SMART ENVIRONMENTS AS AGENTS WORKSPACES

Andrea Omicini, Alessandro Ricci
ALMA MATER STUDIORUM – Università di Bologna
Via Venezia 52, 47023 Cesena, Italy
{andrea.omicini,a.ricci}@unibo.it

Giuseppe Vizzari
Dipartimento di Informatica, Sistemistica e Comunicazione
Università degli studi di Milano–Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
giuseppe.vizzari@disco.unimib.it

Abstract

The pervasive computing scenario provides a diffused presence of technological appliances distributed in the environment and interacting by means of wired or wireless networks. These computational units should be able to fruitfully exploit interactions with other components (e.g. information sources) in order to supply context aware services and support advanced forms of interaction among users. The aim of this paper is to show how models and frameworks for Multi-Agent Systems (MAS) can be exploited to design and implement pervasive computing systems, focusing on the Agent and Artifact (A&A) approach. Artifacts are a conceptual, formal and computational framework supporting the realization of function–oriented elements of a MAS. After a description of a scenario, the paper briefly introduces the A&A framework and shows how it can be used to define an infrastructure supporting complex interaction schemes provided by the scenario.

Keywords: MAS environments, context-awareness

1 INTRODUCTION

The current trend of technological innovation, providing an ever growing availability of wireless communication infrastructures and a miniaturization of computational devices, allows to foresee a scenario in which novel distributed applications (that were not conceivable a few years ago) will not only be feasible, but necessary to motivate the investments on said technological infrastructures. Devices and hosted applications, must be able to opportunistically exploit the growing possibility to interact with their environment, offering communication facilities but also additional services (e.g. storage, printing, resource to other hardware/software resources). While the technology is evolving very rapidly computational models, software infrastructures and innovative applications are still needed to really investigate end understand the new situation [23]. Much attention to the issues raised by this scenario is being paid by the research community, and some relevant results in the area have been developed in the context of Multi-Agent Systems (MAS) research. Some relevant examples are represented by the TOTA middleware for context–aware pervasive systems [9], but also by relevant experiences in the Computer Supported Collaborative Work area [17], and even complete hardware/software solutions employ (intelligent) agents [6] to manage flows of information and services to end users. Autonomous agents represent in fact a natural way of conceiving systems, such as those provided by the pervasive computing scenario, made up of (at least partially) autonomous entities situated in an environment which influences their possibility to act and interact. The central aim of this paper is to show that research on MASs represents a source of abstractions, models and instruments (both conceptual and computational) for the design and development of complex context-aware pervasive systems.

The above introduced scenario requires the possibility of the overall system to trigger and to possibly manage a set of autonomous but interdependent and related actions carried out by distinct entities, according to both contextual factors and to the entities' individual goals and tasks. According to [8], coordination is "the act of managing interdependencies between activities", so this scenario seems relevant to Coordination Theory. Ciancarini's general coordination model [3] comprises entities, a coordination medium and coordination laws. But due to the dynamic nature of the scenario (e.g. some of the involved entities are hosted in devices that can physically move, join or leave the network) and to the fact that activities are set up according to the current context of the involved entities, the coordination model should consider the fact that its three components are dynamically determined by the current situation of the environment associated to the overall system.

Building on these considerations this paper proposes agent *workspaces* as first class abstractions to model, design and implement smart environments. Workspaces represent one of the fundamental elements of a programming model called A&A (Agents and Artifacts) [15] which aims at directly modelling and engineering working environments in the context of cognitive multi-agent systems. This effort can be set in a wider perspective depicted by recent efforts in AOSE (Agent-Oriented Software Engineering) that remark the fundamental role of the environment for the engineering of MAS [20]. Workspaces are suitable structures aggregating artifacts that either encapsulate resources or reify elements of an actual working environment that we want to represent in order to create contexts for coordinating people, supply awareness information, support the discovery of services and resources. The A&A programming model also provides the notion of *agent body* as a medium for through which an agent *mind* can sense and affect a working environment. In this specific scenario, only some entities are associated to a synthetic mind, while for human actors (or human driven computer application) the body serves as a *proxy* inside the workspace. The following section will discuss related works, while Section 3 will briefly introduce a specific scenario which will be used to show how the workspace approach can be suitably adopted to model, design and implement an infrastructure for smart environments. The notion of workspace, artifacts and agent bodies as building blocks for their construction will be introduced in Section 4, and Section 5 will describe the introduced scenario in terms of workspaces in order to show how they can be applied in this context. Conclusions and future developments will end the paper.

2 RELATED WORKS

A first category of research efforts and systems providing the adoption of agents and MASs for the design and implementation of context-aware pervasive systems mainly adopted agents and agent oriented infrastructures as design abstractions and middleware.

