Knowledge Networks for Pervasive Services
Nicola Bicocchi, Marco Mamei, Franco Zambonelli
University of Modena and Reggio Emilia, Italy
{name.surname}@unimore.it

Mathias Baumgarten, Maurice Mulvenna
University of Ulster, United Kingdom
{m.baumgarten,md.mulvenna}@ulster.ac.uk

ABSTRACT
Technologies to pervasively acquire information about the physical and social worlds – as needed by services to achieve context-awareness – are becoming increasingly available. Paradoxically, the risk is to make pervasive services overwhelmed by growing amounts of contextual data, and unable to properly exploit them. This calls for specific approaches to automatically organize and aggregate such data before delivering it to services.

Contextual data items should form a sort of self-organized ecology within which they autonomously link and combine with each other into sorts of “knowledge networks”. This can produce a compact and easy-to-be-managed higher-level knowledge about situations occurring in the environment, and eventually can make services able to easily acquire “situation-awareness”. In this paper, after having framed the key concepts and motivations underlying “situation-awareness” and our “knowledge networks” approach, we present the design and implementation of a “knowledge networks” prototype, intended as a tool to support self-organization and self-aggregation of contextual data item to facilitate their exploitation by pervasive services. A representative case study in the area of adaptive pervasive advertisement is introduced to clarify the concepts expressed, to exemplify the actual functioning of the toolkit and of some specific algorithms integrated within it, as well as to evaluate its effectiveness.

Keywords
Pervasive Computing, Self-organization, Context Awareness, Knowledge Networks.

I. INTRODUCTION
The mass diffusion of pervasive computing technologies such as sensor networks [Est02], sensing-enabled smart-phones [Bic08, Cam08], RFID tags [Wan06], localization tools [HigB01, Bel08], will soon make pervasively available an incredible amount of real-time information about the physical world, its processes, and its objects. In addition, the success of participatory Web tools is feeding the Web with information of any kind about any topic. For instance, tools such as Google Earth and Google Latitude get continuously enriched by geo-located contextual information coming from very diverse communities and related to a variety of facts and events situated in the physical and social worlds [Hon08, UliG07, Cas07].

Overall, the above trends contribute to increase the amount of contextual information that can be exploited by pervasive services in order to achieve high degrees of contextual awareness and self-adaptability, as well as to deliver brand new innovative services for interacting with the world around us. However, the fruitful exploitation of the above described information by services calls for: (i) notable communication efforts to retrieve, from a variety of diverse devices (and possibly from remote sources) all needed information; (ii) notable computation efforts to analyze available information, with the goal of making them more meaningful (i.e., associated to situations) and ultimately machine understandable.

In this context, there is an urgent need to investigate principles, algorithms, and tools, to organize, aggregate, and to enrich this growing amount of distributed information to make it easier to access, more meaningful and, consequently, better understandable, by pervasive services. In particular, we believe that there must be an evolution from a model of simple context-awareness [Dey00], in which the focus is providing services with simple interfaces to access heterogeneous context providers, leaving to services themselves the burden of understanding the retrieved information, towards a model of situation-awareness [Bau06] in which a middle layer is in charge of organizing sparse pieces of information in order to provide services with a pre-digested and more comprehensive higher-level knowledge related to a “situation” of interest.

To pave the way to situation-awareness, our vision considers the existence of a sort of fully self-organizing and self-adaptable middleware layer in charge contextual data management, i.e., a “Knowledge Networks”, and mediating between contextual data producers and pervasive services. Within the KNs layer, the increasing mass of contextual information assumes the form of a sort of “self-organized data ecosystem”, in which individual items of contextual information self-relate, self-organize and self-aggregate with each other. This can lead to structured and meaningful collections of related knowledge items and, possibly, to the self-production of higher-level concepts than those expressed by individual items. Thus, KNs may effectively support services by allowing them to reach, with reduced efforts, a comprehensive understanding of “situations” and, consequently, a higher-degree of adaptability and autonomicity.

