

Ubiquitous computing for mobile environments

Jose M. Molina, Juan M. Corchado and Javier Bajo

Abstract. The increasing role and importance of ubiquitous computing and mobile environments in our daily lives implies the need for new solutions. The characteristics of agents and multi-agent systems make them very appropriate for constructing ubiquitous and mobile systems. The aim of this chapter is to present some of the advances in practical and theoretical applications of multi-agent systems in the fields of ubiquitous computing and mobile environments carried out by several AgentCities.ES research groups.

1. Introduction

Intelligent environments, also known as Ambient Intelligence, have become increasingly important in recent years. These environments are characterized by certain capacities (all or some of them) such as ubiquity, transparency and intelligence. The multi-agent systems (MAS) have become increasingly relevant for developing distributed and dynamic intelligent environments. One of the advantages of the agents is their adaptability to work in mobile devices, so they support wireless communication (Wi-Fi, Bluetooth, WiMAX, UMTS, etc.) which facilitates their portability to a wide range of mobile devices. This advantage makes the agents and multi-agent systems very appropriate to be applied to the development of ubiquitous and mobile environments.

Agents can be characterized through their capacities in areas such as autonomy, reactivity, pro-activity, social abilities, reasoning, learning and mobility [78]. These capacities make the multi-agent systems very appropriate for constructing intelligent environments. An agent can act as an interface between the user and the rest of the elements in the intelligent environment. Furthermore, given the adaptability of agents to mobile devices (with low memory and processing resources), it is possible to provide an ubiquitous and transparent interaction, even personalizing

⁰Estefania Argente, Juan A. Botia, Sergio Ilarri, Vicente Botti, Emilio S. Corchado, Virginia Fuentes, Manuel Gonzalez, Arantza Illarramendi, Vicente Julian, Eduardo Mena and Nayat Sanchez

the user access. An intelligent agent can adapt itself to environmental changes or make predictions based on previous knowledge or experience. In this sense an agent is context-sensitive and can take decisions allowing it to automatically adapt itself to the changes in its surroundings. An agent usually integrates within a multi-agent system, or agent society, exchanging information and resolving problems in a distributed way. It requires an organization-oriented perspective to model this kind of problems, identifying the roles that every agent plays in the society or organization. These characteristics facilitate both ubiquitous communication and computation.

This chapter presents some advances in practical and theoretical applications of multi-agent systems in the fields of ubiquitous computing and mobile environments, and is structured as follows. In the next section a brief summary of the technologies used to construct intelligent environments is presented. Then, some real developments in ambient intelligence using agent technology are enumerated. In the third section the importance of the context definition for ubiquitous computing is emphasized. The main strategies for both representation and managing of context information are presented. Section four carries on with the importance of context information and focuses on ontology definition of context-based applications based on agents. The deployment of an ontology for environment definition is explained. Then ontology definition for agent communication and ontology definition for agent interaction are presented. The fifth section focuses on the need for social organization in ubiquitous systems. In this sense this section reviews the organizational model in agent societies and presents a new organization oriented multi-agent platform, which can be used in mobile devices. Finally, the last section of this chapter focuses on mobile agents. The advantages and benefits of mobile agents are discussed, the existing mobile agent platforms are studied and, as a conclusion, the possibilities for mobile agents in the future are shown.

2. Ambient Intelligence

2.1. Technology for Ambient Intelligence

Ambient Intelligence proposes a new way to interact between people and technology, where the technology is adapted to individuals and their context within which they live. This new way has the following goals:

- Promote a vision of people surrounded by intelligent interfaces that merge into daily life;
- Foster a computer-literate environment of intelligent data processing and communication by creating a simple, natural and effortless human-system interface;
- Develop an array of intelligent and intuitive systems and interfaces;
- Develop the capability to recognize and respond to individual user's requirements in a comprehensive manner;

- Create technologically complex environments in numerous contexts, such as medicine, academia, social structures, etc.

From this perspective, agents must be able to respond to events, take the initiative according to their goals, communicate with other agents, interact with users, represent and manage context information and make use of reasoning mechanisms to find the best solutions to achieve goals.

New approaches for Ambient Intelligence agent-based systems propose the use of context aware agents that handle a set of technologies and the incorporate mechanisms for representing and managing context information that provide the agents with the flexibility and adaptation to survive on dynamic environments and accomplish the Ambient Intelligence vision. The growing use of wireless devices (especially hand-held devices) in recent years has led to new requirements as well as to a great opportunity to extend traditional wired communication techniques. In this section, the main wireless technologies used to construct intelligent environments are presented. In Sections 3 and 4 the context aware middleware and ontologies are studied in detail.

The aim of ambient intelligence (AmI) is to construct intelligent environments that facilitate a ubiquitous access with independence of the physical location [19]. Wireless networks are location-independent (in the sense that wires are not needed) and provide a wide range of coverage. Protocols used to communicate in wireless technologies are mainly classified in the 802.1x.x protocol family for Bluetooth, infrared and Wi-Fi, and protocols used in mobile phones within the GPRS or UMTS technologies. Other wireless technologies that must be taken into account are GSM, GPS, RFID or ZigBee. Wireless LANs, also known as Wi-Fi (Wireless Fidelity) networks can be used as a replacement or as an extension of wired LANs [36]. They provide reduced infrastructure and low installation cost, and also give more mobility and flexibility by allowing workers to stay connected to the network as they roam among covered areas, increasing efficiency by allowing data to be entered and accessed on site [36]. Infrared connections require optic signals and the principal inconvenience is the need of direct-vision between devices. Bluetooth is a wireless technology that utilizes a short-range radio link and operates in the 2.4 - 2.48 GHz frequency band. Bluetooth is a technology that facilitates the interaction between near devices providing a high reliability and low energy consumption. GPRS uses a packet-switched system which provides data transfer services on mobile phone networks. UMTS is a universal mobile telecommunications system that operates in the 2 GHz frequency band and emphasizes the compatibility. This great amount of technologies, requires compatibility solutions. Some works, like the RASCAL system [41], allow mobile devices to autonomically self-manage connection endpoints and data transmission over available network technologies, providing a contingency manager system with autonomic capabilities which enables collaboration among a set of ubiquitous services deployed in the infrastructure and/or in the ad-hoc network. Other works, like ASAM [9], Adaptive Service Access Management, enables

effective delivery of next generation ubiquitous services by dynamically combining end user requirements and service provisioning policies with network-facing management and control functionality.

RFID technology is used to identify and receive information about humans, animals and objects on the move. An RFID system contains basically four components: tags, readers, antennas and middleware [71]. Tags with no power system (batteries) integrated are called passive tags or *transponders*. The reader is a device that interrogates or sends electromagnetic waves. RFID systems typically operate in three different frequency ranges: low frequency (30KHz - 500 KHz) and ultra high frequency (850 MHz - 950 MHz and 2.4 GHz - 2.5 GHz). Systems operating on low frequency are less costly, but have a shorter reading range. The middleware consists of processing software and hardware required to convert the tag signals into valid data [71]. The tag or transponder is placed on the object itself. As this object moves into the reader's capture area, the reader is activated and begins signaling via electromagnetic waves (radio frequency). The transponder subsequently transmits its unique ID information number to the reader, which transmits it to a device or a central computer where the information is processed and displayed. This information is not restricted to the location of the object, and can include specific detailed information.

One of the most important issues using wireless Technologies is to provide security guarantees when a communication between mobile devices is established. At the moment some services are provided: authentication services (WPA, WPA2, CCKM), data encoding services (RSN, TKIP, WEP, CKIP, CMIC), access control services (PEAP, smart card) or private virtual network services (IP-SEC or SSL) [45].

