ReSpecT [82] generally addresses time dependency by capturing time events, providing time-aware coordination abstractions, and supporting the definition and enforcement of *timed coordination policies* [83]—so, ReSpecT-programmed tuple centres can work as time-dependent abstractions for the coordination of distributed processes [84].

Stochastic behaviour, too, was a concern for a number of tuple-based coordination models. STOKLAIM [85], for instance, adds distribution rates to coordination primitives—thus making it possible to model nondeterministic real-life phenomena such as failure rates and interarrival times. Also, Swarm-Linda [53] enhances LINDA implementation with swarm intelligence to achieve features such as scalability, adaptiveness, and fault tolerance—by modelling tuple templates as ants, which feature probabilistic behaviour when looking for matching tuples in a distributed setting. Finally, STOPKLAIM [81] integrates a probabilistic version of LINDA—PLINDA [86]—with STOKLAIM, thus featuring a probabilistic and time-stochastic coordination model.

In the overall, however, while they provide some of the basic mechanisms to implement the typical features of self-organising coordinated systems, the aforementioned models either do not capture all the essential features of nature-inspired coordination models, or fail to properly answer to the need of today complex software systems. This is why many of the novel research lines on nature-inspired coordination stretch existing tuple-based models to achieve the expressive power required to model and build complex self-organising natural systems [16]. Along this line, in the following section

we describe some of the hottest trends in the field and provide some vision over the future evolution of the field.

4. Trends in Nature-Inspired Coordination

Nature-inspired systems represent one of the most prominent topics in the research on coordination models and technologies, nowadays [16]. This is why several promising trends in the field have emerged in the last years, which are likely to lead to relevant results in the next decade.

4.1. Expressing the Full Dynamics of Natural Systems. Capturing just *some* of the principles and mechanisms of natural systems does not ensure to catch their *essence*, whatever this might be. Extracting patterns for coordination [87] or self-organisation [88] is a fundamental step towards the understanding of the basic mechanisms underlying self-organising coordination; however, misinterpreted patterns obviously lead to unsatisfactory results. For instance, mapping pheromones over tuples for tuple-based stigmergic coordination may not be enough to capture the behaviour of ant-based systems—if not coupled with suitable scent diffusion and decay mechanisms for tuples, as well as with some stochastic behaviour for ant motion [53]. Instead, fundamental properties such as adaptiveness of systems can be achieved by fully capturing the pheromone-environment-ant scenario, as discussed in [61] for SwarmLinda [53].

Correspondingly, a fundamental trend in nature-inspired coordination is understanding and capturing *all* the mechanisms and patterns that *altogether* determine the features of natural systems we are interested in. As already mentioned in this paper, this is exactly what happens in the case of chemical coordination models. On one hand, the original models—namely, Gamma and CHAM—exploit the raw schema of computation as a chemical reaction, but are *not* expressive enough to fully reproduce any nontrivial chemical system. In fact, for example, notions such as reaction rate, concentration, and probabilistic execution are required to reproduce even the simplest model of real chemical reactions: given that neither Gamma nor CHAM provide for such notions, they are not expressive enough to fully match the behaviour of real chemical systems. Instead, (bio)chemical tuple spaces, as defined in [40, 41], fully exploit the chemical metaphor by providing time-dependent and stochastic chemical laws; as a result, they are perfectly capable of fully reproducing the dynamics of a real chemical reaction and also of putting it to use in the coordination of complex adaptive pervasive systems [89].

This is also why probabilistic extensions to classical coordination models are going to be more and more relevant, following already existing examples such as the aforementioned STOKLAIM [85, 90]—a *stochastic* extension of the KLAIM model for mobile coordination [91]—as well as the many probabilistic extensions of LINDA [92]—among which PLINDA [86] and Probabilistic KLAIM [93]. Formally, this will also require probabilistic formal models for the specification of the semantics of probabilistic coordination models,

possibly by combining already existing approaches—such as [45, 46, 49, 90]—with newly developed ones.

4.2. Blending Patterns and Metaphors. According to [94], “there are many possible nature-inspired metaphors that can be adopted, and choosing one may require a careful analysis of the pros and cons of the different metaphors.” There, the source of inspirations are classified according to four possible sorts—physical, chemical, biological, and social—and the general features of each one are discussed and compared. Among the drawn conclusions, what is relevant here is the fact that there are apparently no single general purpose solutions addressing all the potential issues of nature-inspired computing—not a single source of inspiration could help facing all the problems and providing all the desired properties.

Along this line, *mixing* abstractions and mechanisms coming from different conceptual sources is apparently a conspicuous approach when trying to go beyond toy-like “paper-oriented” case studies. This is for instance the approach taken by the Co-fields model [36], where the physical inspiration of force fields—used as computational fields to coordinate collective spatial motion—is combined with stigmergic notions such as diffusion and decay—based on the TOTA middleware used for the implementation of Co-fields [35].

More notably, this is also the motivation behind the SAPERE coordination model for pervasive service ecosystems [77, 78]. SAPERE exploits

- (i) the *chemical* metaphor for driving the evolution of coordination abstractions, where *ecolaws* are conceived as chemical-like laws;
- (ii) *biochemical* abstractions for topology and diffusion;
- (iii) the notion of *ecosystem*—typical of *social* systems—in order to model the overall system structure and dynamics.

Altogether, the main point here is to be able to blend all the diverse metaphors and mechanisms in a coherent conceptual and technical framework satisfying also the basic software engineering principle of *conceptual integrity*. While the fact that the examples discussed previously actually achieve such a result could be debatable, it is anyway unclear, yet, which principles should drive such a blending in general.

