Multiagent and Grid Systems – An International Journal 5 (2009) 1–18
DOI 10.3233/MGS-2009-0116
IOS Press

A context-sensitive infrastructure for coordinating agents in ubiquitous environments

Manfred Bortenschlager*, Gabriella Castelli, Alberto Rosi and Franco Zambonelli

Abstract. The combination of contextual information about the real world (e.g., collected by sensors) with information coming from the virtual world (e.g., the Web 2.0), may represent an enormous enrichment particularly for services organized and provided by agents in ubiquitous environments. To address the challenging need for coordination in such environments, and to provide the user a high-level of service quality, an engineered approach to exploit such information is required. Such an approach should generate added-value by offering means for combining diverse data sources, should allow delivering context-sensitive information and, hence, should promote context-dependent coordination of entities in ubiquitous environments. In this paper, we report the scenario-based analysis of key requirements for ubiquitous environments that we have used as the basis for the design of our proposal. Following, our proposal for the Ubiquitous Coordination Model (UbiCoMo) and its associated infrastructure is detailed. The UbiCoMo model covers an expressive data model based on four-field tuples to represent contextual information, data distribution and management concepts based on tuple spaces, and the integration of coordination patterns to resolve reoccurring coordination problems in ubiquitous scenarios. The UbiCoMo infrastructure integrates the fundamental mechanisms for agent-based coordination in ubiquitous environments, and is well-suited to provide the specific means required to offer high-level services and context-sensitive functionalities. A concrete ubiquitous application, the *living diary*, is assumed as a case study both to illustrate the requirements analysis and to exemplify the usage and the suitability of UbiCoMo.

Keywords: Ubiquitous environments, coordination media, coordination theory, context-sensitive infrastructure, tuple spaces

1. Introducing the problem scope

The work we present in this paper is based on an newly arising but very natural paradigm which promises to be useful for a number of ubiquitous services. In the coming years, we will not only browse the Web, but we will be able to "browse the world" around us. Users and software agents will be able to seamlessly request contextual information dependent on our current or historical physical position (e.g., by pervasive sensors or actuators) which can be enriched by correlation with additional information coming from the virtual world (i.e., the Web). An infrastructure that integrates this comprehensive information about the world can serve as an environment for context-dependent mediated interactions between users and agents, i.e., as a shared context-sensitive coordination medium.

To foster the coordinative activities of agents, the environment in which they are embedded needs to be engineered [30] by appropriate concepts and models. Coordination theory [20], per se, is independent from any concrete discipline and, hence, can be exploited to deduce valuable mechanisms also for coordination in ubiquitous environments. This theory defines coordination as the "act of managing

*Corresponding author. E-mail: mborten@salzburgresearch.at.

ISSN 1574-1702/09/$17.00

interdependencies between activities". Many efforts have to be invested in defining mechanisms to resolve the interdependencies, which is the key to improve coordination [20]. In order to accomplish this, usually coordination laws are established and applied. In our infrastructure we use coordination theory as an input and build a coordination model (and the associated infrastructure) for agent and service coordination in ubiquitous environments, which finally shall improve the service quality and performance of such ubiquitous information systems.

The proposed Ubiquitous Coordination Model (UbiCoMo) provides comprehensive concepts for context-sensitive coordination of agents in ubiquitous environments. The UbiCoMo infrastructure is implemented as a layered architecture that shall serve as an enabler for the vision we call *Ubiquitously Browsing of the World* (introduced in more details in [6]), by providing the appropriate and necessary coordination media.

As the basis for the design of UbiCoMo we conducted a scenario-based analysis to capture the general requirements of a coordination model and infrastructure for ubiquitous scenarios. As a consequence, we derived a minimum set of specifications for a ubiquitous coordination model, which – as realized in our UbiCoMo approach – represents our specific contribution to MAS-based environments engineering. In particular, a ubiquitous coordination model has to:

– provide an expressive data model that makes the integration and correlation of real world as well as virtual world data possible, and that allows for context-sensitive queries. In UbiCoMo, we propose to represent the facts and related contextual information according to four dimensions (the *W4 Concept*), which enable dealing with heterogeneous data sources in a context-dependent way.
– offer accordant information management and distribution means to address the specific characteristics and dynamics of ubiquitous environments. In order to provide a great degree of flexibility, we incorporate a tuple space approach which inherently de-couples information exchange and coordination activities, required to tackle dynamics, and also covers several tuple replication strategies to tackle efficiency.
– integrate the insights of coordination theory and particularly of coordination patterns. In UbiCoMo, we provide the proposal for several coordination patters that resolve frequently reoccurring coordination problems.

The remainder of this paper is organized as follows: Section 2 analyzes the requirements for coordination in ubiquitous environments. The *living diary* application example is assumed as a case study to ground the discussion. Based on the above analysis, Section 3 introduces the UbiCoMo coordination model and details the layered architecture of the implemented UbiCoMo infrastructure. Section 3 gets back to the living diary example and uses it to exemplify the usage of UbiCoMo and its suitability in ubiquitous environment. Finally, in Section 5 we place our proposal in the context of related work, and summarize and discuss further work issues in Section 6.

2. Requirements analysis

In order to design an architecture to address the broad spectrum of future information systems for ubiquitous environments, we found it fundamental to examine several well-known examples of ubiquitous scenarios. From the analysis, we derived relevant commonalities which were eminent in each of these scenarios, and took these as the basis for the derivation of requirements. These requirements express characteristics that must be addressed by any proposal aimed at effectively and successfully providing coordination services in ubiquitous environments.

As sources for our examinations we took the Ambient Intelligence (AmI) scenarios of the IST Advisory Group (ISTAG)[1] and the EU projects Amigo,[2] DAIDALOS,[3] WORKPAD,[4] and CASCADAS.[5] The detailed description of these scenarios is available on the Web. In this paper, to better ground the discussion and for the sake of synthesis, we only focus on a single representative scenario: the living diary from the CASCADAS project.

A living diary is an individual human-centric application that realizes a sort of a digital auto-composing diary for the activities and experiences of a user. The living diary considers the automatic collection of facts describing the situation in which a user is in (real-time data), as they can be inferred by sensors and RFID tags in the surroundings, location information generated by GPS or WiFi, access to Web data, etc. Then, it considers the storing of such data and the possibility for a user (or for a user agent) to access it for browsing his own history, browsing the world around him, as well as for coordinating with other users on a context-sensitive way (e.g., based on the users' relative positions and common interests). For instance, a living diary may be very relevant in the context of tourism, where tourists can use this diary to document their activities, share experiences, and where tourists in a group can coordinate their movements and future activities based on shared access to diary data.

To discuss the requirements derived from the analysis of the investigated scenarios, for each of the requirements we summarize the requirement in general terms (as derived from the analysis of different scenarios), and put it in the specific context of the living diary example. Each requirement is related to an identifier *Rx*, where *x* is a consecutive number. Table 1) shortly summarizes the above identified requirements.

R1: Enrichment of the physical world with (virtual) data sources

A user while being on the move might be interested in a certain object of the physical world and wishes to get further information delivered. From that we derive: *Ubiquitous environment representations must be able to enrich information coming from the physical world by merging it with information from other (virtual) data sources.* (*R1*) In terms of the living diary, this means that, for instance, by applying reverse geo-coding (i.e., correlating geo-coordinates to an actual position), a user (e.g., tourist) can get an additional information to his/her current location.

