Towards Situated Awareness in Urban Networks: A Bio-inspired Approach

Graeme Stevenson*, Jose Luis Fernandez-Marquez†,
Sara Montagna‡, Alberto Rosi§, Juan Ye*,
Akla-Esso Tchao†, Simon Dobson*,
Giovanna Di Marzo Serugendo†, and Mirko Viroli‡
*School of Computer Science, University of St Andrews
†CUI, University of Geneva
‡Università di Bologna
§Università di Modena e Reggio Emilia
graeme.stevenson@st-andrews.ac.uk

Abstract—The possibility to have millions of computational devices interconnected across urban environments opens up novel application areas. In such highly distributed scenarios, applications must gain awareness as a result of opportunistic encounters with co-located devices, a departure from traditional reasoning approaches. We envision situated awareness as an emergent property of such networks, where bio-inspired algorithms are employed to coordinate interactions between devices through managing the lifecycle, distribution, and content of data. A congestion-aware, crowd-steering example illustrates this vision.

I. Introduction

Smartphones and public displays, which embed sensing and computation capabilities, are among the first classes of device to become truly ubiquitous. The promise of opportunistically networking such devices offers untapped potential to support the delivery of pervasive services that address users' information needs [3].

Context awareness, the process that infers application-level knowledge from low-level sensor data, is a pervasive service cornerstone. However, traditional approaches to its realisation, in the form of standalone components or static device configurations, do not apply to such dynamic and highly distributed scenarios.

In the SAPERE project [28], we re-imagine awareness as an intrinsic characteristic of the network, resulting from continuous, ad-hoc interactions between software components, layered on top of a chemically-inspired coordination model (section III). A set of biologically-inspired algorithms support these interactions, providing levers to manage the life and shape of data within the network (section IV).

Here we empower bio-inspired design patterns with powerful sensing and intelligent reasoning capabilities by introducing two added-value components – *situation awareness* and *social sensing*. We illustrate this integration as a vision of city traffic navigation that highlights the roles each aspect plays towards developing a system to dynamically adapt users' routes towards their destinations (section V), and present some conclusions from this early work (section VII).

II. Motivation

The pervasive computing vision assumes the presence of computational capability always and everywhere. Environments are potentially large in scale with sensors widely deployed, applications are hosted by mobile devices, and new applications, sensing and reasoning capabilities may be introduced by third parties. To realise situation awareness in such an environment requires a shift from a traditional system engineering methodology where a centralised reasoner digests all the sensor data and serves all applications [17].

Consider a traffic management scenario with the goal of routing drivers towards their destination, dynamically adapting routes to avoid traffic congestion. Assume that (1) cameras that monitor road conditions (e.g., congestion and accidents) are installed at each road junction, (2) users' cars report their real-time GPS location, and (3) Social 'sensors' (here we use Twitter as an example) monitor information from trusted authorities regarding traffic conditions [8]. As the number of cars and cityscape increases in size, it quickly becomes impossible to provide the necessary computational power and network infrastructure to feed data from all these sources to a centralised reasoning server to analyse traffic flow on each street and guide each individual user to their desired destination in real time.

One potential solution is to deploy reasoners local to each junction to detect local traffic conditions, to process video data, and to use GPS data to calculate the average speed of cars passing through the junction. Twitter reasoners in each district analyse relevant traffic information by filtering tweets sent within, or that mention the district. The inferences made by these localised reasoners can be aggregated along the path back to each cars' navigation application, with aggregation accounting for uncertainty; that is, sensors' operational constraints and imprecision introduced by delay in transmitting data from source to sink.

By combining the results from junction and tweet reasoners, we can form application relevant views of the traffic flow in a city, including current incidents and congestion

predictions based on car position, direction of travel, and historic movements. Using the inferred traffic flow, navigation applications can direct users appropriately. [22] presents a small-scale exploration of some of these ideas in the form of simulated crowd-steering within a museum setting.