2.2. Agent Technology and Real Developments in Ambient Intelligence

In this section a brief summary of real developments in ambient intelligence using agent technology is presented. The incorporation of artificial intelligence techniques has led to further studies and to the modeling of shopping and leisure time in shopping malls in terms of multi-agent systems [4, 5, 15]. These authors focus on the shopping problem and on the recommendations that can be made to users. The growing use of hand-held devices in recent years has led to new requirements as well as to a great opportunity to extend traditional commerce techniques and apply new techniques. These new devices facilitate the use of new interaction techniques, for instance, some systems focus on facilitating users with guidance or location systems [15] by means of their wireless devices. The application of intelligent environments to health care and elderly care is one of the priorities in ambient intelligence. In this way different agent-based applications have been developed [16]. These applications make important contributions to traditional care techniques and improve the patient's quality of life. Another important field of application for ambient intelligence is housing. Nowadays it is usual to find services based on home automation in our homes. There are some agent-based home automation intelligent environments [60] which facilitate daily life at home. Ambient

intelligence has also been highly accepted in mobility and transportation problems. As a good example we can cite the navigation systems installed in our cars. Some examples of agents applied to the development of intelligent environments are navigation, delivery or route optimization [7, 11, 15, 80]. Finally, another important field where the application of ambient intelligence has been successful is education and learning [38], as well as culture, leisure and entertainment [4, 5, 12]. All these real developments have obtained promising results and demonstrate the importance of agents and multi-agent systems in the construction of intelligent environments. Agent technologies applied to ambient intelligence open a new research line which offers new interesting possibilities.

3. Context Definition

3.1. Context aware applications

A generic definition of context could be the following:

Context information is the set of useful data that, in a concrete instant of time, describe the elements which surround the user and some interesting aspects of the user itself.

Figure 1 represents all the typical elements which might be part of user context in a conventional ubiquitous computing system (which includes those based on agents software).

This information has static and dynamic components. Static information embraces all details related to the user that do not change through time, or at least do not frequently vary. User profile is an example of such information. In the profile, the birth date does not change but preferences in music, for example, may change through time. Dynamic information constantly changes and provides a description of an up-to-date snapshot of the user and his surroundings. For example, the location of the user in a building is a good example of such kind of information.

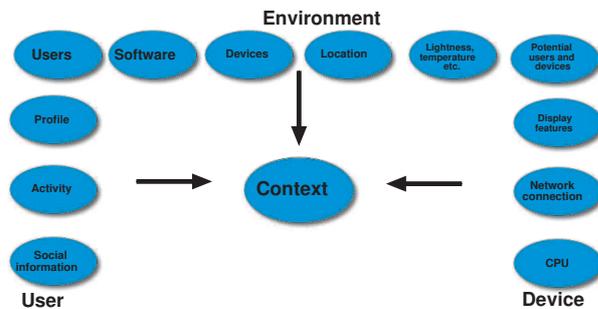


FIGURE 1. Information composition for the user context in a conventional ubiquitous computing system.

But, what is actually a context-aware service? We define a context-aware service as the kind of service which modifies its behavior according to information it has, related to user context. Hence, a music search service which implicitly takes into account user profile could be a context-aware service. Another example is a service which looks for restaurants for a user, taking into account his proximity to the possible options. An interesting concept related to this kind of service is context-aware service provisioning. It consists of the necessary mechanisms to dynamically provide users or software with services. The set of offered services will depend on time and context information for their selection. Think, for example, of an application in which, depending on the direction you take in a cross road of a highway, the software which interacts with the user through a hand held device offers a Theaters search service if going downtown or a sports facility booking menu if going to the suburbs.

Why should context be incorporated into ubiquitous computing information systems? Context information is crucial to guarantee that services can be personalized depending on user profile, user location, user state (i.e., working, not to be disturbed, out of office) and user's device. These elements compose a set of contextual information whose availability at any time and anywhere will allow the development of flexible services in such systems. Typical advantages obtained by incorporating context into a system are the following:

- increasing user satisfaction as services are more adapted to his preferences and profile;
- automating some functions: by means of behavior rules defined by the user, some activities can be automated;
- right information, in the right time and at the right place: a semantic model of context information makes it possible to filter incoming information, depending on the user situation;
- low obtrusive software: as it is capable of deciding when is more appropriate, and how, to interact with the user and
- increased personalization.

3.2. Strategies for representation of context information

In order to represent context information, we need to define a correct life cycle for it. Examples of this can be found in [76] and [57]. The success of an application relying on context information depends, in a high percentage, on what technology we use for its representation and management. Nowadays, most extended approaches to this problem are based on the use of ontologies.

By using an ontology, the existence is guaranteed of a common model for all the software entities of the ubiquitous system. Main advantages include a common model for information included in the context of the user and the possibility of reusing previously defined ontologies for these kinds of systems. Examples of systems which use ontologies to process delivered information in ubiquitous computing systems are [75, 33]. Its basic functioning is based on OWL [64] language and related technologies.

The first work which used OWL to describe information entities in a context-awareness based information system can be found in the CoBrA system [13]. Within this work, a first standard ontology to model ubiquitous computing information was proposed, Cobra-Ont. This ontology reused SOUPA¹ (Standard Ontology for Ubiquitous and Pervasive Applications) [14]. SOUPA is a shared ontology for ubiquitous computing applications. SOUPA delivers a common vocabulary for pervasive computing application developers. It combines a number of different basic vocabularies whose origins are found in commonly accepted ontologies. This ontology is divided into two different but related groups of vocabularies: the kernel or nucleus, the SOUPA core and extensions to the core. SOUPA reuses other more or less standard ontologies like, for example DAML-time and others. In the CoBrA system, as it could be expected, we already find an element which performs reasoning on contextual information using the advantages offered by ontologies. However, there are no explicit rules to identify interesting events like that cited above [75]. Another interesting detail about this architecture is that it is based on the agent metaphor [79], as it uses concepts like role, belief, desire and intention which belong to the BDI model of agency.

3.3. Managing context information

Managing context information includes operations for register, search, delete and update context information. These are basic operations, although other less direct operations include delivery, reasoning and aggregation. In order to have an up to date context information registry, a repository of context information is needed. And this repository should allow for registry, search, delete and update operations with pieces of context for a single user or a group of them.

Context delivery (i.e., the process of making context available and up to date for interested users, services and/or applications) might be a delicate task, depending on the kind of system architecture we have (i.e., if it is distributed, with a centralized directory or with no central directory at all). Moreover, the coordination model for delivery must be taken into account (i.e., if we use a blackboard model or a publish/subscribe notification strategy and so on.). A review is out of the scope of this chapter but more details on this issue might be found in [54].

Context reasoning and aggregation are two tasks which are strongly inter-related. Reasoning on context is the process of using a deductive process to infer new interesting situations from a basic context definition. For this, a logic theory with foundational facts and axioms is needed. In most of the cases, descriptive logic and user defined **if-then** rules are used for concrete implementations. Context aggregation is the process of defining mechanisms for dealing with the same representation of context but seen at different levels of abstraction. Aggregation is used to get more convenient representations depending on the application.

¹<http://ebiquity.umbc.edu/paper/html/id/165/The-SOUPA-Ontology-for-Pervasive-Computing>

4. Ontology definition of context-based applications based on agents

Nowadays, communication between software systems, organizations, and persons causes difficulties of interoperability, re-use and communication, due to the existing differences of each one, with respect to concepts, models and structures. Defining ontologies help to solve these problems, since the main motivation relies on integrating different domains into a coherent framework, providing a common vocabulary definition and interoperability between heterogeneous systems [70].