Within such a vision, the key contributions of this paper are:

1. To motivate and substantiate our vision via a simple, yet general and representative, case study in the area of pervasive advertisement displays. The case study, which is also adopted as a running example through the paper, exemplifies the need of modern pervasive services to evolve from models of
To identify the general key general concepts behind our idea of KNs, together with a general reference architecture of knowledge networks. Such concepts and architecture are of a general nature, and can be useful to clarify some general concepts related to the management of contextual data, behind our specific experience;

3. To describe the design and implementation of a knowledge KNs network prototype we have developed. In order to evaluate the effectiveness of the prototype and to exemplify some algorithms for self-organization integrated within it, knowledge networks are put at work within the introduced case study.

The remainder of this paper is organized as follows. Section 2 introduces a motivating case study in the area of pervasive advertisement services. Section 3 introduces the key concepts related to KNs and rational behind the design of the KNs toolkit. Section 4 details the current prototype implementation of the KNs. Section 5 presents, by applying them to the case study, some representative algorithms for knowledge management currently integrated in the KNs toolkit. Section 6 presents some experiments to evaluate the KNs architecture and its algorithms. Section 7 discusses related works and Section concludes.

2. CASE STUDY SCENARIO

To guide the discussion, let us consider the scenario of a modern exhibition center, like a big museum or a stadium. In this kind of scenarios, it is realistic to assume the presence of a pervasive infrastructure of embedded devices such as sensors of various types, lots of WiFi access points, RFID tags and location systems such as GPS devices. In fact, exhibition centers may afford the costs of deploying such infrastructures if this allows them to provide better services that consequently may attract a higher number of visitors, which in turn may lead to higher revenues. Furthermore, the same type of infrastructure may be used to increase security and to provide pervasive safety and communication mechanisms.

As a specific example of a service that can be attractive to visitors and that can also attract revenues, we consider the presence of a number of advertising screens that can be used to display to visitors information about the exhibition itself as well as commercials. Today, such advertising screens display generic information in a simple cyclic way that is independent of the situation they operate in (i.e., independent of who is actually in the proximity of that screen). Instead, a “smart” service can decide what information to display on the basis of the available contextual information (e.g., capturing user profiles by accessing Bluetooth-enabled PDAs owned by users, or by reading RFID tags worn by them). This would increase the value of the displayed advertisement both for users and for advertising companies (see Figure 1).

The problem is that in a large exhibition center with many thousands of people, and with a large number of devices that produce contextual information, a single software component on a screen would have to manage an incredible amount of information to get a clue of what to do. Such information may include: (i) thousands of possibly incomplete user profiles that have to be synchronized with statistical information available elsewhere or with some information extracted from other sources, (ii) a multitude of sensorial data detailing what users are currently doing, (iii) historical data detailing what they have done in the past to be possibly used for understanding what they will do in the future. Also, the components on dispersed screens may have to coordinate their actions to, e.g., limit the amount of commercials of a given company to show.

In addition, screens have to deal with other context data, for example coming from a sensor network monitoring environment conditions and Web content to decide the advertisement to show. For example, should a sensor network determine that the environment around the screen is very noisy, the screen can decide automatically to insert subtitles in the advertisement. Similarly, should the sensor network determine that the environment is very hot and humid, they system could favor commercials for iced soft-drinks rather than other products.

The case study outlines the potential for the emergence of the following paradox: the large amount of information available around, instead of being able to provide useful information can make services unable to act properly. That is, being able to access contextual information does not imply becoming aware of what’s happening around and, ultimately, be able to act accordingly.

It is also worth noticing that the presented case study scenario presents challenges and issues that are common to a wide number of other applications. Therefore, our considerations and proposals can be applied in a lot of other circumstances.
3. KNOWLEDGE NETWORKS DESIGN PRINCIPLES

To make contextual information meaningful and useful, some tools must be made available to pervasive services that can properly correlate and pre-digest contextual information so as to provide them with a higher-level understanding of situations around, without forcing them to access and manage large amounts of data internally.

![Figure 2: The Knowledge Networks approach.](image)

In particular, to avoid pervasive services to access and digest large amounts of data directly, a sort of "middle-layer" must be placed in between the data sources and the services. Such a middle-layer will be in charge of digesting data items, analyze them and build a compact, higher-level view of the context.