On the other hand, one should notice that the coexistence of diverse layers, abiding by different sorts of principles and laws, is what could be actually observed in many interesting natural systems. An insect colony, roughly speaking, is a social system when observed at its top level, but has also an obvious “organism” layer, when observed at the “insect-level,” along with some biochemical, chemical, and physical layers under them. More generally, “diverse hierarchies play a role in modelling and simulation for computational biology” [95], so hierarchical/layered models, where different layers obey to different sorts of laws, represent a functional example of how different metaphors and mechanism could be successfully mixed up and work together in a complex system.

4.3. Semantic Coordination. While tuple-based models and technologies are manifesting their potential in the coordination of complex computational systems [16], there are at the same time showing their limits in the context of open and knowledge-intensive scenarios, where coordination policies typically need to deal with heterogeneous knowledge sources, partially-specified information, and uncertainty. In fact, as observed in [96], associative access to tuples in standard tuple-based models is based on a *tuple-matching mechanism* that is purely syntactic. Correspondingly, in most tuple-based models, coordination occurs in a merely syntactic fashion: no semantics is associated to the information exchanged via tuples, so no coordination policies can be in principle based on the interpretation of the information exchanged.

Instead, many complex software systems nowadays need to deal with large, even huge amounts of data, information, and knowledge. In particular, intelligent distributed systems operating within knowledge intensive environments, as well as complex sociotechnical systems, typically require automatic understanding and manipulation of knowledge chunks of many sorts. This is why current research on tuple-based models is focussing on ontology languages to semantically describe information in tuple spaces—so to allow coordinables and coordination laws to talk about the domain of discourse in a semantically well-founded way. In fact, the generative nature of tuples makes it easy to associate semantics to tuples and tuple spaces, add a space for shared ontologies, and suitably extend the matching mechanism.

As a result, many relevant approaches—altogether known as *semantic tuple space computing*—aim at augmenting tuple spaces with semantics, as discussed in [97]. For instance, Triple Space Computing (TSC) [98] extends the tuple-based model with Semantic Web technology and relies on RDF to couple tuple-based coordination with the Semantic Web: there, in fact, tuples are triples of the form *subject predicate object*—as typical of the RDF approach. Similarly to many other semantically enhanced tuple-based coordination models—such as Conceptual Spaces (CSpaces) [99], Semantic Web Spaces [100], and sTuples [101], semantic tuple centres [96, 102], and description spaces [103]—, TSC aims at overcoming the limitations of standard tuple spaces at least in two ways. On one hand, it sets coordinated components free of the hassles of a rigid definition of communication syntax at design time, by charging the coordination abstractions of the burden of semantic interpretation. On the other hand, it promotes richer and more expressive forms of communication and—mostly—coordination, where coordination policies can also be based on semantic criteria.

In [1], the author observes that “nature might provide the most direct inspiration of all by letting us build devices that affect direct information processing.” Integrating semantic coordination within nature-inspired systems is then quite a regular step in the research on tuple-based models—in particular because many (if not most) complex systems nowadays need to operate within knowledge-intensive environments [16]. For instance, the already mentioned SAPERE model features LSA as semantically well-founded abstractions, and ecolaws work in principle on the basis of semantic interpretation of the items in the LSA spaces [78–80].

A more radical approach is instead represented by *Molecules of Knowledge* (MoK), a tuple-based nature-inspired coordination model integrating the basic principles and mechanisms of semantic coordination and self-organisation. MoK focusses on knowledge management [104, 105], exploiting the full power of the biochemical metaphor to achieve *knowledge self-organisation* within knowledge-intensive environments. There, knowledge sources produce *atoms* of knowledge in *biochemical compartments*, which then diffuse and aggregate in *molecules* by means of *biochemical reactions*, acting locally within and between such spaces. Knowledge consumer's workspaces are mapped into such compartments, which reify information-oriented user actions in terms of *enzymes* influencing atoms *aggregation* and molecule *diffusion*—for instance, making potentially-relevant knowledge atoms and molecules *autonomously move* toward the interested users.

4.4. Further Trends. Other potential trends may be foreseen, although based more on some visions of the things to come rather than on some visible direction in the literature.

4.4.1. Understanding Core Mechanisms. The measure of the expressiveness of languages is a particularly interesting problem in the field of coordination models, where a reference framework as widely understood and accepted as the Turing machine for sequential languages does not exist, yet [106]—and consequently, it is subject to research and explorations [107, 108]. As far as tuple-based models are concerned, several reference frameworks have been exploited in order to assess the expressive power of the basic LINDA language [63] and to compare it to its extensions [64] as well as to other coordination models [109].

In this context, one of the main research concern is to understand the basic elements of the expressiveness of the coordination language. In fact, LINDA is a glaring example of a minimal set of coordination primitives capable of expressing a wide range of coordination behaviours: mapping sets of primitives onto classes of coordination systems and understanding the minimal set required to build each class of coordination systems are then quite relevant research topics. In particular, understanding the minimal set of coordination primitives required to build complex stochastic behaviours would be important in the design of nature-inspired coordination languages.

For instance, *uniform coordination primitives*—that is, LINDA-like coordination primitives returning tuples matching a template with a uniform distribution [110]—seemingly capture the full-fledged dynamics of real chemical systems within the coordination abstractions. Also, probabilistic extensions like PLINDA [86] and STOPKLAIM [81] apparently provide most of the required linguistic mechanisms for injecting stochastic behaviour in coordination systems. However, suitable formal frameworks making it possible to measure the expressiveness of probabilistic coordination languages will be required in order to find out the essential linguistic mechanisms are actually required to build the whole range of nature-inspired coordinated systems.