The AmbieSense project[1], for example, was aimed at "providing relevant information to the right situation and user" in a pervasive computing scenario; it adopted agents [6] to reify and encapsulate significant elements of the overall system that had to interact to manage, supply or exploit contextual information (e.g. user's context, contents and services that are present in the environment, recommender service to suggest relevance of a given content to a user and his/her context). Contextual information was supplied by an infrastructure made up of tags, used to determine the proximity of a device to a relevant reference point in the environment, and a common wireless communication facility.

A different experience described in [17] was aimed at facilitating the coordination and collaborative work of employees in a hospital ward. In this case, agents we adopted as wrappers managing the access to specific sources of information (e.g. legacy database servers) or shared resources and devices (e.g. large screens and public displays, maps). As for the AmbieSense project, also these agents manage, exchange and exploit context information to carry out their actions.

The notion of context, however, is multifaceted and characterized by different aspects, ranging from the current spatial position of an entity, to his/her preferences; but they also depend of the activity that is currently being carried out, to the point that something may or may not be contextually relevant in relation to a particular activity [5]. In this framework, agent models can provide not only an infrastructural support to the realization of distributed systems managing contextual information, but also abstractions and mechanisms supporting the representation and management of non trivial notions of context.

The aforementioned TOTA approach, for instance, represents an answer to the need of considering the spatial context of devices in order to devise spatially grounded interaction and coordination schemes by means of a computational field based approach [11]. Spatial aspects represent indeed a relevant element of what is generally conceived as the context of an entity, but there can be many different ways of representing them, with distinct levels of granularity and goals. A typical benchmark application in the pervasive computing area is described by the *smart environment* scenario (see, e.g., [22]), and it provides context and location aware services such as automatic discovery and exploitation of the resources that are nearest to the client and most suited to its requirements and needs. In particular, in this case, the spatial representation requires abstractions related to offices, meeting or printer rooms and so on. In other words, in this scenario it is generally not relevant to have a finer level of granularity (e.g.

[1] http://www.ambiesense.net/

where *exactly* an actor or a resource is located inside a room). To model this situation suitable abstractions are needed to reflect this representational requirement, and specific mechanisms to construct infrastructures made up of these abstractions and to govern the information flows must be defined. Some related works that describe relevant efforts in this direction are represented by the HiMAT system [4] and by the awareness component of the CASMAS model [2, 7]. The former provides the interpretation of the Internet as a collection of hierarchical domains and it represents portions of it as a topology in which mobile agents are situated, can move, perceive their environment and act to carry out specific tasks (such as the search for information in a distributed setting). HiMAT maps domains (and sub-domains) to TuCSoN [13] tuple centres, that encapsulate and give access to the resources related to that type of "spatial" abstraction, according to specific access rules. In CASMAS, the awareness management module is also based on topologies, that are the basic structure of layers representing specific aspects of physical environments and social communities. These topologies are exploited to guide the distribution of awareness information about the state (and location) of human actors, electronic devices, other resources present in the environment. The overall goal is supply actors situated in it contextual information that can be exploited to carry out specific tasks requiring interaction among them.

3 A MOTIVATING EXAMPLE: THE SMART ENVIRONMENT SCENARIO

The specific smart environment scenario that is discussed in this section provides an infrastructure supporting the activities of different people working or frequenting a University Department. To do so, we will not only consider human actors, but also resources (digital or not) that are relevant to their activities, as well as some specific contextual information.

This organization is situated in a building covered by a wired and wireless network infrastructure. In particular, we consider interactions among people (to simply exchange messages, but also to arrange meetings and perform specific patterns of coordination) as well as interactions among people and services or resources (both technological and physical). All interactions are context dependant, where the term context refers to: the abstract *localization* of the entities involved in the interaction; their *role* in the organization (which defines specific rights of access to resources and services, policies for interactions, and organizational aspects in general); *additional elements* related to their profile and current state.

The first element does not represent a problem for relevant static elements of the environment (such as rooms) or objects (such as department's laser printers) whose positions do not change often. On the other hand, such system requires to track the positions of people moving in the environment. While outdoor localization can be easily performed by means of GPS devices, the typical error (in some cases over 2 meters) characterizing these devices is not acceptable for an indoor localization. However, to effectively be able to tell in which room (or corridor) a person is located, an infrastructure of RFID readers could be set up in order to reveal the passage of people having specific "smart badges" through specific gateways such as relevant doors (see, e.g., [19] or [10] for this kind of RFID application). Of course, specific "hot spots" providing access to the smart environment to people not having these means of localization (such as occasional visitors) could also be considered. Organizational roles and policies represent abstractions that are commonly used in this kind of scenario. For instance, in order to reserve a meeting room provided with an overhead projector one should ask to a specific secretary, which controls the resource timetable. Of course, not everyone is allowed to reserve such a resource (the reserver's role must be professor), and not all requests have the same relevance according to the reasons of the meeting. The last contextual element is related to the profile of a user, a resource or a service involved in an interaction, and his/her/its current state. In particular, these elements are relevant to tell whether an interaction should take place in a specific moment or if it should be better to postpone it. For instance, people involved in an important meeting should not be disturbed with interactions which do not require an immediate management.