With reference to the case study, such a layer could facilitate the aggregation of user profiles, possibly merging them with sensorial information, in order to provide situation-specific knowledge to services and enabling them to immediately act on its basis. For instance, one can think of aggregating individual data items describing users with similar interests. This can define a new, higher-level, aggregated data item eventually, representing in a compact way the overall situation of users around a screen, e.g., "there are 70% of women who are interested in modern art" or "80% of visitors are approaching the cafeteria". By correlating such information with other sources (i.e., ambient sensors), one can easily infer, for example, "80% of visitors are approaching the cafeteria AND it is very warm and humid". In the case study, having the possibility of accessing information of this kind can be very useful for quickly deciding what advertisement to show on a screen.

Clearly, to be effective in pervasive scenarios, the envisioned middle-layer has to rely on a distributed and lightweight architecture and must strongly exploit self-organization within. In particular, data organization, aggregation, and generation of higher-level data items must occur in an adaptive way and without requiring human intervention. That is, the envisioned middle-layer must define an underlying distributed “ecosystem” that is populated with data items that interact and aggregate autonomously with each other. In particular, such a middle-layer should not be constructed with a predefined structure in mind (e.g., hierarchical). Instead, a "liquid" structure should emerge on the fly from the relationships among data items.

Our idea of knowledge networks is fully in line with the above perspective (see Figure 2). KNs are a kind of lightweight "middle-layer" concept in which atomic units of knowledge are automatically processed, related with each other, combined into high-level concepts, and eventually made available to services via a dedicated querying interface.

When considering that even relatively small network scenarios can generate enormous amounts of knowledge, it is necessary that KNs can provide different views and perspective on the data, as well as flexible means of correlating and managing knowledge. Furthermore, different kinds of services may have different needs in terms of type, scope and format of knowledge required.

Accordingly, one has to consider the possibility of a multiplicity of KNs to be created and co-exist within a distributed knowledge space where each network is limited by clearly defined knowledge boundaries in order to serve application-specific and/or service-specific goals. Although the context is the same for all situations (and thus the basic contextual information is the same) the way in which this has to be perceived and elaborated by services may depend on the specific type of service one wants to deploy. For instance, in the case study, a service to display commercials may be more interested in the gender distribution around in order to decide whether to advertise ties or perfumes, while a service to display information about specific events may be more interested in the cultural distribution of people around in order to decide whether to inform about a poetry lecture or about an on-going comedy show.

Obviously, it is illusory to identify all possible dimensions in which knowledge may be organized. However, it is feasible to identify a given subset that is useful for various applications. This includes:

- **A semantic dimensions**, in which knowledge atoms that are related to a situation relate to each other according to the concepts that are available or inferred from e.g. a shared ontology. This can be the case for knowledge that facilitates and supports spontaneous interoperability in pervasive computing and service-orientated computing, or of knowledge related to inferring users’ activities from a variety of heterogeneous sensorial information.

- **A spatial dimensions** in which knowledge atoms that are related to a local fact can network to knowledge atoms at different locations (or distribute/replicate themselves in different locations). This can be of use to express e.g. distributed situations, in which spatiality actually refers to physical spatiality, and which can be of great use for pervasive services. Also, we could conceive any class of spatially distributed P2P structures to distribute knowledge across a network and to facilitate access to knowledge (as in the case of e.g. knowledge brokers).

- **A temporal dimensions**, in which knowledge atoms express facts, which have occurred (or are about to occur) at different times. This can be the case for elaborating knowledge for predictive purposes: starting from a situation at a given moment in time, then analyzing and extracting new knowledge in the form of a KNs expressing the most likely future situation.

These considerations summarize into the conceptual reference architecture for KNs depicted in Figure 2. The figure also shows that KNs can also be organized around additional application-specific dimensions in which knowledge atoms may be organized.
in multiple KNs that serve different purposes, and possibly overlap with each other (as in the application example, where we have exemplified how a service may need to be aware of the gender situation and another of the cultural situation).

A possible general criticism of the proposed middle layer to manage context information approach is that it does not eradicate the problem of analyzing large amounts of information, but simply passes it to a different component layer. Although this may be true to some extent, one should consider that: (i) the approach promotes a clear separation of concerns that – as always in software engineering – can notably reduce the complexity of developing and maintaining services; and (ii) in a distributed setting, KNs can take care of knowledge management duties that would have been otherwise replicated inside each service. The latter point in particular has the potential to optimize the process of managing knowledge in a distributed environment.