Originally, the term *ontology* comes from the existence concept defined by philosophers, but it has been adopted by Artificial Intelligence with the idea of representing the real world, viewed as a set of concepts (entities, attributes, and processes), their definitions and relationships between them. This representation is achieved by the ontology conceptualization mechanism [70].

Applying the ontology definition to the context of a multi-agent system, ontologies are defined as a common vocabulary to share information in the exchanged queries and assertions messages between participating agents [34]. The ontology role in a communication process is to avoid ambiguous definitions of terms for facilitating agents to share knowledge between them inside a domain.

The main reasons for using an ontology in a context-aware multi-agent system are the following [59]: the ontology development allows sharing knowledge, ontology universe allows context reasoning, for composing complex contextual information and reasoning about it, and finally, ontologies detect inconsistencies in contextual information, since it can be highly imperfect.

Normally, ontology represents a conceptualization of particular domains. However in case of context-aware applications, the context is not limited to a specific environment, since it can be whatever domain (airport, fairground, university, shopping center etc.) [26]

Following the categorization defined by Schilit [62] that divided contextual information in: computing context (network, devices, etc.), user context (preferences, location, and social situation) and physical context (temperature, traffic, etc.), a contextual information of a context-aware system for dynamical environments can be defined by ontologies. The ontology definition must gather all concepts and their properties and relationships for accomplishing this contextual definition.

For building ontologies, Noy and McGuinness propose an iterative process based on the methodology proposed by Gruninger and Fox [35] who defined the competency questions used in the scope and goal step, and the development of the classes hierarchy based on Top-Down and Bottom-Up strategies. The steps for developing an ontology are described as follows [55]:

1. Determine ontology goal: it is important to have clear requirements and the intention of the ontology use. The scope of the ontology can be limited by a question-answer iterative process, making several questions about the domain that ontology may cover, what is the use of ontology, etc.

2. Consider the integration of existing ontologies: reusing ontologies is a requirement in order to interact with other applications that base on particular ontologies or controlled vocabularies. In the case that no relevant ontology can be reused, the better option is to develop a new ontology from scratch.
3. Ontological acquisition: defining the ontology implies a process of ontological acquisition, which consists of the identification of the key concepts and relationships of interest domain. A Top-Down strategy or a Bottom-Up strategy or a combination of both of them can be used for the ontological acquisition step.
4. Codification process: it consists of specific and formal representation of the conceptualization gathered in the capture phase and it allows selecting representation language. In this step the ontology could be created using the Protege tool [58]

The proposed steps set up the basis of the ontology development. However there are other alternatives of methodologies for ontology development as Gomez-Perez [31] and Uschold [70] present alternative ontology-development methodologies. The Ontolingua tutorial [23] discusses some formal aspects of knowledge modeling and Ontolingua for portable ontologies defined by Gruber presents a system for describing ontologies using multiple representation languages.[34]

4.1. Developing an ontology for environment definition

Following defined steps, there is a proposal of a meta-ontology [25] that focuses mainly on the definition of all the concepts in order to be valid for any environment or domain. These ontological high level concepts (Figure 2) are considered to be meta-concept or meta-object for composing the environmental model in context-aware systems, and can be described as follows:

- Framework is the general application concept which includes high level system concepts and defines what is the current environment or domain of the system. It has two slots: Sector and Event, that represents, system sector (technology, entertainment, market etc.) and the current event (fairground, conference, congress, exhibition etc.), respectively. These slots are properties of whole subclasses of Framework. Sector can take *mobile* value and Event can take *fairground* value in mobile fairground domain, for instance.
- Location represents the (x, y) coordinates of any place, participant or object.
- Spatial region and temporal region concepts define the environment area and temporal system information about users in any location in spatial region, respectively. Spatial region represents the map or NxM area, and it is composed of segments with a range of positions for each one. For example, segment1 is a segment with the range of positions: (3, 5) (3, 6). Temporal region represents user date (dd/mm/yyyy) and hour (hh:mm) when he is in a specific position inside the map. This spatial and temporal representation is shared for all system domains.

- Place concept represents interest points in the environment. Places can be participant company's expositors like Nokia, Siemens etc. in a fairground domain, for instance.
- Participant concept refers to people or companies that play a role in the system. In mobile fairground domain, participants can be visitors and companies (Nokia, Siemens, Motorola etc). A preference is a Participant's subclass, and it gathers preferential product, firm, price, model etc.
- Service concept can be any kind of system provision offered to users referred to contextual information. A service could be a notification in a user device about preferential user product.
- Product represents any kinds of information or object that users require to be informed. A product in the fairground domain is a mobile, for instance.
- Device concept gathers information about different user's devices in which the system works. An example of a user device is a PDA or a smart phone.

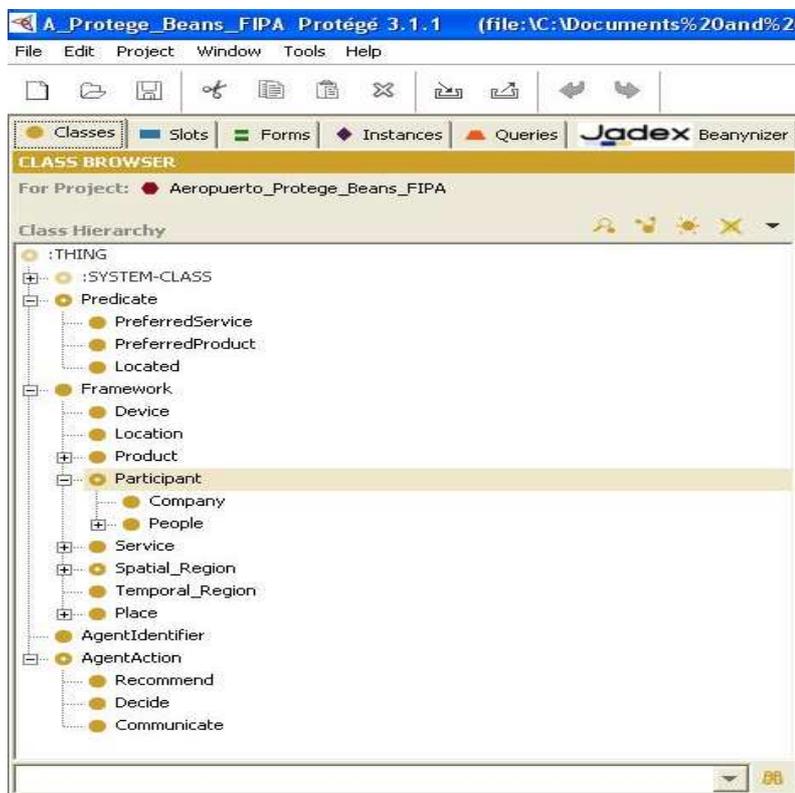


FIGURE 2. Ontology High Level Conceptualization.

4.2. Ontology definition for agent communication

The use of domain ontology [34] is one of the most promising approaches to model the distributed agents' knowledge, constituting the common ground of an entire multi-agent system. This ontology describes, in a natural way, ontological commitments for a set of agents so that they might be able to communicate about a domain of discourse without a necessary operation of a globally shared theory.

When an agent A communicates with another agent B, a certain amount of information I is transferred from A to B by means of an ACL (Agents Communication Languages) Message. Inside the ACL Message, I is represented as a content expression consistent with a proper content language and encoded in a proper format [8].