3.1 Human actor-resource interaction

The smart environment represents a uniform access to the discovery of resources and services available in the Department. Traditional electronic resources and services (such as printers) are endowed with additional computational capabilities that allow them to advertise themselves by "publishing" their profile in the smart environment. The latter can thus be inspected by users searching for particular resources or services, and the environment is able to route them towards the most suitable one, given their specific request and context. This means that users' computational devices must also be provided with additional computational capabilities (e.g. a specific software layer) supporting interaction with the environment. A simple example of this facility is the possibility of discovering printers available in the Department, being suggested which is considered most suited to the specific printing task, due to contextual considerations, such as type of printed documents (e.g. color or black and white), time of day, location of the printer and distance from user's current location. The resources that are advertised and (at least partly) managed by the smart environment, however, are not only of technological nature. For instance, rooms themselves as well as overhead projectors, beamers, notebooks shared by the Department personnel for their activities, can be considered as resources. The management of the reservations, including the enforcement of the related organizational rules, is delegated to the environment itself. While routine reservation activities can be simply managed, the environment also supports the resolution of conflicts: users trying to reserve a resource are notified that the request generates a conflict and according to the situation the environment reacts in a different way. For instance, it checks for alternatives that are compatible with the request (e.g. a user requiring a specific beamer could be proposed a different one if the required one is already reserved), and if the request cannot be managed it uses organizational policies to decide what to do (e.g. fail reporting to the user, start a negotiation between users that originated the conflicting requests).

3.2 Interaction among human actors

The smart environment also enables new forms of interaction among users, ranging from simple context–aware message passing, to possible virtual notice boards (or environment based bulletin board systems), but also supporting more complex forms of coordination based for instance on specific workflows. With the expression context–aware message passing we refer to a set of possible heterogenous interaction schemes that are available to users. First of all, users may decide to publish information related to their state and context (e.g. current activities and location), and this information can be inspected by other users to decide whether or not to try to establish an interaction. In this case, a "synchronous" form of interaction could start (e.g. a sort of chat session, but also voice over IP calls), and it could also involve more than two communication partners. However, according to the personal configuration of the interaction with the environment, users in specific contexts can be shielded by incoming communication requests. More traditional forms of interaction (electronic mail service) are also available. In addition to these services, the smart environment also allows users to mark places with messages that can be observed by other people present in the same area (sort of virtual sticky notes) or by other people inspecting the environment (for instance looking for forthcoming seminars held in the Department). More complex forms of interaction provide the coordination of several users involved in specific activities, such as the assignment of stages to students. First of all, a professor of the Department proposes a stage program, which can be inspected by students. When a student applies for the stage, a secretary performs a first check by inspecting the student curriculum, then a commission made up of several Professors has to check the stage program in order to formally accept the stage. The environment keeps track of the procedure evolution, and signals the involved actors when their actions (inspection of the documents and decision) are needed.

4 WORKSPACES AND ARTIFACTS

4.1 Artifacts and workspaces

A working environment in A&A is defined as the part of the MAS that is designed and programmed by MAS programmers and dynamically instantiated and used by agents to support their working activities. A working environment is conceived as a dynamic set of artifacts, organized in workspaces. The latter are the logical containers of artifacts, useful to define the topology of the working environment. A workspace provides a notion of locality for agents: an agent can participate to one or multiple workspaces, and can work with those artifacts belonging to such workspaces only (see Fig. 1 – right). This concept can be used to define the distribution model of an application at an abstract level: a working environment – which corresponds to a (possibly distributed) application or MAS – can account for one or multiple workspaces, and an individual workspace can be either mapped onto a single node of the network or spread among multiple nodes. The notion of artifact is the core abstraction of the programming model: it is meant to represent any passive entity belonging to the working environment – hence existing outside the agent mind – that is created, shared & used (and eventually disposed) by agents to carry on their activities, in particular social ones. An artifact (type) is typically meant to be explicitly designed by MAS engineers so as to encapsulate some kind of function, here synonym of "intended purpose". More on this notion can be found in [14].