This is a novel approach to achieving decentralised situation awareness, where a population of independent data items are perceived to autonomously, interact and aggregate to arrive at higher-level information (i.e., situations), which disseminate to the locations in the network where the information is requested. As no entity plays the role of 'coordinator' in this interaction, decentralised reasoning is affected less by computational bottlenecks and communication bandwidth limits than a centralised approach. It is robust to dynamic changes in network topology, in that no global knowledge of the network, sensors, and reasoners is assumed. The interactions between sensors, reasoners, and applications are based on matching their capabilities and requirements. This approach is built atop bio-inspired design patterns that support a modular composition towards a more complex system behaviour. Such modularity supports evolution of a system and facilitates maintainability [20].

III. THE SAPERE COORDINATION APPROACH

In this nature-inspired approach to encounter-based coordination, pervasive services are deployed as autonomous individuals in an ecosystem of other services, data sources, and devices [28]. Central to this are the ideas of supporting *situated adaptation* of software, grounded on interactions with the immediately surrounding physical and social world [18], *participant diversity* through open models of service production and usage, and *long term evolution* of network structure and participants with minimal re-engineering.

Within the *SAPERE* project, we are developing a middleware [29] that sits between physical and application concerns and governs, on a strictly local basis, a limited portion of the overall infrastructure. Deployed across a densely populated network of mobile nodes, the middleware forms the foundation upon which a very dynamic set of spatially-situated components discover, interact, and coordinate with each other, moderated by local rules embedded in each computational node. The abstract components of our pervasive ecosystem architecture include [22]:

- **LSAs** Each individual (device, physical or virtual resource, knowledge or data provider, software service, or user agent) is manifest by a *Live Semantic Annotation* (LSA) - a uniformly represented, structured, semantically founded, and continuously updated annotation (a nested property-value structure) that reflects an individual's information relevant to the coordination of an ecosystem e.g., events, state, interfaces, or agent goals.
- **LSA Spaces** LSAs reside inside LSA spaces - containers situated in each deployed node within which middleware services apply eco-laws and support the
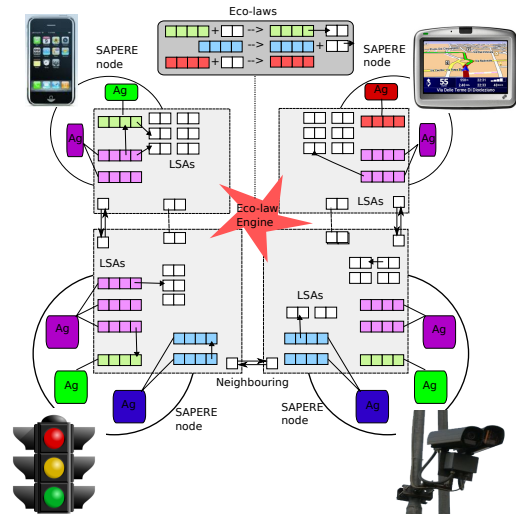

Figure 1: An architectural view of a pervasive ecosystem.

execution and lifecycle of individuals and their spatial interactions. We refer to the set of interconnected LSA spaces as the execution *environment*.

- **Eco-Laws** The way in which individuals interact is determined by the set of fundamental *eco-laws* that regulate the ecosystem model. In the same vein as chemical laws, eco-laws take a set of reactant LSAs in an LSA-space, and substitute them with a set of product LSAs, which are placed either locally or in the neighbourhood. Such transformations can lead to: creation or deletion of LSAs, semantic manipulation of LSA state, interaction through the creation and deletion of bonds between LSAs, and relocation of LSAs to support state perception across a region of space.