![Figure 3: A Conceptual reference architecture for KNs.](image)

4. THE KNOWLEDGE NETWORKS IMPLEMENTATION

The KNs toolkit has been implemented on the ACE (Autonomic Communication Element) model developed within the CASCADAS EU project [Hof06].

This model leverages existing autonomic component-models and adaptive agent-based model with features conceived to facilitate the design and development, in networked scenarios of heterogeneous devices, of complex, self-adaptive and self-organizing, services [Hof06].

In particular, the implementation of the KNs toolkit relies on two basic classes of components, knowledge atoms and knowledge containers [Bau06].

A knowledge atom represents the atomic unit of knowledge, and is typically connected to a data source. A knowledge atom provides a uniform abstraction to access contextual information independently of its type, size or context. This is required to provide generic access to knowledge from within the knowledge network as well as from services and components that are outside, independently of the specific characteristics of the data source (whether, e.g. a sensor, a tag, or a Web atom). In addition, a knowledge atom incorporates relevant descriptions of the knowledge/data associated with it, such as context, system and usage based information, as well as any information relevant for the creation, and maintenance of the knowledge atom itself. This makes each knowledge atom fully self-descriptive and as such provides information relevant for different organizational purposes as provided by the network. For instance, in the case study, each of the user profiles would be represented as individual knowledge atoms. Most importantly, knowledge atoms can possibly link to each other to create clusters and networks of related information.

A knowledge container, on the other hand, is a structure capable of (virtually) encapsulating any number of knowledge atoms as well as other knowledge containers, thus providing a single point of access to multiple knowledge sources. The underlying concept of a knowledge container is similar to that of knowledge atoms (i.e. it encapsulates and makes available contextual information). However, the key point is that knowledge containers can “organize” knowledge, by making it possible to enforce and reify structural and behavioral relations between knowledge atoms and between other knowledge containers, in order to access such structured knowledge as if it were atomic information. Also, other than organizing knowledge, they can encapsulate algorithms and methods to manipulate knowledge, e.g., for analyzing, aggregating, pruning or transforming it. As a simple example, in the case study, a knowledge container could provide the “average profile” of visitors in one of the exhibition rooms, by encapsulating the, possibly large number of, atomic profiles that are required to compose a comprehensive profile. As another example, one can think of a knowledge container that, by analyzing the past and present information related to users, is able to predict and provide to services an estimation on the likely future position of a user.

More generically, by properly relating knowledge atoms and knowledge containers according to specific needs, it is possible to enable services to access contextual information according to different application-specific views and/or at different levels of granularity.

The overall architecture of the implemented knowledge networks toolkit is shown in Figure 3.

At the lowest level, the toolkit considers the presence of a number of components, implementing the concept of knowledge atoms (KA for short in the figure) for specific data sources. The knowledge atom classes that we have implemented so far include: atoms for connecting to GPS devices, to CrossBow Micaz sensors, for accessing system properties in computational devices, for accessing Web information, and generic knowledge atoms for hosting static (pre-loaded) and historical information.

For some kinds of data sources, either too resource constrained or too dynamic and volatile (e.g., RFID tags), it is unreasonable to allocate a dedicated knowledge atom to each of them. In these cases, a special kind of knowledge atom acting as a “Atom Repository” can be instantiated to provide, via a single component, access to a multitude of knowledge atoms. For instance, with regard to the case study, if the user profiles are stored in RFID tags and captured by one RFID reader (rather than being associated to a knowledge atom on the users’ PDA), it is possible to think of accessing individual profiles via a single atom repository associated to the RFID reader rather than allocating a knowledge atom for each of the captured profiles/tags.
At the middle level, we can find a number of knowledge container components that are used to organize, analyze, and manipulate the data provided by knowledge atoms, so as to actually reify the concept of knowledge networks. The KNs toolkit do not prescribe what knowledge containers should be instantiated at this level, nor does it limit the number and type of components.

Depending on the specific needs of specific applications, new knowledge containers can be defined and allocated, also at run-time, to provide specific knowledge management functionalities and/or specific aggregation functions and/or specific knowledge views, also possibly building over existing knowledge containers. For instance, with reference to the case study, a knowledge container may be instantiated to collect all RFID knowledge atoms related to user profiles and then produce an average profile.

All knowledge containers provide a specific interface, centered around a simple "getValue" operation, giving access to the contained information by application-level components and by other containers. However, this does not prevent them from making available richer means of accessing and querying knowledge, as we have made in our case study.