An abstract representation of an artifact is shown in Fig. 1 (left) and it is very similar to artifacts as found in human society: artifact function – and related artifact behaviour – is partitioned in a set of operations, which agents can trigger by acting on artifact usage interface. The usage interface provides all the controls that make

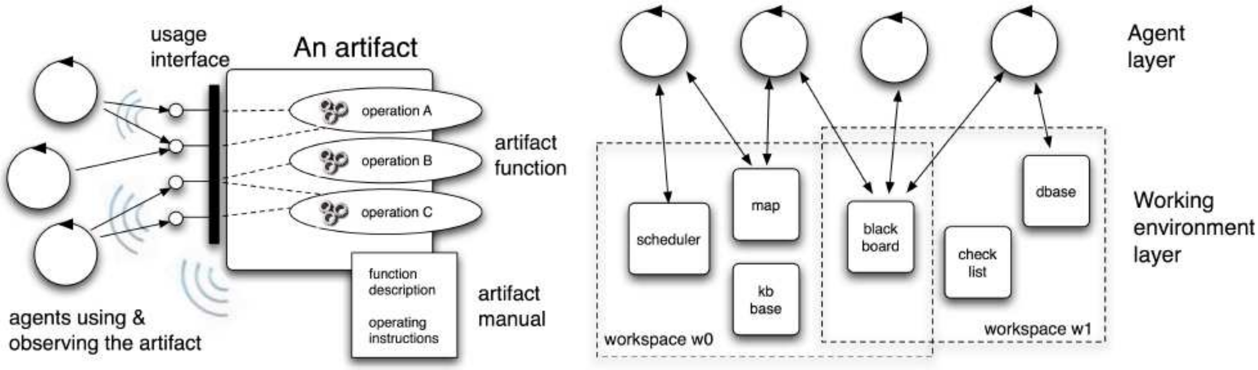

Figure 1: *(Left)* Abstract representation of an artifact. *(Right)* Abstract representation of a working environment with two workspaces, with some artifacts of different kinds inside.

it possible for an agent to interact with an artifact, triggering and controlling the execution of operations and perceiving observable events generated by the artifact itself, as a result of operation execution and evolution of its state. Such a model strictly mimics the way in which humans use their artifacts: a simple example is the coffee machine, whose usage interface includes suitable controls – such as the buttons – and means to make (part of) the machine behaviour observable – such as displays – and to collect the results produced by the machine – such as the coffee can. Analogously to the human case, in A&A each artifact type can be equipped by the artifact programmer with a manual composed essentially by the *function description* – as the formal description of the purpose intended by the designer – , the *usage interface description* – as the formal description of artifact usage interface and observable states – , and finally the *operating instructions* – as the formal description of how to properly use the artifact so as to exploit its functionalities. Usage interface description is just a description of the controls – analogously to the description of which buttons and displays the coffee machine has – , while operating instructions describe the usage protocols, to exploit such controls. Such a manual is meant to be essential for creating open systems with intelligent agents that dynamically discover and select which kind of artifacts could be useful for their work, and then can use them effectively even if they have not preprogrammed by MAS programmers for the purpose. It is worth remarking here the similarities and differences between the artifact abstraction and the general notion of service. On the one side, artifacts can be seen indeed as a natural way to implement services, without the need to agentify them as typically happens in service-based agent approaches. On the other side, typically services are conceived as a purely architectural concept: here instead, artifacts are meant to be the basic building block complimentary to the agent abstraction defining a new extended agent programming model. Then, typically a service can be designed and implemented on top of multiple artifacts and agents.

4.2 Agent bodies

In this overall picture, nothing is said about the specific (cognitive) model of the agent: actually A&A is meant to be orthogonal to this aspect: agents are simply conceived as autonomous entities executing some kinds of working activity, either individually or collectively – typically in order to achieve some individual or social goal, or to fulfill some individual or social task. Such activities – from an abstract point of view – are seen as the execution of sequences of actions, which according to the A&A model can be roughly classified as: (i) internal actions, (ii) communicative actions, involving direct communications with one or more agents through some kind of ACL, and (iii) pragmatical actions, as interactions within the working environment that concern construction, sharing, and use of artifacts. Despite of their specific model / architecture, in order to execute actions over an artifact and perceive observable events, agents must be situated in the working environment: for this purpose, the notion of agent body is introduced. The agent body functions as the medium through which the agent mind – i.e. the part that is designed and programmed according to a certain kind of cognitive model / architecture – can sense and affect a working environment. Agent bodies are essential to decouple – for engineering purposes – the agent mind from the working environment in which the agent is situated, so as to be able to use A&A with different kinds of programming model for the agent mind, including both intelligent agent and reactive / mobile / general agent models.