R2: Context-sensitive queries for integrated data sources

Various occurring data sources may be represented by different kinds of physical or virtual sensors, exhibiting different formats, quality, or quantity. Only a generalizing intermediary data model can help to harmonize and provide an interface for context-sensitive queries. We derive that: *Ubiquitous environment representations must provide an expressive data model that allows to appropriately integrate various data sources and context-sensitive queries.* (*R2*) In the living diary, a tourist walking in a park may read an RFID tag mounted on monuments or buildings. ID information coming from environmental RFID tags are integrated in a common data model with different web accessible information repositories which deliver the queried information based on the tourist's current context.

R3: Context-dependent data refinement

Ubiquitous services require for personalized information. In order to provide this information relevant to specific situations of a user, an appropriate model allowing context-sensitivity is necessary which at

[1]http://cordis.europa.eu/ist/istag.htm, Document: ftp://ftp.cordis.lu/pub/ist/docs/istagscenarios2010.pdf.
[2]http://www.hitech-projects.com/euprojects/amigo, Document: Deliverable D1.2.
[3]http://www.ist-daidalos.org, Document: Deliverable D111.
[4]http://www.workpad-project.eu, Document: Deliverable D1.3.
[5]http://www.cascadas-project.org, Document: Deliverable D5.1.

the same time has to be simple to be able to contain data from most diverse sources. This means that: *Ubiquitous environment representations must allow for a refinement of data according to the specific context of the user's current situation*. (*R3*) For this, consider a tourist in a museum. The living diary guides the tourist to the rooms hosting his/her preferred objects accordingly to his/her personal profile and interests, and may deliver additional and user-targeted information about the various objects to the tourist's handheld device.

R4: Increased flexibility through de-coupled communication

The dynamics of pervasive and mobile environments are particularly challenging from the reliability viewpoint. On the one hand, communications are subject to frequent changes and prone to steady disconnections. On the other hand, the data itself is of a highly dynamic nature, and one cannot guarantee neither access to specific data nor reliability and stability of data. Thus, to provide the necessary flexibility it is essential to avoid both tightly-coupled communications and tightly-coupled access to data. This results to: *Ubiquitous environment representations must address flexibility and dynamics through de-coupled communication and information distribution*. (*R4*) In a tourism scenario, the living diary can be used to access very diverse and dynamic information, as well as to interconnect tourists and let them share data or experiences. Thus, tight access to specific data sources as well as tight communication channels between them are very inappropriate.

R5: Ability to handle network disconnections

If uplinks to centrally maintained services are not available due to a disconnection, some sort of redundancy of services and information must be envisaged. If a network access is not possible, at least the locally available devices should be able to interact, respectively, devices and selected services must be able to function in a stand-alone mode. Hence, we derive that: *Ubiquitous environment representations must provide a minimum set of services/information even when the network connection is not ideal*. (*R5*) Living diaries can make local copies of user profiles, locally store information related to visited places, and allow for data exchange between tourists in an ad-hoc manner. We are aware this may lead to using not up-to-date data, but it at least ensures continuity of services.

R6: Addressing diversity to tackle complexity

Due to technological advances, hardware is getting more powerful and smaller at the same time. Various hardware entities already exist and the number and diversity is increasing. Ubiquitous information systems have to take this into account. This means that: *Ubiquitous environment representations must address the increased complexity resulting from the magnitude and diversity of involved entities and technologies*. (*R6*) A living diary application can integrate a arbitrary amount of data sources such as information coming from the Web or sensors such as RDIF tags or satellite-based technologies.

R7: Explicit integration of coordination mechanisms

In the examined scenarios, coordination is either explicitly or implicitly steadily reoccurring. Environments must be accordingly engineered to integrate appropriate concepts to facilitate coordination. We conclude that: *Ubiquitous environment representations must address the increasing need for coordination of coordinables by appropriate mechanisms as it is a precondition for the delivery of the required service quality*. (*R7*) The living diary can assist in coordinating touristic activities. It can be differentiated between individual coordination (e.g., the living diary takes on the purchasing process of several tickets, books guided visits and arranges the chronology of the activities), or collective coordination (coordinative processes among the tourists of a group).

R8: Reusable coordination solutions

A considerable number of problems related to coordination activities are common to all scenarios and exhibit a high degree of similarity. Thus: *Ubiquitous environment representations must provide*

Table 1
Listing of Derived Requirements

ID	Requirement
R1	Ubiquitous environment representations must be able to enrich information coming from the physical world by merging it with information from other (virtual) data sources.
R2	Ubiquitous environment representations must provide an expressive data model that allows to appropriately integrate various data sources and context-sensitive queries.
R3	Ubiquitous environment representations must allow for a refinement of data according to the specific context of the user's current situation.
R4	Ubiquitous environment representations must address dynamics and flexibility through de-coupled communication and information distribution.
R5	Ubiquitous environment representations must provide a minimum set of services/information even when the network connection is not ideal.
R6	Ubiquitous environment representations must address the increased complexity resulting from the magnitude and diversity of involved entities and technologies.
R7	Ubiquitous environment representations must address the increasing need for coordination of coordinables by appropriate mechanisms as it is a precondition for the delivery of the required service quality.
R8	Ubiquitous environment representations must provide standards and generally applicable models to (re-)use for reoccurring problems.

standards and generally applicable models to (re-)use for reoccurring problems. (*R8*) A living diary can arrange meetings based on other tourists position, unassigned time slots, and interest, and furthermore guide and coordinate them to reach the ideal meeting place.

3. UbiCoMo – The model for ubiquitous coordination

Based on our investigation on the requirements for coordination in ubiquitous environments (Section 2), we defined a ubiquitous coordination model (UbiCoMo) and implemented a layered coordination infrastructure accordingly.

To address the potential complexity of ubiquitous environment representations (as from requirement *R6*), UbiCoMo is organized as a layered architecture where each of the six layer encapsulates specific functionalities (see Fig. 1). The factual data layer offers the means for capturing and keeping data from the physical and virtual world (*R1*). To combine both worlds and to have means to beneficially integrate data coming from diverse sources the data model layer exploits the W4 concept (*R2*). Furthermore, as the W4 concept allows to address context information according to several dimension (actor, activity, space, and time), we try to satisfy *R3*. Due to the adoption of a tuple space-based approach, the tuple management is capable of covering the provision and up-to-dateness of the in W4 integrated data (*R4*). As the tuple management layer is responsible for acting locally on a device, the tuple distribution layer provides added-value by interconnecting different entities. This layer is responsible for establishing a network and hence, a federated tuple space that is constituted by the available nodes (*R5*). *R6* is addressed by the openness of the W4 concept and the tuple-space approach which can account for arbitrary types of information.

As UbiCoMo reflects a ubiquitous coordination model and *R7* is the "coordination requirement", several layers are responsible for addressing this requirement. To provide improved coordination various aspects are necessary and for this a combination of several UbiCoMo layers is needed: data model, tuple management and distribution layer. The coordination layer accomplishes the coordinative activities of the potentially great magnitude of occurring entities (*R6*) according to defined laws (*R7*). Reoccurring coordination problems are subsumed in pattern-based solutions (*R8*) on this layer. The application

Table 2
Overview of Design Decisions Related to Defined Requirements

Design Decision	Requirement(s)	Comment	Section
Layered architecture	R6	Appropriate method to show the dependencies of various concepts and to de-compose different concerns	–
Factual data laye	R1	Provides the factual data sets coming from the physical and the virtual world	3.1
Data model layer	R2, R3, R7	Proposes the W4 concept as the data schema for context-sensitive queries according to four dimensions	3.2
Tuple management layer	R4, R7	Integrates the usage of tuple spaces	3.3
Tuple distribution layer	R4, R7	Covers appropriate information management (e.g., replication strategies)	3.4
Coordination layer	R6, R7, R8	Encapsulates the resolution of interdependencies through laws within the business logic or patterns	3.5
Application layer	R6, R7	Subsumes ubiquitous entities, agents and/or services	3.6

layer, finally, represents the interface to the user and provides the requested services (*R6, R7*). Table 2 summarizes these decisions based on requirements.