4.4.2. Predicting Complex Behaviours. The problem of predicting the behaviour of complex systems is hard: interaction typically makes complex systems unpredictable in principle [11]. Complex computational systems are often unpredictable even for the engineers designing them; nonetheless, understanding their possible evolution over time is typically required, for instance in order to match some application requirements. Being generally unpredictable does not mean, for a computational system, being *completely* unpredictable: while predicting all the states that a system evolution will pass through could be unfeasible, properties like the reachability of a required state (possibly, associated to some probability value) or the avoidance of undesired ones may be instead within the reach of available techniques such as model checking and data mining [111, 112].

Coordination models and technologies are typically in charge of harnessing the complexity of articulated computational systems [13]: in particular, coordination abstractions are often used at the core of complex systems [113]. Also, coordination abstractions are typically well defined and computationally predictable: while this does not make coordinated complex system generally predictable, it makes it possible in principle to make them *partially predictable*, based on the predictability of the core coordinative behaviour.

A promising research line could then focus on suitably-formalised coordination abstractions—along with a suitably defined engineering methodology [114]—that could in principle ensure the predictability of given system properties within generally unpredictable coordinated systems—such as complex nature-inspired systems.

4.4.3. Coordination for Simulation. Simulation of complex systems is a multidisciplinary issue, ranging from physics to biology, from economics to social sciences, and so on: no complex system of any sort can be studied nowadays without the support of suitable simulation tools. In fields like biology, for instance, experiments done *in silico*—that is, simulating on a computer—are nowadays at least as relevant as those *in vitro* and *in vivo*.

Given the prominence of the interaction issues of complex systems, coordination models and technologies have the potential to work as the core of nontrivial simulation frameworks. In particular, self-organising nature-inspired coordination models are seemingly well suited for the simulation of complex systems, being capable to capture in principle even the most articulated stochastic behaviours. So, it is not difficult to envision future research scenarios where coordination middleware plays a central role in the development of rich simulation frameworks [115].

5. Conclusion

It might be debatable whether the emergence of nature-inspired coordination models and technologies actually represents a change of paradigm in the modelling and engineering of complex software systems [116]. Nonetheless, it is a fact that *nature-inspired models of coordination*, starting from early chemical and stigmergic approaches, have recently

evolved to become potentially the *core of complex computational systems*—such as pervasive, knowledge-intensive, intelligent, and self-* systems. In this paper, we surveyed some of the most remarkable examples in the literature, devise their main issues, and point out the most promising trends, focussing in particular on *tuple-based coordination models*.

In the overall, nature-inspired models of coordination have a long history behind them and a huge *potential* for development which is currently under exploration. In the near future, we may reasonably expect that many research activities will be aimed at designing nature-inspired coordination models and technologies that could work as the sources of the fundamental abstractions and mechanisms required in the engineering of complex nature-inspired computational systems.

Acknowledgments

This work has been supported by the EU-FP7-FET Proactive Project SAPERE Self-aware Pervasive Service Ecosystems, under Contract no. 256873. The author would like to thank first of all his collaborators and friends at the Alma Mater Studiorum in Bologna and Cesena for all the work done together on the topic of coordination models and languages during the last two decades. Also, friendly thanks go to Giancarlo Fortino for inviting him to held the keynote speech at IDC 2012, thus forcing him to reflect further and deeply on the subject illustrated in this paper.

