

Awareness in Collaborative Ubiquitous Environments: the Multilayered Multi-Agent Situated System Approach

MARCO P. LOCATELLI and GIUSEPPE VIZZARI

Department of Informatics, Systems and Communication

University of Milan–Bicocca

Viale Sarca 336, U14 Building, 20126 Milano, Italy

{locatelli, vizzari}@disco.unimib.it

Collaborative Ubiquitous Environments (CUEs) are environments that support collaboration among persons in a context of ubiquitous computing. This paper shows how results of the research in the Multi-Agent System (MAS) area, and in particular on MAS environments, can be used to model, design and engineer CUEs, with specific reference to the management of context-awareness information. After a description of the reference scenario, the Multilayered Multi-Agent Situated System model will be introduced and applied to represent and to manage several types of awareness information (both physical and logical contextual information). Finally, three different approaches to the design and engineering of CUEs will then be introduced and evaluated.

Categories and Subject Descriptors: I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*; C.2.4 [**Distributed Systems**]: Distributed applications; H.1.2 [**User/Machine Systems**]: Human factors

General Terms: Design, Human Factors, Languages

Additional Key Words and Phrases: MAS environments, context awareness

1. INTRODUCTION

Collaborative Ubiquitous Environments (CUEs) are environments that support collaboration among persons in a context of ubiquitous computing [Weiser 1993; Satyanarayanan 2001]. This means that this kind of environments supports human activities in a different manner than the traditional desktop metaphor adopted by the personal computers. They allow a “natural” interaction for the persons, which are not forced to sit down in front of a PC, the single spot of the environment in which both computational power and connectivity were concentrated. Persons must be supported while interacting with each other or with the environment in which they live and work. The latter, in fact, is increasingly endowed with an ever growing

Authors’ address: Department of Informatics, Systems and Communication, University of Milan–Bicocca, Viale Sarca 336, U14 Building, 20126 Milano, Italy; email: {giuseppe.vizzari, marco.locatelli}@disco.unimib.it.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0000-0000/20YY/0000-0001 \$5.00

number of sensors, information sources, hidden (or disappearing) computational devices. These devices can exploit the pervasive presence of wired and wireless communication infrastructures, and the full potential of this new technological setting is still not fully explored and exploited by current applications. According to [Zambonelli and Parunak 2002] this new scenario, with its related possibilities and challenges, will even necessarily lead to a revolutionary shift of paradigm in the way computer systems are modeled and engineered.

One of the characterizing features of the introduced scenario is the presence of devices that must interact in order to provide a rich and articulated support to human activities, but at the same time they must be able to provide a reasonable support autonomously. The need to establish interactions in an opportunistic way, leads to consider Multi-Agent Systems [Ferber 1999] as an effective way to analyze, model and design computer systems realizing CUEs. In fact, a device that is part of the overall system can be enhanced by the presence of an agent that is able to perceive the local context of the device itself, detect relevant information sources, communicate its presence, and in general establish meaningful interactions. Besides the single agent “reasoning” capabilities, the abstractions and mechanisms supporting agent mutual perception and interaction are fundamental, in the general setting of MAS research and especially in our context. These abstractions and mechanisms are functions that can be typically ascribed to a MAS environment [Weyns et al. 2007]. Most agent definitions emphasize the presence of an environment in which agents are situated, may perceive each other, act and interact. On the other hand, research in agent and MAS context has often been focused on specific aspects of agency, such as the individual reasoning and decision making processes, the possibility to communicate by sending and receiving messages expressed into a shared language according to a given domain ontology. Recent events and research efforts (i.e. the Environment for MAS workshop [Weyns et al. 2005; 2006] and the Environment TFG¹ in the context of the AgentLink Coordination Action) have highlighted the general relevance of the environment, not only as a communication infrastructure, but as a first class concept that is on one hand a necessary element of a MAS, and on the other a useful source of abstractions and mechanisms to support the analysis, modelling and design of a MAS to solve a specific problem or face a specific requirement of the desired system. The Multilayered Multi-Agent Situated System (MMASS) [Bandini et al. 2002] represents a formal and computational framework supporting the modeling of agents as well as the environment in which they are situated.

One of the aims of this paper is to show how this model, and in general MAS environment abstractions and mechanisms, can be adopted to design an infrastructure for the management and distribution of awareness information in CUEs. The following section will introduce in a more extensive way the CUEs scenario and its specific requirements, while section 3 will describe the MMASS model, and section 4 will show how this model can be applied to effectively face these requirements. In this framework, this work differs from other existing proposals that employ agents and agent-based infrastructures simply as a middleware for the design and implementation of pervasive computing systems (see, e.g., [Lech and Wienhofen 2005;

¹<http://www.cs.kuleuven.ac.be/~distrinet/events/e4mas/tfg2005/index.php>

Favela et al. 2002]). Section 5 will present a selection of the available platforms developed in this context and it will discuss their suitability to support the development of CUEs. Conclusions and future developments will end the paper.

2. REFERENCE SCENARIO: COLLABORATIVE UBIQUITOUS ENVIRONMENTS

As previously introduced, a Collaborative Ubiquitous Environment (CUEs) is a composite “mixed” system designed to support collaboration among persons in a context of ubiquitous computing. In fact, a CUE is composed of objects (we also include electronic and computational devices in this term) that show a variety of computational capabilities: from sensors to wallboards, from actual documents to pieces of furniture, from desks to doors, and so on, up to traditional general purpose computers that can play the hidden role of servers or the visible role of terminals supporting individual work. Apart from PCs, each object is dedicated to a specific functionality that can be provided by local computational power or by the interaction with a computational environment offering (wireless) connectivity. We suppose that each object can have specialized computational capabilities making them reactive and proactive in relation to actors and/or other objects that are close in their surroundings.

The need of a support for collaboration in ubiquitous computing environments has emerged during the last years [Campbell 2005; Bardram 2005; Cabitza et al. 2006], in fact, the ubiquitous computing literature is generally more focused on the individual dimension. CUEs require support both for *coordination* [Beaudouin-Lafon 1999] to manage expected patterns of interactions among users and/or devices, *must* policies such as an alarm system that has to activate fire fighting systems when sensors detect fire, and *awareness* [Schmidt 2002] to manage contextual information, *may* policies devoted to supplying users information about opportunities that can arise in the CUE, such as the “nearby” presence and availability of a technological appliance or a possible partner of interaction. The collaborative dimension, in fact, adds further requirements to recognize and support the *communities* acting in the CUE. People freely move in the physical space carrying and approaching objects that altogether provide different forms of computational connectivity, as well as meeting other people and establishing with them various forms of cooperation. Moreover, considering the logical dimension, which is the dimension where information and coordination resources are managed to support these forms of cooperation, we can recognize a similar kind of dynamism: actors own, make available, approach and coordinate their access to these resources in a flexible way according to their needs, interests, duties or simply because they realize that an opportunity is offered to them or that the current state of affairs requires their coordinated intervention. Additional motivations to the need of managing both *coordination* and *awareness* dimensions are presented in [Cabitza et al. 2006]. Our goal, however, is to maintain a strong separation between the computational abstractions (i.e. modeling concepts) and mechanisms devoted to them, in terms of their behavior, their language and the design approach; in particular, we actually have different interaction-space models and there is thus no reason to mix them.