A further insight conceived during the examinations was that due to the overwhelming amount of diverse data, coming from heterogeneous sources, and ubiquitous services, which shall be accessible in an anytime, anywhere, and anyhow manner, future ubiquitous information systems are inescapably going to be more complex. We derived that such systems will require for sophisticated coordination strategies (see requirements R6, R7, and R8) in order to appropriately address the rising magnitude of user requirements, constraints, and interrelations and hence, the need for context-sensitivity. As a consequence, we studied the abstract theory of coordination (see also Section 1) in more details and integrated appropriate mechanisms. It has been observed that several types of coordinative behaviors are highly similar, often reoccurring, and typical for specific situations [11,12]. Consequently, various such coordination laws and strategies can be subsumed to so-called patterns (see the coordination layer, Section 3.5), which shall provide standardized means to assist in these situations. A detailed listing of various coordination patterns can be found in [3].

The following sections describe each layer of UbiCoMo in more details and relate them back to the defined requirements.

3.1. Factual data layer

In order to support the idea of *Ubiquitously Browsing the World* [6] as mentioned in the introduction to this paper, the possibility of navigating information spaces that can represent a detailed model of the world has to be considered, comprising both present and historic data about its entities, its processes, and its social structure. For this, a proper merging and integration of information coming from two distinct sources must be supported: (*i*) from the *real world* (i.e., through pervasive devices such as sensors), which due to their proximity to the (mobile) user can be directly accessed and any environmental characteristic of the physical world can be acquired and information generated or processes triggered; or (*ii*) from the *virtual world* (i.e., the Web), which shall enable the actors (agents, users etc.) to dynamically and on-demand retrieve additional information related to their current situation (i.e., context-sensitivity). We denote these either physical or virtual data as “facts”. Also, the actors within the environment can be described by facts. Consequently, facts can also represent information about actors (e.g., software agents) such as their state or activities, which is a necessary prerequisite for our coordination environment to enable either direct or indirect coordination between entities in an de-coupled way.

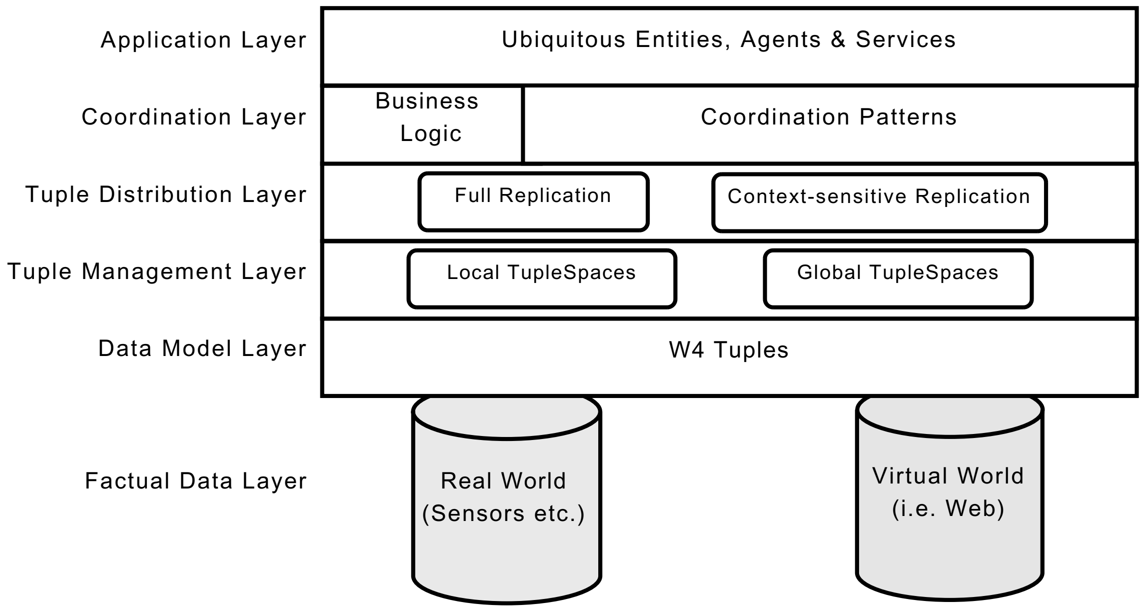

Fig. 1. Architecture Overview of the Ubiquitous Coordination Model (UbiCoMo).

To build a comprehensive data model that allows for integrating various sources, the separation into facts coming from real and virtual world sources is necessary. By this, real world information can be enriched on-demand with information from virtual sources (*R1*).

3.2. Data model layer: The W4 concept

The great magnitude and heterogeneity of potentially resulting amount of facts require for an expressive, yet simple concept to represent this information about the two worlds of the factual data layer. Such a data model must provide means to *model* and *combine* information coming from heterogeneous sources, enable ease of querying and processing, and should account for adaptation to context and incomplete information. Our proposed "context ontology" considers that diverse data about facts can be expressed by means of a 4-fields tuple (*Who, What, Where, When*): "someone or something (*Who*) does/did some activity (*What*) in a certain place (*Where*) at a specific time (*When*)". We refer to this data model as the W4 concept representing an ontology with simple and extensible semantics.

The *Who* field associates a subject to a fact and represents an actor. *Who* may represent a human being, an unanimated part of the context acting as a data source (e.g., the ID of an RFID tag), or the name of a web resource. The *Who* field is represented by a type-value string with an associated namespace that defines the "type" of the entity that is represented. For example, valid entries for this field are: "user:Gabriella", "tag:tag#567".

The *What* field describes the activity performed by the subject. This information can either come directly from the data source (e.g., a sensor is reading a temperature value) or be inferred from other context parameters (e.g., an accelerometer on a PDA can reveal that the user is running), or it can be explicitly supplied (e.g., content of a web page). This field is represented as a string containing a predicate-complement statement. For example, valid entries for the *What* field are: "read:book", "work:pervasive_computing_group", "read:temperature = 23".

The *Where* field associates a location to the fact and thus represents a spatial description. In our model the location may be a physical point represented by its coordinates (e.g., longitude, latitude), a geographic region, or it can also be a logical place. Web resources, for instance, are also expressed as logical places by using an URI. In addition, context-dependent spatial expressions like "here" or "within:300m" can be used for context-aware querying.

The *When* field associates a time or a time range to a fact and is referred to as a temporal description. Also in the case of the *When* field, context-dependent expressions can be defined (e.g., "now", "today", "yesterday", "before") and used for context-sensitive querying.