References

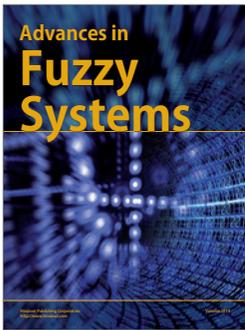
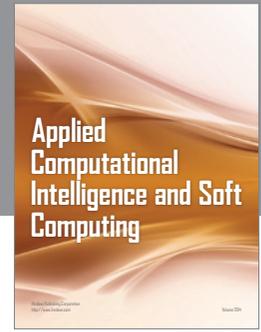
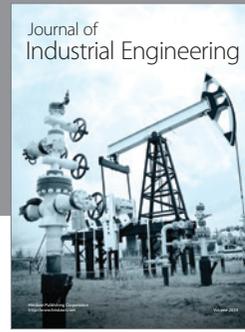
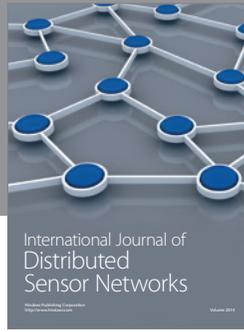
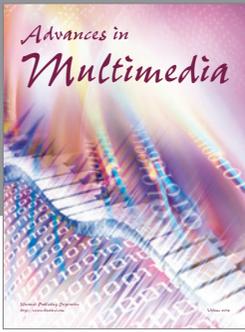
- [1] N. Shadbolt, “Nature-Inspired Computing,” *IEEE Intelligent Systems*, vol. 19, no. 1, pp. 2–3, 2004.
- [2] J. Liu and K. C. Tsui, “Toward nature-inspired computing,” *Communications of the ACM*, vol. 49, no. 10, pp. 59–64, 2006.
- [3] G. A. Papadopoulos and F. Arbab, “Coordination models and languages,” *Advances in Computers*, vol. 46, pp. 329–400, 1998.
- [4] A. Omicini, “Coordination models and languages: state of the art. Introduction,” in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds., pp. 3–5, Springer, 2001.
- [5] N. Busi, P. Ciancarini, R. Gorrieri, and G. Zavattaro, “Coordination models: a guided tour,” in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds., chapter 1, pp. 6–24, Springer, 2001.
- [6] S. Ossowski and R. Menezes, “On coordination and its significance to distributed and multi-agent systems,” *Concurrency Computation Practice and Experience*, vol. 18, no. 4, pp. 359–370, 2006.
- [7] P. Ciancarini, A. Omicini, and F. Zambonelli, “Coordination technologies for Internet agents,” *Nordic Journal of Computing*, vol. 6, no. 3, pp. 215–240, 1999.
- [8] A. Omicini and G. A. Papadopoulos, “Editorial: why coordination models and languages in AI?” *Applied Artificial Intelligence*, vol. 15, no. 1, pp. 1–10, 2001.
- [9] D. Q. Goldin, S. A. Smolka, and P. Wegner, *Interactive Computation: The New Paradigm*, Springer, 2006.
- [10] A. Omicini, A. Ricci, and M. Viroli, “The multidisciplinary patterns of interaction from sciences to Computer Science,” in *Interactive Computation: The New Paradigm*, D. Q. Goldin, S. A. Smolka, and P. Wegner, Eds., pp. 395–414, Springer, 2006.
- [11] P. Wegner, “Why interaction is more powerful than algorithms,” *Communications of the ACM*, vol. 40, no. 5, pp. 80–91, 1997.
- [12] P. Ciancarini, “Coordination models and languages as software integrators,” *ACM Computing Surveys*, vol. 28, no. 2, pp. 300–302, 1996.
- [13] P. Ciancarini, A. Omicini, and F. Zambonelli, “Multiagent system engineering: the coordination viewpoint,” in *Proceedings of the 6th International Workshop on Intelligent Agents VI. Agent Theories, Architectures, and Languages (ATAL ’99)*, N. R. Jennings and Y. Lespérance, Eds., vol. 1757 of *Lecture Notes in Artificial Intelligence*, pp. 250–259, Orlando, Fla, USA, July 1999.
- [14] D. Gelernter, “Generative communication in Linda,” *ACM Transactions on Programming Languages and Systems*, vol. 7, no. 1, pp. 80–112, 1985.
- [15] N. Carriero and D. Gelernter, “How to write parallel programs. A guide to the perplexed,” *Computing Surveys*, vol. 21, no. 2, pp. 323–357, 1989.
- [16] A. Omicini and M. Viroli, “Coordination models and languages: from parallel computing to self-organisation,” *Knowledge Engineering Review*, vol. 26, no. 1, pp. 53–59, 2011.
- [17] D. Rossi, G. Cabri, and E. Denti, “Tuple-based technologies for coordination,” in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds., chapter 4, pp. 83–109, Springer, 2001.
- [18] D. Gelernter and N. Carriero, “Coordination languages and their significance,” *Communications of the ACM*, vol. 35, no. 2, pp. 97–107, 1992.
- [19] J. A. Scott Kelso, “The complementary nature of coordination dynamics: self-organization and agency,” *Nonlinear Phenomena in Complex Systems*, vol. 5, no. 4, pp. 364–371, 2002.
- [20] S. E. Page, “Self organization and coordination,” *Computational Economics*, vol. 18, no. 1, pp. 25–48, 2001.
- [21] P. P. Grassé, “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*, vol. 6, no. 1, pp. 41–80, 1959.
- [22] H. Van Dyke Parunak, “‘Go to the ant’: engineering principles from natural multi-agent systems,” *Annals of Operations Research*, vol. 75, pp. 69–101, 1997.
- [23] S. Camazine, J.-L. Deneubourg, N. R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau, *Self-Organization in Biological Systems*, Princeton Studies in Complexity, Princeton University Press, Princeton, NJ, USA, 2001.
- [24] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.
- [25] G. Theraulaz and E. Bonabeau, “A brief history of stigmergy,” *Artificial Life*, vol. 5, no. 2, pp. 97–116, 1999.
- [26] E. Bonabeau, “Editor’s introduction: stigmergy,” *Artificial Life*, vol. 5, no. 2, pp. 95–96, 1999.
- [27] J. J. Odell, H. Van Dyke Parunak, M. Fleischer, and S. Brueckner, “Modeling agents and their environment,” in *Proceedings of the 3rd International Workshop on Agent-Oriented Software Engineering (AOSE ’02)*, F. Giunchiglia, J. J. Odell, and G. Weiss, Eds., vol. 2585 of *Lecture Notes in Computer Science*, pp. 16–31, Springer, Bologna, Italy, July 2003.

