Chapter 30

Agents writing on walls: Cognitive stigmergy and beyond

Andrea Omicini

Abstract One of the fundamental factors driving the (self-)organisation of complex social systems – such as human organisations, animal societies, and multi-agent systems – is the interaction of individuals mediated by the environment. Sharing a workspace, mutually perceiving (the effect of) each other's actions, modifying a common environment: all these are simple yet powerful mechanisms that enable social coordination. We shortly recall and classify some of the main sorts of environment-based coordination, and discuss their impact on the engineering of self-organising socio-technical systems.

1 Introduction

The complexity of today and tomorrow artificial systems is seemingly too huge for traditional programming and software engineering techniques, exploiting deterministic, "fully-controlled" approaches—where the behaviour of a system is essentially determined once and for all at design time. The main reasons for this are basically two: *unpredictability* and *size*.

On the one hand, *openness* is an almost unescapable features of today software systems: so that the number, nature, behaviour and dynamics of the components of a system cannot be predicted once and for all at design time. Also, non-trivial computational systems are typically *situated*: they are "immersed" in an environment – that is, their evolution is strictly dependent on their surrounding environment – which is usually unpredictable in its behaviour and dynamics—then, in the way in which it affects the overall behaviour of computational systems. Finally, most of the interesting computational systems are *socio-technical systems*: that is, they involve humans as essential components, whose behaviour – and then, their influence on the systems – cannot be foreseen at design time.

Andrea Omicini
Department of Computer Science and Engineering (DISI), University of Bologna, Italy
e-mail: andrea.omicini@unibo.it

On the other hand, the articulation of computational systems is generally destined to grow: the *amount* of computational units, software components, people involved, knowledge used and produced, services required and provided by software systems, is endlessly *increasing* over the years. Simply put, handling such a huge amount of things altogether within a software development process is often cognitively unfeasible, even adopting the most advanced techniques for managing project complexity.

Nowadays, the only way out of this is apparently to resort to the same mechanisms that allow complex natural and social systems to work: in particular, to rethink complex computational systems as *self-organising systems* (SOS), based on simple-yet-powerful nature-inspired *coordination* mechanisms [11]. Roughly speaking, SOS exploits size (number of components, resources, . . .) to autonomously deal with unpredictability (of the environment, of the components, . . .), with no need to predetermine its structure and behaviour at design time.

Along this line, in this paper we first review the basic mechanisms of *self-organising coordination* (SOC) [13] as they emerge from social and natural SOS: stigmergy, cognitive stigmergy, behavioural implicit communication—generally speaking, the mechanisms for *environment-based coordination* (Section 2). Then, we elaborate on the architecture and behaviour of the software systems exploiting SOC in the engineering of computational SOS, focussing in particular on the role of the *environment* in computational systems (Section 3), and on the different *layers* where SOC mechanisms can intervene and affect the systems' dynamics, driving self-organisation (Section 4).

2 Self-Organisation via Environment-based Coordination

In the context of social systems – both human and animal ones, both natural and artificial ones – environment plays a fundamental role in supporting cooperation, collaboration, competition, and, more generally, complex coordination activities [18]. Self-organisation in such systems is typically the result of the interaction among the system's components that is *mediated* by means of some structures in the environment. Through its structure and dynamics, the environment affects component's interaction and influences the system's self-organisation.

In the remainder of this section, we shortly recall (and conceptually re-order) the main elements of SOS, and focus on environment-mediated interaction.

2.1 Agents Leaving Traces: Self-Organisation through Stigmergy

Historically, the original notion of self-organising coordination is grounded in studies on the behaviour of social insects, like ants or termites. The key concept here is *stigmergy*, introduced by [5] as an explanation for the coordination observed in termites societies, where *"The coordination of tasks and the regulation of constructions are not directly dependent from the workers, but from constructions themselves."*

Namely, the notion of stigmergy generally refers to a set of coordination mechanisms mediated by the environment. For instance, in ant colonies, chemical substances – namely *pheromone* – act as environment markers for specific social activities and drive both the individual and the social behaviour of ants.

