Self-organizing Knowledge Networks for Pervasive Situation-Aware Services

Matthias Baumgarten[1], Nicola Bicocchi[2], Rico Kusber[3],
Maurice Mulvenna[1], Franco Zambonelli[2]

1) School of Computing and Mathematics, University of Ulster, Belfast, Ireland
2) DISMI - Università di Modena e Reggio Emilia, Reggio Emilia, Italy
3) ComTec, University of Kassel, Kassel, Germany

Abstract

Adapting to current context of usage is of fundamental importance for pervasive computing services. As the technology for acquiring contextual information is increasingly available and as it is producing growing amounts of data, there is the need for tools to organize such data before delivering it to services. This produces a sort of "knowledge networks" representing comprehensive knowledge related to a "situation" in an expressive yet manageable way. In this paper, also with the help of a simple case study, we motivate the need for situation-awareness and for knowledge networks, introduce a reference architecture for knowledge networks, and exemplify a prototype implementation thereof. Finally, current and future research directions are discussed.

Keywords: *pervasive computing services, knowledge networks, self-organization, sensor networks.*

1. Introduction

In pervasive computing scenarios, the capability of services to autonomously adapt to the context from which they are requested and in which they execute is necessary to achieve effective autonomic behavior to efficiently satisfy increasingly demanding users [11]. This requires both the technologies to capture contextual data and the capability of services to exploit such data at the best.

Much of the technology to acquire contextual information is already becoming available, and it will soon become pervasive due to the increasing deployment of sensors, location systems, users and organization profiles [9, 12, 15]. What is still in its infancy and as such needs to be properly resolved is the investigation of the principles and the algorithms with which this growing amount of distributed information can be properly organized, aggregated, and made more meaningful, so as to facilitate the successful exploitation thereof by pervasive services [1].

In other words, we think there must be an evolution from a model of simple context-awareness, in which services access isolated pieces of contextual data and are directly in charge of digesting them, towards a model of "situation-awareness", in which services access properly structured and organized information, which reflect comprehensive knowledge that is related to a "situation" of interest and which can be exploited in a standardized fashion [8, 14]. This is where the idea of knowledge networks (KNs from now on) arises: providing models and tools to represent, analyze and self-organize contextual information into sorts of structured collections of related knowledge items. Thus KNs may support applications and services in reaching, with reduced efforts, a comprehensive understanding of "situations" around and, consequently, a higher-degree of adaptability and autonomicity.

In this paper, we firstly motivate (in Section 2) the above outlined need for pervasive services in order to leverage context-awareness to situation-awareness and introduce the basic concepts of KNs. A case study in the area of adaptable pervasive advertisement services is also introduced to clarify the concept expressed, and will be adopted as a running example throughout the paper. Section 3 introduces a reference architecture for KNs, sketches the current prototype implementation of a KN tool that we have implemented in the context of the European Project "CASCADAS", and presents the knowledge aggregation algorithm currently integrated within it. Future research directions are discussed in Section 4. Section 5 concludes this paper.

2. From Context-awareness to Situation-awareness

According to most assessed user-centric definitions [3, 8] "*context is any information that can be used to characterize the situation of an entity*" (a service or a software component) and that can be considered as relevant to adapt/improve the interaction between such entities and its users (exploit context-awareness to maximize its functional benefits). Recently, more software-centric viewpoints on context and context-awareness have emerged, which focus on context-awareness as a mean for services to improve quality and reliability via autonomicity and adaptability (exploit contextual information to self-monitor, self-configure, self-reconfigure, etc.) [7, 11, 13].

While we fully commit to the above, we also perceive that technological advances are creating a notable gap between "*context is any information*" and "*that can be used to characterize the situation of an entity*". That is, acquiring contextual information does by no means imply the capability of understanding situations, especially in the presence of an overwhelming amount of information and a lack of relations between them.

The imminent mass diffusion of pervasive technologies such as sensor networks [6], RFID tags [15], localization tools [10], will soon make pervasively available an incredible amount of real-time information about the physical world, its processes, and its objects. Further, the dramatic success of participatory Web tools, aka Web 2.0, is feeding the Web with information of any kind about any topic. In particular, tools such as Google Earth get continuously enriched by geo-located and localized contextual information coming from very diverse social communities and related to a variety of facts and events situated in the world [4].

Overall, the above trends contribute to accumulate information that can be potentially used by software to achieve context-awareness. However, for software services to fruitfully exploit the above described information calls for *(a)* notable communication efforts to retrieve (possibly from remote) all needed information and *(b)* for notable computation efforts to analyze all available information with the goal of making them more meaningful and ultimately machine understandable.