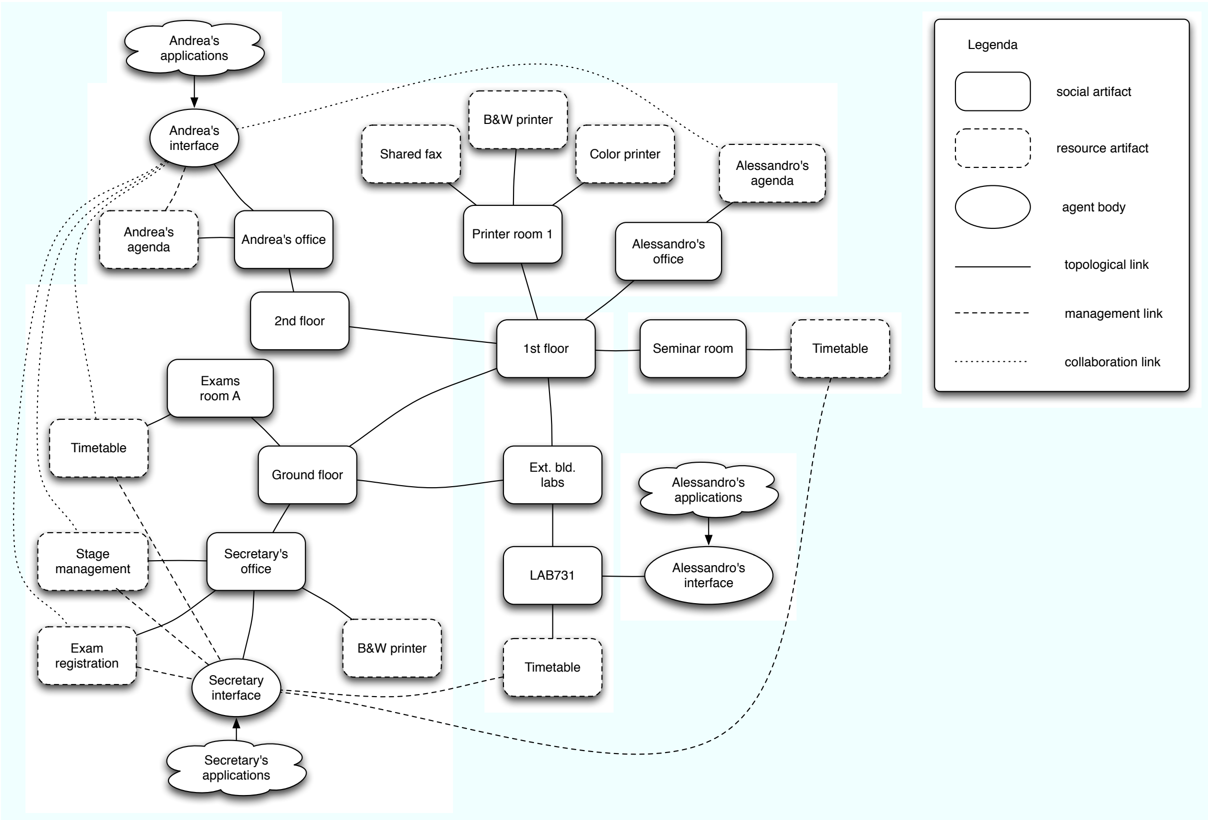

Figure 2: A diagram showing a partial modelling of the workspace related to the introduced smart environment through various artifacts.

An infrastructure supporting the implementation of environments based on the A&A approach (CArtAgO) is described in [16] and it provides the basic elements of this programming model.

5 WORKSPACE WIEW OF THE SMART ENVIRONMENT

5.1 Structuring and exploiting the workspace

Section 3 has briefly introduced a particular smart environment scenario. Figure 2 depicts a particular example of this kind of smart environment modeled according to the A&A approach. In particular, this approach provides the encapsulation of resources and services into specific artifacts that can be suitably connected to define a workspace. The latter can be accessed by agents' bodies and thus, in this specific scenario, by synthetic agents as well as proxies of humans or human driven computer applications. As discussed in [21], different types of artifacts can be defined according to their function in the workspace. In particular, in this case we adopted *resource artifacts* to wrap relevant objects of the environment that can be exploited by agents and users (e.g. faxes and printers) and to reify particular services and functions offered by the smart environment.

In particular, for instance, we defined an artifact related to the timetable of a seminar room; user applications and agents can thus access the workspace through their respective bodies and they can identify this kind of artifact inspecting the workspace. Andrea can thus inspect this artifact and book the seminar room for a given period of time; moreover he could indicate that the information about the seminar to be held in the room should be advertised in the Department the day of the seminar, so as to provide a reminder to those who might be interested. The information associated to the booking could thus be propagated along the links from the seminar room timetable artifact to the other linked artifacts, that can be defined as *social* since their main aim is to support and regulate the interaction among agent bodies.

Later on Andrea is engaged with some exams, and he will have to use another resource artifact, that is, a register for the results of the exams that was previously created by the Department Secretary. The latter will be able to eventually close it for persistent storage once verified that Andrea correctly filled it with the required data. In this case, thus, the artifact must not only manage the enclosed information and provide suitable functionalities for the different roles, but it should also provide some form of coordination function among the actors involved in its management.