As of now, we have already implemented a number of classes for knowledge containers, to test with a variety of knowledge organization algorithms/models. These include, other than simple containers applying simple aggregating functions (e.g. average, maximum, minimum) to an ensemble of knowledge atoms (as it can be the case for the user profiles in the case study), algorithms to facilitate advanced models of semantic knowledge organization, of spatial knowledge aggregation, as well as advanced models for knowledge consistency verification. Some of these algorithms are described in more detail in Section 5.

It is worth noticing that, although for the sake of simplifying the presentation, the organization of the architecture is presented as a layered one (knowledge atoms at the lower level, knowledge containers and supplementary components at the higher level), from the conceptual viewpoint, such layering does not exist. Instead, the various components that compose the toolkit, apart from being of different classes, are peers with each other, and only the dynamic patterns of interactions that are established among them can eventually lead to some sort of structured (e.g. layered) organization. In particular, we can identify two main interaction patterns that will structure the KNs:

1. Links between knowledge atoms and knowledge containers. Knowledge containers will link to a number of atoms creating hierarchical relationship among concepts. Self-organizing mechanisms to create these links will be described in Section 5.1
2. Links between knowledge atoms. These links allow to organize and represent relationships among atoms and possibly instantiate knowledge containers. We used them in Section 5.3 to spatially self-organize distributed information atoms.

Concerning the allocation of the various components over a network, it is to be said that the components of a knowledge networks can execute and interact in a distributed setting,
independently of their actual allocation. Thus, it is possible to have instantiated distributed KNs in which, for example, knowledge atoms are allocated directly on the data source devices (e.g., PDAs or sensors) and different knowledge containers are allocated on different nodes. For instance, in the case study, one knowledge container performing user profiles aggregation can execute directly on the advertisement screens, while a knowledge container devoted to monitor users’ for security reason can be allocated on the central server of the security office.

Considering the case study, it is possible to see how the KNs architecture can be applied in a straightforward way. User profiles (stored in RFID tags), sensor data and Web data are wrapped into knowledge atoms. Knowledge containers organize and aggregate such context data. Application components access to the knowledge atoms that are in the proximity. The knowledge container aggregates and computes an average of the user profiles. Knowledge containers to gather a compact representation of the context and show advertisements on that basis (see Fig. 4).

5. KNOWLEDGE NETWORKS ALGORITHMS

In this section, we highlight the algorithms at the basis of the data aggregation performed by the knowledge network in the case study application. In particular, we will illustrate the mechanism to aggregate user profiles (this is encoded in the User Profile Knowledge Container of Fig. 4) and to aggregate data coming from the sensor network (this is encoded in the Sensor Data Knowledge Container of Fig. 4).

![Figure 5. Knowledge Atom describe the user profile. Knowledge container are used to collect together and create a compact average representation of the profiles of the users near a screen.](image)

5.1 Aggregating User Profiles

Figure 5 illustrate the KN components to aggregate user profiles. The RFID tags containing user profile information are represented by means of knowledge atoms. Each screen (connected with a RFID reader) creates a knowledge container to reference all the knowledge atoms that are in the proximity. The knowledge container aggregates and computes an average of the user profiles, thus offering to the screen a synthetic description of the population that will watch the advertisement. On the basis of such a compact context representation the screen is facilitated in selecting the most suitable advertisement.

This mechanism is rather simple. Each knowledge atom describes the associated user profile by a set of keywords. In our implementation we adopted a simple dictionary containing the terms that can be used to describe the user. The adoption of such a fixed dictionary provides two key advantages. On the one hand, the RFID tag carried on by the user can contain only a reference to a term in the dictionary rather than the whole string, this complies with the stringent memory constraints of RFID tags. On the other hand, the adoption of a fixed dictionary notably simplifies the clustering of profiles described next.

A knowledge container, associated to an advertisement screen, simply computes a statistic of the population nearby by counting the terms describing the user profiles. The result is weighted list of keywords describing the population. As a trivial example, supposing that the population in front of the screen is composed by 5 women and 3 men, and that they all work in the academia, then the knowledge container will describe the current population as: ["woman",5], ["man",3],["academia",8].