Ontology application to a multi-agent system describes agents knowledge in the communication process and this communication is achieved by FIPA-ACL using ontological concepts for messages [8]. The model of communication FIPA [69] is based on the assumption that two agents share a common ontology for the domain of discourse. It ensures an agents mutual understanding because they describe the same meaning for the symbols used in the messages. In order to perform the proper semantic checks on a given content expression it is necessary to classify all possible elements in the domain of discourse according to their generic semantic characteristics. This classification is derived from the ACL language defined in FIPA that requires the content of each ACL Message to have a proper semantics according to the performative of the ACL Message. To satisfy the content of FIPA-ACL messages, ontology should define a set of different type of schemes: predicates, concepts and actions. Predicates are expressions that say something about the status of the world and can be true or false, e.g.,

(Belongs-to (Mobile: model NOKIA 6230)(Company: NOKIA))

stating that *the Mobile NOKIA 6230 belongs to the company NOKIA*. Agent actions, i.e., special concepts that indicate actions that can be performed by some agents, e.g.,

(Sell (Mobile: model NOKIA 6230)(Person: name Anne))

Concepts are expressions that indicate entities with a complex structure that can be defined in terms of slots, e.g.,

(Person: name Anne: age: 30)

Concepts typically make no sense if used directly as the content of an ACL message. They are generally referenced inside predicates and other concepts such as in

(Mobile: model NOKIA 6230: Belongs-to(Person: name Anne))

A fully expressive content language should be able to represent and distinguish between all the above types of elements. An ontology for a given domain is a set of schemes defining the structure of the predicates, agent actions and concepts that are pertinent to that domain.

4.3. Ontology definition for agent interaction

Ontology must define predicates, agent actions etc., for gathering the appropriated semantic according to ACL messages, so these ontology concepts are related with

the different kinds of message in FIPA. Predicates can be used as the content of an INFORM or QUERY-IF message and Agent Actions can be used in REQUEST ACL message [8].

The interaction model is used to represent the dependencies and relationships between agent roles in the multi-agent system, according to the protocol definitions, that are actions that involve interaction between two roles played by agents in a multi-agent system. In [27] an interaction model is proposed as a phase of the analysis and design process, according to Gaia methodology. Some protocols between agents are defined for the goals of providing context-based services in any environment, for instance:

- Receive-Registry-Profile: an agent receives a REQUEST message of registry from another agent, and its profile.
- Agree-Registry: Send an AGREE message to confirm the agent registry.
- Warn-provider: Send an INFORM message to the closer provider role for alerting the presence of other agent.
- Offer-Service: Send a PROPOSE message for offering contextual-information services to other agents.

5. Social Networks

5.1. Agent Societies

Ubiquitous systems need methodologies, frameworks and software that take care of situatedness, openness, locality in control and locality in interactions [81]. Situatedness implies that software components execute in the context of an environment and can influence it or be influenced by it. Moreover, systems can dynamically change their structure, so its elements can enter and leave the system through time. Furthermore, there is locality both in control (with autonomous and proactive control) and interaction (components interact with each other according to local, geographical or logical, patterns). Finally, emerging societies can be formed in which teams, coalitions or other organizational structures are needed. For example, in mobile-commerce settings, personalized information agents, each representing a potential business partner, might dynamically form temporary profit-oriented coalitions to enhance a customer's purchasing and negotiating strategies in multiple electronic marketplaces [46]. Thus, ubiquitous systems will need to be modeled and designed in terms of social systems, following an organizational point of view.

An organization provides a framework for activity and interaction through the definition of roles, behavioral expectations and authority relationships [29]. Thus, an *agent organization* is a social entity composed of a specific number of members that accomplish several distinct tasks or functions and that are structured following some specific topology and communication interrelationship in order to achieve the main goal of the organization.

Dynamic agent organizations that self-adjust for making the most of their current environment are more and more important. These organizations could

appear in dynamic or emerging societies of agents such as Grid domains, peer-to-peer networks, or other environments in which the agents coordinate in a dynamic way for offering combined services. So, it can be an appropriate approach to solve problems related with mobile ad-hoc networks. For example, agent-based virtual organisations for the Grid have been employed in the CONOISE-G project [65], in which an infrastructure to support robust and resilient virtual organisation formation and operation is developed. The social factors in the organization of multi-agent systems are also more and more important for structuring interactions in dynamic open environments.

5.2. Towards an organization oriented open MAS

Organizational models have been recently used in agent theory for modeling coordination in open systems and to ensure social order in MAS applications [17, 63].

Three dimensions can be used to describe the MAS from an organizational point of view [3]: its structure; its functionality; and its norms. In [2], a survey of MAS organization-oriented methodologies is detailed. Many MAS methodologies and frameworks, such as Agent-Group-Role [24] or INGENIAS [32], only take into account the structure and functionality view, specially detailing the organization roles, groups and role relationships.

Other methods, such as Tropos [30], go further on detailing more complex and elaborated organizational structures, such as hierarchies, matrices, congregations, federations and so on. They also propose using those organizational structures in the analysis and design phases. A deeper explanation of all those complex structures can be found in [37] [1].

Finally, other approaches are focused on the social norms (SODA [56], Electronic Institutions [22] or OMNI [73], for example). They explicitly define control policies to establish and reinforce them, taking into account the organizational dynamics of the system, but they hardly take advantage of the topological structure of the system and intrinsic relationships of its members.

Regarding agent platforms, the most well-known agent platforms [2] offer generic agents with basic functionalities, which users should extend; and an execution environment that facilitates agent communication at execution time. However, very few agent platforms support agent organizational features, such as AMELI [21], JACK Teams [39] and MOISE+ [40], which do take into account some of the concepts proposed in the organizational approach. More specifically, AMELI and MOISE help designers to control obligations and norms of agents; whereas JACK Teams provide team behaviors.

5.2.1. Organizations in Ubiquitous systems. Over the last few years, only a few testbeds and real applications have been developed and reported in this area. The Universidad Politecnica de Valencia has developed an organization oriented multi-agent platform, called SPADE, which can be employed too in mobile devices.

SPADE (a Smart Python multi-agent Development Environment) [20] is a new agent platform, fully FIPA compliant, that supports a new communication

protocol between agents, based on Instant Messaging systems, that uses a distributed network to route messages from one agent to another. It also supports agent mobility, presence notification between components (this allows the system to determine the current state of the components that are connected to the platform in real-time), multi-user conference (message sharing between a group of agents). These capacities provide new communication capabilities between agents, which make agents more versatile. As commented before, the SPADE platform can be applied in mobile devices. More specifically, SPADE was developed in Python. This allows the execution of the platform in several architectures and operating systems such as Windows, Linux, MacOS, Windows Mobile, PalmOS, SymbianOS for mobile phones, etc.

SPADE also enables organization design, based on the concept of Organizational Unit, which represents the minimum set of members' relationship inside an organization. There are three types of organizational units: simple hierarchy (formed by a supervisor who has control over all other members; and several subordinates that carry out the basic tasks and communicate with each other through the supervisor); a team (in which all members collaborate between them to reach a global and common goal, sharing all their information, and coordination is obtained using mutually accepted decisions and plans) and a flat structure (which represents an anarchy in which there is no fixed structure nor control of one member over another). More complex and elaborated organizational structures can be built in SPADE using those organizational units, such as bureaucracy, matrix, federations, congregations and so on. Moreover, SPADE offers a series of services related to an organizational unit for controlling agent interactions (allowing one to enable/disable communications, bilateral and multiple interactions), unit members (controlling agent identity, quantity of members, admission, expulsion and registration procedures), and organizational units (allowing to create, delete, configure or join an organization).