At the end of the day, Andrea is about to leave the Department, but before leaving he wants to print the last chapter of the thesis of one of his students and wants to check if Alessandro will be available the following morning to discuss about it. First of all he can inspect the workspace, searching for artifacts wrapping printers that provide the current state of the spooling queues, and thus he can select the best printer according to his current needs (i.e. to obtain a reasonably readable draft printed very quickly). Once selected the best printer and launched the print operation, he will select the artifact encapsulating Alessandro's agenda. Of course he will only be able to visualize those details that the specific access control rules defined by Alessandro to secure the access to a private resource allow him to see. In this particular case, Alessandro decided to grant the members of his work group a read only access to his agenda, shadowing the details of events and meetings not described as 'private'. Andrea will thus know that the following morning Alessandro will be busy with a lesson. In this pattern of interaction, the need to consider the role of an agent body (and thus the role of an agent or human actor) in the workspace in order to determine allowed actions is even more significant than in the exam registration, since this kind of access to resources has to deal with privacy issues.

5.2 Specifying workspaces with A&A ReSpecT

To show how the above described scenario can be effectively realized by means of workspaces encapsulating suitably interconnected artifacts we will now describe how A&A ReSpecT [12] (a revised version of the ReSpecT language for the specification of TuCSoN programmable tuple centres behaviours) can be employed to manage the flow of context awareness information in the workspace.

In particular, we could model information related to events that must be forwarded to those artifacts that could be interested as a tuple `event(information,topic)`. The `topic` field allows to specify that the tuple must only be sent to those artifacts for which a specific interest in that topic has been expressed, for instance by means of another tuple `concerned_with(art_a,art_b,topic)`. While the last tuple is used to express the existence of a specific (unidirectional) link between artifacts `art_a` and `art_b`, a reaction rule exploiting it and the format for `event` information must be defined to actually carry out the tuple distribution, and in particular it could be specified as follows:

```
reaction(out(event(information, topic)),(
   rd(concerned_with(art_a, art_b, topic))
   art_b@node ? out(event(information, topic))
   ).
```

In this example we consider that the artifact `art_b` is deployed to the `node` server reachable in the network from `art_a`, that could be hosted by a different machine.

By means of this kind of mechanisms it is also possible to define collaboration links providing the replication of information generated in a specific resource artifact, such as an agenda, into other artifacts, for instance related to individuals which (due to an ongoing collaboration with the owner of the agenda) could be interested in that form of awareness information. It must also be noted that this mechanism is configurable and manageable, as in part due to the presence of tuples, which can be then modified, for instance by the owner of the agenda which can thus manage the grant of rights of access to that resource.

While in this way the distributed information is not modified, the introduction of topological information could require to consider the distances among linked artifacts, in order to model some sort of spatial context awareness. For instance, the description of resources could be modelled as a tuple `resource(state,distance)`, where the second field represents the distance from the current artifact. Adjacency relations among artifacts could be modeled by means of tuples `adjacent(art_a,art_b)`, and thus a more generic kind of link that the previous one. The reaction rule that generates the information diffusion can specified as follows:

```
reaction(out(resource(state,distance)),(
    rd(adjacent(art_a,art_b))
    distance_diff is distance+1
    art_b@node ? resource(state,distance_diff))
    ).
```

The introduced links and mechanisms support thus the definition of multiple layer graph structures (similar to those provided by some field based approaches, such as [1]) insisting on the same set of artifacts, in order to represent different information distribution policies. It must be noted that by means of specific reaction rules, such as the one specified for the distribution of resource related tuples, field based schemes of diffusion could be defined.

5.3 Interfacing applications with workspaces

While the previous section has shown how the workspace internal structure can be realized, the interfaces allowing agents minds (or applications acting on behalf of human users) can access workspaces and the enclosed artifacts will now be introduced. Figure 3 shows a deployment diagram representing the main different types of elements of the system. In particular, the central element of the smart environment can be hosted by a single server, on which the CArtAgO Infrastructure is installed as an execution platform hosting the various artifacts composing the workspace related to the smart environment. Artifacts could also be distributed on different computational nodes on a computer network, each characterized by the presence of the CArtAgO Infrastructure.

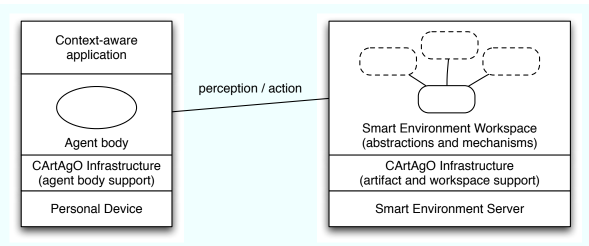

Figure 3: A deployment diagram showing the main different types of elements of the system, namely, devices endowed with agent bodies, on the left, and infrastructure management nodes, on the right.

As shown in Figure 2 to access a workspace requires the adoption of an agent body, that is a container of effectors to perform actions and a dynamic set of sensors to collect stimuli from the working environment. A subset of the infrastructure is thus also necessary for devices hosting agents bodies. In particular, the interaction among agent bodies and the workspace takes place through perceptions and actions. With reference to perception, different types and number of sensors can be attached to an agent body, each one identified by its own sensor identifier, denoted by SID; each sensor can be focused (either in a persistent way, though the `focus` operation or as a "one–shot" action, through the `sense` action) to a specific artifact of the workspace. Artifacts, moreover, can be inspected to obtain operations (in terms of function descriptions and operating instructions) that can be carried out in it through the general `execOp` action. It it not the aim of this paper to provide a complete description of the A&A programming model (such a description can be found in [15]), however the main primitives of this API are reported in Figure 3.