- [28] A. Molesini, A. Omicini, and M. Viroli, "Environment in agent-oriented software engineering methodologies," *Multiagent and Grid Systems*, vol. 5, no. 1, pp. 37–57, 2009.
- [29] D. Weyns, M. Schumacher, A. Ricci, M. Viroli, and T. Holvoet, "Environments in multiagent systems," *Knowledge Engineering Review*, vol. 20, no. 2, pp. 127–141, 2005.
- [30] H. Van Dyke Parunak, S. Brueckner, and J. Sauter, "Digital pheromone mechanisms for coordination of unmanned vehicles," in *Proceedings of the 1st International Joint Conference on: Autonomous Agents and Multiagent Systems*, pp. 449–450, ACM, New York, NY, USA, July 2002.
- [31] H. Van Dyke Parunak, "A survey of environments and mechanisms for human-human stigmergy," in *Proceedings of the 2nd International Workshop on Environments for Multi-Agent Systems (E4MAS '07)*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds., Lecture Notes in Computer Science, pp. 163–186, Springer, 2007.
- [32] J. P. Banâtre and D. Le Métayer, "The gamma model and its discipline of programming," *Science of Computer Programming*, vol. 15, no. 1, pp. 55–77, 1990.
- [33] J.-P. Banâtre, P. Fradet, and D. Le Métayer, "Gamma and the chemical reaction model: fifteen years after," in *Multiset Processing. Mathematical, Computer Science, and Molecular Computing Points of View*, C. S. Calude, G. Păun, G. Rozenberg, and A. Salomaa, Eds., vol. 2235 of *Lecture Notes in Computer Science*, pp. 17–44, Springer, 2001.
- [34] G. Berry, "The chemical abstract machine," *Theoretical Computer Science*, vol. 96, no. 1, pp. 217–248, 1992.
- [35] M. Mamei and F. Zambonelli, *Field-Based Coordination for Pervasive Multiagent Systems. Models, Technologies, and Applications*, Series in Agent Technology, Springer, 2006.
- [36] M. Mamei, F. Zambonelli, and L. Leonardi, "Co-fields: towards a unifying approach to the engineering of swarm intelligent systems," in *Proceedings of the 3rd International Workshop on Engineering Societies in the Agents World (ESAW '02)*, P. Petta, R. Tolksdorf, F. Zambonelli et al., Eds., pp. 68–81, Madrid, Spain, September 2002.
- [37] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, 1999.
- [38] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications with the TOTA middleware," in *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom '04)*, pp. 263–273, Orlando, Fla, USA, March 2004.
- [39] M. Mamei and F. Zambonelli, "Programming stigmergic coordination with the TOTA middleware," in *Proceedings of the 4th International Conference on Autonomous Agents and Multi agent Systems (AAMAS '05)*, pp. 551–558, ACM, New York, NY, USA, July 2005.
- [40] M. Viroli, M. Casadei, E. Nardini, and A. Omicini, "Towards a chemical-inspired infrastructure for self-* pervasive applications," in *Proceedings of the 1st International Workshop on Self-Organizing Architectures (SOAR '09)*, D. Weyns, S. Malek, R. de Lemos, and J. Andersson, Eds., vol. 6090 of *LNCS*, pp. 152–176, chapter 8, Springer, Cambridge, UK, July 2010.
- [41] M. Viroli and M. Casadei, "Biochemical tuple spaces for selforganising coordination," in *Proceedings of the 11th International Conference on Coordination Languages and Models (COORDINATION '09)*, J. Field and V. T. Vasconcelos, Eds., vol. 5521 of *Lecture Notes in Computer Science*, pp. 143–162, Springer, Lisbon, Portugal, June 2009.
- [42] A. Omicini, "Nature-inspired coordination for complex distributed systems," in *Intelligent Distributed Computing VI, 6th International Symposium on Intelligent Distributed Computing (IDC '12)*, G. Fortino, C. Bădică, M. Malgeri, and R. Unland, Eds., vol. 446 of *Studies in Computational Intelligence*, pp. 1–6, Springer, Calabria, Italy, September 2012.
- [43] M. Resnick, *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds*, MIT Press, Cambridge, Mass, USA, 1997.
- [44] A. Ricci, A. Omicini, M. Viroli, L. Gardelli, and E. Oliva, "Cognitive stigmergy: towards a framework based on agents and artifacts," in *Proceedings of the 3rd International Workshop on Environments for MultiAgent Systems (E4MAS '07)*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds., vol. 4389 of *LNAI*, pp. 124–140, Springer, Hakodate, Japan, February 2007.
- [45] M. Bravetti, R. Gorrieri, R. Lucchi, and G. Zavattaro, "On the expressiveness of probabilistic and prioritized data-retrieval in Linda," *Electronic Notes in Theoretical Computer Science*, vol. 128, no. 5, pp. 39–53, 2005.
- [46] M. Bravetti, R. Gorrieri, R. Lucchi, and G. Zavattaro, "On the expressiveness of probabilistic and prioritized data-retrieval in Linda," in *Proceedings of the 2nd International Workshop on Security Issues in Coordination Models, Languages, and Systems (SecCo '04)*, Security Issues in Coordination Models, Languages, and Systems, Electronic Notes in Theoretical Computer Science, vol. 128, no. 5, pp. 39–53, 2005.
- [47] M. Casadei and M. Viroli, "Applying self-organizing coordination to emergent tuple organization in distributed networks," in *Proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO '08)*, pp. 213–222, IEEE Computer Society, Venice, Italy, October 2008.
- [48] M. Viroli, M. Casadei, and A. Omicini, "A framework for modelling and implementing self-organising coordination," in *Proceedings of the 24th Annual ACM Symposium on Applied Computing (SAC '09)*, pp. 1353–1360, Honolulu, Hawaii, USA, March 2009.
- [49] M. Bravetti, "Expressing priorities and external probabilities in process algebra via mixed open/closed systems," *Electronic Notes in Theoretical Computer Science*, vol. 194, no. 2, pp. 31–57, 2008.
- [50] A. Omicini, S. Ossowski, and A. Ricci, "Coordination infrastructures in the engineering of multiagent systems," in *Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook*, F. Bergenti, M.-P. Gleizes, and F. Zambonelli, Eds., vol. 11 of *Multiagent Systems, Artificial Societies, and Simulated Organizations*, pp. 273–296, Kluwer Academic, chapter 14 edition, 2004.
- [51] A. Omicini and S. Ossowski, "Objective versus subjective coordination in the engineering of agent systems," in *Proceedings of the Intelligent Information Agents: An AgentLink Perspective*, M. Klusch, S. Bergamaschi, P. Edwards, and P. Petta, Eds., vol. 2586 of *Lecture Notes in Artificial Intelligence*, pp. 179–202, Springer, 2003.
- [52] A. Omicini, "SODA: societies and infrastructures in the analysis and design of agent-based systems," in *Agent-Oriented Software Engineering 1st International Workshop (AOSE '00)*, P. Ciancarini and M. J. Wooldridge, Eds., vol. 1957 of *Lecture Notes in Computer Science*, pp. 185–193, Limerick, Ireland, June 2000.
- [53] R. Tolksdorf, R. Menezes, and editors, "Using swarm intelligence in Linda systems," in *Proceedings of the 4th International Workshop on Engineering Societies in the Agents World (ESAW '04)*, A. Omicini, P. Petta, and J. Pitt, Eds., vol. 3071 of *Lecture Notes in Computer Science*, pp. 49–65, Springer, 2004.