A way to connect ubiquitous computing and cooperation is through the notion of *context*, since ubiquitous computing and context-aware computing share the same

goal: to make the environment “alive” and its context an important part of what determines the application’s behaviour [Chen and Kotz 2000]. More specifically, we like the idea to view context not as a representational problem but *as an interactional problem*, as proposed in [Dourish 2004]. Context has to be seen as a relational property that holds between objects or activities; in particular, something may or may not be contextually relevant in relation to some particular activity. The scope of contextual features is thus defined dynamically; context is relevant to particular settings or situations, particular instances of action, and particular parties to that action. Finally, context is not “out there” rather it arises from the activity. In fact, this view is more coherent with what we consider the main advantage of ubiquitous computing (once adequately developed): to bring back the notion of context as currently mediated and constrained by the desktop metaphor to its natural connotation, that of physical context, with all its affordances, possibly augmented by computing capabilities to become more significant. In ubiquitous computing the context is naturally composed of a physical component and a logical component, as mediated by the computational capabilities associated to objects of various kinds, and moreover they are or should be smoothly integrated. From the *context awareness* point of view, it is important to represent information about both physical and logical aspects related to actors, because actors move and act in the physical environment and have logical (sometimes social, because it is important and necessary to represent and exploit the relationships among persons) relationships among them. The physical spatial representation is made possible by the use of suitable localization technologies but it is not limited by them in the sense that the representation does not necessarily reflect the technological localization; in fact, physical representation could have different scales and granularity, for example to represent localization in a building by the room even if the localization provided by the technology is finer (e.g. small areas of a room). Of course, the logical dimension by its nature is not related to physical aspects, so the concept of granularity is not related to a physical property. However, we can model different kinds of relationships among actors and sometimes these can represent the “same” information but at a different level of abstraction, so granularity is a valid concept also for the logical dimension. Beyond the representation of spaces, both physical and logical, context awareness is built on the information dynamically diffused in them in accord to the view of context as an interactional problem; the *modulated propagation of information* is a key requirement of CUEs and allows dynamic interaction and perception among actors in terms of perception by proximity, dynamic configuration of the environment, acquisition of contextualized information and so on.

The characteristics of CUEs depicted above as generic requirements can be found immediately in paradigmatic examples such as the ones described in section 4. The Mmass model will be adopted to represent the notions and mechanisms regulating these CUEs and, to this end, the following section will briefly introduce the model.

3. MULTILAYERED MULTI-AGENT SITUATED SYSTEM

The Multilayered Multi-Agent Situated Systems (Mmass) model is a formal and computational framework for the definition of systems made up of a set of autonomous entities acting and interacting in a structured environment. The model

has been successfully applied to different contexts, from the simulation of complex systems (e.g. crowds [Bandini et al. 2004]), to the support of human interactions in the Computer Supported Cooperative Work area [Simone and Bandini 2002]. This section does not represent a formal description of the model (that can be found in [Bandini et al. 2002]), but will briefly introduce its main concepts, specifically focusing on the environmental structure. In fact the latter deeply influences agents behaviour, as the environment is the source of their perceptions, a constraint limiting their actions (e.g. their movement), but it also provides them a medium to interact with other entities.

In particular MMass agents are situated in *sites*, that is, nodes of the graph related to a single layer of the environment (i.e. a single Multi-Agent Situated System). The overall agent environment can be in fact composed of several interacting layers, each one representing a different aspect such as, for instance, a physical spatial abstraction, or the relationships among specific actors living in the environment. We will now focus on a single MMass to introduce its structure, as well as the abstractions and mechanisms supporting the definition of agents, their behaviours and interactions. The spatial structure of a MMass is an *undirected graph*; the presence of an edge connecting two sites indicates possibility of agents to move from one of those sites to the other, to perceive the presence of signals (i.e. *fields*, which will be introduced later on) and agents in the adjacent site. Adjacency between sites, besides enabling agents' movements and local perception, also support the diffusion of fields, according to specific rules managing their distribution in the environment (i.e. *diffusion function*). In the original definition of the model, edges were not weighted; however, in this case, since it is possible to adopt one MMass layer to represent the social relationships among individuals, it can be useful to support the definition of graph including weighted vertices. Social relationship representations will in fact adopt a weight to indicate the strength of a given relationship between two individuals. Another difference between the adopted version of the model and the original definition is the possibility of a site to host more than one agent. In fact, since there can be several coordinated spatial representations of the environment, for instance related to different levels of granularity, it can be necessary to relax the non-interpenetration principle that limited to one the maximum number of agents present in a site. For instance, if the representation of a physical environment provides the mapping of one room to a single MMass site it can be necessary to allow different agents related to different electronic devices currently positioned in the room to be situated in that site. The modified constraints and principles were aimed at reflecting actual constraints of physical environments, for sake of realism, and these modifications do not have any impact on the model properties in this context.

Agents inherit spatial relations from the sites they occupy. The adjacency relation among agents is a necessary condition for the applicability of *reaction*, the first kind of interaction mechanism defined by the MMass model. In fact this operation involves two or more agents that are placed in adjacent sites and allows them to synchronously change their state, after they have performed an agreement. The second interaction mechanism defined by the MMass model provides the possibility for agents to emit *fields*, that are signals able to diffuse through the environment

that can be perceived by other agents according to specific rules. This indirect interaction mechanism is also related to stigmergic approaches to agent communication, but fields are not just associated to an intensity value that persists through time in a given position, but may rather convey more complex kind of information also through the spatial dimension of the environment. In particular, for every field type a *diffusion function* must be specified in order to define how related signals decay (or are amplified) during their diffusion in the environment, from the source of emission to destination sites. Other functions specify how fields of the same kind can be *composed* (for instance in order to obtain the intensity of a given field type at a given site) or *compared*. From a semantic point of view fields themselves are neutral even if they can have related information in addition to their intensity; they are only signals, with an indication on how they diffuse in the environment, how they can be compared and composed. Different agent types may be able to perceive them or not and, in the first case, they may have completely different reaction, according to their behavioural specification. With reference to perception, an agent may perceive a field with a non-null intensity active in the site it is situated on according to two parameters characterizing its type and related to the specific field type. The first one is the *sensitivity threshold*, indicating the minimum field intensity that an agent of that type is able to perceive. The second is the *receptiveness coefficient* and it represents an amplification factor modulating (amplifying or attenuating) field value before the comparison with the sensitivity threshold. A specific *perception function*, associated to an agent type, maps the state of an agent to these parameters, dynamically defining the current perceptive capabilities of an agent according to its state. In this way, for instance, the same agent that was unable to perceive a specific field value could become more sensitive (increase its own receptiveness coefficient) as a consequence of a change in its state. This allows to model physical aspects of perception, but also conceptual ones such as agent interests. Reaction and field emission are two of the possible actions available for the specification of agent behaviour, related to the specification of how agents may interact. Other actions are related to the possibility to move (*transport* operation) and change the state upon the perception of a specific event in their local context (*trigger* operation).