The W4 concept relies on the assumption that software agents are associated with data sources and are in charge of creating W4 tuples and inserting them in a shared data spaces (see tuple management layer, Section 3.3). Data are sensed from different available sources (e.g., RFID tags, GPS devices, Web services) and information is inferred by combining them in a W4 tuple. Due to the nature of ubiquitous environments (see also *R4*), this sensed data may change frequently. W4 tuples can be generated continuously and thus can account for keeping dynamically changing data up-to-date. The *When* field is responsible for providing this information. By interpreting this field, agents can deduce the timeliness of a W4 tuple. Moreover, a history of generated tuples can be constructed and kept in the data spaces.

Let us consider a simple examples to clarify the W4 concept: Gabriella is walking in the campus park. An agent running on her GPS-equipped PDA can periodically (e.g., every n seconds) create a tuple corresponding to her activities. An example could be the following:
<user:Gabriella;walk:4kmh;area:Uni_Modena_campus;2006/10/17,1753>

The *Who* is entered implicitly by the user at the login, *What* and *Where* can be derived by the GPS (e.g., the speed of Gabriella as measured by the GPS can be used to deduce that she is walking), *When* can be provided both by the PDA or by the GPS.

Moreover, the model is not restricted to human beings but in an abstract sense every coordinable entity can represent an actor. Consequently, also states or activities of agents can be modeled in W4 which is exploited to use UbiCoMo as the coordination environment for agents. A W4 tuple of an agent may be composed as follows: <agent:FactCollector;request:RFID-tag;OOI:food_hall_menu;now>. This example illustrates that an agent called FactCollector (*Who*) (see also Fig. 2) is now (*When*) requesting the ID of an RFID tag (*What*) mounted at the entrance of the food hall (*Where*).

According to coordination theory, to resolve interdependencies a description of four (abstract) entities is necessary: actors, activities, constraints, and interdependencies itself. By using W4, we provide a means to accordingly model these entities: *Who* reflects the actors, *What* the activities, and the constraints can be defined by *Where* and *When*. The interdependencies cannot be directly modeled by W4 but are addressed in the higher coordination layer (see Section 3.5).

The W4 concept can be perfectly exploited for context-sensitive coordination along four different dimensions of context (actor, activity, space, and time), which – as they are accordingly represented in the data model – can, in turn, be exploited and combined as needed to address the specific context of an actor in a particular situation. By adopting W4 tuples, we provide a way to semantically combine diverse data from different sources (*R2*) into one data model, which in turn can be exploited to provide additional and context-dependent information to the user (*R3*). It represents a unique interface for upper layers of UbiCoMo to access heterogeneous facts (*R7*).

3.3. Tuple management layer

The tuple management layer further takes into account the separation between a local and a global element and addresses each differently:

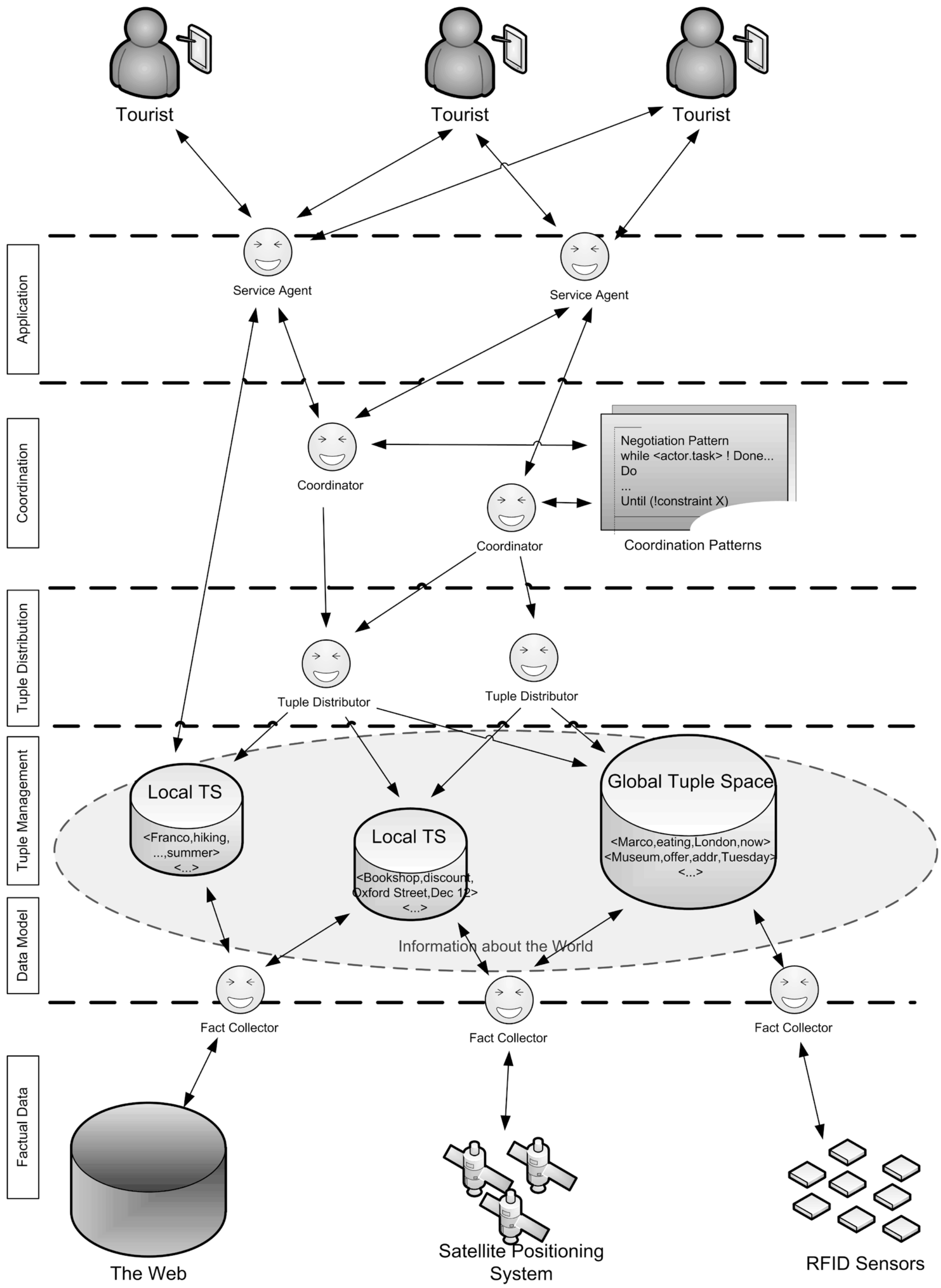

Fig. 2. UbiCoMo applied to the tourism living diary: The yellow "faces" represent agents which are deployed on several levels within UbiCoMo.

Local Element: Due to the inherent properties of decentralized (i.e., peer-to-peer-based) architectures – such as scalability and fault resilience [1] – and their appropriateness for mobile environments [18], the local element of the UbiCoMo architecture is entirely based on unstructured P2P concepts. Concretely, we deploy a decentralized space-based computing approach (SBC) [23], which is very similar to Linda-like systems [14]. The characteristics of mobile and ubiquitous environments like unpredictable, sporadic connectivity can be addressed by the high degree of decoupling, which is provided by the SBC approach. These decoupling mechanisms are threefold: (*i*) *spatial* decoupling: two processes can reside in completely different computational environments and communicate via the space, (*ii*) *temporal* decoupling: the processes do not have to be available at the same time in order to communicate (i.e. asynchronous communication), and (*iii*) *referential* decoupling: entities can communicate with each other and request data without addressing each other directly. As opposed to the idea of the original Linda system [14], which assumed an existing central and persistent data space, this approach is completely decentralized. Every involved node carries a local space and may be replicated with the other available participants of the MANET according to variable strategies, which are covered by a higher layer (tuple distribution layer).

