This is a pre print version of the following article:

Engineering environment-mediated coordination via nature-inspired laws / Zambonelli, F.. - 9068:(2015), pp. 63-75. (4th International Workshop on Agent Environments for Multi-Agent Systems, E4MAS 2014 Paris; France May 6, 2014) [10.1007/978-3-319-23850-0_5].

Springer Verlag
Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

13/11/2026 05:54

(Article begins on next page)

Engineering Environment-mediated Coordination via Nature-inspired Laws

Franco Zambonelli

Dipartimento di Scienze e Metodi dell'Ingegneria,
University of Modena and Reggio Emilia
e-mail: franco.zambonelli@unimore.it

Abstract SAPERE is a general multiagent framework to support the development of self-organizing pervasive computing services. One of the key aspects of the SAPERE approach is to have all interactions between agents take place in an indirect way, via a shared spatial environment. In such environment, a set of nature-inspired coordination laws have been defined to rule the coordination activities of the application agents and promote the provisioning of adaptive and self-organizing services.

1 Introduction

Progresses in mobile and ubiquitous computing are making possible to conceive a variety of innovative general-purpose pervasive computing services for interacting with the physical and social worlds around us [5,2,12]. However, the effective design and development of such services requires the capability of promoting flexible and adaptive interactions among a multiple of distributed devices and software components.

To support the vision, a great deal of research activity in pervasive computing has been devoted to meet the requirements of pervasive service systems, i.e.: supporting self-configuration and context-aware composition; enforcing self-adaptability and self-organization; and ensuring that service frameworks can be highly flexible and long-lasting [29]. The SAPERE ("Self-aware Pervasive Service Ecosystems") approach [27,28] tackles the problem at the foundation, conceiving a radically new way of modeling integrated pervasive services and their execution environments, such that the apparently diverse issues of context-awareness, dependability, openness, flexibility, can all be uniformly addressed.

SAPERE models a pervasive service framework as a distributed multiagent system, in which the coordination between the application agents rely on spatially-situated and environment-mediated interactions [21]. In particular, in the SAPERE environment, a set of simple yet very expressive nature-inspired interaction laws dictates how agents will interact with each other, e.g., how they will compose and orchestrate their activities and how they will exchange information.

As it will be backed up in the following of this paper, the SAPERE approach effectively supports the provisioning of adaptive self-organizing services,

suitable to meet the requirements of pervasive service systems. In addition, with SAPERE, we have somewhat answered to some long-standing research questions related to environment engineering in multiagent systems. For instance:

- SAPERE defines a set of general-purpose interaction laws, embedded and enforced in the abstract spatial environment, which shows that valuable application-independent approaches can be defined as far as environmental abstractions are concerned [23].
- The SAPERE approach, with its peculiar environment-mediated coordination model, makes it possible to properly encapsulate any kind of resources and services within the environment, yet preserving the full observability of such resources and of the related coordination events [22]
- In general, SAPERE properly frames the needed abstractions and architectural building elements to engineer environment-mediated (and nature-inspired) multiagent systems [20].

To elaborate on the above, we introduce the SAPERE environment-centered coordination architecture (Section 2) and the key characteristics of the SAPERE middleware and its programming model (Section 3). Following, we present the set of nature-inspired coordination laws (Section 4) and discuss how they can be used to enforce a variety of self-organization scheme (Section 5), also with the help of a simple application example. Finally, we discuss related work (Section 6) and conclude (Section 7).

2 The SAPERE Approach and its Reference Architecture

SAPERE takes its primary inspiration from nature, and starts from the consideration that the dynamics and decentralization of future pervasive networks will make it suitable to model the overall world of pervasive services, data, and devices as a sort of distributed computational *ecosystem*.

As from Figure 1, SAPERE conceptually architects such pervasive service ecosystem as a non-layered *spatial environment*, laid above the actual pervasive network infrastructure [8]. The environment embeds the basic *interaction laws* (which we also call *eco-laws*) that rule the activities of the system. The environment mediates all interactions and represents the ground on which components of different species indirectly interact and combine with each other. Such interactions take place in respect of the eco-laws and typically based on the spatial relationships between components, so as to serve their own individual needs as well as the sustainability of the overall ecology. Users can access the ecology in a decentralized way to use and consume data and services, and they can also act as "prosumers" by injecting new data or service components, possibly also for the sake of controlling the ecology behavior.