As previously noted, the overall agent environment can be composed of several interacting layers. In order to allow this interaction (i.e. intra-MASS interaction) the model introduces the notion of *interface*. An interface of a given MASS specifies fields imported from and exported into each MASS, and more precisely it is defined as $Interface(MASS_i, export : E; import : I)$ where E and I are respectively the set of fields exported into and imported from $MASS_i$. Imported fields, that must be correspondingly exported by another MASS, can be used in specifying agent actions exactly as fields that are internal to the agent MASS. By definition, the value of an external field in any site of the local *Space* of a MASS is the value specified at its emission. Moreover, the receiving MASS has to define whether and how this value has to be internally propagated by means of local fields defined for this purpose. In fact, their distribution function is highly dependent on the structure of the local *Space* which is completely hidden to external MASSes.

In order to effectively exploit an MMASS model representing the awareness as-

ACM Journal Name, Vol. V, No. N, Month 20YY.

pects of a CUE the model itself must be open to the influences and signals generated by sensors and devices spread in the actual environment. Moreover, the computation carried out in the MMass environment, and more precisely the actions of agents situated in it, must be able to have an influence on the world, not only through the MMass environment, but also by communicating with the devices that are hosting them, for instance by displaying a message on a screen (either personal, for instance on a PDA, or public, for instance in a large screen). It is thus necessary to introduce both a new input primitive that can be included in the behavioural specification of a given agent type and the possibility to invoke an external input operation when expressing conditions triggering agents' actions, in order to represent both *intentional* and *reactive* behaviours. Both operations must support the interaction of the MMass agent with the coordination world (through the Manager agent as explained in section 4). The new primitive $request(s, o)$ is defined as the specific intentional request operation for an external object o . The $input(e)$ function can be used instead as a conditional element for the activation of a primitive, specifying that an external object e can be used to express further conditional elements. On the other hand, in order to support the possibility of an agent to provide awareness information to the coordination world, an output primitive must also be provided by the model. The $output(s, o)$ primitive specifies that the agent triggers the Manager agent by sending an o message. The information specified by o is determined on the basis of the information locally accessible to the current agent, and precisely its state s and the elements of its local context.

4. MMass AND CUES

As introduced in section 2, CUEs treat both coordination and awareness facets. Of course they are tightly coupled to provide support to collaboration; thus, the entities of the two “worlds” must have a way to interact. Moreover, the awareness side of a CUE can encompass several different aspects, related to the physical position of an actor (it could be a person as well as a device endowed with an agent managing awareness information) in a given representation of the CUE as well as the logical position, e.g. the role in an organization or project. Figure 1 depicts the reference model for MMass-based awareness management in a CUE. In particular, beyond k MMass layers (the topological spaces related to the various aspects of represented awareness information) there are some additional elements:

- *Awareness agent* ($A_{i,j}$): this kind of agent is situated on the awareness graph j and it is associated to actor i of the CUE; its behaviour is described in accord to the MMass model thus this agent reasons on fields, by perceiving and emitting them.
- *Manager agent* (M_i): the agent that bridges the awareness and coordination worlds for actor i ; it translates information expressed in term of fields (exported by the A_i agents) into *awareOf* statements (the kind of information that the agents of the coordination world can understand) and, in the other direction, imports fields in the proper $A_{i,j}$ agent when requests of propagation come from the agents of the coordination world.
- *Personal fulcrum* (fulcrum as the support for coordination): the space of interaction of the agents of the coordination world.

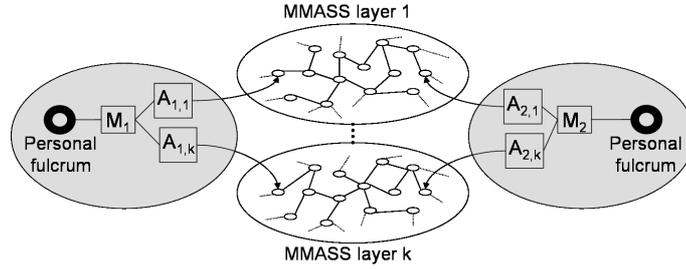


Fig. 1. Reference model for MMASS-based awareness management in a CUE.

In this paper we do not provide details on the coordination component of a CUE (see [Cabitza et al. 2006; Locatelli and Simone 2006] to have more information on this topic) because it is not strictly related to topological spaces and their representation, instead it is focused on information and behaviour exchange. However, to let the reader better understand the following sections and the section 5 here we put in evidence that the coordination part of the model is implemented by adopting DJess [Cabitza et al. 2005] because it is a middleware that easily allows the sharing of facts (to represent statements) and rules (to describe behaviours). In the following we will show how to model four scenarios that exploit the characteristics of CUE depicted in section 2 by applying the MMASS model.

4.1 Modulated propagation of information and contextual perception

The modulated propagation of information on the topological spaces allows entities to be aware of information in relation to their context. Entity’s context is reflected in the model in terms of localization on the topological space and the perception capabilities (type of field perceived and perception threshold) of the entity’s Agent. This scenario describes how awareness information can be propagated on the topological spaces and how the agents can perceive it in a manner related to their own context in a situation where the information is addressed to people that are in a particular state, that is being in some position of the smart department [Cabitza et al. 2006].

The smart department (designed in accord to the pervasive computing view) provides support for workshop management, for example to let attendees be aware of the fact that a workshop is going to start. This is done by deploying wall-mounted displays (WM in figure 2) around the department and by providing a localization device to the attendees (Mark, Sue, Sarah and Mr. Brown in figure 2). Both devices and attendees are localized and so situated on the physical localization graph; the sites of the graph reflect the logical area of the department (A51 is the “PCC workshop” room). The field *diffusion function* decreases its intensity by one unit for each edge crossed in path from the emission site. Displays perceive a field when it has an intensity greater than a threshold; instead, interested attendees perceive the field when the intensity is lower than (possibly another) threshold. Because Sarah is not interested in this workshop her A-agent, differently from A-agent of other attendees, is configured to not perceive the field.