- [54] T Spaces, <http://www.almaden.ibm.com/cs/tspaces/>.
- [55] P. Wyckoff, S. W. McLaughry, T. J. Lehman, and D. A. Ford, "T spaces," *IBM Systems Journal*, vol. 37, no. 3, pp. 454–474, 1998.
- [56] E. Freeman, S. Hupfer, and K. Arnold, *Java Spaces Principles, Patterns, and Practice: Principles, Patterns and Practices*, The Jini Technology Series, Addison-Wesley Longman, 1999.
- [57] TuCSoN, <http://tucson.apice.unibo.it/>.
- [58] A. Omicini and F. Zambonelli, "Coordination for internet application development," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 3, pp. 251–269, 1999.
- [59] GigaSpaces, <http://www.gigaspace.com/>.
- [60] N. Carriero and D. Gelernter, "Linda in context," *Communications of the ACM*, vol. 32, no. 4, pp. 444–458, 1989.
- [61] R. Menezes and R. Tolksdorf, "Adaptiveness in Linda-based coordination models," in *Engineering Self-Organising Systems. Nature-Inspired Approaches to Software Engineering*, G. D. M. Serugendo, A. Karageorgos, O. F. Rana, and F. Zambonelli, Eds., vol. 2977, pp. 212–232, Springer, 2003.
- [62] N. Busi, R. Gorrieri, and G. Zavattaro, "A process algebraic view of Linda coordination primitives," *Theoretical Computer Science*, vol. 192, no. 2, pp. 167–199, 1998.
- [63] N. Busi, R. Gorrieri, and G. Zavattaro, "On the expressiveness of Linda coordination primitives," *Information and Computation*, vol. 156, no. 1-2, pp. 90–121, 2000.
- [64] M. M. Bonsangue, J. N. Kok, and G. Zavattaro, "Comparing coordination models based on shared distributed replicated data," in *Proceedings of the 14th ACM Symposium on Applied Computing (SAC '99)*, pp. 156–165, March 1999.
- [65] A. I. T. Rowstron, *Bulk primitives in linda run-time systems [Ph.D. thesis]*, The University of York, 1996.
- [66] E. Denti and A. Omicini, "Designing multi-agent systems around a programmable communication abstraction," in *Formal Models of Agents, ESPRIT Project Model Age Final Workshop*, J.-J. Ch. Meyer and P.-Y. Schobbens, Eds., vol. 1760 of *Lecture Notes in Artificial Intelligence*, pp. 90–102, Springer, 1999.
- [67] S. Ahuja, N. Carriero, and D. Gelernter, "Linda and friends," *Computer*, vol. 19, no. 8, pp. 26–34, 1986.
- [68] M. Viroli and A. Omicini, "Coordination as a service," *Fundamenta Informaticae*, vol. 73, no. 4, pp. 507–534, 2006.
- [69] A. Ricci and M. Viroli, "Coordination artifacts: a unifying abstraction for engineering environment-mediated coordination in MAS," *Informatica*, vol. 29, no. 4, pp. 433–443, 2005.
- [70] A. Omicini, A. Ricci, M. Viroli, C. Castelfranchi, and L. Tummolini, "Coordination artifacts: environment-based coordination for intelligent agents," in *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '04)*, pp. 286–293, New York, NY, USA, July 2004.
- [71] A. Ricci, M. Viroli, and A. Omicini, "Environment-based coordination through coordination artifacts," in *Proceedings of the 1st International Workshop on Environments for Multi-Agent Systems (E4MAS '04)*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds., vol. 3374 of *Lecture Notes in Computer Science*, pp. 190–214, New York, NY, USA, February 2005.
- [72] D. Kirsh, "Distributed cognition, coordination and environment design," in *Proceedings of the 3rd European Conference on Cognitive Science (ECCS '99)*, S. Bagnara, Ed., pp. 1–11, Istituto di Psicologia, Consiglio Nazionale delle Ricerche, Siena, Italy, 1999.
- [73] A. Omicini, "Agents writing on walls: cognitive stigmergy and beyond," in *The Goals of Cognition. Essays in Honor of Cristiano Castelfranchi*, F. Paglieri, L. Tummolini, R. Falcone, and M. Miceli, Eds., chapter 29, pp. 543–556, College Publications, London, UK, 2013.
- [74] P. Ciancarini, "Distributed programming with Logic Tuple Spaces," *New Generation Computing*, vol. 12, no. 3, pp. 251–283, 1994.
- [75] E. Denti, A. Natali, A. Omicini, and M. Venuti, "Logic tuple spaces for the coordination of heterogeneous agents," in *Proceedings of the 1st International Workshop on Frontiers of Combining Systems (FroCoS '96)*, F. Baader and K. U. Schulz, Eds., vol. 3 of *Applied Logic Series*, pp. 147–160, Kluwer Academic, Munich, Germany, March 1996.
- [76] M. Viroli and F. Zambonelli, "A biochemical approach to adaptive service ecosystems," *Information Sciences*, vol. 180, no. 10, pp. 1876–1892, 2010.
- [77] F. Zambonelli, G. Castelli, L. Ferrari et al., "Self-aware pervasive service ecosystems," *Procedia Computer Science*, vol. 7, pp. 197–199, 2011.
- [78] M. Viroli, D. Pianini, S. Montagna, and G. Stevenson, "Pervasive ecosystems: a coordination model based on semantic chemistry," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12)*, S. Ossowski, P. Lecca, C.-C. Hung, and J. Hong, Eds., pp. 26–30, ACM, Riva del Garda, Italy, March 2012.
- [79] M. Viroli, E. Nardini, G. Castelli, M. Mamei, and F. Zambonelli, "A coordination approach to adaptive pervasive service ecosystems," in *Proceedings of the 5th IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops (SASOW '11)*, pp. 114–119, SASO 2011, Ann Arbor, Mich, USA, October 2011.
- [80] M. Viroli, E. Nardini, G. Castelli, M. Mamei, and F. Zambonelli, "A coordination approach to adaptive pervasive service ecosystems," in *Proceedings of the 1st Awareness Workshop Challenges in Achieving Self-Awareness in Autonomous Systems (AWARE '11)*, 2011.
- [81] M. Bravetti, D. Latella, M. Loreti, M. Massink, and G. Zavattaro, "Combining timed coordination primitives and probabilistic tuple spaces," in *Trustworthy Global Computing*, C. Kaklamani and F. Nielson, Eds., vol. 5474 of *Lecture Notes in Computer Science*, pp. 52–68, Springer, 2009.
- [82] A. Omicini and E. Denti, "From tuple spaces to tuple centres," *Science of Computer Programming*, vol. 41, no. 3, pp. 277–294, 2001.
- [83] A. Omicini, A. Ricci, and M. Viroli, "Time-aware coordination in ReSpecT," in *Proceedings of the 7th International Conference on Coordination Models and Languages (COORDINATION '05)*, pp. 268–282, Namur, Belgium, April 2005.
- [84] A. Omicini, A. Ricci, and M. Viroli, "Timed environment for web agents," *Web Intelligence and Agent Systems*, vol. 5, no. 2, pp. 161–175, 2007.
- [85] R. De Nicola, D. Latella, J.-P. Katoen, and M. Massink, "StoKlaim: a stochastic extension of Klaim," Tech. Rep. 2006-TR-01, Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" (ISTI), 2006.
- [86] M. Bravetti, R. Gorrieri, R. Lucchi, and G. Zavattaro, "Quantitative information in the tuple space coordination model," *Theoretical Computer Science*, vol. 346, no. 1, pp. 28–57, 2005.
- [87] N. Busi, P. Ciancarini, R. Gorrieri, and G. Zavattaro, "Coordination models: a guided tour," in *Coordination of Internet Agents: Models, Technologies, and Applications*, A. Omicini, F. Zambonelli, M. Klusch, and R. Tolksdorf, Eds., chapter 1, pp. 6–24, Springer, 2001.