Figure 1 shows an architectural view, based on the above abstractions, of a portion of an ecosystem featuring: a smartphone (carried by people), a navigation device (hosted in a car) and two devices (a traffic light and a camera) placed on a junction. They form a network of 4 computational nodes; a local LSA-space and some agents running in each node (e.g., recommendation agents, advertising agents, visualisation agents in displays, profile agents and sensor agents in smartphones); LSAs through which agents manifest (in colour); additional LSAs representing data, knowledge, and contextual information like the existence of neighbouring nodes (in white); bonds between LSAs; and a set of eco-laws executed by an underlying engine working over the global LSA-space. In a more general case, one should think at a very large and mobile set of devices connected to each other based on proximity creating a distributed "space" – ideally a pervasive continuum – where LSAs form spatial structures evolved over time.

Interactions between components are based solely on the existence of bonds between their LSAs. Once established, all

interaction is indirect, through observing changes made to the structure and content of each partner's LSA. This embedding of core interaction principles within the environment represents a shift away from traditional coordination models, where interactions are externally shaped and executed. No component, service or device is permanent, everything can change and evolve, self-adapting over space and time, without undermining the ecosystem.

IV. Achieving awareness via bio-inspired collaboration

In biological self-organising processes, such as Quorum Sensing, Chemotaxis, Morphogenesis and Ant Foraging [7], the environment plays a key role, allowing organisms to achieve very complex context-aware coordination tasks through the use of a simple set of rules, and having only local perception of the world. In these biological processes, the environment is not only a shared space that allows indirect communication, but also an active entity that spreads, aggregates and evaporates substances, making these basic mechanisms the basis of the more complex self-organising processes [7]. Indeed, the role of the environment in self-organising systems has been previously emphasised [25].

While several middleware platforms supporting the storage, propagation and maintenance of distributed tuple-based data structures are proposed [13], [11], [15], BIO-CORE [6] is the first where the responsibilities and functionality of the environment are clearly defined by a simple set of low-level services enacting self-organising mechanisms. In BIO-CORE, data is held under the laws of the environment and, like the laws of nature act upon materials and substances, the properties of the information affect how the laws apply. The mechanisms proposed in BIO-CORE are spreading, aggregation, gradient, and evaporation. These belong to a larger catalog [7], presented previously, where a set of bio-inspired self-organising mechanisms are presented analogously to software engineering design patterns—as reusable solutions to commonly occurring scenarios, structured to detail the problems that each mechanism can solve, the specific solution that each brings, the dynamics among participating entities, and their implementation.

In this section we overview the BIO-CORE design patterns, highlight the role that they play in achieving situation awareness within a decentralised environment, and overview the process by which we implement out approach within a framework of semantic technologies.

A. BIO-CORE Patterns

The spreading pattern allows entities to increment the global knowledge of a system by periodically sending information from one entity to another. To avoid an explosion of information in the system, the aggregation pattern synthesises this information, extracting meaningful information. Additionally, the evaporation decreases information's
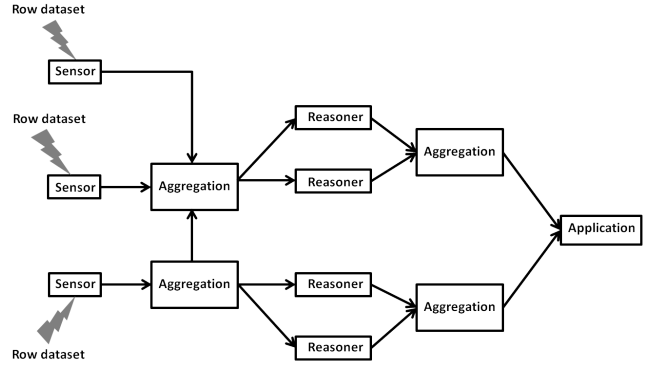

Figure 2: Interactions between sensors and reasoners when sending a reasoning result back to the application

relevance over time (i.e., making recent information more relevant than earlier information). Finally, The gradient pattern, composed by spreading, aggregation, and evaporation patterns, provides information about the distance and direction of the node where the gradient was created.

In BIO-CORE, we propose these four patterns as low level services executed by the middleware – enacted in SAPERE through the application of eco-laws to LSAs – that can be activated on demand by higher level services. For example, an LSA whose spreading property takes the value *true* will be spread over the network, while information in an LSA whose evaporation property takes the value *true* will be deemed less relevant over time.