When Mr. Brown emits the “PCC workshop is going to start” field on the A51

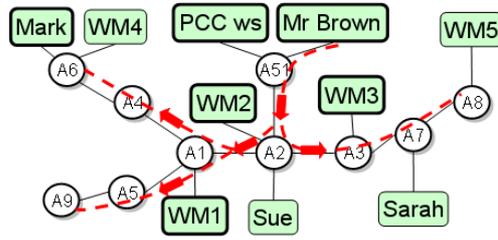


Fig. 2. Modulated propagation of information and contextual perception.

site the wall-mounted displays near this area perceive the field (which has a higher intensity) and show the message, instead the displays further (such as WM4 and WM5 in figure 2) do not perceive it. On the other side, attendees near the A51 area do not perceive the field, because its intensity is too high, but attendees far from the workshop room perceive the field and can be aware of the beginning of the “PCC workshop”. In this manner, the information is shown on the wall-mounted displays close to the workshop room and notified (through their personal device) only to persons who are far from it, so person near the room are not notified both from the wall-mounted display and their personal device. By applying this approach it is possible to design systems able to reduce the information overload in the spirit of calm technologies [Weiser and Brown 1996].

4.2 Proximity perception

This scenario describes how the perception of entities in the neighborhood can be realized by the propagation of a *presence field*. This puts in evidence the need of management of the persistence of the information in the environment to guarantee the consistence of the information. Each entity situated on the topological space emits a presence field in order to let the other ones be aware of its presence. The presence field is characterized by the *emitter id* and by a *diffusion function* that decreases the field intensity (to obtain a small neighborhood a quickly decreasing function must be defined). When the entity moves around the space, it emits the presence field again and the neighborhood changes consequently; in fact, the presence field emitted before disappears and the new one defines the new neighborhood.

In a department people can perceive the presence of the other ones thanks to the propagation of the presence field; it is important to put in evidence that the presence field could be propagated both on physical and logical topological spaces, so the neighborhood is not necessary driven by the physical proximity. A person in the department, e.g. Marcello, emits his presence field (figure 3a); when he goes to a new localization area (figure 3b) he emits the field again and now he can be perceived also by other persons, e.g. Marco and Federico.

A smart board in a room of the department is used to support opportunistic collaboration [Campbell 2005; Favela et al. 2002] among persons by exploiting the perception of the presence field. Marcello is in his office and is working at the smart board; Giuseppe is working at a project with Marcello so when he comes in Marcello’s office his presence is perceived by the entities in the room and the project-support software application use the smart board to show the summary

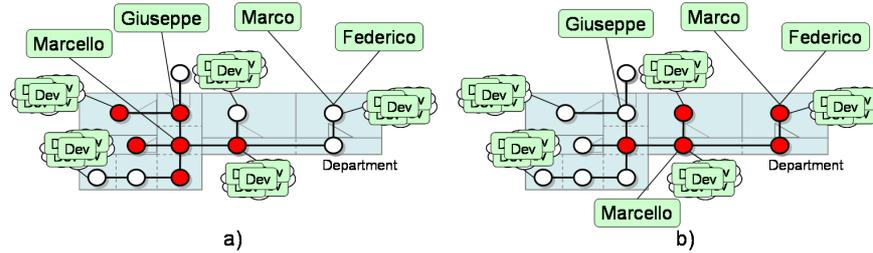


Fig. 3. Perception of actors in the neighborhood.

information about the last project meeting. Now one of them can pay attention to these information and use them to start to collaborate with the other one. Of course, by being an awareness information it is a *may* and it is in accord to the mean of supporting opportunistic collaboration: to provide the useful information (and, of course, tools) to promote collaboration among persons.

4.3 Social relationships among persons

Since a CUE is designed to support the activities of human beings and their interaction and collaboration, it is necessary to represent and exploit the relationships among persons. Mmass allows the representation of this kind of information as a specific layer and moreover it allows us to define “actions over this kind of network” through the field propagation and perception mechanism. In fact, once the network is created and agents are positioned in it, they can propagate information on it to reach other members. An example of social relationships is provided by the *support care network for elderly* case study [Locatelli and Simone 2006]. There are many people that take care of an elder, from family members to specifically paid staff (e.g. nurses); each one has a direct relationship with the elder (arcs from a member to the elder in figure 4) and could have a relationship related to the care of the elder with some other members of the network (arcs among members in figure 4). In this kind of Mmass layer each site is dedicated to one person. Weights on arcs represent the *distance* between two persons, so a low number means that there is a tight relationship.

When a relevant event related to the elder happens a field is propagated on the graph by the Elder agent and it reaches the members of the network close to the elder such as the two sons. In this way, the proper members are reached to provide support; but what happens if the two sons are unavailable or if one (e.g., the first son) is unavailable and the other one (Elder’s son 2) does not have enough knowledge to solve a problem? The Elder’s son 1 agent can autonomously (when the son does not respond to the request) propagate a field that reaches the members linked to him; in this manner the FarMember is reached, which is the person that has the tight relationship with the first son and is supposed to own the better knowledge about the elder, although she is far from the elder.

The distance among members and between the elder and members can be dynamically maintained by the whole system that observes and interprets the interaction among person by the information provided by proper technologies [Tao et al. 2005] (a more thorough discussion on the actual representation of physical/logical po-

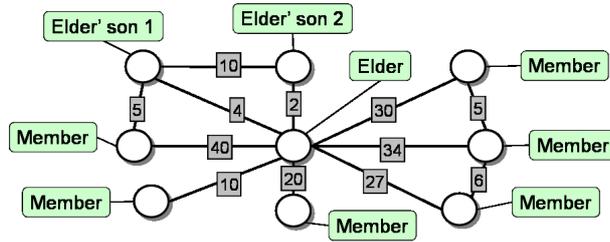


Fig. 4. The support care network for elderly.

sitions of agents and environmental structures in the system can be found in the following section).

5. TOWARDS MMass-BASED COLLABORATIVE UBIQUITOUS ARCHITECTURES

The main aim of this section is to evaluate some of the currently available computational supports and middlewares supporting the design and engineering of MAS environments for a possible adoption in MMass-based Collaborative Ubiquitous Architectures. First of all, it must be noted that a Collaborative Ubiquitous Environment is largely (and not surprisingly) a distributed system. However, a single layer of environmental representation, such as those introduced in the previous section, can be suitably hosted by a single computational node of the network, acting as an awareness server for the specific aspect of the CUE represented in the hosted layer. Devices scattered in the actual environment that are part of the CUE are assumed to be able to communicate with these nodes², in order to signal their presence and perceive the signals coming from the CUE. Every layer is responsible to maintain updated the current state of the related aspect of the overall environment (e.g. reflecting the actual physical position of a device on the discrete representation of the environment, manage the diffusion of fields). Information related to the current position of an agent could be provided by some *sensor installed on the device* hosting it (e.g. a GPS receiver, in case of physical position), by an *infrastructure of sensors* installed in the environment (e.g. a wireless network or a RFID infrastructure [Mamei and Zambonelli 2006]), or by a *user* (e.g. its current availability to interact or carry out a task, in case of a conceptual position). In other words, according to the specific aspect of the environment represented in a given layer, the information on agent position could be provided by the agent, obtained by an infrastructure or even inferred. However, nodes managing environmental layers are currently assumed to be well known by every device that is part of the CUE (their addresses could be set during the configuration of the adopted middleware).