For the *components* living in the ecosystem, all of which can be abstracted as autonomous software agents (and whether being sensors, actuators, services, users, data, or resources in general), SAPERE adopts a common modeling and a common treatment. Each of them has an associated semantic representation

which we call "LSA" (*Live Semantic Annotations*), to be injected in the spatial environment as it it were a sort of shared spatial memory (or tuple space [11]). This is a basic ingredient for enabling dynamic environment-mediated interactions between components. To account for the high dynamics of the scenario and for its need of continuous adaptation, SAPERE defines LSAs as living, active entities, tightly associated to the agent they describe, and capable of reflecting its current situation and context. This supports semantic and context-aware interactions both for service aggregation/composition and for data/knowledge management. In the case of pure data items, the entity and its LSA coincide.

The approach based on LSA makes it possible to encapsulate within the

The *eco-laws* define the basic interaction policies among the LSAs of the various agents of the ecology. In particular the idea is to enforce on a spatial basis, and possibly relying on diffusive mechanisms, dynamic networking and composition of data and services by composing their LSAs and exchanging data via them. Data and services (as represented by their associated LSAs) will be sort of chemical reagents, and interactions and compositions will occur via chemical reactions, relying on semantic pattern-matching between LSAs.

As detailed later on, the set of eco-laws includes: *Bond*, which is the basic mechanism for local interactions between components, and acts as a sort of virtual chemical bond between two LSAs (i.e., their associated agents); *Spread*, which diffuses LSAs on a spatial basis, and is necessary to support propagation of information and interactions among remote agents; *Aggregate*, which enforces a sort of catalysis among LSAs, to support distributed data aggregation; *Decay*, which mimics chemical evaporation and is necessary to garbage collect data. As discussed in Section 5, set of eco-laws are general enough to be applicable to a wide range of application domains.

Adaptivity in SAPERE is not in the capability of individual components, but in the overall self-organizing dynamics of the ecosystem. In particular, adaptivity will be ensured by the fact that any change in the system (as well as any change in its components or in the context of the components, as reflected by dynamic changes in their LSAs) will reflect in the firing of new eco-laws, thus possibly leading to the establishment of new bonds or aggregations, and/or in the breaking of some existing bonds between components.

3 The SAPERE Middleware and its Programming Interface

In this section we shortly overview how SAPERE applications can be programmed, by introducing the API of the SAPERE middleware and exemplifying its usage.

3.1 The Middleware

The execution of SAPERE applications is supported by a middleware infrastructure [26] which reifies the SAPERE architecture in terms of a lightweight