A real application of a multi-agent system architecture to offer services in the tourism industry has also been developed [48], following both organizational and ubiquitous concepts. Users can access the system using a Java-enabled mobile phone or PDA anytime, so they can obtain up-to-date information about the places they will visit and to plan a specific day. Currently, a hierarchical approach has been implemented, in which a broker agent is in charge of establishing and controlling communication between user and sight agents. But other more complex structures are faced, such as sight coalitions, in which places with similar activities offer services in common.

6. Mobility

In a wireless environment, most of the assumptions that guide the definition of the traditional client/server architecture are not valid: 1) fast, reliable and cheap communications; 2) robust and powerful devices; and 3) fixed locations of the

participating devices. Thus, the client/server architecture is not adequate anymore for wireless environments, and several other agent-based software models have been proposed [67]:

- *Client/agent/server*. It is a three-tier architecture that introduces an agent on the server side (i.e., in the wired network). The agent becomes an intermediate for the interactions between the client (the mobile device) and the server. In this way, the server can communicate with the server agent even if the mobile device is unreachable at that moment, and the server agent will communicate appropriately with the client when the wireless connectivity is recovered.
- *Client/agent/agent/server* (also called *client/intercept/server*). Regarding the previous model, this one proposes the addition of a client-side agent. The purpose is to abstract the client from the intricacies of the wireless communications, as a server agent does for its server. The client and server agents interact to reduce the wireless communications and can divide the tasks among themselves according to the existing conditions. The client-side agent can also include optimizations such as *view materialization* [77] and an *asynchronous-disconnected mode* (requests that cannot be satisfied by the view are queued when connectivity is lost and resumed later when it is available again).
- *Mobile agents*. They are agents that have the ability to move autonomously from computer to computer to perform their tasks [51, 52, 47].

In the rest of this section we focus on the last model, as it is a general and flexible model which presents very interesting features for mobile environments.

6.1. Client/Server vs. Mobile Agents

In the traditional *client/server* architecture, a server at a certain computer offers a set of services to interested parties. Then, three steps take place: 1) a client located at another computer requests the execution of a service by interacting with the server, 2) the server performs the requested service, and 3) the server returns the result to the client.

Mobile agents arise as a promising alternative (and also as a complementary approach) to client/server for mobile environments. A mobile agent is a program that has the capability to move to other execution environments: it can decide itself when and where to move to perform its tasks. Mobile agents execute on context denominated *places* and can autonomously travel from *place* to *place* (usually, on different computers) resuming their execution there. Mobile agents are not bound to the computer where they are created; instead, they can move freely between *places* on different computers. Two types of mobility can be considered for mobile agents. If the whole agent's execution state is saved before a trip and restored at the target computer, we have *strong mobility*: on arrival at a new computer, the agent would resume by executing the statement that follows the movement statement. If the execution state is not saved, we have *weak mobility* [28]: on arrival at a new computer, the agent executes a certain *callback method* (predefined or specified by the programmer). While strong mobility is difficult to support, weak mobility

has proven to be sufficient in most scenarios [6]. In the following, we analyze why mobile agents are beneficial to mobile environments.

6.2. Advantages of Mobile Agents

Due to their mobile nature, mobile agents offer many interesting benefits [47]. From the point of view of wireless networks, we would like to highlight the following:

1. *They simplify the maintenance of servers/devices.* As they can move to remote computers to achieve their goals, they avoid the need for installing specialized server processes on every machine to fulfill the requirements of all types of mobile devices and wireless applications. Instead, *only one server process* (the *mobile agent platform*, as explained in Section 6.3) *needs to be running* on a computer, and *many different agents* can travel from the mobile devices to that computer at any time carrying the required functionalities. Similarly, a mobile agent can travel with a the mobile device to provide a required service.
2. *They reduce the network load and latency.* A mobile agent can travel to the computer or mobile device that holds the necessary data, access them locally, and filter out the data that do not need to be sent over the wireless network. Moving the computation to the data, instead of the other way around, can save many wireless resources when large volumes of data must be analyzed. Besides, it improves the network latency, as the mobile agent obtains data quickly via local interactions.
3. *They are asynchronous and autonomous.* In traditional synchronous client/server architectures, the client must keep the connection active while its request is being processed by the server. If the connection fails (which may happen frequently with unreliable wireless connections), the client has to send the request to the server again, which will process it from the beginning. Alternatively, a mobile agent does not need to keep contact with its source computer while performing its tasks: a mobile device can send a mobile agent to a computer on the fixed network, and then go off-line or even be powered off. The agent becomes independent of its originating device, and thus it allows to dispatch a task into the network easily. When the device re-establishes the connection, it can collect the mobile agent and/or its results.

Furthermore, mobile agent technology also exhibits a *good performance* compared with the traditional client/server approach. For example, in [68] they evaluate the savings introduced by mobile agents when interacting with a remote database in a wireless environment, and in [50] they evaluate several strategies to download files from a wired network and show how mobile agents exhibit similar performance to client/server approaches. Due to all these benefits, mobile agents have been claimed to be very interesting for mobile and pervasive computing environments [68, 10, 72].

6.3. Mobile Agent Platforms

A mobile agent platform is an environment that allows agents to execute and provides them with different services, such as communication and mobility facilities. There are *many* available mobile agent platforms [49, 66, 18, 43], some developed by research groups and others by private companies. Aglets, Voyager, Grasshopper, and Tryllian are among the most popular alternatives. Most of the existing platforms have been developed in standard Java because it provides benefits such as platform independence (a key condition to allow agents to travel among heterogeneous mobile devices), secure execution, dynamic class loading, multithreading, and object serialization. Due to the impossibility of saving and restoring the execution stack in a standard Java system, they usually implement weak mobility.

The communication and mobility services that a platform must provide are interrelated. Particularly, mobile agents must be able to communicate among themselves, via remote method invocation or message passing, even if they move across computers. *Location transparency*, defined as the ability to communicate with mobile agents independently of their current locations, is a desirable feature. This is specially so in mobile environments (enough challenging on their own!), where placing this responsibility on the programmer should be avoided.

Some platforms are based on a remote invocation model for communications (e.g., Voyager, Grasshopper, and SPRINGS), through the idea of *proxy*, which is an abstraction used to communicate with an agent (similar to the idea of *stub* in *RMI*). On the contrary, others follow a message passing paradigm (e.g., Aglets, Tryllian and JADE²). Regarding message passing, there is an interest in the community to follow the FIPA standard (<http://www.fipa.org/>) to ensure agent interoperability. There is also a proposal specifically designed for mobile agents (*MASIF*, the *Mobile Agent System Interoperability Facility* [51]), adopted by OMG in 1998 to enable interoperability among different mobile agent platforms. However, the future of this specification is uncertain, since only a few platforms (Aglets and Grasshopper) implement it and it has not been modified since 2000.

6.4. The Future of Mobile Agents

Mobile agents have stirred up a lot of interest and research efforts during the last few years. However, despite their benefits, they have not been adopted outside the research area. In fact, the initial hype during the late nineties was followed by a more moderate period; while mobile agents continue to be an important focus of attention [82, 61], some doubts arise about their applicability and performance [74, 44, 42]. Thus, several issues have yet to be solved to increase the confidence of developers when pondering mobile agents as a practical approach to their problems.