While the notion of stigmergy has undergone a large number of generalisations / specialisations / extensions [14, 16], and even a larger number of implementations in systems of many sorts, its main features can still be recapitulated quite easily. In short, stigmergic coordination could be roughly represented as follows: some interacting *agents* perform some *action* on the *environment* that leaves some *traces*, or markers, which can then be perceived by other agents and affect their subsequent behaviour. Interaction among agents is mediated by the environment, through traces – like ant's pheromones –: emission of traces is generative – once they are produced, traces' life is independent of the producer –, but their evolution over time depends on their relation with the environment—as in the case of pheromone diffusion, aggregation, and evaporation in ant colonies. Such a sort of interaction among agents is what produces self-organisation: whereas it occurs on a *local* basis, its effect is *global* in terms of the system's *global* structures and behaviours it originates [4].

Research on stigmergic coordination produced a number of interesting results—including relevant approaches in multi-agent systems (MAS), see [15, 22] among the many others. However, it also emphasised two typical main biases of stigmergy-based approaches: *(i)* the agent model is usually quite simple – ant-like agents do not exploit any cognitive ability of theirs –, and *(ii)* the environment model is often quite elementary—featuring pheromone-like traces along with simple environmental evolution mechanisms.

By contrast, a number of relevant works in the field of cognitive sciences point out the role of stigmergy as a fundamental coordination mechanism also e.g. in the context of human societies and organisations [20, 29]. In this context, as noted in [16]:

- modifications to the environment are often amenable of an interpretation in the context of a shared, conventional system of signs:
- the interacting agents feature cognitive abilities that can be used in the stigmergy-based interaction:

Based on this, the notion of *cognitive stigmergy* was introduced in [16] as a first generalisation of stigmergic coordination exploiting agent's cognitive capabilities to enable and promote self-organisation of social activities.

2.2 *Agents Writing on Walls: Self-Organisation via Cognitive Stigmergy*

When traces have meaning, stigmergic coordination adds a novel dimension to self-organisation. This does not necessarily lead to a dichotomy: the intensity of the scent of a pheromone trail has a sort of implicit "meaning" that most ants autonomously "interpret" in order to set their own behaviour. Instead, the point here is that when agents leave traces in the environment, other agents can perceive and interpret them according to a conventional system of signs: such as agents writing symbols on (MAS) walls, with other agents reading their writes, understanding them, and reacting appropriately.

Social systems present very clear examples of traces with a meaning—traces becoming *signs*. Road signs work exactly like that, enforcing traffic laws to drive the self-organisation of traffic systems. First of all, they are put in visible places, so as to demand attention – and reaction – from "driver agents". Also, they refer to the surrounding environment—so, their meaning is local, and such is the interaction they produce, too. Drivers react to their presence, but also to their colour and shape, and possibly to the encoded symbols that can be read and interpreted: red signals, for instance, are typically signs of prohibitions, whereas danger warning signs are usually yellow. Colours are here used to activate immediate driver's reaction at a non-symbolic level, whereas the full interpretation of road signs is deferred to a slower but richer symbolic interpretation: so, road signs work as both traces and signs for (cognitive) driver agents.

When traces become signs, stigmergy becomes *cognitive stigmergy* [16]. There, self-organisation is based on marks amenable of a symbolic interpretation, and involves intelligent agents, able to correctly interpret signs in the environment, and to react properly. For instance, in the case of traffic laws, road signs play the twofold role of traces and signs, so that both (classic) stigmergy and cognitive stigmergy are in place as fundamental mechanisms for self-organisation in traffic systems.

Among the many sorts of computational models for coordination, *tuple-based models* [19] can be taken as a computational specimen for stigmergic coordination—including cognitive stigmergy [13]. There, multiple tuple spaces physically/logically distributed in a computational system could be seen as the building blocks of the system's environment—the "walls" upon which coordinating agents can read and write signs in the form of *tuples*. Suspensive coordination primitives – like `in` or `rd` – could be used either by "stupid" processes to synchonise upon pre-determined tuple patterns, or by intelligent agents properly interpreting the symbolic content of tuples—as in the case of logic tuples, for instance [3]. So, self-organising patterns based on either stigmergy or cognitive stigmergy can be in principle built upon tuple-based coordination models and technologies.