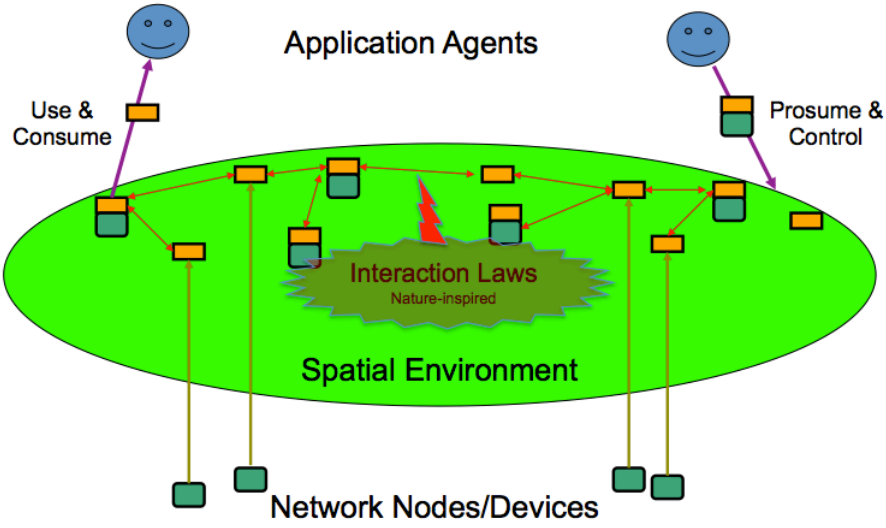


Figure 1. The SAPERE Architecture based on environment-mediated interactions

software support, enabling a SAPERE node to be installed in tablets and smartphones. Operationally, all SAPERE nodes (whether fixed at the infrastructure level or mobile) are considered at the same level since the middleware code they run could support the same services and provides the same set of functions.

Each SAPERE node hosts a local tuple space [11], that acts as a local repository of LSAs for local agents, and a local eco-laws engine. The LSA-space of each node is connected with a limited set of neighbor nodes based on spatial proximity relations. Such relations consequently determine the spatial shape of the SAPERE substrate. From the viewpoint of individual agents (that will constitute the basic execution unit) the middleware provides an API to access the local LSA space, to advertise themselves (via the injection of an LSA), and to support the agents' need of continuously updating their LSAs. In addition, such API enables agents to detect local events (as the modifications of some LSAs) or the enactment of some eco-laws on available LSAs.

Eco-laws are realized as a set of rules embedded in SAPERE nodes. For each node, the same set of eco-laws applies to rule the dynamics between local LSAs (in the form of bonding, aggregation, and decay) and those between non-locally-situated LSAs (via the spreading eco-law that can propagate LSAs from a node to another to support distributed interactions).

From the viewpoint of the underlying network infrastructure, the middleware transparently absorbs dynamic changes at the arrival/dismissing of the supporting devices, without affecting the perception of the spatial environment by individuals.

3.2 The SAPERE API

In the SAPERE model, each agent executing on a node takes care of initializing at least one LSA (representing the agent itself), of injecting it on the local LSA space, and of keeping the values of such LSA (and of any additional LSA it decides to inject) updated to reflect its current situation. Each agent can modify only its own LSAs, and eventually read the LSAs to which it has been linked by a proper eco-law. Moreover LSAs can be manipulated by eco-laws, as explained in the following sections.

At the middleware level, a simple API is provided to let agents inject LSA – `injectLSA(LSA myLSA)` – and to let agents atomically update some fields of an LSA to keep it "alive" – `updateLSA(field = new-value)`. In addition, it is possible for an agent to sense and handle whatever events occur on the LSAs of an agent, e.g., some match that triggers some eco-laws. E.g., it is possible to handle the event represented by the LSA being bound with another LSA via the `onBond(LSA mylsa)` method.

The eco-laws assure self-adaptive and self-organizing activities in the ecosystems. Eco-laws operate on a pattern-matching schema: they are triggered by the presence of LSAs matching with each other, and manipulate such LSAs (and the fields within) according to a sort of artificial chemistry [29].

3.3 LSAs

LSAs are realized as descriptive tuples made by a number of fields in the form of "name-value" properties. By building over tuple-based models and extending upon them [11], the values in a LSA can be: *actual*, yet possibly dynamic and changing over time (which makes LSAs live); *formal* not tied to any actual value unless bound to one and representing a dangling connection (typically represented with a "`?`").

Pattern matching between LSAs – which is at the basis of the triggering of eco-laws – happens when all the properties of a description match, i.e., when for each property whose names correspond (i.e., are semantically equivalent) then the associated values match. As in classical tuple-based approaches, a formal value matches with any corresponding actual value.

For instance, the following `LSAa:(sensor-type = temperature; accuracy = 0.1; temp = 45)`, that can express the LSA of a temperature sensor, can match the following `LSAb:(sensor-type = temperature; temp = ?)`, which can express a request for acquiring the current temperature value. LSAa and LSAb match with each other. The properties present in LSAa (e.g., accuracy) are not taken into account by the matching function because it considers only inclusive match.

4 The Eco-laws Set

Let us now detail the SAPERE eco-laws and discuss their role in the SAPERE ecosystem.