This kind of network structure had an influence on the preliminary selection of the frameworks and middlewares to consider and evaluate. The TOTA middleware [Mamei and Zambonelli 2004], for instance, offers a rich and sophisticated support to the design and engineering of Pervasive Computing applications exploiting the abstractions of agents and MAS environment. One of TOTA's most distin-

²This is a basic assumption of the Ubiquitous Computing scenario, which is rapidly becoming a reasonable one at least in several urban contexts and built environments.

guishing features is the possibility to diffuse and keep updated context-awareness information in a dynamic environment, and in particular it offers the possibility to maintain the structure of Co-Fields over a changing network. In this case, however, the structure of the network is not very relevant in determining the context of a given node, and instead the reference scenario requires the capability to integrate several spatial representations, related to different aspects of the CUE.

The relevance of environment related abstractions and mechanisms highlighted in the previous section (i.e. spatial structure in which agents are situated, fields and related functionalities and services) in the representation of contextual information led us to focus on those MAS approaches actually providing some specific support to environmental aspects. The adoption of a middleware providing a communication infrastructure supporting direct agent interaction (e.g. JADE [Bellifemine et al. 2005]), or even a more traditional distributed object or message oriented infrastructure, would require the design and implementation of substantial additional structures and interaction mechanisms or the realization of an agent (or a set of agents) specifically devoted to the management of environment related abstractions and mechanisms to enable context-aware forms of agent interaction.

All these considerations led to consider three approaches:

- the adoption of a *framework for Mmass based applications* [Bandini et al. 2006], which provides a computational support to the abstractions and mechanisms defined by the model, integrated with a suitable middleware (the MAIS reflective architecture [Adorni et al. 2006]) supporting network communication;
- the adoption of an *artifact-based approach* [Omicini et al. 2004] to the design and engineering of MAS environment, through the adoption of TuCSoN tuple centres [Omicini and Zambonelli 1999];
- the adoption of DJess a *distributed inferential engine* [Cabitza et al. 2005] supporting the sharing of contextual information and reactive behaviours.

In the following sections these approaches will be discussed, both with reference to the support to abstractions and mechanisms defined by the Mmass model, and with reference to the specific type of application scenario.

5.1 Mmass framework MAIS reflective architecture

The first approach that was evaluated for the design of an architecture for Mmass-based CUEs is based on a framework for Mmass-based applications [Bandini et al. 2006] integrated with a middleware supporting network communication, as well as many other functionalities which are not relevant to this specific application (described in details in [Adorni et al. 2006]). In particular, a prototype of such a system was developed in the context of the Multichannel Adaptive Information System (MAIS) project³, to evaluate the adoption of the MAIS reflective architecture in an ubiquitous-computing context, and to test its suitability to the integration with a MAS-oriented modeling approach and the related tools. The CUE application scenario provided the design of a system supporting the collaboration of nurses in a hospital ward. The environment was supposed to be provided with a wireless communication infrastructure and technological support for the localization of nurses

³<http://www.mais-project.it>

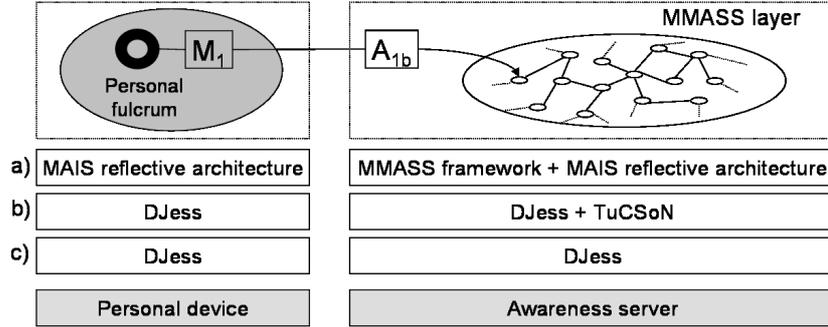


Fig. 5. A deployment diagram for the different analyzed middleware solutions for the CUE architecture.

(e.g. a network of RFID readers able to detect the presence or passage of a nurse wearing or carrying a tag). Nurses were supposed to have a PDA on which they could obtain information on the state of the ward, for instance pending request calls from patients. In addition, the system had to supply awareness information on the current state of other nurses and also support them in coordinating their activities: for instance, through the system a nurse had the possibility to “accept a call”, informing the others of her decision.

Figure 5a shows a deployment diagram of the prototype. In particular, the role of the MAIS reflective architecture in this specific application was to transparently update information about remote devices and objects on the computational node hosting the awareness server related to a given MMass layer and environmental aspect. To do so, the awareness server had to host a MAIS sub-layer comprising a “mirrored” version of the remote objects and the mechanisms supporting their updating according to a given update strategy. For instance, the information about the current state of a nurse is available on his/her personal device, but it is needed to update his/her position on a logical spatial representation layer in an awareness server; analogous considerations can be done for localization information. In a similar way information generated by a MMass layer can also be transparently updated on a remote computational node, such as a user device: from the user’s point of view, the developed application could be defined as a context-aware, rich form of instant messenger.

The information generated by users, devices and support infrastructures is exploited by M agents related to system users and other relevant modelled entities, that act as bridges between the actual environment and the representation of one of its aspects encapsulated in a MMass layer. The mirrored remote objects are accessed by A agents through the *input* primitive, introduced in section 3, and the effects of these operations can produce, for instance, the movement of the executing agent in the spatial structure of the layer, or the emission of a field. The MMass framework offers in fact computational elements and mechanisms matching the abstractions defined by the model (e.g. agents, layers in terms of graphs of sites, fields and the mechanisms for their diffusion), and represents thus a very simple way to effectively implement systems modelled through MMass. A more thorough

description of this framework can be found in [Bandini et al. 2006].

The integration between the MAIS reflective architecture and the MMass framework did not present particular issues; however some considerations aimed at improving this prototype can be done:

- the MAIS reflective architecture supported a transparent updating of remote information, and a very simple form of network communication. However this middleware is not just a communication infrastructure, even if in this specific case it was only exploited in this way; the possibility to employ a different and simpler middleware should be evaluated;
- the original usage of the MMass framework was in the context of Multi-Agent Based Simulation, and thus it presents specific mechanisms for the management of agents and system dynamics which required specific modifications for this application. In particular, in this case the environment cannot control the evolution of the agents, and in particular their activation, while this is the typical situation in a simulation application; the framework must rather reflect in a timely fashion the signals coming from the environment, triggering the movement of agents, the possible diffusion of fields and enabling agent perception, to support the updating of awareness information. To this end, agents were supplied with a thread of control of their own, and the environment had to be endowed with specific threads to manage field diffusion, agent reactions, and other MMass mechanisms;
- the modifications to the underlying model, described in section 3, required minor customizations of the framework, for instance allowing more than one agent to occupy the same site. Specific environments and diffusion mechanisms had to be defined to support the adoption of layers having a spatial structure including weighted edges.