However, again, social systems provide a clear example of the fact that environment-mediated coordination is not limited to traces or signs left in the environment. Instead, as noted in [21]:

- agents acting in a shared environment often use their actions to (implicitly) communicate their intentions or goals to other agents;
- a shared environment allowing observation of other agent's actions, along with intelligent agents interpreting them, works the basis for self-organising coordination in many social systems.

Based on this, the notion of *behavioural implicit communication* (BIC) was introduced by Cristiano Castelfranchi *et al.* [21, 1] as a wide generalisation of stigmergic coordination, where the observation of agent's actions within a shared environment is the essential interaction mechanism which the self-organisation of complex social (and socio-technical) systems depends upon.[1]

2.3 *Agents in a Workspace: Self-Organisation via Behavioural Implicit Communication*

When actions in the environment have meaning, new forms of communication and coordination can contribute to the (self-)organisation of complex (computational, social, socio-technical) systems. Here, this is the foremost contribution of the notion of BIC, introduced in [21] and fully described in [1] as the form of (self-organising) coordination that stems from the observation of agent actions in a shared environment. As claimed in [21], BIC could be considered as a generalisation of the notion of stigmergy (and cognitive stigmergy, too), in the sense that self-organising coordination among agents is based on the observation and interpretation of actions as wholes, rather than of their mere effects on the environment in terms of either traces or signs.

BIC occurs whenever an agent uses its own actions to communicate and coordinate with other agents observing the agent's actions through a shared environment. As in the case of stigmergy, this requires an environment making the observation of other agent actions possible—under the obvious assumption that observing agents are intelligent enough to interpret actions. Also, more structured properties of the environment are often exploited, as the ability to select actions to be made observable to other agents, or to hide some actions from other agents' view.

BIC is a critical coordination mechanism that is mainly responsible for the overall social order of human societies. As a very simple example, it is customary in Italy for pedestrians that aim at crossing a busy road to signal their intention to car drivers by simply making a first step upon zebra crossing, then waiting for drivers indicating their will to allow pedestrian crossing by braking in response, thus clearly slowing the incoming cars. In the field of artificial systems, it is apparent that RoboCup teams could easily rely on the mutual observation of both teammates and adversary actions to promote ef-

[1] *Sematectonic stigmergy* [14] is a form of stigmergic coordination that could be taken the missing link between classic (marker-based) stigmergic coordination and BIC-based coordination. However, this distinction, while fundamental in the classification of nature-inspired SOC mechanisms, is of no use here.

fective team coordination—as teamwork in general seems to deeply rely upon many forms of BIC.

In the end, what clearly emerges from the notion of BIC, as well as from the many enlightening case studies discussed by Castelfranchi and colleagues [21, 1], is that an essential mechanism for the self-organisation of complex social system is the observation of the agent's actions by other agents sharing the same working environment, and their interpretation in order to understand intentions, goals, attitude, beliefs, mental states of the agents with which one has to coordinate—in a word, to *mind-read* other coordinating agents.

2.4 *Self-Organisation via Environment-based Coordination in the Overall*

In the overall, computational, social and socio-technical systems have the potential to exploit a number of simple-yet-powerful environment-based mechanisms of interaction. They could exploit the perception of traces left by agent actions on the environment (stigmergy), their interpretation as signs belonging to a conventional symbolic system (cognitive stigmergy), the observation of actions as wholes within the environment (BIC): or, more generally, any working and meaningful combination of these mechanisms. Also, they could involve agents of any sorts – "stupid" or intelligent, human or animal, natural or artificial – interacting and working altogether in possibly-heterogeneous settings.

What all of these mechanisms have in common is that the environment – with its structure and dynamics – is essential for mediating / enabling / governing the interaction, and to make it viable to promote effective self-organisation of the systems. In the next section, we briefly recall the fundamental requirements for the engineering of the environment in computational systems exploiting the above-mentioned mechanisms for self-organisation.

3 Computational Environment for Self-Organising Coordination

When focussing on environment-mediated interaction as the basis for SOC in computational systems, the question to be answered is clearly how a computational environment should be structured and engineered in order to make it possible and promote interactions of that sort.