- [88] L. Gardelli, M. Viroli, and A. Omicini, "Design patterns for self-organising systems," in *Multi-Agent Systems and Applications V: 5th International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS '07)*, H.-D. Burkhard, R. Verbrugge, and L. Z. Varga, Eds., vol. 4696, pp. 123–132, Springer, Leipzig, Germany, September 2007.
- [89] M. Viroli, M. Casadei, S. Montagna, and F. Zambonelli, "Spatial coordination of pervasive services through chemical-inspired tuple spaces," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 6, no. 2, article 14, 2011.
- [90] M. Bravetti, D. Latella, M. Loreti, M. Massink, and G. Zavattaro, "Combining timed coordination primitives and probabilistic tuple spaces," in *Proceedings of the 4th International Symposium on Trustworthy Global Computing (TGC '08)*, vol. 5474, pp. 3–4, Barcelona, Spain, November 2008.
- [91] R. De Nicola, G. L. Ferrari, and R. Pugliese, "Klaim: a kernel language for agents interaction and mobility," *IEEE Transactions on Software Engineering*, vol. 24, no. 5, pp. 315–330, 1998.
- [92] A. Di Pierro, C. Hankin, and H. Wiklicky, "Probabilistic Linda-based coordination languages," in *Proceedings of the 3rd International Conference on Formal Methods for Components and Objects (FMCO '04)*, F. S. de Boer, M. M. Bonsangue, S. Graf, and W. P. de Roever, Eds., vol. 3657 of *Lecture Notes in Computer Science*, pp. 120–140, Springer, Berlin, Heidelberg, 2005.
- [93] A. Di Pierro, C. Hankin, and H. Wiklicky, "Probabilistic KLAIM," in *Proceedings of the 6th International Conference on Coordination Models and Languages (COORDINATION '04)*, R. De Nicola, G.-L. Ferrari, and G. Meredith, Eds., vol. 2949 of *Lecture Notes in Computer Science*, pp. 11924–13427, Springer, Pisa, Italy, February 2004.
- [94] F. Zambonelli and M. Viroli, "A survey on nature-inspired metaphors for pervasive service ecosystems," *International Journal of Pervasive Computing and Communications*, vol. 7, no. 3, pp. 186–204, 2011.
- [95] C. Maus, M. John, M. Rohl, and A. Uhrmacher, "Hierarchical modeling for computational biology," in *Formal Methods for Computational Systems Biology*, M. Bernardo, P. Degano, and G. Zavattaro, Eds., vol. 5016 of *Lecture Notes in Computer Science*, pp. 81–124, Springer, 2008.
- [96] E. Nardini, A. Omicini, and M. Viroli, "Semantic tuple centres," *Science of Computer Programming*. In press.
- [97] L. J. B. Nixon, E. Simperl, R. Krummenacher, and F. Martin-Requerda, "Tuplespace-based computing for the Semantic Web: a survey of the state-of-the-art," *Knowledge Engineering Review*, vol. 23, no. 2, pp. 181–212, 2008.
- [98] D. Fensel, "Triple-space computing: semantic web services based on persistent publication of information," in *Proceedings of the IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM '04)*, F. A. Aagesen, C. Anutariya, and V. Wuwongse, Eds., vol. 3283, pp. 43–53, Bangkok, Thailand, November 2004.
- [99] F. Martn-Requerda, "Towards Cspaces: a new perspective for the Semantic Web," in *Proceedings of the 1st IFIP WG12.5 Working Conference on Industrial Applications of Semantic Web*, M. Bramer and V. Terziyan, Eds., vol. 188, pp. 113–139, Springer, Jyväskylä, Finland, August 2005.
- [100] R. Tolksdorf, L. Nixon, and E. Simperl, "Towards a tuplespace-based middleware for the Semantic Web," *Web Intelligence and Agent Systems*, vol. 6, no. 3, pp. 235–251, 2008.
- [101] D. Khushraj, O. Lassila, and T. Finin, "sTuples: semantic tuple spaces," in *Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '04)*, pp. 268–277, Boston, Mass, USA, August 2004.
- [102] E. Nardini, M. Viroli, and E. Panzavolta, "Coordinating e-health systems with TuCSoN semantic tuple centres," *Applied Computing Review*, vol. 11, no. 2, pp. 43–52, 2011.
- [103] E. Nardini, A. Omicini, and M. Viroli, "Description spaces with fuzziness," in *Proceedings of the 26th Annual ACM Symposium on Applied Computing (SAC '11)*, M. J. Palakal, C.-C. Hung, W. Chu, and W. E. Wong, Eds., vol. 2, pp. 869–876, Tunghai University, TaiChung, Taiwan, March 2011.