In real cases, in which tens or hundreds of keywords are involved, some knowledge containers (KC’s) can be dynamically created to describe different aspects of the population. For example, in the above situation, a “Gender”-KC can be dynamically created to contain only the information ["woman",5], ["man",3], describing the gender distribution of the population. Each of these dynamically created KCs offer a specialized view about the current situation.

The advertisement application will then try to pick an advertisement that suits the given weighted list of keywords.

5.2 Aggregating Sensor Data

We have also explored the possibility of extracting high-level and compact knowledge about the structure of an environment as sensed by a sensor network [Bic07]. The basic idea is to have knowledge atoms executing a distributed gossip-based algorithm. They periodically exchange data with neighboring atoms. A logical link between two neighbors is re-enforced if the environmental characteristics are similar and weakened otherwise. When the status of the links reach a sufficient degree of stability, the network of knowledge atoms is able to self-organize itself into a set of distinct partitions each associated to a knowledge container and corresponding to a region of the environment characterized by a specific sensing pattern, e.g. a room with a specific temperature or light levels (see Figure 6).

In this way, the possibly large amount of sensorial data generated by the network can be not necessarily perceived as a multiplicity of unrelated information. Instead, the algorithm makes it possible to perceive the sensor network as if it were made up of a more limited number of “macro sensors” (i.e., the knowledge containers), each associated to a well-characterized region of the physical environment. To some extent, the algorithm provides for the automatic construction in the KNs of specific spatial views by aggregating data to represent the overall “situation” of a region of the environment. This can facilitate usage by services.

Moreover, each knowledge container could provide application-specific aggregation functions on its region. In this way the KN is not only a way to perceive the environment in a simplified fashion, but can also act as a computing layer able to produce
derived knowledge. In the case study, one could think of realizing this kind of aggregation algorithm on the data collected by the sensor network to provide the advertisement application with a compact representation of the environment surrounding the screen. Finally, since the KNs can be realized both in a fully-distributed fashion (in sensor networks) as well as a centralized solution (in the Web-based repository), services can dynamically decide how to access information.

From a complementary perspective, we are also studying how location information about users (e.g. as provided by RFID localization [Hah04]) can be organized so as to infer spatial relationships among users and services/resources. For example, we can easily infer how close a user is to a given resource (e.g. a restaurant). Such kind of information may be useful in several applications. For instance, if a person enters a supermarket one may want to observe its movements in order to provide specific services, such as person-centric advertisement, seasonal offers, etc. To realize such services, once a person has entered a shopping mall, the persons profile could be registered with a dedicated knowledge container that employs a different, more specific means of organization. Such a container could analyze the specific location of the person within the shopping mall identifying if the person is for instance near an advertisement screen or if the person is entering a specific shop. Other forms of organization could rely on evaluating the surroundings of the person’s location, and on creating a virtual orb around the persons position and querying other sources if they are within this orb or not. Theoretically, due to the fact that KNs can be strongly distributed among several computational resources, there are unlimited possibilities about how such knowledge sources can be organized and related with each other.

5.3 Integrating Situations

Information about user profiles and sensor data can be integrated into a more comprehensive view on the current situation. For example, relying on aggregated sensor data the knowledge network might infer that the an area of the environment is rather noisy. Moreover, relying on user profiles, it might infer that the population in that area is interested in movie trailers advertisements. Combining the two views on the present situation, the knowledge network may decide to subtitle the movie trailers to compensate for the noise in the area.

As another example, sensor data might indicate that the area is very warm, while profile information and time of day indicate that users might be interested in having lunch. Such an integrated view on the current situation may induce the advertisement service to offer recommendation on soft-drinks and ice creams.

We again emphasize that: (i) application services, such as the exemplary advertisement provider, would be notably facilitated by such an integrated representation of the system situation, (ii) similar arguments can be applied to a wide range of application scenarios.

6. EXPERIMENTAL RESULTS

We conducted a set of experiments to evaluate the KNs framework. Here we present experiments related to some of the main aspects of the KN approach, i.e., (i) results illustrating performances of the implemented KNs toolkit; (ii) results analyzing one of the self-organizing algorithms we have in KNs.

6.1 Infrastructure

To assess our implementation, we performed some experiments to show the performances of the presented toolkit. In particular, we tested the response time of the KNs to queries.