In general, the pivotal role of environment in complex computational systems has been generally recognised, already: in particular, the environment is nowadays considered as a first-class abstraction in the modelling and engineering of MAS, as discussed in [26]. There, the authors put forward the environment as an explicit part of MAS, assigning it a twofold role: *(i)* determining the surrounding conditions for agents to exist, and *(ii)* providing exploitable design abstractions for around which MAS can be designed and built. When dealing with SOC, the latter role could be translated into "providing agents with exploitable coordination abstractions enabling and promoting self-organisation".

Middleware and software infrastructure is how a computational environment is put in place. According to [25], software infrastructures are required to provide MAS designers with the means to develop MAS environment, and thus to structure MAS applications. There, authors "endorse a general viewpoint where the environment of a multi-agent system is seen as a set of basic bricks we call *environment abstractions*, which *(i)* provide agents with services useful for achieving individual and social goals, and *(ii)* are supported by some underlying software infrastructure managing their creation and exploitation".

Accordingly, in the remainder of this section we try to devise out the main requirements and the architecture for a computational environment supporting SOC, focussing in particular on the form of the "basic bricks"—the coordination abstractions provided by the environment to enable and promote self-organisation.

3.1 Requirements for Environment in Self-Organising Systems

Following the step-by-step presentation of the mechanisms for environment-based coordination of Section 2, the first requirement for an environment supporting SOC is to make it possible the "recording" of agent's traces, so as to enable stigmergic coordination. Also, the ability to properly react to the emission of agent's traces is mandatory for the environment: for instance, pheromone-like traces should interact with the environment where they are deposited so as to enact the basic mechanisms of diffusion, aggregation, and evaporation. In addition, traces should be available to other agents for perception, so the environment should allow for multiple agents sharing traces, as well as perceiving their relevant properties—the equivalent of pheromone-like scent and intensity.

Furthermore, in order to allow for local interaction – a fundamental requirement for SOS –, the environment should feature a *topology*, that is, a coherent and expressive set of spatial abstractions. To cope with the basic needs of SOC, topological abstractions should enforce at least some notion of *locality*—allowing agents and resources in the environment to be associated to local spaces of some sort, and enabling a well-founded notion of local interaction among agents, and between agents and the environment as well. As obvious, more refined and structured topological models could be adopted to implement and exploit more articulated environment structures for SOC.

The above requirements for environment features suffice for both stigmergy and cognitive stigmergy. Of course, it may be the case that environment structures include tools supporting agents in the symbolic interpretation of traces as signs, such as dictionaries and ontologies: even though this may help enforcing cognitive stigmergy coordination, it is not strictly needed—whereas agent intelligence is a necessary pre-condition for cognitive stigmergy.

SOC mechanisms based on the observation and interpretation of agent actions within a shared environment call instead for more stringent requirements. In the case of BIC, agent's actions aimed at behaviourally expressing coordination issues need to be made observable to other agents sharing the same environment. Relevant properties of such actions need to be visible to target agents—and possibly hidden to other agents, for instance by means of virtual "walls" properly partitioning agent's groups.

3.2 Structuring the Environment in Self-Organising Systems

Suitable environment abstractions should then be provided by the software infrastructure so as to satisfy the above-defined requirements.

For instance, as already suggested in Subsection 2.2, tuple-based middleware naturally support stigmergic coordination—with tuples working as traces, tuple spaces as environment abstractions, and agents interacting by writing and reading tuples in tuple spaces. However, two issues need to be addressed before stigmergy could be effectively implemented in terms of tuple-based coordination: environment behaviour and topology.

On the one hand, essential mechanisms implementing effective environment dynamics against traces – like pheromone diffusion, aggregation, and decay – cannot be easily implemented in terms of standard tuple space behaviour. The ability to associate specific coordinative behaviours to tuple spaces is then mandatory—as it happens for instance with coordination abstractions like *tuple centres*, that is, programmable tuple spaces [2]. On the other hand, topological distribution of tuple spaces in the typical tuple-based models suffices to build basic SOC mechanisms. However, more refined topological abstractions, such as A&A workspaces [17], clearly promote more articulated and expressive forms of SOC.

When marks become signs, tools like dictionaries, vocabularies, and ontologies can be of help in order to give meaning to traces. For instance, notions like *function description* and *operating instructions* for artefacts [12] enable effective interaction of intelligent agents with the environment abstractions, which could work as informative containers of signs.