By employing this API specific agent bodies can be realized in order to (i) interface and integrate in the system existing applications (e.g. personal information management systems, for the management of calendars) or also (ii) to realize new ones, if necessary, to effectively exploit the above described smart environment to support innovative context-aware services. In addition, other existing resources like shared calendars will be encapsulated by proper artifacts, instead of being wrapped by some "service–agent".

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper presented an approach based on the A&A programming model to the analysis, design and development of workspaces as integrated environments for Multi-Agent Systems. The approach has been applied to a smart environment scenario, which provides the integration of resources and services that are made available by the environment itself, which also supports context aware forms of interaction among users, ranging from simple message passing schemes to complex workflows providing the collaboration of several agents. The paper did not present a complete modelling of the whole scenario, but was aimed at showing the adequacy of the approach, testing the expressiveness of the selected support infrastructure, selecting specific examples of paradigmatic patterns of interaction. The described workspace structure, the abstractions defined to represent events and relevant aspects of the smart environment, as well as the introduced mechanisms to manage the distribution of contextual information in the workspace, were successfully tested exploiting the TuCSoN infrastructure.

A new infrastructure (CArtAgO) supporting the implementation of environments based on the A&A approach is described in [16] and it provides the basic elements of this programming model. Further experiments on the smart environment scenario will employ this infrastructure. Some of the most relevant future developments of CArtAgO will concern (i) artifact composition – support for linking together existing artifacts to compose a complex workspace, (ii) integration of existing services as kinds of artifacts – in particular it would be useful to encapsulate in an artifact existing coordination facilities such as those offered by TuCSoN, (iii) integration of a comprehensive access control model – drawing on previous work on this topic in the development of the TuCSoN infrastructure, it is planned to

Artifacts construction, discovery and disposal

- createArtifact(Name,TypeID,Conf,WspID) - to create a new artifact called Name, of type TypeID, with a starting configuration represented by Conf, in the workspace identified by WsdID;
- getArtifactID(Name,WspID):AID - to get the identifier of an existing artifact, given its name;
- disposeArtifact(AID) - to dispose an existing artifact.

Workspace construction, discovery and disposal

- createWorkspace(Name)- to create a new workspace called Name in the working space;
- getWorkspaceID(Name):WspID - to obtain the identifier of an existing workspace, given its logical name;
- disposeWorkspace(WspID) - to dispose an existing workspace.

Artifact use and observation

- execOp(AID,Op(Name,Params),SID) - to trigger the execution of an operation on an artifact given its identifier AID, specifying operation name and parameters, and the sensor, identified by SID, to collect corresponding events (if any) as generated by the artifact;
- sense(SID,Filter,Timeout):Perception - to actively perceive a possible observable event collected by the sensor SID, applying some kind of filter Filter, for a timeframe of maximum Timeout time units;
- focus(AID,SID) - to start observing persistently an artifact identified by AID, collecting all the possible observable events generated by the artifact on the sensor identified by SID;
- unfocus(AID,ID) - to stop observing an artifact, previously focussed by a focus action.

Artifact inspection

- getFD(AID):FD - to retrieve the function description representation of a specific artifact;
- getOI(AID):OI - to retrieve the operating instructions representation of a specific artifact;
- getObsState(AID):ObsState - to retrieve the observable state representation of a specific artifact.

Sensors management

- linkSensor(SensorType):SID - to link a new sensor of type SensorType to the agent body, getting an identifier of it;
- unlinkSensor(SID) - to unlink a previously linked sensor from the agent body.

Figure 3: Workspaces management API.

introduce a Role–Based Access Control model inspired by RBAC [18]) to regulate the access to artifacts and enclosed resources to agent bodies, and thus autonomous agents and human actors, situated in the workspace.

References

[1] S. Bandini, S. Manzoni, and G. Vizzari. A spatially dependant communication model for ubiquitous systems. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *The First International Workshop on Environments for Multiagent Systems, E4MAS 2004*, volume 3374 of *Lecture Notes in Computer Science*, pages 74–90. Springer-Verlag, 2005.

[2] F. Cabitza, M. P. Locatelli, M. Sarini, and C. Simone. CASMAS:sSupporting collaboration in pervasive environments. In *4th IEEE International Conference on Pervasive Computing and Communications (PerCom 2006)*, pages 286–295. IEEE Computer Society, 2006.

[3] P. Ciancarini. Coordination models and languages as software integrators. *ACM Computing Surveys*, 28(2):300–302, 1996.