5.2 Coordination Artifacts (TuCSoN)

Artifacts have been proposed as first-class abstractions to design and engineer environments for MASs. They can be viewed as computational modules providing a *function* that agents exploit to carry out individual or collective (social) activities. While agents typically represent the proactive, goal-driven components of a MAS, artifacts can be used to define *objects* of the MAS environment, elements that can be endowed with reactivity but lacking proactivity and goal-orientation. Sample artifacts are components used by agents to access legacy elements (e.g. a database) or to perform collective tasks, by observing and modifying a shared medium. In particular, coordination artifacts [Omicini et al. 2004] provide functionalities related to managed and constrained forms of agent interaction; typical examples of coordination artifacts are blackboards, tuple spaces and workflow engines.

In our case a single MMass layer can be viewed as a specific coordination artifact, providing the rules for the interaction among agents with respect to a specific aspect of their environment. The rules for the management of agent interaction in a CUE, introduced in section 4, are expressed in terms of MMass concepts and mechanisms, and they must be coded into a specific coordination artifact model. To do so, a specific model for artifacts must be selected. The TuCSoN (Tuple Centres Spread over Networks) coordination infrastructure [Omicini and Zambonelli 1999]

```

% Base environmental spatial representation, configuration and state
site(idSite)
adjacency(idSiteA, idSiteB, weight)
field-type(type, diff-type, diff-step)          % diff-type: down | up
presence(idAgent, idSite)
emission(f-type, site-src, intensity, payload) % agents must "out" this kind
                                                % of tuple to perform an emission
field-val(f-type, intensity, payload, curr-site, site-src)

% Basic field-diffusion mechanism
reaction( out(emission(f-type, site-src, intensity, payload), (
  in_r(emission(f-type, site-src, intensity, payload),
  out_r(field-val(f-type, intensity, payload, site-src, site-src))))).
reaction( out(field-val(f-type, intensity, payload, curr-site,
  site-src), (
  (adjacent(curr-site, diff-site, weight)),
  no_r(field-val(f-type, _, _, diff-site, site-src)),
  field-type(f-type, down, diff-step),
  intensity > diff-step * weight,
  diff-int is intensity - diff-step * weight,
  out_r(field-val(field-type, diff-int, payload, diff-site, site-src)))).

```

Fig. 6. Tuples representing the configuration of a MMass layer and the elements that make up its current state, and sample ReSpecT reaction rules managing field diffusion.

provides a clearly defined model for the specification of coordination artifact as well as a middleware for the design and engineering of infrastructures supporting agent interaction. The basic element of these infrastructures is represented by tuple centres, that build on the concept of tuple space and tuple-based coordination, originated by Linda [Gelernter 1985]. Tuple centres add to the basic tuple spaces the notion of a behavioural specification, that is, the possibility to specify the way a tuple centre reacts to the insertion of tuples by agents using it. TuCSoN adopts the ReSpecT language [Omicini and Denti 2001], which will be adopted in the remainder of this section to describe and implement the elements and mechanisms defined by the MMass model.

First of all, specific tuples must be defined so as to specify information on the configuration and state of the specific MMass layer. In particular, configuration information is related to the spatial structure of the layer and to the types of fields that can be diffused in it. State information specifies instead the position of agents in the spatial structure of the layer, as well as the presence of field values in its sites. Figure 6 shows an example of tuples representing this information. Having specified the format of tuples representing the spatial structure of the environment, it is now possible to encode the mechanisms defined by the MMass model into specific ReSpecT reactions. Figure 6 also shows an example of how basic reaction operations can be combined to perform field diffusion. The two reactions specify respectively the fact that a field emission triggers the beginning of a field diffusion and that the presence of a field in a given site should trigger its diffusion on adjacent sites on which the field has still not been diffused. The recursive triggering of the second reaction propagates the field diffusion to sites adjacent to the one on which the emission took place on which the field was not already diffused, and

the overall process ends when the field cannot be diffused anymore, due to the fact that the diffused value would be void. It must be noted that this mechanism is sound only for fields whose diffusion function provide a constant decrement of field intensity for every edge crossed in the field propagation process. The introduced mechanisms should thus only be considered a proof of concept, the results of a feasibility analysis, rather than a complete specification/implementation of MMass concepts and interaction model in TuCSoN. Figure 5b shows a deployment diagram for this architectural choice: in particular, the presence of a DJess module in the awareness server represents a simple way of supporting the communication between the A-agents (agents managing specific aspects of the awareness side of CUEs) present in the related MMass layer and remote M-agents, which represent the bridge between awareness and coordination aspects. In fact, as introduced in section 4, coordination aspects of CUEs are managed through specific modules implemented in DJess.

5.3 Distributed Inferential System (DJess)

The last considered architectural solution is based on DJess [Cabitza et al. 2005], a communication middleware whose main aim is to support the remote and transparent interaction of computational nodes hosting instances of the Jess rule based inference system. DJess is essentially an extension of Jess that adds a communication layer underneath its inferential capabilities. In particular, it is possible to define a web of inferential systems transparently sharing facts and rules among them across a network. For this specific application only the first functionality was exploited, in particular to support the possibility of A-agents to communicate with the related M-agents by means of the adoption of the *input* and *output* primitives introduced in section 3. This is necessary to support the interaction between the awareness and the coordination aspects of the CUE. It must be noted that the rule-sharing function was used in the modules that manage the coordination aspects of the CUE, that are implemented in DJess. The overall deployment diagram of this architecture is shown in figure 5c.

The role of DJess, in this framework, is similar to the one assumed by TuCSoN in the previous section: it was considered an alternative approach to implementation of MMass-based environmental representations for CUEs. In particular, it is possible to define fact templates for basic configuration and state of a specific MMass layer. Figure 7 shows how the `deftemplate` command can be used to define a specific structure for a given fact type. In this case, templates are defined for facts representing the building blocks of a MMass layer (i.e. sites and edges connecting them) and other information related to the configuration of the layer (e.g. the specification of a field type). Other templates are used to represent information about the current state of the environment, such as the position of an agent, the presence of fields in sites, and so on.