Moving towards BIC-based coordination, the most natural approach in a computational environment is to rely on *reification* of actions within some environment abstraction. So, for instance, relevant actions, along with their most meaningful properties, could be made available to other agents by reifying them as suitably-descriptive tuples in a shared tuple space. Accordingly, artefacts like dashboards, logs, diaries, and note-boards [16] work by keeping track of both actions and annotations made by individual agents during a working session. This would not suffice, however, without some computational mechanism able to promote the agent *awareness* of other agents' actions. For instance, the notion of *focus* of attention as modelled in CArtAgO [17] works essentially along that direction, by allowing agents to direct their attention to specific artefacts recording the agent's actions they are interested in, and with respect to which they intend to react and coordinate.

4 The Many Faces of Self-Organising Coordination

When observing real-world instances of stigmergic coordination – or, more generally, of SOS – it is quite natural to focus on the single most-evident pattern of self-organisation, to devise it out, and describe the SOS based on that pattern. However, this is not how SOS really work: there, self-organisation actually takes many forms, and affects many levels of the SOS architecture. For instance, living systems are typically hierarchical, layered systems [6], where self-organisation mechanisms work essentially at every level—from the molecular up to the organ level.

In the same way, "real" software systems facing the many faces of complexity should be expected to exploit many different coordination mechanisms for self-organisation at different architectural levels.

4.1 Layers in Self-Organising Coordination: MAS as a Reference Model

MAS represent the most natural paradigm for the design and development of complex software systems [28]. Accordingly, in the following we take MAS as the conceptual reference for computational SOS, and discuss the way in which coordination mechanisms for self-organisation may intervene at many different levels in the architecture of a SOS.

In particular, we refer to the A&A meta-model [12]: there, agents are responsible of activities, while artefacts are assigned required functions in MAS. In particular, artefacts shape MAS environment, mediating and governing the intra- (social) and inter-MAS (environmental) interaction. MAS topology is organised around workspaces, which agents and artefacts belong to, and which the notion of locality in the A&A meta-model is based on. According to [9], artefacts are conceptually layered into three classes, ruling three different sorts of MAS interaction: individual artefacts, ruling the interaction of individual agents within the MAS; social artefacts, ruling social interaction among agents; and environment artefacts, ruling MAS-environment interaction.

As interaction (and its management, that is, coordination) is the core of any self-organisation process, it is apparent that self-organisation dynamics could originate from – and then, be governed by – any of the above-mentioned sorts of artefacts in a MAS. Correspondingly, SOC could originate from the interaction of individual agents with the system, from the social interaction among agents, and also from the interaction between MAS and the environment.[2] So, for instance, SOC mechanisms like stigmergy could be exploited for regulating MAS-environment interaction, cognitive stigmergy for social interaction, and BIC for individual interaction.

As a case study, in the remainder of this section we focus on knowledge-intensive systems: in particular, on those socio-technical systems where agents are represented by human knowledge workers, dwelling within knowledge intensive environments, where huge amounts of information and data cannot be handled without the help of some SOC mechanism.

4.2 Layers in Self-Organising Coordination: A Case Study with Knowledge-Intensive Environments

As discussed in [10], knowledge-intensive environments (KIE) represent an essential dimension in modern computational systems, featuring rich information contexts and knowledge-oriented human-computer interaction. However, the (too) many possible ways of human-computer knowledge-intensive systems, along with the huge, unordered amount of relevant information available, could make the design of knowledge-intensive workspaces a nightmare.

A computer-based working environment for human knowledge workers like researchers, journalists, writers, lawyers, physicians, politicians, administrators – all professionals dealing with critical amounts of information resources that need to be suitably reviewed, organised, and used within complex, ever-changing contexts – requires on the one hand that all the relevant information sources are made available to the user in a complete yet usable format, on the other hand that the working environment autonomously evolves and adapts to the individual uses and work habits, as well as to the diverse goals and contexts defined by the everyday activity. Both requirements could be addressed by facing the following key issues: the suitable definition and representation of all the relevant information; the self-organisation (including creation, propagation, and dissipation) of knowledge; and the self-organisation of the working environment.