4.1 Bond

Bonding is the primary form of interaction among co-located agents in SAPERE (i.e., within the same LSA space). In particular, bonding can be used to locally discover and access information, as well as to get in touch and access local services. All of which with a single and unique adaptive mechanism. Basically, the bond eco-law realizes a sort of a virtual link between LSAs, whenever two LSAs (or some SubDescriptions within) match.

The bond eco-law is triggered by the presence of formal values in at least one of the LSAs involved. Upon a successful pattern matching between the formal values of an LSA and actual values of another LSA, the eco-law creates the bond between the two. The link established by bonding in the presence of the "`?`" formal fields is bi-directional and symmetric.

Once a bond is established, the agents holding the LSAs are notified of the new bond and can trigger actions accordingly. After bond creation, the two agents holding the LSAs can read each other LSAs. This implies that once a formal value of an LSA matches with an actual value in an LSA it is bound to, the corresponding agent can access the actual values associated with the formal ones. For instance, with reference to the `LSAa` and `LSAb` of the previous subsection, the agent having injected LSAb, upon bonding with `LSAa` (which the agent can detect with the `onBond` method) it can access the temperature measure by the sensor represented by `LSAb`.

As bonding is automatically triggered upon match, debonding takes place automatically whenever some changes in the actual "live" values of some LSAs make the matching conditions no longer holding.

We emphasize that bonding can be used to enable two agents to spontaneously get in touch with each other and exchange information, all of which with a single operation and with both having injected an LSA in the space. That is, unlike in traditional discovery of data and services, without distinguishing between the roles of the involved agents and subsuming the traditionally separated phases of discovery and invocation.

4.2 Aggregate

The ability of aggregating information to produce high-level digests of some contextual or situational facts is a fundamental requirement for adaptive and dynamic systems. In fact, in open and dynamic environments, one cannot know *a priori* which actual information will be available (some information source may disappear, other may appear, etc.) and the availability of ways to extract a summary of all available information (without having to explicitly discover and access the individual information sources) is very important.

The aggregate eco-law is intended to aggregate LSAs together so as to compute summaries of the current system's context. An agent can inject an LSA with the *aggregate* and *type* properties. The aggregate property identifies a function to base the aggregation upon. The type property identifies which LSAs to

aggregate. In particular it identifies a numerical property of LSAs to be aggregated. In the current implementation, the aggregate eco-law can perform most common order and duplicate insensitive aggregation functions [4].

The aggregate eco-law supports separation of concern and allows to re-use previous aggregations. On the one hand, an agent can request an aggregation process without dealing with the actual code to perform the aggregation. On the other hand, the LSA resulting from an aggregation can be read (via a proper bond) by any other agent that needs to get the pre-computed result.

4.3 Decay

The decay eco-law enables the vanishing of components from the SAPERE environment. It applies to all LSAs that specify a decay property to update the remaining time to live according to the specific decay function, or actually removing LSAs that, based on their decay property, are expired.

The Decay eco-law therefore is a kind of garbage collector capable of removing LSAs that are no longer needed in the ecosystem or no longer maintained by an agent, for instance because they are the result of a propagation.

4.4 Spread

The above eco-laws act on a local basis, i.e., on a single LSA space. Since the SAPERE model is based on a network of interaction spaces, it is fundamental to enable non-local interactions, by providing a mechanism to send information to remote LSA spaces and making it possible to distribute information and results across a network of LSA spaces.

To this end, in SAPERE we defined the spread eco-law to diffuse LSAs to remote spaces. One of the primary usages of the spread eco-law is to enable searches for components that are not available locally, and vice versa to enable the remote advertisement of services. For an LSA to be subjected to the spread eco-law, it has to include a `diffusion` field, whose value (along with additional parameters) defines the specific type of propagation.

Two different types of propagation are implemented in the SAPERE framework: *(i)* a direct propagation used to spread an LSA to a specified neighbor node, so as to make it possible to realize gossiping schemes and multicasts; *(ii)* a general diffusion capable of propagating an LSA to all neighboring SAPERE nodes, possibly recursively applying such propagation up to a maximum distance form the source node.