[4] M. Cremonini, A. Omicini, and F. Zambonelli. Building mobile agent applications in HiMAT. In *IEEE 1999 International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE'99)*, pages 174–180. IEEE Computer Society, 1999.

[5] P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004.

[6] T. C. Lech and L. W. M. Wienhofen. AmbieAgents: a scalable infrastructure for mobile and context-aware information services. In *4rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), July 25-29, 2005, Utrecht, The Netherlands*, pages 625–631. ACM Press, 2005.

[7] M. P. Locatelli and G. Vizzari. Awareness in collaborative ubiquitous environments: the Multilayered Multi-Agent System approach. *ACM Transactions on Autonomous and Adaptive Systems*, 2(4):1–21, 2007.

[8] T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, 1994.

[9] M. Mamei and F. Zambonelli. Programming pervasive and mobile computing applications with the TOTA middleware. In *2nd IEEE International Conference on Pervasive Computing and Communication (Percom2004)*, pages 263–273. IEEE Computer Society, 2004.

[10] M. Mamei and F. Zambonelli. Augmenting the physical environment through embedded wireless

technologies. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *The Second International Workshop on Environments for Multiagent Systems, E4MAS 2005*, volume 3830 of *Lecture Notes in Computer Science*, pages 235–250. Springer–Verlag, 2006.

[11] M. Mamei, F. Zambonelli, and L. Leonardi. Cofields: a physically inspired approach to motion coordination. *Pervasive Computing*, 3(2):52–61, 2004.

[12] A. Omicini. Formal ReSpecT in the A&A perspective. *Electronic Notes in Theoretical Computer Sciences*, 175(2):97–117, June 2007. 5th International Workshop on Foundations of Coordination Languages and Software Architectures (FOCLASA'06), CONCUR'06, Bonn, Germany, 31 Aug. 2006. Post-proceedings.

[13] A. Omicini and F. Zambonelli. Coordination for Internet application development. *Autonomous Agents and Multi-Agent Systems*, 2(3):251–269, Sept. 1999. Special Issue: Coordination Mechanisms for Web Agents.

[14] A. Ricci, M. Viroli, and A. Omicini. Programming MAS with artifacts. In R. P. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors, *3rd International Workshop "Programming Multi-Agent Systems" (PROMAS 2005)*, pages 163–178, 2005.

[15] A. Ricci, M. Viroli, and A. Omicini. A general purpose programming model & technology for developing working environments in MAS. In M. Dastani, A. El Fallah Seghrouchni, A. Ricci, and M. Winikoff, editors, *5th International Workshop "Programming Multi-Agent Systems" (PROMAS 2007)*, pages 54–69, AAMAS 2007, Honolulu, Hawaii, USA, 15 May 2007.

[16] A. Ricci, M. Viroli, and A. Omicini. CArtAgO: A framework for prototyping artifact-based environments in MAS. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *Environments for MultiAgent Systems*, volume 4389 of *LNAI*, pages 67–86. Springer, Feb. 2007. 3rd International Workshop (E4MAS 2006), Hakodate, Japan, 8 May 2006. Selected Revised and Invited Papers.

[17] M. D. Rodríguez, J. Favela, E. A. Martínez, and M. A. Muñoz. Location-aware access to hospital information and services. *IEEE Transactions on Information Technology in Biomedicine*, 8(4):448–455, 2004.

[18] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 2002.

[19] V. Tabak, B. de Vries, J. Dijkstra, and J. Jessurun. *Progress in Design & Decision Support Systems in Architecture and Urban Planning*, chapter User Simulation Model: Overview & Validation. Capturing human behaviour in the built environment using RFID, pages 117–132. Eindhoven University of Technology, 2006.

[20] M. Viroli, T. Holvoet, A. Ricci, K. Schelfthout, and F. Zambonelli. Infrastructures for the environment of multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 14(1):49–60, 2007.

[21] M. Viroli, A. Omicini, and A. Ricci. Engineering MAS environment with artifacts. In D. Weyns, H. V. D. Parunak, and F. Michel, editors, *2nd International Workshop Environments for Multi-Agent Systems' (E4MAS 2005)*, pages 62–77, 2005.

[22] J. Weissman and Z.-L. Zhang. Smart environments: middleware building blocks for pervasive network computing (a position paper). In B. Knig-Ries, K. Makki, S. Makki, N. Pissinou, and P. Scheuermann, editors, *Developing an Infrastructure for Mobile and Wireless Systems*, volume 2538 of *Lecture Notes in Computer Science*, pages 90–105, 2002.

[23] F. Zambonelli and H. V. D. Parunak. Signs of a revolution in computer science and software engineering. In P. Petta, R. Tolksdorf, F. Zambonelli, editors *Proceedings of Engineering Societies in the Agents World III (ESAW2002)*, volume 2577 of *Lecture Notes in Computer Science*, pages 13–28. Springer–Verlag, 2002.