A key problem is how to provide an efficient *location transparency* (i.e., supporting calls to target agents independently of their locations, as explained before)

²<http://jade.tilab.com/>

in environments with a high number of mobile agents. Location transparency requires a mechanism to keep track of the *places* where the agents are executing at every moment. This is important and challenging in distributed environments in general; even more in wireless environments, where other difficulties appear (e.g., loss of connectivity or unreliable/slow communications). Some platforms (such as Voyager, Grasshopper, and SPRINGS) provide location transparency through the use of *dynamic proxies* (proxies which continue being valid independently of agents' migrations). With dynamic proxies, building applications based on mobile agents is easier: the need of searching an agent every time a remote call is performed is avoided, as the same proxy is always used to route the communications as needed. Voyager implements dynamic proxies through *forwarding chains of proxies* (as agents move, they leave a trail of *footprints*). In Grasshopper, *region servers* are in charge of routing the calls on the proxies to their target agents. Other platforms (e.g., Tryllian and Aglets) do not offer location transparency. Some platforms, such as Tryllian or JADE, do not support proxies³. Not only keeping track of the current locations of the agents is challenging, but also how to ensure a reliable communication with agents that move very frequently [53] (especially in wireless environments, where connectivity can be lost at any time). This is also the concern of the platform *SPRINGS*⁴ [43]. Some experimental results show that this platform achieves a good scalability, which is key for wireless environments where there may be many mobile users; thus, in a wired network SPRINGS supports several thousands of agents continually moving and calling among themselves [43].

While some issues need to be resolved to enable a massive adoption of mobile agent technology, we believe that it is a very useful paradigm for building applications for mobile environments. Thus, as we have explained in this section, mobile agents present very interesting advantages over the traditional client/server approach in a wireless context. However, mobile agent platforms have been designed mainly with fixed distributed environments in mind, and there is not enough experience with the use of mobile agents in real wireless networks. For example, some mechanisms used to keep dynamic proxies up-to-date may need to be adapted to the peculiarities of wireless contexts. We expect that promising developments will occur in the future.

7. Conclusions

In this chapter we have shown the appropriateness of multi-agent systems to the development of Ambient Intelligence applications. The importance of Ambient Intelligence environments, characterized by capacities such as transparency, ubiquity and intelligence, has notably grown in recent years and has become deeply-rooted

³In JADE, a message is sent to a certain agent by specifying its identifier. In Tryllian, the target address is also needed.

⁴*Scalable Platform foR movING Software*, it allows software agents to *spring* among computers, see <http://sid.cps.unizar.es/SPRINGS>.

in the information society. New solutions are required and the multi-agent systems are an excellent alternative for developing Ambient Intelligence applications.

In this sense, the main characteristics of Ambient Intelligence have been studied, illustrating the operation of different technologies. Particularly, this chapter focuses on agent technology. In this way, a brief summary of real developments in ambient intelligence using agent technology has been presented. Moreover, the advances on practical and theoretical applications of multi-agent systems in the fields of ubiquitous computing and mobile environments are presented.

One of the most important tasks that has to be accomplished for an agent to live in an intelligent environment is obtaining a context definition. In this chapter the importance of obtaining a good context definition has been emphasized, and different context aware applications have been presented. Moreover, different strategies for representing context information have been studied, focusing on ontologies. The importance of managing context information has also been noted, and tasks such as context delivery and context reasoning and aggregation have been illustrated.

Given the importance of the communication in multi-agent systems in Ambient Intelligence applications, defining ontologies helps to solve problems of interoperability, re-use and communication and differences about concepts, models and structures. The main concepts of an ontology have been presented and a proposal of a meta-ontology that focuses mainly on the definition of all the concepts in order to be valid for any environment or domain has been detailed. Furthermore, the ontology definition for agent communication and agent interaction have been shown, illustrating both processes with examples.

Ambient Intelligence environments require ubiquitous communication and ubiquitous computation. That implies the need of methodologies, frameworks and software that take care of situatedness, openness, locality in control and locality in interactions. An agent organization is a social entity composed of a specific number of members that accomplish several distinct tasks or functions. Dynamic agent organizations that self-adjust for making the most of their current environment are more and more important. In this chapter, the organizational models that have been recently used in agent theory for modeling coordination in open systems are presented. Moreover, an organization oriented multi-agent platform called SPADE is presented. The SPADE platform can be applied in mobile devices.

Mobile devices supporting wireless communication have become increasingly relevant in the development of multi-agent systems. They provide a framework for obtaining ubiquitous communication and building applications for mobile environments. Mobile agents present very interesting advantages over the traditional client/server approach in a wireless context. In this chapter, the advantages of mobile agents have been presented and mobile agent platforms have been studied. Finally, a discussion about the future of mobile agents is established.

Some issues need to be resolved to enable adoption multi-agent systems to develop Ambient Intelligence applications. The main tendencies and current issues have been presented within this chapter, but there is still a lot of work to do.

References

- [1] E. Argente, V. Julian, and V. Botti. From Human to Agent Organizations. In *First International Workshop on Coordination and Organization (CoOrg'05)*, G. Boella and L. Van der Torre eds., pages 1–11, 2005.
- [2] E. Argente, A. Giret, S. Valero, V. Julian, and V. Botti. Survey of MAS methods and platforms focusing on organizational concepts. *Frontiers in Artificial Intelligence and Applications*, 113:309–316, 2004.
- [3] A. Baciú, and A. Nagy. Coordination and Reorganization in Multi-Agents Systems. In *Informatica*, vol. XLVIII, number 2, Studia Univ. Babeş-Bolyai, pages 53–60, 2003.
- [4] J. Bajo, Y. de Paz, J.F. de Paz, Q. Martín, and J.M. Corchado. SMas: A Shopping Mall Multiagent Systems. In *Proceedings of IDEAL'06*, LNAI, vol 4224 pp. 1166–1173, Springer Verlag, 2006.
- [5] J. Bajo, J.M. Corchado, and L.F. Castillo. Running Agents in Mobile Devices. In *Proceedings of IBERAMIA'06*, LNAI, vol 4140 pp. 58–67, Springer Verlag, 2006.
- [6] L. Bettini and R. De Nicola. Translating strong mobility into weak mobility. In *5th International Conference on Mobile Agents (MA'01), Atlanta, Georgia, USA*, pages 182–197. Springer, 2001.
- [7] Bohnenberger T., Jameson A., A. Krger, and A. Butz. Location-aware shopping assistance: Evaluation of a decision-theoretic approach. In *Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices*, Pisa, 155.169, 2002.
- [8] G. Caire. JADE Tutorial. *Application-Defined Content Languages and Ontologies*, 2002
- [9] M. Calisti, and D. Greenwood. Adaptive Service Access Management for Ubiquitous Connectivity In *4th International Workshop on Management of Ubiquitous Communication and Services, MUCS, May 21-25, 2007, Munich*
- [10] R.S. Cardoso. Mobile agents: A key for effective pervasive computing. In *Second Pervasive Computing Workshop of ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'02), Vancouver, British Columbia, Canada*, 2002.
- [11] C. Carrascosa, J. Bajo, V. Julin, J.M. Corchado, and V. Botti. Hybrid multi-agent architecture as a real-time problem-solving model. *Expert Systems with Applications*. Pergamon-Elsevier Science LTD. doi:10.1016/j.eswa.2006.08.031, 2006.
- [12] A. Chavez, D. Dreilinger, R. Guttman, and P. Maes. A Real-Life Experiment in Creating an Agent Marketplace. In *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, London, UK, 1997.
- [13] H. Chen, T. Finin, and A. Joshi. An ontology for contextaware pervasive computing environments. *Knowledge Engineering Review. Special Issue on Ontologies for Distributed Systems*, 2003.
- [14] H. Chen, F. Perich, T. Finin, and A. Joshi. Soupa: Standard ontology for ubiquitous and pervasive applications. In *International Conference on Mobile and Ubiquitous Systems: Networking and Services*, August 2004.