General diffusion of an LSA via the spread eco-law to distances greater than one is a sort of broadcast that induces a large number of replicas of the same LSA to reach the same nodes multiple times from different paths. To prevent this, general diffusion is typically coupled with the aggregate eco-law, so as to merge together such multiple replicas.

5 From Eco-laws to Distributed Self-organization and its Application to Pervasive Services

The above presented eco-laws form a necessary and complete set to support self-organizing environment-mediated interactions, which can be exploited in a variety of application scenarios.

5.1 Realizing Self-organizing Schemes with Eco-laws

The eco-laws are necessary to support decentralized adaptive behaviors for pervasive service systems. Bonding is necessary to support adaptive local service interactions, subsuming the phases of discovery and invocation of traditional service systems. Spreading is necessary to diffuse information in a distributed environment and to enable distributed interactions. Aggregation and decay are necessary to support decentralized adaptive access to information without being forced to dynamically deploy code on the nodes of the system, which may not be possible in decentralized environments.

Further, and possibly of more software engineering relevance, the eco-law set is sufficient to express a wide variety of interaction schemes (or "patterns"), there included self-organizing ones. Bonding and spreading can be used to realize local and distributed client-server scheme of interactions as well as asynchronous models of interactions and information propagation. Coupling spreading with aggregation and decay, however, makes it possible to realize also those distributed data structures necessary to support all patterns of nature-inspired adaptive and self-organizing behaviors, i.e., virtual physical fields, digital pheromones, and virtual chemical gradients [4].

In particular, aggregation applied to the multiple copies of diffused LSAs can reduce the number of redundant LSAs so as to form a distributed *gradient* structures, also known as *computational force fields*. As detailed in [14], many different classes of self-organized motion coordination schemes, self-assembly, and distributed navigation can be expressed in terms of gradients.

In addition, spreading and aggregation can be used together to produce distributed self-organized aggregations, i.e., dynamically computing some distributed property of the system and have the results of such computation available at each and every node of the system. Distributed aggregation is a basic mechanism via which to realize forms of distributed consensus and distributed task allocation and behavior differentiation.

By bringing also the decay eco-law into play, and combining it with spreading and aggregation, one can realize pheromone-based data structures, which makes possible to realize a variety of bio-inspired schemes for distributed self-organization [4]. In particular, while general diffusion and progressive decay can be used to realize diffusible and evaporating pheromone-like data structures, direct propagation can be used to navigate by following pheromone gradients.

5.2 Application Areas

Over the course of the SAPERE project, we have shown how the SAPERE approach can be effectively exploited in a variety of application scenarios. We overview here three representative examples.

As a first example, we have used SAPERE to realize a distributed tool for helping people move around in complex and crowdy environment, such as big museum and exhibitions [16]. There, SAPERE eco-laws have been exploited to realize a number of distributed computational force fields reflecting the density of people in the different areas of the environment. Then, a mobile app has been implemented capable of directing people along the gradient of decreasing fields, i.e., in less crowded areas.

As a second example, we have implemented a "social feedback" application [1], where public displays receive LSAs by users expressing their personal food preferences, and are able guide users to individually optimized food providers (e.g., restaurants, pub, cafeteria, etc.). This selection is based on *(i)* the food preferences of users and on*(ii)* the estimated waiting time at the various lunch locations, computed by aggregating (via the aggregate eco-law) the overall number of users in that location; Users can then be directed towards the most proper locations by following the gradients of appropriate computational fields.

Finally, we have developed a number of applications for helping people socializing and coordinating in unknown environment [7,12]. The general idea is to exploit public displays to collect information about users and, by using eco-laws to properly aggregate and diffuse information, be able to put in touch people with similar interests.

6 Related Works

6.1 Environment-mediated Coordination Models

The issue we face in this article can be framed as the problem of finding the proper coordination model for enabling and ruling interactions of pervasive services, starting from the key consideration that environment-mediated coordination is well suited to the scenarios of interest.