It is now possible to specify rules granting the layer the possibility to react to the stimuli represented by agent actions, such as a field emission: figure 8 shows two sample rules managing field diffusion. The first one generates the field value on the site in which the emission took place and starts the diffusion process. The second rule analyzes the surrounding of a site in which a field value was just created, it evaluates the possibility to propagate the field value, modulated according to the

```

; Base environmental spatial representation and layer configuration
(deftemplate site "A given MMass site"
  (slot Site))
(deftemplate adjacency "A given arc connecting two sites"
  (slot SiteA)(slot SiteB)(slot Weight))
(deftemplate field-type "Information related to a give field type"
  (slot F-type)(slot Diff-type) ; diff-type: down | up
  (slot Diff-step))

; Base facts for the state of the environment
(deftemplate presence "The position of an agent in the environment"
  (slot Agent)(slot Site))
(deftemplate emission "A fact related to a specific field emission"
  (slot F-type)(slot Agent)(slot Intensity)(slot Payload))
(deftemplate field-val "The value of a field at a given site in space"
  (slot F-type)(slot Intensity)(slot Payload)(slot Curr-site)(slot Site-src))

```

Fig. 7. Fact templates representing the configuration of the related MMass layer and its current state.

diffusion function (the field is propagated when it is not voided by the diffusion function and the destination site was not already visited by the diffusion process). By modifying the working memory, adding new field value, this rule recursively triggers further activations on different sites, stopping only when the emitted field has been propagated to all sites that must be reached according to the spatial structure of the layer and to the diffusion function of the related field type.

5.4 Discussion

The first prototypal implementation of a MMass based CUE, adopting the MAIS reflective architecture and the MMass framework, represented a useful experience that provided useful insights on the adequacy of the modeling approach as well as motivations and directions for a further exploration on different CUE implementation approaches. In particular, two middleware approaches to the design and implementation of environment oriented agent communication infrastructures (i.e. TuCSoN and DJess) were evaluated, in a sort of feasibility analysis. Both the approaches employ a sort of “scripting language” (ReSpecT and Jess) for the description of relevant entities having a role in defining the interaction space, as well as mechanisms for the definition of reactive behaviours of the communication infrastructure. The two languages were adopted to realize basic mechanisms supporting the implementation of MMass based CUEs (i.e. field diffusion). In order to effectively realize a complete prototype, such as the one based on the MMass framework and the MAIS reflective architecture, additional relevant aspects (e.g. timing and synchronization of agents’ actions as well as fields and related mechanisms) should be explicitly managed; however, these tools proved to be adequately expressive to manage this kind of application. Moreover, the presence of specific languages for the definition of the behaviour of a given communication infrastructure supports a *simple configuration* of the structure, rules and laws for the management of awareness in a specific CUE, provided that the basic common interaction mechanisms (i.e. those described by the MMass model) are present. This aspect is particularly

```

; Basic field-diffusion mechanisms
(defrule emission-to-value "Translates emission to a starting field value
and triggers its diffusion"
  ?emit<- (emission (F-type ?f) (Agent ?a) (Intensity ?i) (Payload ?p))
  (presence (Agent ?a) (Site ?s))
  =>
  (retract ?emit)
  (assert (field-val (F-type ?f) (Intensity ?i) (Payload ?p) (Curr-site ?s)
    (Site-src ?s)))
  (assert (diffusing ?f ?s)))
(defrule diffuse-from-site "Diffuse a field to adjacent sites still not reached by
the diffusion (down)"
  (diffusing ?f ?s)
  (field-val (F-type ?f) (Intensity ?i) (Payload ?p) (Curr-site ?c) (Site-src ?s))
  (field-type (F-type ?f) (Diff-type down) (Diff-step ?ds))
  (adjacency (SiteA ?s) (SiteB ?d) (Weight ?w))
  (not (field-val (F-type ?f) (Intensity ?i) (Payload ?p) (Curr-site ?d)
    (Site-src ?s)))
  (test (> ?i (* ?ds ?w)))
  =>
  (assert (field-val (F-type ?f) (Intensity (- ?i (* ?ds ?w)) (Payload ?p)
    (Curr-site ?s) (Site-src ?s))))
(defrule end-diffuse "Ends diffusion, when no other diffusion rule can be fired"
  (declare (salience -100))
  ?tmp<-(diffusing ?f ?s)
  =>
  (retract ?tmp))

```

Fig. 8. Sample Jess rules supporting field diffusion.

important due to the specific nature of this kind of application: in fact, even if a thorough analysis of methodological aspects of CUE definition must still be carried out, we envisage a specific role of *CUE designer* to manage the phases leading from an analysis of the specific application scenario (including the involved actors and devices, as well as the relevant facets of the specific notion of context), to the definition of a set of spatial and logical layers of representation, each one endowed with an internal structure, active entities and patterns of interaction among them.

It must also be noted that both TuCSoN and DJess also provide a degree of *dynamism* in the structure and definition of laws of the CUE that is not as easy to achieve with the approach based on the MMass framework and the MAIS reflective architecture: by altering simple tuples or facts it is in fact possible to modify the structure of the CUE or even the nature of specific interaction mechanisms (e.g. fields and related rules of diffusion) while the system is running, for instance to better reflect changes in the modeled reality. It is thus also possible to envisage a role of *CUE manager*, a function that could be delegated to a human but that could also be (at least partly) carried out by a software agent.

6. CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper has described a Multi-Agent approach to the modelling and design of CUEs, that are environments that support collaboration among persons in a context of ubiquitous computing. In particular, the paper has shown how results of the

research in the topic of MAS environment has provided both modeling abstractions and concrete computational supports for the analysis, design and engineering of CUEs. In particular, the MMass model was applied to represent and to manage several types of awareness information (both physical and logical contextual information), which is an essential part of a CUE. Moreover, three different approaches to the design and engineering of CUEs (the MMass framework integrated with the MAIS reflective architecture, the TuCSoN coordination infrastructure, DJess) were introduced and evaluated.

Future works are mainly aimed at identifying a concrete situation for a prototypal implementation adopting DJess and an effective evaluation of a CUE. This effort will integrate the results of the presented work with the parallel research on the coordination side of CUEs. This experience will lead, on one hand, to further evaluate the adequacy of the proposed approach, but it will also allow to explore methodological issues related to the analysis, design, implementation and management of solutions for this innovative field of community-oriented ubiquitous-computing systems that we consider a necessary direction for the development of the new generation solutions in this area of application.