While relevant information could be handled by adopting many well-known techniques in the area of knowledge representation and extraction, the main point here is the explicit representation, memorisation, and exploitation of user actions in the workspace, to be exploited for the self-organisation of the working environment—à la BIC. Also, user actions should leave traces on the working environment which – as in the case of cognitive stigmergy – could be used for the self-organisation of both the relevant knowledge and the workspace. Furthermore, suitably-represented information chunks could autonomously combine to create new knowledge memes that could propagate towards the user – using for instance nature-inspired SOC mech-

[2] Whereas, one should not forget that also the dynamics of individual components of a MAS (agents and artefacts) may in its turn be determined by (internal) self-organising mechanisms—as in the case of chemical coordination [24].

anisms like biochemical coordination [23, 27] –, thus providing users with a self-organising KIE. As an example, Molecules of Knowledge (MoK) [8] is a self-organising knowledge-oriented model where knowledge sources produce atoms of knowledge in biochemical compartments, which then autonomously diffuse and aggregate in molecules by means of biochemical reactions.

Self-organisation of knowledge would then involve both the individual workspace (where it should be mainly used for the self-organisation of the working environment) and the information infrastructure, and would in principle affect all the available levels: individual, intra-organisation, and inter-organisation. Correspondingly, SOC mechanisms would then regulate individual, social, and environment interaction, according to the A&A meta-model discussed above.

5 Conclusion

The patterns for self-organising coordination as they can be devised from both natural and social SOS – from stigmergy and cognitive stigmergy to behavioural implicit communication – clearly suggest the potential of environment-based coordination in the engineering of complex computational SOS. In this article, we first review in short the main mechanisms for SOC, then discuss the role of the environment in supporting such mechanisms. Finally, we elaborate on the different levels at which self-organisation may affect the architecture and dynamics of computational SOS, by providing a reference model in terms of MAS, and discussing a case study in KIE.

Work is currently ongoing on both theoretical and practical projects where the above concepts and mechanisms are developed and exploited within complex application scenarios—in particular, in the context of pervasive computing scenarios and knowledge-intensive environments.

Acknowledgements I would like to thank the many people that worked with me in the last years, and inspired me with ideas, observations, objections, and criticisms: first of all, Alessandro Ricci and Mirko Viroli, whose constant and persistent stimulus always worked as a inexhaustible form of inspiration, as well as Enrico Denti and Antonio Natali. Then, in the last months, I shared many ideas with Marco Roccetti, who provided me with an excellent dialectical framework for my contribution to focus, and Stefano Mariani, who helped me developing a new perspective on self-organising coordination. Finally, the inspiration provided by Cristiano Castelfranchi and his group – including Rino Falcone, Luca Tummolini, Michele Piunti, and the rest of the Rome's bunch – was essential since the early years of my career as a researcher – even long before I firstly met Cristiano in Barcelona several years ago.