On this base, SAPERE takes as ground the archetypal environment-mediated coordination model, namely LINDA, which simply provides for a blackboard with associative matching for mediating component interactions through insertion/retrieval of tuples.Then, we followed the idea of engineering the coordination space of a distributed system by some policy "inside" tuple spaces, following the pioneer works of approaches like Tucson[15] and MARS [6]. However, our proposal tries to extend these models to include bio-inspired ecological mechanisms, by fine-grained and well structured chemical-like reactions.

In particular, the coordination approach we propose in this paper originates from the chemical tuple space model in [19], though with some notable differences: *(i)* here we provide a detail notational framework to flexibly express eco-laws that work on patterns of LSAs and affect their properties; *(ii)* the

chemical concentration mechanisms proposed in [19] to exactly mimic chemistry is not mandatory here—though it can be achieved by a suitable design of rate expressions; *(iii)* the way we conceive the overall infrastructure, and relationship between agents and their LSAs goes beyond the mere definition of the tuple-space model.

6.2 Situatedness and Context-Awareness in Pervasive Services

Considering the issues of situatedness and context-awareness, which is central to pervasive services, many of the approaches in the literature have approaches the problem by extending and modifying service-oriented approaches and architectures [13].

In PLASTIC [3], service descriptions are coupled with dynamic annotations related to the current context and state of a service, to be used for enforcing adaptable forms of service discovery. This is somewhat in line with the idea of LSAs that we in SAPERE. However, due to its environment-mediated approach, SAPERE goes further and gets rid of traditional discovery services and enforces dynamic and adaptive service interaction via simple natural reactions and a minimal middleware.

In many proposals for pervasive computing environments and middleware infrastructures, the idea of "situatedness" has been promoted by the adoption of shared virtual spaces for services and components interactions. The pioneering system Gaia [18] introduces the concept of active spaces, that is active blackboard spaces acting as the means for service interactions.

Later on, a number of Gaia extensions where proposed to enforce dynamic semantic pattern-matching for service composition and discovery [10] or access to contextual information [9]. However, the concept of active space is more that of a shared memory, rather than that of an active environment in which to reify the existence of the pervasive service components and via which to enable and rule their interactions.

6.3 Self-organization

Several recent works exploit the lessons of adaptive self-organizing natural and social systems to enforce self-awareness, self-adaptivity and self-management features in pervasive computing systems.

At the level of individual component modeling, these proposals take the form of either situated reactive agents or proactive and goal-oriented ones [17]. At the level of interaction models, these proposals typically take the form of specific nature- and socially-inspired interaction mechanisms [4], enforced either at the level of component modeling or via specific middleware-level mechanisms.

We believe the SAPERE framework integrates and improves these works in three main directions: *(i)* it abstracts from the specific internal characteristics of components (no matter whether they are simple reactive components or complex goal-oriented ones) and rather proposes an approach that seamlessly applies to

both cases; *(ii)* it tries to identify an environment-centered interaction model that is able to represent and subsume the diverse nature-inspired mechanisms under a unifying self-adaptive abstraction (i.e. the semantics chemical reactions); *(iii)* the ecological approach we undertake goes beyond most of the current studies that limit to ensembles of homogeneous components, defining a suitable framework for supporting the vision of novel pervasive and Internet scenarios as made up of self-adaptive devices and services, that autonomously cooperate for the creation of global services.

7 Conclusions and Open Challenges

SAPERE can be somewhat considered the result of several years of experience and research in the area of environment-mediated coordination. The innovative nature-inspired approach of SAPERE is effective to enforce, via environment-mediated interactions, a variety of self-organizing schemes for pervasive computing services, and properly answer to a number of open challenges identified by multiagent systems community with regard to the role of the environment in multiagent systems. These challenges include: the definition of an application-independent approach to handle environmental abstractions are concerned [23]; the possibility of encapsulating any kind of resources and services within the environment yet preserving the observability of the environment of such resources and of the related coordination events [22]; and the proper framing of the needed abstractions to engineer environment-mediated (and nature-inspired) multiagent systems [20].