Global Element: The infrastructure may also involve a number of Web-accessible tuple spaces enabling more global queries. In general, an application agent performing a query accesses the local tuple space and/or may refer to remote spaces, which are accessible via long-range transmission technologies provided through infrastructure networks.

Agents can retrieve knowledge facts via simple pattern-matching query mechanisms; i.e., through associative addressing of information stored in tuples which through the W4 concept allows for context-dependent queries even when the information is incomplete. A general challenge for this infrastructure is to identify strategies for evaluating which information (i.e., tuples) to send to some global tuple space and which to keep locally. Such decisions may depend on many factors, such as privacy issues (e.g., a user may not be comfortable of constantly sending his/her GPS location to a public Web server) or performance and scalability reasons.

Due to the inherent decoupling mechanisms of space-based approaches, the tuple management layer ideally addresses *R4*, i.e. flexibility, which is a prerequisite for delivering services in appropriate quality (*R7*).

3.4. Tuple distribution layer

In order to appropriately and reliably distribute information – which in UbiCoMo is stored in form of W4 tuples – a space overlay was developed where agents residing on each of the involved participants are responsible for the distribution of the tuples among the tuple spaces according to appropriate replication strategies. Basically, two diverse types of tuple distribution respectively replication have been designed: (*i*) full, or (*ii*) context-sensitive replication. The former strategy is adopted within the local element when a relatively low number of participants is involved. On a global basis, full replication mechanisms might lack scalability. It is, however, not excluded that also global tuple spaces deploy this full replication strategy in order to distribute some necessary and indispensable information (e.g., addressing) if it is “lightweight” data.

The second strategy allows for tuple replication based on the four context dimensions covered in the W4 data model (i.e., actor, activity, space, and time) and also a combination of these. Consequently, tuples can be distributed more effectively, resulting in less data transmission overhead, less costs, and increased performance. Moreover, information can be provided to the requesting entity in a context-sensitive way through exploiting the four fields of the W4 data model. Hence, the delivered information can be much better targeted to the exact needs of the information requester.

The tuple distribution layer extends the functionalities of the underlying tuple management layer and adds the connection between the local spaces to constitute a federated "quasi-global" space. Hence, this layer is related to covering the same requirements as the tuple management layer, namely *R4* and *R7*.

3.5. Coordination layer

The coordination layer serves for the resolution of the most important element of coordination theory: the *interdependencies*. These are usually implicitly addressed by encapsulating the coordination laws within the business logic of a multi-agent system or application. We propose, however, to provide ways of explicitly addressing the resolution of interdependencies by deploying coordination patterns in order to address general coordination problems. This generalization is a cognitive process where people naturally look for similarities in the world to derive patterns which subsequently can be deployed as best-practices for problems of the same or similar type. As a consequence, the problem solving process can be alleviated and accelerated by using a well-known method.

To deploy a pattern, the coordination problem has to be understood first. Second, the potential solution to the problem can be looked up in a coordination pattern catalogue and the most suitable solution can be chosen. Every pattern comprises different entities, describes the interdependencies and how to resolve them. Patterns denote the least common denominator such that it can be deployed generally. To address specific requirements, it has to be adapted accordingly. The patterns always have to be supplied with the necessary input and, subsequently, according to its internal coordination laws, it generates the output: a solution to the coordination problem.

The patterns concept in UbiCoMo can be deployed in two ways: (*i*) Available reference implementations of coordination patterns in the form of programming libraries can be adopted by invoking the relevant APIs such as the supervisor/worker pattern introduced in [4]. (*ii*) Specifications of patterns as in [3] can be accordingly implemented, which is very similar to the way design patterns are exploited in software engineering [13]. Other examples of coordination patterns would be location-oriented coordination, meeting, or negotiating.

If, however, one problem cannot be resolved by patterns because it is too specific or unique it has to be addressed by implementing the appropriate laws within the business logic, which as a negative effect, may lead to poor extensibility, interchangeability, or re-usability.

Standardized solutions alleviate problem solving processes through re-usability. Hence, the coordination layer covers *R8*. Furthermore, pattern-based coordination is directly linked to optimizing coordination and consequently the service quality (*R7*). The issues raised in *R6*, in turn, are highly depending on coordination and coordination patterns.

3.6. Application layer

The application layer encapsulates the calling agents which are necessary to provide ubiquitous services or applications to requesting users. Agents or, in more general, application-level services can make use of the underlying UbiCoMo layers and encapsulated functionalities as it is required in an anywhere and anytime fashion. By this, a user can interact with the world according to the *Ubiquitously Browsing the World* vision and the responsible agents coordinate with other agents in a context-sensitive fashion to deliver the requested information to the user. We deal with one such application in Section 4.

The application layer comprises the heterogeneous ubiquitous entities (e.g., mobile users) or their surrogate (software agents) (= *R6*); respectively denote the interface to these. Finally, this layer shall provide the requested services to the according quality as claimed in *R7*.

4. An application example using UbiCoMo

In order to demonstrate the ubiquitous coordination model, we provide an example where we apply UbiCoMo to the living diary scenario, which has been introduced in Section 2, in the context of tourism. We accomplish this by modeling the scenario according to the layers of UbiCoMo, providing some implementation details, and discussing the application of the UbiCoMo approach and its benefits. Moreover, some of the coordination patterns frequently appear in such examples and thus, can be exploited beneficially, which is also subject to the discussion.

4.1. Ubiquitous coordination assessment

The goal of the living diary application in this showcase is to generate an added-value by providing context-sensitive information to tourists with respect to users' profiles (i.e., interests) and their current position. Relevant functionalities would be the delivery of additional information to specific objects-of-interests, the generation of recommendations, or user navigation.

To apply UbiCoMo to the tourism living diary example as shown in Fig. 2, the facts about the real world are represented by geographic positions of various objects of interests (e.g., geo-referenced by sensors such as RFID), which, in turn, can be correlated with the real-time position of a user acquired through GPS, respectively with a calculated estimation according to a certain probability of where a tourist is heading to while being on the move. Data coming from the virtual world (the Web) can serve as any additional information that may represent an added-value to the tourist in his/her current situation (first and lowest layer in Fig. 2). Fact Collector agents are interfacing the first and the second layer and are responsible for collecting these data.

On the data model layer, W4 tuples may encapsulate the following information: The *Who* field may be represented by the tourist itself, by other tourists having a similar profile, or by objects-of-interests; *What* is either the current activity a tourist intends to do or the information an object-of-interest can offer; *Where* is the geographical position; and *When* describes the point in time or time range when the information was, is or will be relevant.

In the tourism living diary example, most of the data may be requested from remote servers using (wireless) infrastructure networks. Hence, the tuple spaces local to a MANET may not often offer any further benefit to this scenario (see layer three and four in Fig. 2). It may only make sense for a group of tourists which in certain circumstances may be disconnected from any infrastructure network. In that case, at least the local MANET may be used to share locally available data or invoke corresponding services and hence ensure a minimum availability of the living diary. In addition, a full replication strategy on the tuple distribution layer, which is accomplished by Tuple Distributor agents, may also only make sense with respect to, for instance, sharing contact information. Context-sensitive replication, however, can be exploited in this scenario to its full potential, due to the fact that context can be examined based on all four dimensions of the W4 data model. Consequently, not all profile data and correlating experiences must be exchanged but only relevant ones – according to one's context.