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

References

1. 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). DOI 10.4018/jaci.2010010101
2. Denti, E., Natali, A., Omicini, A.: Programmable coordination media. In: D. Garlan, D. Le Métayer (eds.) Coordination Languages and Models, *LNCS*, vol. 1282, pp. 274–288. Springer-Verlag (1997). DOI 10.1007/3-540-63383-9_86. 2nd International Conference (COORDINATION'97), Berlin, Germany, 1–3 Sep. 1997. Proceedings
3. Denti, E., Omicini, A.: Designing multi-agent systems around a programmable communication abstraction. In: J.J.C. Meyer, P.Y. Schobbens (eds.) Formal Models of Agents, *LNAI*, vol. 1760, pp. 90–102. Springer-Verlag (1999). DOI 10.1007/3-540-46581-2_7. ESPRIT Project ModelAge Final Workshop. Selected Papers
4. Di Marzo Serugendo, G., Foukia, N., Hassas, S., Karageorgos, A., Kouadri Mostéfaoui, S., Rana, O.F., Ulieru, M., Valckenaers, P., Van Aart, C.: Self-organisation: Paradigms and applications. In: G. Di Marzo Serugendo, A. Karageorgos, O.F. Rana, F. Zambonelli (eds.) Engineering Self-Organising Systems: Nature-Inspired Approaches to Software Engineering, vol. 2977, pp. 1–19. Springer (2004). DOI 10.1007/b95863
5. 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). DOI 10.1007/BF02223791
6. Grene, M.J.: Hierarchies in biology. American Scientist **75**, 504–510 (1987)
7. Luck, M., Mařík, V., Štěpánková, O., Trappl, R. (eds.): Multi-Agent Systems and Applications, *Lecture Notes in Computer Science*, vol. 2086. Springer (2001). DOI 10.1007/3-540-47745-4. 9th ECCAI Advanced Course ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School (EASSS 2001), Prague, Czech Republic, 2-13 Jul. 2001, Selected Tutorial Papers
8. Mariani, S., Omicini, A.: Molecules of Knowledge: Self-organisation in knowledge-intensive environments. In: Intelligent Distributed Computing VI, Studies in Computational Intelligence. Springer, Calabria, Italy (2012). Intelligence, vol. 446, pp. 17–22. Springer (2012). DOI 10.1007/978-3-642-32524-3_4. 6th International Symposium on Intelligent Distributed Computing (IDC 2012), Calabria, Italy, 24-26 Sep. 2012. Proceedings
9. Molesini, A., Omicini, A., Denti, E., Ricci, A.: SODA: A roadmap to artefacts. In: O. Dikenelli, M.P. Gleizes, A. Ricci (eds.) Engineering Societies in the Agents World VI, *LNAI*, vol. 3963, pp. 49–62. Springer (2006). DOI 10.1007/11759683_4. 6th International Workshop (ESAW 2005), Kuşadası, Aydın, Turkey, 26–28 Oct. 2005. Revised, Selected & Invited Papers
10. Omicini, A.: Self-organising knowledge-intensive workspaces. In: A. Ferscha (ed.) Pervasive Adaptation. The Next Generation Pervasive Computing Research Agenda, chap. VII: Human-Centric Adaptation, pp. 67–68. Institute for Pervasive Computing, Johannes Kepler University Linz, Austria (2011). URL http://www.perada.eu/essence/#item:80
11. Omicini, A.: Nature-inspired coordination for complex distributed systems. In: Intelligent Distributed Computing VI, Studies in Computational Intelligence, vol. 446, pp. 1–6. Springer (2012). DOI 10.1007/978-3-642-32524-3_1. 6th International Symposium on Intelligent Distributed Computing (IDC 2012), Calabria, Italy, 24-26 Sep. 2012. Proceedings. Invited paper
12. Omicini, A., Ricci, A., Viroli, M.: Artifacts in the A&A meta-model for multi-agent systems. Autonomous Agents and Multi-Agent Systems **17**(3), 432–456 (2008). DOI 10.1007/s10458-008-9053-x. Special Issue on Foundations, Advanced Topics and Industrial Perspectives of Multi-Agent Systems