- [104] S. Mariani and A. Omicini, "Molecules of Knowledge: self-organisation in knowledge-intensive environments," in *Intelligent Distributed Computing VI, Proceedings of the 6th International Symposium on Intelligent Distributed Computing (IDC '12)*, G. Fortino, C. Badica, M. Malgeri, and R. Unland, Eds., vol. 446, pp. 17–22, Springer, Calabria, Italy, September 2012.
- [105] S. Mariani and A. Omicini, "Self-organising news management: the Molecules of Knowledge approach," in *Proceedings of the 1st International Workshop on Adaptive Service Ecosystems: Natural and Socially Inspired Solutions (ASENSIS '12)*, J. L. Fernandez-Marquez, S. Montagna, A. Omicini, and F. Zambonelli, Eds., SASO 2012, Lyon, France, September 2012.
- [106] P. Wegner and D. Goldin, "Computation beyond Turing machines," *Communications of the ACM*, vol. 46, no. 4, pp. 100–102, 2003.
- [107] P. Wegner and D. Goldin, "Mathematical models of interactive computing," Tech. Rep., Brown University, Providence, RI, USA, 1999.
- [108] D. Q. Goldin, "Persistent Turing Machines as a model of interactive computation," in *Proceedings of the 1st International Symposium on Foundations of Information and Knowledge Systems (FoIKS '00)*, pp. 116–135, Springer, London, UK, 2000.
- [109] G. Zavattaro, "On the incomparability of Gamma and Linda," Tech. Rep. SEN-R9827, CWI, Amsterdam, The Netherlands, 1998.
- [110] L. Gardelli, M. Viroli, M. Casadei, and A. Omicini, "Designing self-organising MAS environments: the collective sort case," in *Proceedings of the 3rd International Workshop on Environments for Multiagent Systems (E4MAS '07)*, D. Weyns, H. Van Dyke Parunak, and F. Michel, Eds., vol. 4389 of *Lecture Notes in Artificial Intelligence*, pp. 254–271, Springer, Hakodate, Japan, May 2007.
- [111] M. Casadei and M. Viroli, "Toward approximate stochastic model checking of computational fields for pervasive computing systems," in *Proceedings of the 1st International Workshop on Adaptive Service Ecosystems: Natural and Socially Inspired Solutions (ASENSIS '12)*, J. L. Fernandez-Marquez, S. Montagna, A. Omicini, and F. Zambonelli, Eds., pp. 59–64, SASO 2012, Lyon, France, September 2012.
- [112] M. Alberti and A. Omicini, "Towards the analysis & prediction of complex system behaviour in SAPERE," in *Proceedings of the 1st International Workshop on Adaptive Service Ecosystems: Natural and Socially Inspired Solutions (ASENSIS '12)*, pp. 5–10.
- [113] L. Gardelli, M. Viroli, and A. Omicini, "On the role of simulations in engineering self-organising MAS: the case of an intrusion detection system in TuCSoN," in *Proceedings of the 3rd International Workshop on Engineering Self-Organising Systems (ESOA '05)*, S. A. Brueckner, G. Di Marzo Serugendo, D. Hales, and F. Zambonelli, Eds., vol. 3910 of *Lecture Notes in Artificial Intelligence*, pp. 153–166, Utrecht, The Netherlands, July 2005.

- [114] S. Miles, M. Joy, and M. Luck, "Towards a methodology for coordination mechanism selection in open systems," in *Proceedings of the 3rd International Workshop on Engineering Societies in the Agents World (ESAW '02)*, P. Petta, R. Tolksdorf, F. Zambonelli et al., Eds., pp. 241–256, Madrid, Spain, September 2002.
- [115] N. Cannata, F. Corradini, E. Merelli, A. Omicini, and A. Ricci, "An agent-oriented conceptual framework for systems biology," in *Proceedings of the 4th International Workshop on NETWORK Tools and Applications in Biology (NETTAB '04)*, E. Merelli, P. P. Gonzalez Perez, and A. Omicini, Eds., vol. 3737 of *Lecture Notes in Bioinformatics*, pp. 105–122, Springer, Camerino, Italy, September 2004.
- [116] F. Zambonelli and H. Van Dyke Parunak, "Towards a paradigm change in computer science and software engineering: a synthesis," *Knowledge Engineering Review*, vol. 18, no. 4, pp. 329–342, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