- [15] J.M. Corchado, J. Pavn, E.S. Corchado, and L.F. Castillo. Development of CBR-BDI Agents: A Tourist Guide Application. In *Proceedings of the European Conference on Case-based Reasoning 2004 (ECCBR'04)*. LNAI vol. 3155 pp. 547-559, Springer Verlag, 2005.
- [16] J.M. Corchado, J. Bajo, Y. de Paz, and D. I. Tapia. Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. *Decision Support Systems*. Elsevier Science. doi: 10.1016/j.dss.2007.04.008, 2007.
- [17] V. Dignum, J. Meyer, H. Wiegand, and F. Dignum. An organization-oriented model for agent societies. In *Proc. of International Workshop on Regulated Agent-Based Social Systems (RASTA'02)*, 2002.
- [18] M. Dikaiakos and G. Samaras. A performance analysis framework for mobile-agent systems. In *Revised Papers from the International Workshop on Infrastructure for Agents, Multi-Agent Systems, and Scalable Multi-Agent Systems*, volume 1887, pages 180–187. Springer, 2001.
- [19] Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.. That's what friends are for. Ambient Intelligence (AmI) and the IS in 2010. In *Innovations for an e-Society. Congress Pre-prints, Innovations for an e-Society. Challenges for Technology Assessment*. Berlin, Germany, 2001.
- [20] M. Escrivà, J. Palanca, G. Aranda, A. García-Fornes, V. Julian, and V. Botti. A Jabber-based Multi-Agent System Platform In *Proc. AAMAS06*, pages 1282–1284, 2006.
- [21] M. Esteva, B. Rosell, J. A. Rodriguez, and J. L. Arcos. AMELI: An agent-based middleware for electronic institutions. In *Proc. of AAMAS04*, pages 236–243, 2004.
- [22] M. Esteva, J.A. Rodriguez, C. Sierra, P. Garcia, and J.L. Arcos. On the formal specification of electronic institutions. *Agent Mediated Electronic Commerce*, 1991:126–147, 2001.
- [23] A. Farquhar. *Ontolingua tutorial*. <http://ksl-web.stanford.edu/people/axf/tutorial.pdf>, 1997.
- [24] J. Ferber, O. Gutknecht, and F. Michel. From agents to organizations: an organizational view of multi-agent systems. In *Proc. AAMAS03 - Workshop 4*, 2003.
- [25] V. Fuentes, J. Carbo, and J.M. Molina. Heterogeneous domain ontology for location based information system in a multiagent framework. In *7th International Conference on Intelligent Data Engineering and Automated Learning*, Burgos, Spain, 2006
- [26] V. Fuentes, N.Sanchez, J. Carbo, and J.M. Molina. Reputation in User Profiling for a Context-Aware Multi-Agent System. In *Fourth European Workshop on Multi-Agent Systems*, Lisbon, Portugal, 2006.
- [27] V. Fuentes, N.Sanchez, J. Carbo, and J.M. Molina. Generic Context-Aware BDI Multi-Agent Framework with GAIA methodology. In *International Workshop on Agent-Based Ubiquitous Computing ABUC2007 in AAMAS 2007 Conference*, Honolulu. Hawaii, 2007.
- [28] A. Fuggetta, G. P. Picco, and G. Vigna. Understanding code mobility. *IEEE Transactions and Software Engineering*, 24(5):342–361, 1998.
- [29] L. Gasser. An Overview of DAI In *Distributed Artificial Intelligence: Theory and Praxis*, L. Gasser and N.M. Avouris eds., Kluwer Academic Publishers, pages 9–30, 1992.

- [30] P. Giorgini, M. Kolp, and J. Mylopoulos. Multi-Agent Architectures as Organizational Structures. In *Journal of Autonomous Agents and Multi-Agent Systems*, Kluwer Academic Publishers, 2003.
- [31] A. Gomez-Perez. Knowledge sharing and reuse. *Handbook of Applied Expert Systems*. Liebowitz, editor, CRC Press, 1998.
- [32] J. J. Gomez Sanz. Modelado de Sistemas Multi-Agente. PhD dissertation. Universidad Complutense de Madrid, 2002.
- [33] Tao Gu, Hung Keng Pung, and Da Qing Zhang. Toward an osgi-based infrastructure for context-aware applications. *IEEE PERVASIVE Computing*, pages 66–74, October–December 2004.
- [34] T.R. Gruber. A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition* 5:88 (1993), 199-220.
- [35] M. Gruninger, and M.S. Fox. Methodology for the Design and Evaluation of Ontologies. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95*, Montreal, 1995.
- [36] Hewlett-Packard: Understanding Wi-Fi. <http://www.hp.com/rnd/library/pdf/>, 2002.
- [37] B. Horling, and V. Lesser. A Survey of multiagent Organizational Paradigms. In *The Knowledge Engineering Review*, Cambridge University Press, vol.19, pages 281–316, 2004.
- [38] M. Hospers, E. Kroezen, A. Nijholt, R. op den Akker, and D. Heylen. Developing a generic agent-based intelligent tutoring system and applying it to nurse education. In *Proceedings IEEE International Conference on Advanced Language Technologies (ICALT '03)*, Athens, Greece, 2003.
- [39] N. Howden, R. Ronnquist, A. Hodgson, and A. Lucas. JACK Intelligent Agents-Summary of an Agent Infrastructure. In *Proc. 5th International Conference on Autonomous Agents*, 2001.
- [40] J.F. Hubner, J. S. Sichman, and O. Boissier. S-Moise+: A Middleware for developing Organised Multi-Agent Systems. In *Proc. Int. Workshop on Organizations in Multi-agent Systems, from Organizations to Organization Oriented Programming in MAS*, pages 64–78, 2005.
- [41] R.Ghizzioli, and D.Greenwood. The RASCAL System for Managing Autonomic Communication in Disruptive Environments In *1st IEEE Workshop on Autonomic Communication and Network Management (ACNM)*, Munich, May 21-25, 2007.
- [42] R. S. Gray. Mobile agents: Overcoming early hype and a bad name. In *Fifth International Conference on Mobile Data Management (MDM'04)*, Berkeley, California, USA, pages 302–303. IEEE Computer Society, 2004.
- [43] S. Ilarri, R. Trillo, and E. Mena. SPRINGS: A scalable platform for highly mobile agents in distributed computing environments. In *4th International WoWMoM 2006 workshop on Mobile Distributed Computing (MDC'06)*, Buffalo, New York (USA), pages 633–637. IEEE Computer Society, ISBN 0-7695-2593-8, June 2006.
- [44] D. Johansen. Mobile agents: Right concept, wrong approach. In *Fifth International Conference on Mobile Data Management (MDM'04)*, Berkeley, California, USA, pages 300–301. IEEE Computer Society, 2004.