REFERENCES

- ADORNI, M., ARCELLI, F., BALDONI, R., ALIMONTA, A., LOCATELLI, M., LOSI, P., MARCHETTI, C., MILANI, A., QUERZONI, L., RAIBULET, C., SARINI, M., SIMONE, C., TISATO, F., PIERGIOVANNI, S. T., VIRGILITO, A., AND VIZZARI, G. 2006. *Mobile Information Systems: Infrastructure and Design for Adaptivity and Flexibility*. Springer, Chapter Middleware and Architectural Reflection, 81–114.
- BANDINI, S., MANZONI, S., AND SIMONE, C. 2002. Heterogeneous Agents Situated in Heterogeneous Spaces. *Applied Artificial Intelligence* 16, 9-10, 831–852.
- BANDINI, S., MANZONI, S., AND VIZZARI, G. 2004. Situated Cellular Agents: a Model to Simulate Crowding Dynamics. *IEICE Transactions on Information and Systems E87-D*, 3, 669–676.
- BANDINI, S., MANZONI, S., AND VIZZARI, G. 2006. Towards a Platform for Multilayered Multi-Agent Situated System Based Simulations: Focusing on Field Diffusion. *Applied Artificial Intelligence* 20, 4–5, 327–351.
- BARDHAM, J. E. 2005. Activity-Based Computing: Support for Mobility and Collaboration in Ubiquitous Computing. *Personal and Ubiquitous Computing*.
- BEAUDOUIN-LAFON, M. 1999. *Computer Supported Co-operative Work*. Trends in Software, vol. 7. John Wiley and Sons.
- BELLIFEMINE, F., BERGENTI, F., CAIRE, G., AND POGGI, A. 2005. JADE - a Java Agent Development Framework. In *Multi-Agent Programming*, R. H. Bordini, M. Dastani, J. Dix, and A. E. Fallah-Seghrouchni, Eds. Multiagent Systems, Artificial Societies, and Simulated Organizations, vol. 15. Springer, 125–147.
- CABITZA, F., LOCATELLI, M. P., SARINI, M., AND SIMONE, C. 2006. Casmas: Supporting Collaboration in Pervasive Environments. In *Pervasive Computing and Communications 2006 (PerCom 2006)*. IEEE, 286–295.
- CABITZA, F., LOCATELLI, M. P., AND SIMONE, C. 2006. Cooperation and Ubiquitous Computing: an Architecture Towards their Integration. In *COOP 2006*. IOS Press.
- CABITZA, F., SARINI, M., AND DAL SENO, B. 2005. Djess - a Context-sharing Middleware to Deploy Distributed Inference Systems in Pervasive Computing Domains. In *Pervasive Services, 2005 (ICPS '05)*. IEEE, 229–238.
- CAMPBELL, R. H. 2005. Beyond Global Communications: the Active World. In *PerCom*. 211.
- CHEN, G. AND KOTZ, D. 2000. A Survey of Context-Aware Mobile Computing Research. Tech. Rep. TR2000-381, Dept. of Computer Science, Dartmouth College. November.

- DOURISH, P. 2004. What We Talk About When We Talk About Context. *Personal and Ubiquitous Computing* 8, 1, 19–30.
- FAVELA, J., RODRÍGUEZ, M., ALBA, M., AND MORÁN, A. L. 2002. Supporting Opportunistic Interactions with People, Resources and Agents in Ubiquitous Environments. In *Mobile Ad-hoc Collaboration Workshop at CHI 2002*. Minneapolis, Minnesota, USA.
- FERBER, J. 1999. *Multi-Agent Systems*. Addison-Wesley.
- GELERTNER, D. 1985. Generative Communication in Linda. *ACM Trans. Program. Lang. Syst.* 7, 1, 80–112.
- LECH, T. C. AND WIENHOFEN, L. W. M. 2005. AmbieAgents: a Scalable Infrastructure for Mobile and Context-Aware Information Services. In *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005)*. ACM Press, 625–631.
- LOCATELLI, M. P. AND SIMONE, C. 2006. Supporting Care Networks through an Ubiquitous Collaborative Environment. In *Smart Homes and Beyond*, C. Nugent and J. Augusto, Eds. Assistive Technology Research, vol. 19. IOS Press.
- MAMEI, M. AND ZAMBONELLI, F. 2004. Programming Pervasive and Mobile Computing Applications with the TOTa Middleware. In *International Conference on Pervasive Computing and Communications 2004 (PerCom 2004)*. IEEE Computer Society, 263–276.
- MAMEI, M. AND ZAMBONELLI, F. 2006. Augmenting the Physical Environment through Embedded Wireless Technologies. In *The Second International Workshop on Environments for Multiagent Systems, E4MAS 2005*, D. Weyns, H. V. D. Parunak, and F. Michel, Eds. Lecture Notes in Computer Science, vol. 3830. Springer-Verlag, 235–250.
- OMICINI, A. AND DENTI, E. 2001. From Tuple Spaces to Tuple Centres. *Science of Computer Programming* 41, 3, 277–294.
- OMICINI, A., RICCI, A., VIROLI, M., CASTELFRANCHI, C., AND TUMMOLINI, L. 2004. Coordination Artifacts: Environment-Based Coordination for Intelligent Agents. In *3rd international Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, ACM Press, 286–293.
- OMICINI, A. AND ZAMBONELLI, F. 1999. Coordination for Internet Application Development. *Autonomous Agents and Multi-Agent Systems* 2, 3, 251–269.
- SATYANARAYANAN, M. 2001. Pervasive Computing: Vision and Challenges. *IEEE Personal Communications* 8, 4 (August), 10–17.
- SCHMIDT, K. 2002. The Problem with ‘Awareness’: Introductory Remarks on ‘Awareness in CSCW’. *Computer Supported Cooperative Work* 11, 3-4, 285–298.
- SIMONE, C. AND BANDINI, S. 2002. Integrating Awareness in Cooperative Applications through the Reaction-Diffusion Metaphor. *Computer Supported Cooperative Work* 11, 3-4, 495–530.
- TAO, J., TAN, T., AND PICARD, R. W., Eds. 2005. *Affective Computing and Intelligent Interaction, First International Conference, ACII 2005*. Lecture Notes in Computer Science, vol. 3784. Springer.
- WEISER, M. 1993. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM* 36, 7 (July), 75–84.
- WEISER, M. AND BROWN, J. S. 1996. Designing Calm Technology. *PowerGrid Journal* 1, 1.
- WEYNS, D., MICHEL, F., AND PARUNAK, H. V. D., Eds. 2005. *Environments for Multi-Agent Systems, First International Workshop (E4MAS 2004)*. Lecture Notes in Artificial Intelligence, vol. 3374. Springer-Verlag.
- WEYNS, D., MICHEL, F., AND PARUNAK, H. V. D., Eds. 2006. *Environments for Multi-Agent Systems, Second International Workshop (E4MAS 2005)*. Lecture Notes in Artificial Intelligence, vol. 3830. Springer-Verlag.
- WEYNS, D.,OMICINI, A. ODELL, J. 2007. Environment as a First Class Abstraction in Multiagent Systems. *Autonomous Agents Multi-Agent Systems* 14, 1 5–30.
- ZAMBONELLI, F. AND PARUNAK, H. V. D. 2002. Signs of a Revolution in Computer Science and Software Engineering. In *Engineering Societies in the Agents World III (ESAW2002)*, P. Petta, R. Tolksdorf, and F. Zambonelli, Eds. Lecture Notes in Computer Science, vol. 2577. Springer-Verlag, 13–28.