Two coordination patterns (layer five in Fig. 2) seem to be beneficial in the living diary scenario: First, in order to purchase tickets, receive discounts, or establish dynamic packages of various products, the negotiation pattern [3] may be deployed: Agents operating on behalf of the user release proposals for a purchase. They receive counter-proposals from agents representing the suppliers which are evaluated by the user-agents according to his/her profile until a proper offer is found. The user only has to decide.

Second, in order to actually engage touristic products, a user needs to be at a certain place at a certain time. To organize these meetings effectively, the meeting pattern [3] may assist. This holds also true

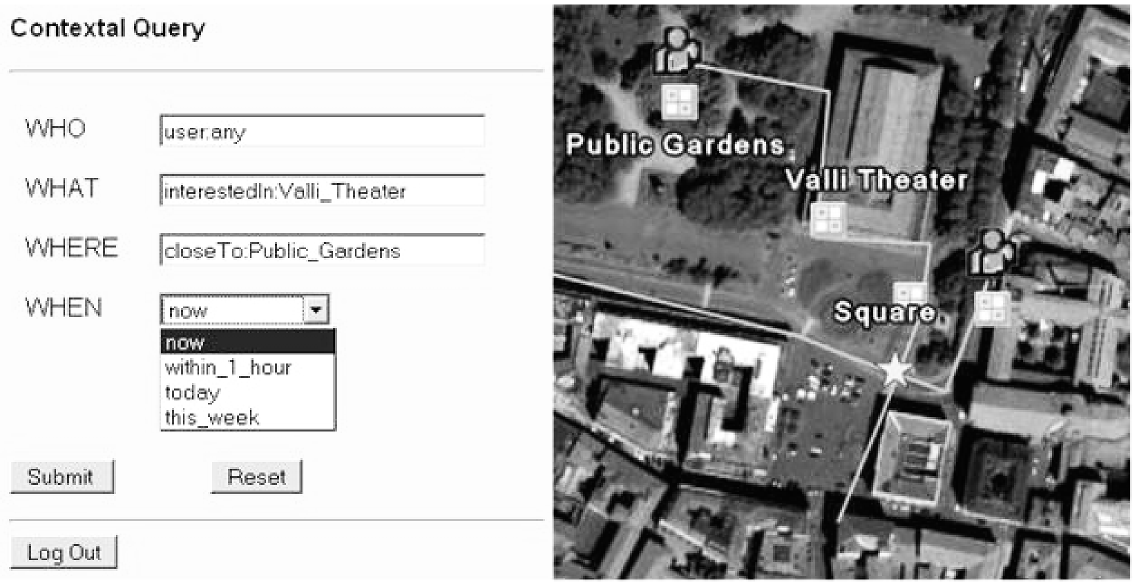

Fig. 3. Map-based representation of the tourism living diary: A simple web-based form is used to access the diary (left). The context information can be presented by the integrated Google Earth interface (right). Here, two users with similar profiles (=interested in the Valli Theater) met close to the public gardens in Reggio Emilia (the star indicates the meeting place, the squares are objects-of-interests).

for a group of tourists who wants to meet at certain points (in time and space), or also for people with similar profiles in order to exchange experiences. The meeting pattern takes constraints such as preferred time and location and the involved tourists and their profiles as input and calculates and proposes an ideal meeting place as the output (i.e., the solution to the coordination problem). Figure 3 illustrates an example where traces of two tourists and their meeting point (indicated by a star) are shown. Figure 2 shows the agents (Coordinators) that are in charge of the pattern deployment.

Finally, ubiquitous services – mediated and provided by Service Agents – may be outdoor and indoor navigation, delivering context-sensitive information, recommendations, ticketing, content or experience sharing, and "social" applications (e.g., communication, chat, photo exchange).

4.2. Implementation details

The concrete implementation of the prototypical application example was done equipping mobile users (simulated tourists) with PDAs with RFID readers and GPS receivers. Users can be localized by their GPS position which is stored in W4 tuples. Additional information to certain objects-of-interests can be retrieved by reading RFID tags which are merged with the corresponding information about the tag's object coming from web-accessible sources to constitute the W4 tuple. The tuple spaces local to each involved peer and the tuple distribution strategies are developed by using the open-source framework LighTS[6] [25] as the basis. Our current implementation of a remote tuple space consists of a Tomcat Web server giving access to a Postgres database that stores the W4 tuples. We developed JSP and Servlets implementing a W4 interface. As a demo user interface, we integrated Google Earth into our example

[6]See http://lights.sourceforge.net.

as a showcase of the tourism living diary (see Fig. 3). Stored W4 tuples can be accessed and displayed on a Google Earth representation.

4.3. Discussion

In this living diary scenario, the Ubiquitous Coordination Model can offer benefits by providing the means of correlating real facts with virtual facts represented by W4 tuples, which can be exploited to provide – through appropriate data transmission tech-nology – an added-value to the tourist. The delivered information can be targeted to the user according to the four dimensions of the W4 data model and hence, the value of this information is increased due to high correlation with the actual real context the user is currently in and his/her profile.

Activities modeled in W4 may be represented by the tourist's interests and what he/she wishes to experience during the holiday. Interdependencies or conflicts resulting from a tourist's intended activities, the timely order, the offers of the site, but also, for instance, potentially conflicting interests of several tourists within a group can be resolved by the appropriate coordination means offered by UbiCoMo. In this example, two particular coordination patterns of UbiCoMo can assist in resolving general problems of negotiating prices or coordinating appropriate meeting points between mobile users.

5. Related work

In the area of context-awareness, most focus so far has been on acquiring individual pieces of contextual information and less on making it available in a more formalized model [28]. As a consequence, due to the lack of expressive models, complex ubiquitous situations cannot be effectively processed. More structured approaches to modeling context consider sets of environmental variables which can be queried [27], or structured models in which contextual information can be aggregated and enriched by features related to information imperfection, e.g., temporal aspects [16]. These approaches tend to make the context model quite complex, and it is difficult to effectively manage services based on these models. The W4 concept avoids this problem by structuring all characteristics of the context in four well-defined fields. A similar proposal in [7] adopts a seven-field data structure to describe the context, four of which corresponding to our W4 fields. However, the purpose is for managing consistency between data from multiple and heterogeneous sources rather than to support contextual activities.

Context-sensitive data structures CSDS [24] provide a unified interface to contextual information. This interface concept is similar as in W4. In CSDS, context items are defined as pieces of data generated by agents. Every agent, however, can generate its own CSDS describing his current context arbitrarily. We believe that it is more beneficial to specify a certain structure a priori (as the four fields in W4) in order to facilitate interoperability between heterogeneous systems. Similarly, in [17] the notion of *views* was introduced denoting a individualized projection of all data available to one specific reference agent. This is a beneficial concept as the large amount of data can be tailored to the real specific needs of an agent according to its current context. Such a mechanism is not explicitly provided In UbiCoMo, but similar results can be acquired through context-dependent queries formulated by the agent itself. Schelfthout et al. [26] define *views* based on the network connectivity within a MANET specifying context as the configuration of participating nodes and the corresponding available data. They use ObjectPlaces as the coordination middleware to manage context which is subject to change because information on reachable nodes changed, or the set itself of reachable nodes changed.