13. Omicini, A., Viroli, M.: Coordination models and languages: From parallel computing to self-organisation. The Knowledge Engineering Review **26**(1), 53–59 (2011). DOI 10.1017/S026988891000041X. Special Issue 01 (25th Anniversary Issue)
14. Parunak, H.V.D.: A survey of environments and mechanisms for human-human stigmergy. In: D. Weyns, H.V.D. Parunak, F. Michel (eds.) Environments for Multi-Agent Systems II, *LNCS*, vol. 3830, pp. 163–186. Springer (2006). DOI 10.1007/11678809_10
15. Parunak, H.V.D., Brueckner, S., Sauter, J.: Digital pheromone mechanisms for coordination of unmanned vehicles. In: C. Castelfranchi, W.L. Johnson (eds.) 1st International Joint Conference on Autonomous Agents and Multiagent systems, vol. 1, pp. 449–450. ACM, New York, NY, USA (2002). DOI 10.1145/544741.544843
16. Ricci, A., Omicini, A., Viroli, M., Gardelli, L., Oliva, E.: Cognitive stigmergy: Towards a framework based on agents and artifacts. In: D. Weyns, H.V.D. Parunak, F. Michel (eds.) Environments for MultiAgent Systems III, *LNCS*, vol. 4389, pp. 124–140. Springer (2007). DOI 10.1007/978-3-540-71103-2_7. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers
17. Ricci, A., Piunti, M., Viroli, M., Omicini, A.: Environment programming in CArtAgO. In: R.P. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni (eds.) Multi-Agent Programming II: Languages, Platforms and Applications, Multiagent Systems, Artificial Societies, and Simulated Organizations, chap. 8, pp. 259–288. Springer (2009). DOI 10.1007/978-0-387-89299-3_8
18. Ricci, A., Viroli, M., Omicini, A.: Environment-based coordination through coordination artifacts. In: D. Weyns, H.V.D. Parunak, F. Michel (eds.) Environments for Multi-Agent Systems, *LNAI*, vol. 3374, pp. 190–214. Springer (2005). DOI 10.1007/b106134. 1st International Workshop (E4MAS 2004), New York, NY, USA, 19 July 2004. Revised Selected Papers
19. Rossi, D., Cabri, G., Denti, E.: Tuple-based technologies for coordination. In: A. Omicini, F. Zambonelli, M. Klusch, R. Tolksdorf (eds.) Coordination of Internet Agents: Models, Technologies, and Applications, chap. 4, pp. 83–109. Springer (2001)
20. Susi, T., Ziemke, T.: Social cognition, artefacts, and stigmergy: A comparative analysis of theoretical frameworks for the understanding of artefact-mediated collaborative activity. Cognitive Systems Research **2**(4), 273–290 (2001). DOI 10.1016/S1389-0417(01)00053-5
21. Tummolini, L., Castelfranchi, C., Ricci, A., Viroli, M., Omicini, A.: "Exhibitionists" and "voyeurs" do it better: A shared environment approach for flexible coordination with tacit messages. In: D. Weyns, H.V.D. Parunak, F. Michel (eds.) Environments for Multi-Agent Systems, *LNAI*, vol. 3374, pp. 215–231. Springer (2005). DOI 10.1007/b106134. 1st International Workshop (E4MAS 2004), New York, NY, USA, 19 July 2004. Revised Selected Papers
22. Valckenears, P., Van Brussel, H., Kollingbaum, M., Bochmann, O.: Multi-agent coordination and control using stigmergy applied to manufacturing control. In: Luck et al. [7], pp. 317–334. DOI 10.1007/3-540-47745-4. 9th ECCAI Advanced Course ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School (EASSS 2001), Prague, Czech Republic, 2-13 Jul. 2001, Selected Tutorial Papers
23. Viroli, M., Casadei, M.: Biochemical tuple spaces for self-organising coordination. In: J. Field, V.T. Vasconcelos (eds.) Coordination Languages and Models, *LNCS*, vol. 5521, pp. 143–162. Springer, Lisbon, Portugal (2009). DOI 10.1007/978-3-642-02053-7_8. 11th International Conference (COORDINATION 2009), Lisbon, Portugal, Jun. 2009. Proceedings
24. Viroli, M., Casadei, M., Nardini, E., Omicini, A.: Towards a chemical-inspired infrastructure for self-* pervasive applications. In: D. Weyns, S. Malek, R. de Lemos, J. Andersson (eds.) Self-Organizing Architectures, *LNCS*, vol. 6090, chap. 8, pp. 152–176. Springer (2010). DOI 10.1007/978-3-642-14412-7_8. 1st International Workshop on Self-Organizing Architectures (SOAR 2009), Cambridge, UK, 14-17 Sep. 2009, Revised Selected and Invited Papers
25. Viroli, M., Holvoet, T., Ricci, A., Schelfthout, K., Zambonelli, F.: Infrastructures for the environment of multiagent systems. Autonomous Agents and Multi-Agent Systems

14(1), 49–60 (2007). DOI 10.1007/s10458-006-9001-6. Special Issue: Environment for Multi-Agent Systems
26. 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). DOI 10.1007/s10458-006-0012-0. Special Issue on Environments for Multi-agent Systems
27. 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). DOI 10.1016/j.procs.2011.09.006. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11)
28. Zambonelli, F., Omicini, A.: Challenges and research directions in agent-oriented software engineering. Autonomous Agents and Multi-Agent Systems **9**(3), 253–283 (2004). DOI 10.1023/B:AGNT.0000038028.66672.1e. Special Issue: Challenges for Agent-Based Computing
29. Schmidt, K., Wagner, I.: Ordering systems: Coordinative practices and artifacts in architectural design and planning. Computer Supported Cooperative Work **13**(5-6), 349–408 (2004). DOI 10.1007/s10606-004-5059-3