In concert, these mechanisms provide a means of achieving awareness through environment enacted processes. The environment spreads the presence of agents, aggregates information to get meaningful information (e.g., concerning semantic relations that exist between a group of co-located people) that evaporates as required.

By capturing the environment's responsibilities using these self-organising processes, we provide a means of achieving awareness in a decentralised, robust, adaptable, and scalable manner, evocative of biological self-organising systems, and supporting software agents in adapting to environmental changes.

B. Pattern-based Situation Awareness

Based on the above BIO-CORE patterns, we devise a mechanism for achieving decentralised situation awareness. Consider a typical scenario where an an application requests information about the situation of an entity. The request is broadcast to its neighbours in the system using the *spreading* pattern. As the request is spread throughout the network, a *gradient* is setup to draw data satisfying the query towards the requesting agent (i.e., the application). Typically, a sensor or a reasoner agent receiving the request evaluates whether its capability matches the request. If it matches, the sensor or reasoner agent either retrieves the matched data or starts

reasoning and encapsulates the result in a reply message that is sent back to the source according to the *chemotaxis* pattern (a means of routing data towards the source of a gradient via the shortest accessible path). But if it does not match, the agent passes the request to the neighbouring node. To reduce communication cost, if there are more than one results matching the request, they are aggregated along the path back to the requester using the *aggregation* pattern. Aged information will be progressively discarded following the *evaporation* pattern. Figure 2 shows how the inferred result is progressively aggregated while being sent back from sensors and reasoners to the application.

These interactions are self-organising in that applications or reasoners need not know which reasoners or sensors are available in advance. The routes between an application and a reasoner or sensor are constructed based on the matching mechanism and the gradient pattern. Where the state of intermediate nodes along the transmission path change (due to failure or mobility), an updated gradient allows routing to automatically adapt assuming a path remains open.

C. Semantic Web based Implementation

Returning to the terminology introduced in section III, SAPERE agents, acting on behalf of the low-level sensors, situation classifiers, and applications express their state through LSAs that continuously reflect the state of their associated components. Each LSA is uniquely identifiable, and contains content that its owning agent wants to manifest. They are realised as an RDF-like [14] set of triples that consist of a subject (an LSA-id), a predicate (the property name, a Uniform Resource Identifier – URI) and an object (the assigned value, a literal, URI, or locally scoped identifier.

BIO-CORE is realised through the self-organising *eco-laws* that – by executing actions upon a small set of *co-located* LSAs – make global properties emerge. Eco-laws are structured as chemical-resembling rules [1], [21] of the kind "P+..+P --r--> Q+..+Q". The semantics of an eco-law *reaction* is that of consuming *reactant LSAs* based on left-hand side patterns and producing a set of *product LSAs* based on the right-hand side patterns.

A mapping of eco-laws to SPARQL/SPARUL, discussed in [24], supports their implementation in conjunction with the RDF-based encoding of LSAs and is implementable using functionality provided by standard tooling.

The complete framework is realised as a lightweight and minimal middleware that reifies LSAs in the form of semantic descriptions, dynamically stored and updated in a system of spatially-situated tuple spaces spread over the devices of the network [23].