We are careful in denoting the W4 concept as an ontology for representing context. The semantics are simple and extensible (e.g., by namespaces), which due to adaptability we consider as a strength. Another approach would be the Context Ontology Language (CoOL) [29], which uses the Aspect-Scale-Context model where each aspect can have several scales to express context. This approach is especially useful for quantifying and converting but may suffer from being constricted and inflexible. In [10], a similar concept to model context as in W4 was presented. The work, however, particularly focuses on the context modeling by using well-defined and established ontologies for e-learning scenarios. UbiCoMo, instead, discusses a more comprehensive approach also covering the necessary infrastructure (i.e., middleware).

The idea of coordinating agents via the mediation of a space of coordination artifacts [22] is definitely of relevance to our work. Coordination artifacts are sorts of objects that represent basic units of coordination and that, by encapsulating behavior other than data, can effectively embody coordination laws and can effectively act as virtual representations of objects and facts of the real world. W4 tuples, due to their uniform structuring, are more suitable for open and dynamic interactions than general, unstructured, artifacts. Still, W4 tuples cannot embody behavior and functions, which have to be encapsulated in other layers of UbiCoMo. In our view, this is a different way of enforcing separation of concerns between the factual data layer and the layer of coordination laws.

Sensing and integration of the (physical as well as virtual) environment is an essential part in UbiCoMo. Weyns et al. [30] focus on having the environment participating in the MAS as an active entity to allow for subjective (i.e., intra-agent dependencies towards other agents) as well as objective coordination (i.e., inter-agent coordination issues external to the agents related to the environment the agent is embedded in). As argued in [2], the coordination logic can be encapsulated somewhere on an axis between the environment and the coordinable (i.e., agent), which highly depends on the intended application. The environment in UbiCoMo is not modeled as an active entity but it is integrated by agents sensing and actuating or changing the environment in order to leave cues.

With respect to the tuple management and distribution layers of UbiCoMo, Linda [14] was the result as the first system using a central tuple space in order to store and distribute data. It was proposed that effective systems in the domain of parallel computing have to emphasis a high degree of decoupling among computing resources. The participants share information stored in a globally accessible, persistent data store, implemented as a centralized tuple space. The main difference of this system compared to the decentralized space-based computing approach of this work is the server-based architecture representing a bottleneck and hampering flexibility. The SBC implementation, however, is a peer-to-peer concept with rich replication mechanisms. Furthermore, due to its small footprint, it is possible to deploy the proposed system on mobile and portable devices. These features make it a light-weighted middleware perfectly suitable for mobile ad-hoc networks.

Other proposals focus on middleware infrastructures to facilitate users in gathering information from an environment enriched with pervasive computing devices. TinyLime [9] proposes a tuple-based model to access information from sensors in the proximity of the user, and also supports ad-hoc sharing of accessed information between close users [17] describe EgoSpaces, a context-aware middleware that deploys tuple spaces too but without any pre-defined context dimensions. To some extent, these can be considered as infrastructures for *Ubiquitously Browsing the World*. However, they focus on specific areas whereas our proposal is more general-purpose.

Proposals for architectures or models for coordinative systems are sporadically available such as in [8, 19]. Ciancarini states that coordinative processes can always be described by the triple of $\{\mathtt{E},\mathtt{M},\mathtt{L}\}$ [8]. In this generic model, $\{\mathtt{E}\}$ represents the coordinable entities. These can be processes, services, agents, or human beings. $\{\mathtt{M}\}$ stands for the coordination media (i.e. communication channels), which serve as connectors between the entities and facilitate communication. {L} is referred to the coordination laws defining how the interdependencies which are the focal point in coordination have to be resolved and hence, semantically define the coordination mechanisms. Ciacarini's ideas are reflected in UbiCoMo: {E} is addressed at the application layer (i.e., basically agents and/or services), {M} is covered by the tuple space-based approach that provides tuple management and tuple distribution functionalities, and the coordination layer covers the {L} element in the form of coordination patterns.

Patterns and related concepts covered in this paper, have also been intensively investigated within software agent research [21]. Moreover, patterns have often proven to be helpful tools in order to tackle similar and reoccurring problems such as in architecture design [5], software engineering [13], or coordination [3,11]. In [12], an approach is presented that tries to derive collaboration patterns by analyzing ad-hoc business processes in order to gain more in-depth knowledge and to optimize.

6. Summary and further work

In this work, we propose a Ubiquitous Coordination Model (UbiCoMo) and introduce a corresponding implementation that provides comprehensive concepts for context-sensitive coordination of agents in ubiquitous environments, and show its applicability to an ubiquitous scenario example (the living diary). Hence, UbiCoMo shall serve as an enabler for our *Ubiquitously Browsing the World* vision, which allows mobile users to access information about the physical world in a context-dependent way relative to four dimensions (actor, activity, space, and time) enriched by additional information coming from the virtual world (i.e., the Web). The contribution of this work to engineering environments for multi-agent systems is the provision of a layered coordination model comprising six layers that takes into account the necessary elements to allow for coordination of agents, ubiquitous entities, and services. To summarize, these elements are an expressive data model (W4), flexible tuple space-based information distribution and management concepts, and coordination patterns that can be exploited to resolve numerous standard coordination problems.

In [2] the challenges of coordination in ubiquitous environments are discussed. One major issue is the integration of semantic mechanisms and the adoption of ontology engineering. One limitation of UbiCoMo is that to date no established ontologies have been used. Ontologies, however, are not domain-independent. UbiCoMo is designed to be generally applicable as coordination infrastructure for ubiquitous environments. Further research work will have to be invested in finding ways to integrate ontologies based on the respective application area. For this, also the W4 concepts will need further extensions. At the moment, W4 tuples are organized in a flat structure. For optimization issues, it would be interesting to examine hierarchical structures, too based on either abstract classifications or on W4 tuple combination, aggregation, or derivation. For this, an XML-based representation of W4 tuples may prove to be beneficial. Consequently, semantic analysis shall be alleviated to allow more semantic forms of pattern matching and correlation among W4 tuples and the identification and exploitation of semantic relations between the tuples. Moreover, as environments become more and more pervasive and "intelligent" [2] it will be an engineering requirement not only to integrate them but make them actively participating entities in UbiCoMo.

The UbiCoMo infrastructure to date only provides full replication mechanisms and we are currently implementing context-sensitive strategies which are going to be evaluated subsequently. Hence, no results can be provided yet about the scalability of the context-sensitive replication strategy especially within the local element of the tuple management and distribution layers. Finally, further coordination patterns shall be identified, described, and implemented in order to provide a more comprehensive pattern catalogue; possibly by investigating research efforts and results of the domain of social sciences (e.g., [15]) or by conducting social network analysis as proposed in [11].