- [45] T. Karygiannis, and L. Owens. Wireless Network Security 802.11, Bluetooth and Handheld Devices. National Institute of Standards and Applications. U.S. Department of Commerce. Special Publication 800-48, 2002.
- [46] M. Klusch and A. Gerber Dynamic Coalition Formation among Rational Agents In *IEEE Intelligent Systems*, vol. 1094, pp. 42 - 47, 2002.
- [47] D. Lange and M. Oshima. Seven good reasons for mobile agents. *Communications of the ACM*, 42:88–89, 1999.
- [48] J.S. Lopez, F.A. Bustos, and V.J. Inglada Tourism Services Using Agent Technology: A MultiAgent Approach In *Proc. 1st Workshop on Industrial Applications of Distributed Intelligent Systems (INADIS)*, vol. CD-ROM, 2006.
- [49] K. Ludwig, A. Josef, W. E. Edgar, S. Wolfgang, and G. Franz. Using mobile agents in real world: A survey and evaluation of agent platforms. In *Second Workshop on Infrastructure for Agents, MAS, and Scalable MAS at Autonomous Agents, Montreal, Canada*. AAAI, 2001.
- [50] E. Mena, J.A. Royo, A. Illarramendi, and A. Goñi. Adaptable software retrieval service for wireless environments based on mobile agents. In *International Conference on Wireless Networks (ICWN'02), Las Vegas, Nevada, USA*, pages 116–124. CSREA Press, 2002.
- [51] D. Milojevic, M. Breugst, I. Busse, J. Campbell, S. Covaci, B. Friedman, K. Kosaka, D. Lange, K. Ono, M. Oshima, C. Tham, S. Virdhagriswaran, and J. White. MASIF, the OMG mobile agent system interoperability facility. In *Mobile Agents (MA'98), Stuttgart, Germany*, volume 1477. Springer, 1998.
- [52] D. Milojevic, F. Douglass, and R. Wheeler. *Mobility: processes, computers, and agents*. Addison-Wesley Professional, April 1999.
- [53] A. L. Murphy and G. P. Picco. Reliable communication for highly mobile agents. *Autonomous Agents and Multi-Agent Systems*, 5(1):81–100, 2002.
- [54] I. Nieto, J. Botia, and A. Gomez-Skarmeta. Information and hybrid architecture model of the ocp contextual information management system. *Journal of Universal Computer Science*, 12(3):357–366, 2006.
- [55] N.F. Noy and D.L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. em Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.
- [56] A. Omicini. SODA: Societies and Infrastructures in the Analysis and Design of Agent-based Systems. In *Agent-Oriented Software Engineering*, vol.1957, pages 185–193, 2001.
- [57] J. Pascoe N. S. Ryan and D. R. Morse. Human-computer-giraffe interaction - hci in the field. *Workshop on Human Computer Interaction with Mobile Devices*, 1998.
- [58] Protege. *The Protege Project*. <http://protege.stanford.edu> 2000.
- [59] D. Rios, P. Dockhorn Costa, G. Guizzardi, L. Ferreira Pires, J.G. Pereira Filho, and M. van Sinderen. Using Ontologies for Modelling Context-Aware Services Platforms. In *Workshop on Ontologies to Complement Software Architectures*, Anaheim, CA, USA, 2003.
- [60] U. Rutishauser, J. Joller, and R. Douglas. Control and learning of ambience by an intelligent building, *IEEE Transactions on Systems, Man and Cybernetics Part A*:

Systems and Humans, Special Issue on Ambient Intelligence, Vol. 35, No. 1, pp 121-132, 2005.

- [61] G. Samaras. Mobile agents: What about them? did they deliver what they promised? are they here to stay? In *Fifth International Conference on Mobile Data Management (MDM'04)*, Berkeley, California, USA, pages 294–295. IEEE Computer Society, 2004.
- [62] B.N. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*. IEEE, December 1994.
- [63] W. Scott. *Organizations: rational, natural, and open systems*. Prentice Hall, 2002.
- [64] M.K. Smith, C. Welty, and D.L. McGuinness. Owl web ontology language guide. w3c recommendation. Technical report, W3C, February 2004.
- [65] J. Shao, W. A. Gray, N.J. Fiddina, T.J. Norma, A. Preece, P. Gray, S. Chalmers, N. Oren, N. Jennings, M. Luck, et al. Supporting Formation and Operation of Virtual Organisations in a Grid Environment In *BT Technology Journal*, vol. 24 no 1, 2006.
- [66] A. R. Silva, A. Romão, D. Deugo, and M. M. Silva. Towards a reference model for surveying mobile agent systems. *Autonomous Agents and Multi-Agent Systems*, 4(3):187–231, September 2001.
- [67] C. Spyrou, G. Samaras, P. Evripidou, and E. Pitoura. Wireless computational models: Mobile agents to the rescue. In *Second International DEXA Workshop on Mobility in Databases and Distributed Systems (MDDS'99)*, Florence, Italy, pages 127–133. IEEE Computer Society, 1999.
- [68] C. Spyrou, G. Samaras, E. Pitoura, and P. Evripidou. Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies. *Mobile Networks and Applications*, 9(5):517–528, 2004.
- [69] The Foundation for Intelligent Physical Agents. Available : <http://www.fipa.org>, 2002.
- [70] M. Uschold, and M. Gruninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review* 11(2), 1996.
- [71] U.S. Department of Commerce. Radio Frequency Identification: Opportunities and Challenges in Implementation. Technical Report, 2005.
- [72] L. Vasiu and Q.H. Mahmoud. Mobile agents in wireless devices. *IEEE Computer*, 37(2):104–105, 2004.
- [73] J. Vazquez-Salceda, V. Dignum, and F.Dignum. Organizing Multiagent Systems. Institute of Information and Computing Sciences. Tech. Report. Utrecht University, 2004.
- [74] G. Vigna. Mobile agents: Ten reasons for failure. In *Fifth International Conference on Mobile Data Management (MDM'04)*, Berkeley, California, USA, pages 298–299. IEEE Computer Society, 2004.
- [75] X. Wang. Ontology-based context modeling and reasoning using owl. In *Context Modeling and Reasoning Workshop at PerCom 2004.*, 2004.
- [76] R. Want, B.N. Schilit, N. Adams, R. Gold, K. Petersen, D. Greenberg, J. Ellis, and M. Weiser. An overview of the parctab ubiquitous computing environment. *IEEE Personal Communications*, 2(6), pages 28–43, 1995.

- [77] O. Wolfson, P. Sistla, S. Dao, K. Narayanan, and R. Raj. View maintenance in mobile computing. *SIGMOD Record*, 24(4):22–27, 1995.
- [78] M. Wooldridge, and N.R. Jennings. Agent Theories, Architectures, and Languages: a Survey. In *Wooldridge and Jennings, editors, Intelligent Agents, Springer-Verlag*, pp. 1-22, 1995.
- [79] M. Wooldridge. *An Introduction to MultiAgent Systems*. Wiley & Sons, 2001.
- [80] T. Yamashita, K. Izumi , and K. Kurumatani. Car Navigation with Route Information Sharing for Improvement of Traffic Efficiency. In *Proceedings of 7th Annual IEEE Conference on Intelligent Transportation Systems*, pp. 465-470, Washington, D.C., U.S.A, 2004.
- [81] F. Zambonelli, and H.V.D. Parunak. From Design to Intention: Signs of a Revolution. In *Proc. AAMAS02*, pages 455–456, 2002.
- [82] A. B. Zaslavsky. Mobile agents: Can they assist with context awareness? In *Fifth International Conference on Mobile Data Management (MDM'04), Berkeley, California, USA*, pages 304-. IEEE Computer Society, 2004.

Jose M. Molina
University Carlos III of Madrid,
Avenida de la Universidad Carlos III 22,
28270 Colmenarejo (Madrid),
Spain
e-mail: molina@ia.uc3m.es

Juan M. Corchado
University of Salamanca,
Plaza de la Merced s/n,
37008 Salamanca,
Spain
e-mail: corchado@usal.es

Javier Bajo
Pontifical University of Salamanca,
Calle Compania 5,
37002 Salamanca,
Spain
e-mail: jbajope@upsa.es