One distinct benefit of this implementation approach is that an environment model can be built upon and interlink with existing ontology-based domain knowledge through the principles of Linked Data [2]. Not only does this allow for the adoption of emerging vocabulary standards, for
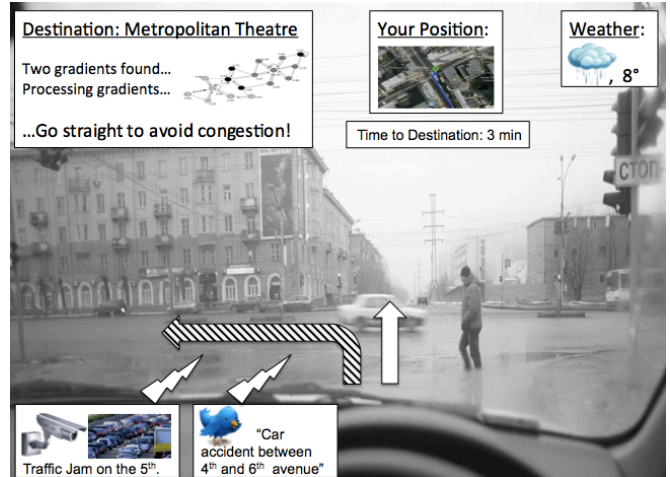

Figure 3: Steering a car towards a point of interest.

example, for representing geospatial information, but this consequently permits data generated within SAPERE to be straightforwardly linked and exposed to the outside world.

V. REVISITING THE MOTIVATIONAL EXAMPLE

While a basic level of context awareness is brought by spontaneous execution of eco-laws happening across LSA Spaces, the recognition of complex situations is supported by application dependent agents, called *reasoners*, that act upon the environment. In this scenario, BIO-CORE supports reasoners in collecting data about real time traffic conditions, which, in conjunction with historical data, are elaborated to define areas of the city that should be avoided. BIO-CORE is further used to distribute this new information. Given these elements, we realise this scenario (see figure 3) by executing five concurrent processes:

1) Each point of interest (PoI) is reified using its position (GPS, street address) as an LSA within its closest SAPERE node. From each PoI, a computational gradient is built across the network of nodes, where the gradient value represents the *accessible* distance of that nodes's location from the gradient source.

2) Entity (e.g., person, car) speed and position are sensed and injected in the SAPERE node as LSAs. Such LSAs are spread through the SAPERE network by ascending similarly established reasoner gradients, and collected by the closest node hosting a reasoner.

3) We assume regions are partitioned into local spaces, and in each local space sit reasoners that proactively collect as input all locally perceived data concerning entity position and velocity and perceive a view on the local space. The perceived view is represented in timestamped LSAs indicating where obstructions exist or are expected, along with a measure of their seriousness, and are then spread to affected locations.

The LSAs will be updated whenever the reasoner produces new result.

4) The aggregation pattern is used to penalise the value of the PoI gradient in each affected location proportionally to the seriousness of the congestion, making those areas less attractive, and allowing cars to be steered through streets with less density of cars. Multiple approaches to penalisation may be adopted; a naive implementation of penalisation could be $g/(\beta c)$, where g is the PoI gradient, c is the congestion degree, and β is the coefficient factor on the correlation between g and c.

5) Users navigate towards a PoI through a smartphone application or car GPS. The application LSA operates by bonding to the PoI's gradient LSA in the local SAPERE node, which stores the direction towards the next node containing the lowest valued gradient (i.e., the shorter distance to the PoI). Given that this gradient value is contextualised with live traffic information and congestion information supplied by reasoners, the lowest value of the gradient intrinsically avoids areas of congestion.

VI. RELATED WORK

Situation awareness, stemming from context awareness, has been widely studied in pervasive computing for over a decade. Context is defined as 'any information that can be used to characterise the situation of an entity' [19], which is usually converted from raw sensor data. A situation is defined as an abstraction of the events occurring in the real world, which is usually abstracted from the aggregation of multiple contexts [27]. Many pervasive computing system supports situation awareness, which has evolved from centralised systems [5], [26] to distributed systems [10], [9] and multi-agent-based systems [4], [12].

An exemplar distributed system is SOCAM, Service-Oriented Context-Aware Middleware, targeted towards the building and rapid prototyping of context-aware services [9]. SOCAM is a distributed middleware that transfers data from various physical spaces within which contexts are acquired, to a semantic space where contexts can be easily shared and accessed by context-aware services. It consists of service components that acquire, interpret, and store contexts as well as providing context-aware services. These five components can be distributed over heterogeneous networks and interact with each other. This is different from our approach in that we not only distribute the data acquisition and service provision but also distribute reasoning capability into *localised reasoners*. Another difference is that we implement interactions between components based on patterns and semantic matching in a modular, self-organising manner.