As the activities within the SAPERE European Project have finished, we will now challenge the SAPERE findings and tools against innovative services in the area of urban computing and smart cities [5]. In particular, we are currently analyzing how to exploit SAPERE to realize the ambitious concept of "urban superorganism" [24,25], i.e., to have all the ICT devices in our urban environment, and the citizens within, be enable to act collectively towards the realization of urban-level objectives aimed at improving quality of life and sustainability. With this regard, and to realize such vision, research on multiagent systems environment will still have to face some open research challenges. For instance:

- How can the adopted environmental abstraction account for the socio-technical nature of future urban computing scenarios, where environment-mediated interactions will involve not only agents but also humans, and will also have to account for their social relationships?
- How can one promote human participation in urban activities by means of a suitable environment-mediated coordination model that is able to effectively incentivize collaboration?

In general, beside the two above open challenges, we think there is still plenty of room for 10 more years of interesting research in the area of environments for multiagent systems.

Acknowledgments: Work supported by the EU project SAPERE, No. 256873.

References

1. Bernhard Anzengruber, Gabriella Castelli, Alberto Rosi, Alois Ferscha, and Franco Zambonelli. Social feedback in display ecosystems. In *IEEE International Conference on Systems, Man, and Cybernetics, Manchester, SMC 2013, United Kingdom, October 13-16, 2013*, pages 2893–2898, 2013.
2. Luigi Atzori, Antonio Iera, and Giuseppe Morabito. From "smart objects" to "social objects": The next evolutionary step of the Internet of Things. *IEEE Communications Magazine*, 52(1):97–105, January 2014.
3. Marco Autili, Paolo Benedetto, and Paola Inverardi. Context-aware adaptive services: The plastic approach. In *FASE '09: Proceedings of the 12th International Conference on Fundamental Approaches to Software Engineering*, pages 124–139, Berlin, Heidelberg, 2009. Springer-Verlag.
4. Ozalp Babaoglu and al. Design patterns from biology for distributed computing. *ACM Trans. Auton. Adapt. Syst.*, 1(1):26–66, 2006.
5. Nicola Bicocchi, Alket Cecaj, Damiano Fontana, Marco Mamei, Andrea Sassi, and Franco Zambonelli. Collective awareness for human-ict collaboration in smart cities. In *21st IEEE International WETICE Symposium*, pages 3–8, 2013.
6. Giacomo Cabri, Letizia Leonardi, and Franco Zambonelli. MARS: A programmable coordination architecture for mobile agents. *IEEE Internet Computing*, 4(4):26–35, 2000.
7. Gabriella Castelli, Marco Mamei, Alberto Rosi, and Franco Zambonelli. Developing social applications in sapere. In *10th IEEE International Conference on Ubiquitous Intelligence and Computing, Vietri sul Mare, Sorrento Peninsula, Italy*, pages 314–320, 2013.
8. Gabriella Castelli, Marco Mamei, Alberto Rosi, and Franco Zambonelli. Engineering pervasive service ecosystems: The SAPERE approach. *TAAS*, 10(1):1:1–1:27, 2015.
9. Patricia Dockhorn Costa, Giancarlo Guizzardi, João Paulo A. Almeida, Luís Ferreira Pires, and Marten van Sinderen. Situations in conceptual modeling of context. In *Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006), 16-20 October 2006, Hong Kong, China, Workshops*, page 6. IEEE Computer Society, 2006.
10. Chien-Liang Fok, Gruia-Catalin Roman, and Chenyang Lu. Enhanced coordination in sensor networks through flexible service provisioning. In John Field and Vasco T. Vasconcelos, editors, *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 66–85. Springer-Verlag, June 2009. 11th International Conference (COORDINATION 2009), Lisbon, Portugal, June 2009. Proceedings.
11. David Gelernter. Generative communication in linda. *ACM Trans. Program. Lang. Syst.*, 7(1):80–112, January 1985.
12. Sara Hachem, Animesh Pathak, and Valérie Issarny. Service-oriented middleware for large-scale mobile participatory sensing. *Pervasive and Mobile Computing*, 10:66–82, 2014.
13. Michael N. Huhns and Munindar P. Singh. Service-oriented computing: Key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, 2005.
14. Marco Mamei and Franco Zambonelli. Programming pervasive and mobile computing applications: the tota approach. *ACM Trans. Software Engineering and Methodology*, 18(4), 2009.