References

[1] S. Androutsellis-Theotokis and D. Spinellis, A survey of peer-to-peer content distribution technologies, *ACM Comput Surv* **36**(4) (2004), 335–371.
[2] M. Bortenschlager, G. Kotsis and M. Mamei, Current Developments and Future Challenges of Coordination in Pervasive Environments, in: *The 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (*WETICE*)*: Workshop on Interdisciplinary Aspects of Coordination Applied to Pervasive Environments: Models and Applications* (*CoMA*), Paris, 2007.
[3] M. Bortenschlager, G. Kotsis and S. Reich, A Generic CoordinationArchitecture as an Enabler for Mobile Collaborative Applications, in: *Distributed and Mobile Collaboration* (*DMC 2006*) *Workshop – WETICE*, Manchester, UK, 2006.
[4] M. Bortenschlager, G. Kotsis and S. Reich, An Architectural Approach to Apply the Supervisor/Worker Collaboration Pattern to Nomadic Workspaces, in: *The 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (*WETICE*)*. Distributed and Mobile Collaboration Workshop*, Paris, 2007.
[5] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad and M. Stal, *Pattern-Oriented Software Architecture: A System of Pattern*, JohnWiley, 1996.
[6] G. Castelli, A. Rosi, M. Mamei and F. Zambonelli, Browsing the World: Bridging Pervasive Computing and the Web, in: *InternationalWorkshop on Ubiquitous Information Systems*, Muenster, Germany, 2006.
[7] X. Chang and S.C. Cheung, Inconsistency Detection and Resolution for Contextaware Middleware Support, in: *4th International ACM Symposium on the Foundations of Software Engineering*, Lisbon, Portugal, 2005.
[8] P. Ciancarini, CoordinationModels and Languages as Software Integrators, *ACM Comput Surv* **28**(2) (1996), 300–302.
[9] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy and G. Picco. Mobile Data Collection in Sensor Networks: The TinyLIMEMiddleware, *Pervasive and Mobile Computing* **1**(4) (2005), 446–469.
[10] R. de Freitas Bulcao Neto and M. da Graca Campos Pimentel, Toward a Domain-Independent Semantic Model for Context-Aware Computing, in: *LA-WEB '05: Proceedings of the Third Latin American Web Congress*, Washington, DC, USA, IEEE Computer Society, 2005, p. 61.
[11] D. Deugo, M. Weiss and E. Kendall, Reusable Pattern for Agent Coordination, in: *A. Omicini, F. Zambonelli, M. Klusch and R. Tolksdorf*, editors, Coordination of Internet Agents: Models, Technologies, and Applications, Springer, 2001, pp. 347–368.
[12] S. Dustdar and T. Hoffmann, Interaction Pattern Detection in Process Oriented Information Systems, *Data and Knowledge Engineering*, 2006.
[13] E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional, 1995.
[14] D. Gelernter, Generative Communication in Linda, *ACM Trans Program Lang Syst* **7**(1) (1985), 80–112.
[15] P.H. Gulliver, *Disputes and Negotiations – A Cross-cultural Perspective*, Academic Press, 1997.
[16] K. Henricksen, J. Indulska and A. Rakotonirainy, Developing Context-aware Pervasive Computing Applications: Models and Approach, *Pervasive and Mobile Computing* **2**(1) (2006), 37–64.
[17] C. Julien and G.-C. Roman, EgoSpaces: Facilitating Rapid Development of Context-Aware Mobile Applications, *IEEE Transactions on Software Engineering* **32**(5) (May 2006), 281–298.
[18] W. Kellerer, R. Schollmeier and K. Wehrle, Peer-to-Peer in Mobile Environments, in: *R. Steinmetz and K. Wehrle*, eds, Peer-to-Peer Systems and Applications, 2005, pp. 401–417.
[19] M. Klein, Coordination science: Challenges and directions, in: *Coordination Technology for Collaborative Applications*, 1996, pp. 161–176.
[20] T.W. Malone and K. Crowston, The Interdisciplinary Study of Coordination, *ACM Comput Surv* **26**(1) (1994), 87–119.
[21] H.S. Nwana, L. Lee and N.R. Jennings, Coordination in Software Agent Systems, *The British Telecom Technical Journal* **14**(4) (1996), 79–88.
[22] A. Omicini, A. Ricci and M. Viroli, Agens Faber: Toward a theory of artefacts for MAS, *Electronic Notes in Theoretical Computer Sciences* **150**(3) (29 May 2006), 21– 36. 1st International Workshop "Coordination and Organization" (CoOrg 2005), COORDINATION 2005, Namur, Belgium, 22 April 2005, Proceedings.
[23] A. Omicini and F. Zambonelli, Tuple centres for the coordination of internet agents, in: *SAC'99: Proceedings of the 1999 ACM Symposium on Applied Computing*, New York, NY, USA, ACM Press, 1999, pp. 183–190.
[24] J. Payton, G.-C. Roman and C. Julien. Context-Sensitive Data Structures Supporting Software Development in Ad Hoc Mobile Settings, in: *Proceedings of the 3rd International Workshop on Software Engineering for Large-Scale Multi-Agent Systems, co-located with ICSE 2004*, May 2004, pp. 34–41.

[25] G.P. Picco, D. Balzarotti and P. Costa, LIGHTS: A Lightweight, Customizable Tuple Space Supporting Context-Aware Applications, in: *Proceedings of the 20th ACM Symposium on Applied Computing* (*SAC05*), Santa Fe (NewMexico, USA), March 2005, ACM Press.

[26] K. Schelfthout, T. Holvoet and Y. Berbers, Views: Customizable Abstractions for Contextaware Applications in MANETs, in: *SELMAS '05: Proceedings of the Fourth International Workshop on Software Engineering for Large-Scale Multiagent Systems*, ACM Press, 2005, pp. 1–8.

[27] B. Schilit, N. Adams and R. Want, Context-Aware Computing Applications, in: *1stWorkshop onMobile Computing Systems and Applications*, Lake District, UK, 1994.

[28] A. Schmidt, K.A. Aidoo, A. Takaluoma, U. Tuomela, K.V. Laerhoven and W.V. de Velde, Advanced Interaction in Context, in: *International Symposium on Handheld and Ubiquitous Computing*, Karlsruhe, Germany, 1999.

[29] T. Strang, C. Linnhoff-Popien and K. Frank, CoOL: A Context Ontology Language to enable Contextual Interoperability, in: *Proceedings of 4th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems* (*DAIS2003*), Paris/France, Springer Verlag, 2003, pp. 236–247

[30] D. Weyns, A. Omicini and J. Odell, Environment as a First Class Abstraction in Multiagent Systems, *Autonomous Agents and Multi-Agent Systems* **14**(1) (2007), 5–30.

Authors' Bios

Manfred Bortenschlager Manfred Bortenschlager is researcher at the Mobile and Web-based Information Systems Group of Salzburg Research. His research interests are focused on coordination models and mechanisms for pervasive information systems applied to telecooperative applications. He concentrates on coordination theory, P2P networks, agent technology, and space-based computing. He is involved in several national and international (applied) research projects related to above mentioned topics. Currently, Manfred conducts his PhD studies at the Telecooperation Department at the Johannes Kepler University Linz on coordination architectures for pervasive environments.

Gabriella Castelli Gabriella Castelli is currently a Ph.D. student at Department of Engineering Sciences and Methods at University of Modena and Reggio Emilia. She obtained the Laurea degree in Computer Science in 2006. Her main research interests include pervasive computing, context-awareness, location aware services and web 2.0 technologies. She is a member of AIIA.

Alberto Rosi Alberto Rosi is PhD Student at the University of Modena and Reggio Emilia in Industrial Innovation Engineering, since 2006. He obtained the Laurea degree in Business Management in 2005 from the University of Modena and Reggio Emilia. His current research interests include distributed and pervasive computing, context-awareness, and sensor networks.

Franco Zambonelli Franco Zambonelli is a professor of computer science at the University of Modena and Reggio Emilia. His research interests include distributed and pervasive computing, multiagent systems, and agent-oriented software engineering. He received his PhD in computer science from the University of Bologna. He is a member of the IEEE, the ACM, AIIA, and TABOO.