Our reasoning approach is agent-based, and one representative example in the area is ACAI – an Agent-based Context Aware Infrastructure [12]. ACAI allows context to be collected, processed, inferred and disseminated to applications. There are two, optional, reasoning-relevant agents in ACAI: a reasoner agent that contains a new hybrid inference system that integrates logical reasoning, fuzzy reasoning, and semantic rule representation into one system; and a system knowledge base agent that acts as the interface to the repository where queries are sent and responses are returned. This infrastructure nicely separates the context management tasks into individual agents and coordinate them to achieve a greater goal. The difference between this work and ours is that we use patterns to manage data acquisition, aggregation, and dissemination in a high-level and modular way. Among them, the aggregation pattern is responsible for merging data from different sources to find a consensus, which is similar to part of the function of the reasoner agent.

SAPERE's architecture is inspired by work from the semantic tuple space community [16], but introduces a different approach to the engineering of distributed systems by strictly binding LSAs to their owner, making LSAs the only mechanism by which software agents can perceive and be perceived in the environment.

The use of bio-inspired mechanisms in pursuit of developing decentralised algorithms has been extensively studied in the self-organisation community; we refer the reader to [7], [20] for further information.

VII. CONCLUSION

This paper explored the deployment of applications atop an urban network of opportunistically-interconnected devices, achieving awareness through coordination grounded on distributed, locally-scoped interactions. We presented a vision of inner city, hazard-aware crowd steering approached in a decentralised, scalable, and robust manner by bringing together several strands of work on bio-inspired patterns, situation awareness, and social sensing.

In [22] we report early work towards realising this vision in the form of a simulated crowd-steering case study within a museum setting. The early results are promising and now we look to developing a framework for evaluating the performance of the approach in areas such as bandwidth consumption, communication latency, and the implication of varying node densities in the network.

REFERENCES

[1] J.-P. Banâtre and T. Priol. Chemical programming of future service-oriented architectures. *JSW*, 4(7):738–746, 2009.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, MarMar 2009.

[3] G. Castelli, A. Rosi, M. Mamei, and F. Zambonelli. A simple model and infrastructure for context-aware browsing of the world. In *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, pages 229–238. IEEE Computer Society, 2007.

[4] H. Chen, T. Finin, and A. Joshi. An Intelligent Broker for Context-Aware Systems. In *Adjunct Proceedings of Ubicomp 2003*, pages 183–184, October 2003.

[5] A. K. Dey. *Providing Architectural Support for Building Context-Aware Applications*. PhD thesis, Department of Computer Science, Georgia Institute of Technology, 2000.

[6] J. L. Fernandez-Marquez, G. Di Marzo Serugendo, and S. Montagna. Bio-core: Bio-inspired self-organising mechanisms core. In *6th International ICST Conference on Bio-Inspired Models of Network, Information, and Computing Systems*, York, England, 5-7 December 2011. LNCS.

[7] J. L. Fernandez-Marquez, G. Di Marzo Serugendo, S. Montagna, M. Viroli, and J. L. Arcos. Description and composition of bio-inspired design patterns: a complete overview. *Natural Computing*, pages 1–25, 2012.

[8] L. Ferrari, A. Rosi, M. Mamei, and F. Zambonelli. Extracting urban patterns from location-based social networks. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, LBSN '11, pages 9–16, New York, NY, USA, 2011. ACM.

[9] T. Gu, H. K. Pung, and D. Q. Zhang. A service-oriented middleware for building context-aware services. *Journal of Network and Computer Applications*, 28(1):1–18, Jan. 2005.

[10] K. Henricksen and J. Indulska. Developing context-aware pervasive computing applications: models and approach. *Pervasive and Mobile Computing*, 2(1):37–64, Feb. 2006.