We put in place several atom repositories either local and connected through the Internet. The remote atom repositories were installed in several European research facilities (member of the CASCADAS project). We ran queries from different networks and monitored minimum and maximum response times of the networked KNs.

Figure 7 (top) reports a similar experiment in a distributed setting. The experiment consisted in querying multiple distributed KNs with the CASCADAS project. We ran queries from different networks and monitored minimum and maximum response times of the networked KNs.

Figure 6: (top) 4 recognizable regions of an environment as identified by a specific property, and a network of Knowledge Atoms immersed on it. (bottom) Knowledge Atoms spatially self-organize into four virtual overlays, reflecting the environment. Each region is then represented by a single Knowledge Container acting as a sort of macro sensor for that region.
KNs connected over the Internet and measuring minimum and maximum response times. Also in this case, the reported results refer to the worst case possible, where all the knowledge atoms are queried.

In contrast with the previous experiment, these graphs show that the response time scales almost linearly with the size of the KNs and can have large fluctuations because of components workload and network delays.

Figure 7: (top) Min. and Max. Response Times among several local KNs. (bottom) Min. and Max. Response Times among several distributed KNs scattered though Europe: UU – University of Ulster (UK), BUTE - Budapest University of Technology and Economics (Hungary), FOKUS - Fraunhofer Institute (Germany), UNIMORE – University of Modena and Reggio Emilia (Italy).

6.2 Spatial Self Organization

We performed other experiments to quantitatively evaluate the algorithm to cluster sensor data. In particular, we conducted some experiments both in a simulation environment using the KNs toolkit to verify the convergence and accuracy level of our approach in large-scale scenarios and in a real sensor network test bed. This example clearly show how KNs could be exploited not only to manage relations between several knowledge sources but also to handle inherent dynamism in physical systems. In particular, we used the algorithm to compute the average value among the data collected by a sensor network in a region. Knowledge atoms installed on sensors compute local averages and propagates them across the network by gossiping. This process leads iteratively to the computation of the average in the region that is made available by a knowledge container [Bic07].

In Figure 8 we summarized the performances of the proposed spatial self-organizing algorithm. In particular we measured the aggregated errors produced by the approach while partitioning a networks with different sizes and densities. Figure 8 (top) shows that the average error tends to decrease increasing the network density because of the number of interactions between knowledge atoms increases as well. More interactions produce an higher convergence speed. Obviously, also the number of messages being exchanged, and then the communication costs, increases. Figure 8 (bottom) shows the scalability of the algorithm varying the network size from 10 to $10^3$ nodes. As expected the average error increases with the number of participating nodes. This happens because the number of iterations needed by the algorithm to converge increases with the number of nodes. However, an almost linear (in the logarithmic domain) trend guarantees good scalability and acceptable errors even if applied on networks with thousands of nodes. Finally, it is possible to see that a decrease in the sensor sampling rate (that is equivalent to an increase in the environment dynamism) slowly degrades performances without altering the overall behavior. These experiments show the effectiveness of the self-organizing algorithm we have implemented to spatially correlate several distributed knowledge atoms.

Figure 8: (top) Average error for different network densities. (bottom) Average error for different network sizes.

7. RELATED WORK

Early works in the area of context-awareness, as from Schmidt et al. [Sch99] and Dey et al. [Dey00], concentrates on the issue of acquiring context data from sensors and of the processing such data to make it available to processes/services in the form of abstract components. Such approaches have two main shortcomings: one the one hand they do not provide a uniform model and a common semantics to describe the data. This forces developers to build new query languages and new components in dependence of the kind of information at hand. On the other hand they neither address the problem of extracting high-level situations from raw sensor readings nor the problems that of providing application-specific views. The KNs framework
overcomes these limitations via a single query interface and by embedding self-organizing mechanisms to analyze data effectively and extract higher-level knowledge according to any needed view.