15. Andrea Omicini and Franco Zambonelli. Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, September 1999.
16. Danilo Pianini, Alois Ferscha, Mirko Viroli, and Franco Zambonelli. Hpc from a self-organisation perspective: the case of crowd steering at urban scale. In *Workshops of the 2014 IEEE Conference on High Performance Computing, Bologna, Italy*, 2014.
17. Alessandro Ricci, Andrea Omicini, Mirko Viroli, Luca Gardelli, and Enrico Oliva. Cognitive stigmergy: Towards a framework based on agents and artifacts. In Danny Weyns, H. Van Dyke Parunak, and Fabien Michel, editors, *Environments for Multi-Agent Systems*, volume 4389 of *LNAI*, pages 124–140. Springer, February 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.
18. Manuel Román, Christopher K. Hess, Renato Cerqueira, Anand Ranganathan, Roy H. Campbell, and Klara Nahrstedt. Gaia: a middleware platform for active spaces. *Mobile Computing and Communications Review*, 6(4):65–67, 2002.
19. Mirko Viroli and Matteo Casadei. Biochemical tuple spaces for self-organising coordination. In John Field and Vasco T. Vasconcelos, editors, *Coordination Languages and Models*, volume 5521 of *LNCS*, pages 143–162. Springer-Verlag, June 2009. 11th International Conference (COORDINATION 2009), Lisbon, Portugal, June 2009. Proceedings.
20. Mirko Viroli, Tom Holvoet, Alessandro Ricci, Kurt Schelfthout, and Franco Zambonelli. Infrastructures for the environment of multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):49–60, 2007.
21. Danny Weyns, Alexander Helleboogh, Tom Holvoet, and Michael Schumacher. The agent environment in multi-agent systems: A middleware perspective. *Multiagent and Grid Systems*, 5(1):93–108, 2009.
22. Danny Weyns, Andrea Omicini, and James Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):5–30, 2007.
23. Danny Weyns, H. Van Dyke Parunak, Fabien Michel, Tom Holvoet, and Jacques Ferber. Environments for multiagent systems state-of-the-art and research challenges. In *Environments for Multi-Agent Systems, First International Workshop, E4MAS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers*, pages 1–47, 2004.
24. Franco Zambonelli. Toward sociotechnical urban superorganisms. *IEEE Computer*, 45(8):76 – 78, 2012.
25. Franco Zambonelli. Engineering self-organizing urban superorganisms. *Eng. Appl. of AI*, 41:325–332, 2015.
26. Franco Zambonelli, Gabriella Castelli, Marco Mamei, and Alberto Rosi. Integrating pervasive middleware with social networks in sapere. In *International Conference on Selected Topics in Mobile and Wireless Networking*, pages 145–150, Shanghai, PRC, 2011.
27. Franco Zambonelli et al. Self-aware pervasive service ecosystems. *Procedia CS*, 7:197–199, 2011.
28. Franco Zambonelli, Andrea Omicini, Bernhard Anzengruber, Gabriella Castelli, Francesco L. De Angelis, Giovanna Di Marzo Serugendo, Simon A. Dobson, Jose Luis Fernandez-Marquez, Alois Ferscha, Marco Mamei, Stefano Mariani, Ambra Molesini, Sara Montagna, Jussi Nieminen, Danilo Pianini, Matteo Risoldi,

Alberto Rosi, Graeme Stevenson, Mirko Viroli, and Juan Ye. Developing pervasive multi-agent systems with nature-inspired coordination. *Pervasive and Mobile Computing*, 17:236–252, 2015.

29. Franco Zambonelli and Mirko Viroli. A survey on nature-inspired metaphors for pervasive service ecosystems. *Journal of Pervasive Computing and Communications*, 7:186–204, 2011.