[11] K. Herrmann. Mesh mdl-a middleware for self-organization in ad hoc networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*. IEEE Computer Society, 2003.

[12] M. Khedr and A. Karmouch. Acai: agent-based context-aware infrastructure for spontaneous applications. *J. Netw. Comput. Appl.*, 28(1):19–44, Jan. 2005.

[13] M. Mamei and F. Zambonelli. Programming stigmergic coordination with the tota middleware. In *Proceedings of the 4th international joint conference on Autonomous agents and multiagent systems*, pages 415–422. ACM, 2005.

[14] E. Miller and F. Manola. RDF primer. W3C recommendation, W3C, Feb. 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/.

[15] A. L. Murphy, G. P. Picco, and G.-C. Roman. LIME: A middleware for physical and logical mobility. In *Proceedings of the The 21st International Conference on Distributed Computing Systems*. IEEE Computer Society, 2001.

[16] L. J. B. Nixon, E. Simperl, R. Krummenacher, and F. Martin-recuerda. Tuplespace-based computing for the semantic web: A survey of the state-of-the-art. *Knowledge Engineering Review*, 23:181–212, June 2008.

[17] A. Padovitz, S. Loke, and A. Zaslavsky. Multiple-agent perspectives in reasoning about situations for context-aware pervasive computing systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(4):729 –742, july 2008.

[18] A. Rosi, M. Mamei, F. Zambonelli, S. Dobson, G. Stevenson, and J. Ye. Social sensors and pervasive services: Approaches and perspectives. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 525 –530, march 2011.

[19] D. Salber, A. K. Dey, and G. D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, CHI '99, pages 434–441, New York, NY, USA, 1999. ACM.

[20] A.-E. Tchao, M. Risoldi, and G. di Marzo Serugendo. Modeling self-* systems using chemically-inspired composable patterns. In *Proceedings of SASO 2011*, pages 109–118, 2011.

[21] 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*, 6(2):14:1 – 14:24, June 2011.

[22] M. Viroli, D. Pianini, S. Montagna, and G. Stevenson. Pervasive ecosystems: a coordination model based on semantic chemistry. In *Proceedings of the 27th ACM Symposium On Applied Computing 2012*, March 2012.

[23] M. Viroli and G. Stevenson. On the space-time situation of pervasive service ecosystems. In *Workshop on Spatial Computing*, Valencia, Spain, June 2012. Informal Proceedings.

[24] M. Viroli, F. Zambonelli, G. Stevenson, and S. Dobson. From soa to pervasive service ecosystems: an approach based on semantic web technologies. In *Adaptive Web Services for Modular and Reusable Software Development: Tactics and Solution*. 2012. To Appear.

[25] D. Weyns, A. Omicini, and J. Odell. Environment as a first class abstraction in multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14:5–30, February 2007.

[26] S. S. Yau, D. Huang, H. Gong, and Y. Yao. Support for situation awareness in trustworthy ubiquitous computing application software. *Software: Practice and Experience*, 36(9):893–921, 2006.

[27] J. Ye, S. Dobson, and S. McKeever. Situation identification techniques in pervasive computing: a review. *Pervasive and mobile computing*, 8:36–66, Feb. 2012.

[28] F. Zambonelli, G. Castelli, L. Ferrari, M. Mamei, A. Rosi, G. Di Marzo, M. Risoldi, A.-E. Tchao, S. Dobson, G. Stevenson, Y. Ye, E. Nardini, A. Omicini, S. Montagna, M. Viroli, A. Ferscha, S. Maschek, and B. Wally. Self-aware pervasive service ecosystems. *Procedia Computer Science*, 7:197–199, 2011.

[29] F. Zambonelli, G. Castelli, M. Mamei, and A. Rosi. Integrating pervasive middleware with social networks in sapere. In *Mobile and Wireless Networking (iCOST), 2011 International Conference on Selected Topics in*, pages 145 –150, oct. 2011.