Several research works get inspiration from tuple space data models to represent contextual information in the form of tuples and access them via associative (i.e., pattern-matching based) query operations. The basic idea is that associative access, within a uniform interface, can facilitate semantic-based access to a variety of knowledge sources, possibly enforcing application-specific views. Egospaces [Jul06] adopts this perspective to access contextual information according to user-specific views. However, it does not commit to a specific pre-defined structure for context tuples, which can make it difficult for services to uniformly deal with different aspects of the context represented in different formats. The Context Fabric model [Hon02] relies on well-structured context tuples each describing a single piece of context data in terms of entities (people, place, thing) and attributes (e.g., the name). However, it does not propose solutions for enforcing application-specific views. Recent proposals focusing on sensor networks suggest exploiting a tuple-based approach to flexibly access information on sensor networks [New04, Mot06]. Although not focusing on specific tuple structures, such proposals are of interest in that they consider the possibility of providing application-specific views in accessing sensorial data. The idea is to have services dynamically inject code into the sensors for aggregating/elaborating data within the sensor network, and eventually enabling services to directly access aggregated data according to their own specific needs. All these systems have commonalities with the ideas presented in our KNs framework, which however strives for more generality. So, while existing approaches focuses on sensor networks or user-centric contextual information, KNs appear a more general-purpose model, suited for diverse devices and scenarios.

A proposal sharing a number of goals with the KNs proposal is the “knowledge plane” approach [Cia03]. There, the idea is to couple the service layer with a (heavyweight) control plane where both tools for the analysis of situational knowledge and sophisticated logic of application control and management are embedded. On the contrary, KNs have the goal of being lightweight, embedding the logic related to information management and only relatively simple logics for their internal unsupervised maintenance. That is, for KNs to be effectively usable, they must rely on simple self-organization algorithms for knowledge management and on simple self-management mechanisms to adapt their internal behavior accordingly.

Apart from collecting context data, another important issue relates to situation-awareness: the idea of having algorithms and tools to digest large amounts of data in order transform it into more meaningful knowledge, is at the very centre of data mining research. However, most data mining research focuses on centralized architectures that do not fit our visions, and a few have been focused on distributed network architectures.

Recently, however, data mining approaches have been proposed to analyze large amounts of contextual data and infer hidden linkages, correlations, rules, and constraints in such data [Fay96, Bar03]. In general, all the mechanisms proposed in this field (and in the wider data mining area) can be potentially employed within the knowledge network layer to extract knowledge from raw data collected by sensing devices.

Sensor networks in prime offer unique challenges and opportunities to distributed data mining, given the potentially large amount of sensors in a network and the consequently large amount of data to be analyzed. Some approaches [Bon05, McS05] focus on mining sensed data for prediction purpose. [Bon05] proposes a framework for data mining upon sensor network for supervised learning (prediction, classification, etc.) based on distributed sensor clustering and aggregation. Similarly, [McS05] proposes a framework for prediction based on the flow of local predictors through the network. Other approaches [Gar04, Kui05] focus on the general problem of identification of pattern by using distributed AI algorithm. Whatever the case, all these approaches: (i) uphold the need for data mining to analyze the vast amount of data in pervasive computing application, (ii) show that decentralized approaches are effective and operable in distributed network with several nodes. Clearly, such approaches are relevant also to the KNs domain and could be well integrated in the structures of our framework.

Another trend of research in applying distributed data mining to pervasive computing scenarios consists in analyzing data coming from wearable sensors to infer and predict user’s behavior and social interactions. The work presented in [GaiP06] applies data mining techniques to automatically identify social structures among a group of people, by making use of radio-based proximity sensors. A similar proposal can be found in [Pen05], where the use of microphones and IR badges is proposed to measure who is talking with whom, and derive social networks and other context information by mining such information. All these algorithms could be implemented within the KNs and complement the existing self-organizing mechanisms for data analysis. For example, they would be extremely useful in the proposed case study application, where advertisements could be delivered also on the basis of the inferred social relationships of the user.

8. CONCLUSIONS

The increasing deployment of pervasive computing technologies such as sensors, tags, location systems, and user profilers will soon form the basis of a globally shared and distributed infrastructure, producing huge amounts of contextual information for the use of general-purpose pervasive services. However, this also introduces the need for novel models and tools to properly prune, organize and aggregate this growing amount of distributed information, so as to facilitate the successful exploitation thereof by pervasive services.

In this context, self-organizing KNs promise to become a very useful tool. By taking care of managing an increasing amount of contextual information in a fully self-organizing and self-managing way, KNs induce a separation of concerns that facilitates the development of pervasive services and that, at the same time, enables them to reach higher degrees of contextual and situational-awareness.

9. REFERENCES