To ground the discussion, let us consider the scenario of a modern exhibition center, like a big museum or a stadium. In contexts of this kind, it is realistic to assume the presence of a pervasive infrastructure of embedded devices such as sensors of various types, lots of WiFi connections, RFID tags and other location systems. In fact, exhibition centers may afford the costs of deploying such infrastructures if this enables to provide good services to visitors (and accordingly attract a higher number of persons) and get 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, consider the presence in the exhibition center 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 independent of the situation (i.e., independent of who is actually close to that screen). A "smart" service devoted to decide what information to display could exploit the availability of contextual information to adaptively decide what information to show on the basis of the people around and of their activities and interests. This would increase the value of the displayed advertisement both for users and for advertising companies.

The problem is that in a large exhibition center with dozens of thousands of people, and with a large number of devices producing 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 thousands of possibly incomplete user profiles that may have to be integrated with statistical information available somewhere or with some information extracted from other sources, a multitude of sensorial data detailing what users are currently doing, 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., bound the amount of commercials of a given company to show.

The KNs approach considers that, 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 internally access and manage large amounts of data. To some extent, KNs act as a sort of middle-layer that eliminates the need for services to directly manage contextual information. With reference to the application example, KNs could provide aggregated user profiles with sensorial information in order to provide situation specific knowledge to a decision making system. For instance "there are 70% of women who are interested in modern art" or "80% of visitors are approaching the cafeteria", represent valuable information to decide what advertisement to show.

Unlike in the renowned "knowledge plane" approach [5], we do not consider KNs as a heavyweight control plane for services, where to embed logics of application control and management. KNs should be light-weight and should only embed logics of information management, and rather simple logics for their internal unsupervised maintenance. That is, for KNs to be effectively usable, they must rely on self-organization algorithms for knowledge management and for self-management mechanisms to adapt their internal behavior accordingly.

A possible criticism of the KN approach is that it does not eradicate the problem of analyzing large amounts of information, but simply passes it to a different component that either exists at application or at KN level. Although this may be true to some extent, one should consider that: *(a)* 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 *(b)* in a distributed setting, KNs can take care of knowledge management duties that would have been otherwise replicated inside each service.

3. Knowledge Networks

Once one has absorbed the general idea of KNs, the question arises of KNs could actually look like, i.e., what kind of reference architecture could be used or how KNs could eventually translate into actual middle-level tools for supporting situation awareness, and of what algorithms for data management one could think in order to make KNs "intelligent".

3.1 Reference Architecture

Let's start with the assumption that there are a lot of various kinds of "sensors" (whether physical sensors, software sensors, or social Web 2.0 sensors) generating large amounts of (mostly) independent atomic units of contextual information (see Figure 1). We call these "knowledge atoms". The KN approach considers exploiting self-organization approaches to aggregate/correlate/prune such knowledge atoms to facilitate their exploitation by services.

In general, one cannot think at constructing a single KN capable of mirroring the universe and provide different views to accommodate for individual situations. On the one hand, when considering that even relatively small network scenarios can generate enormous amounts of knowledge, it is necessary that KNs can provide different levels of abstraction 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 co-exist within a globally accessible 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 this has to be perceived and elaborated by services may depend on the specific type of service one has to enforce. In other words, the context may be in need to be perceived by services as a variety of situations, and one should thus consider that several "dimensions", according to which knowledge atoms can be networked with each other, exist. In the application example, 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 illusionary to identify or reflect all possible dimensions in which knowledge may be organized in. However it is feasible to reflect a given subset thereof to be exploited by various applications. Firstly, we have a purely semantic dimension, in which knowledge atoms that are related to a situation network/relate to each other according to the relations institutionalized in (or inferred from) some shared ontology. This can be the case for knowledge facilitating and supporting spontaneous interoperability in pervasive computing and service-oriented computing [11], or of knowledge related to inferring users' activities from a variety of heterogeneous sensorial information [15]. Secondly, we have a spatial dimension, in which knowledge atoms that are related to a local fact network relate to knowledge atoms at different locations (or distribute/replicate themselves in different locations). This can be of use to express some 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). Thirdly, we may have a temporal dimension, in which knowledge atoms express facts occurred (or about to occur) at different times. This can be the case of elaborating knowledge for predictive purposes: starting from the situation at current time, analyze and extract new knowledge in the form of a KN expressing the most likely future situation.

These considerations summarize into a sort of conceptual reference architecture for KNs (see Figure 1). The figure does not show that KNs can be organized around other application-specific dimensions in which knowledge atoms may be organized in variously shaped KNs serving different purposes, and possibly overlapping 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).

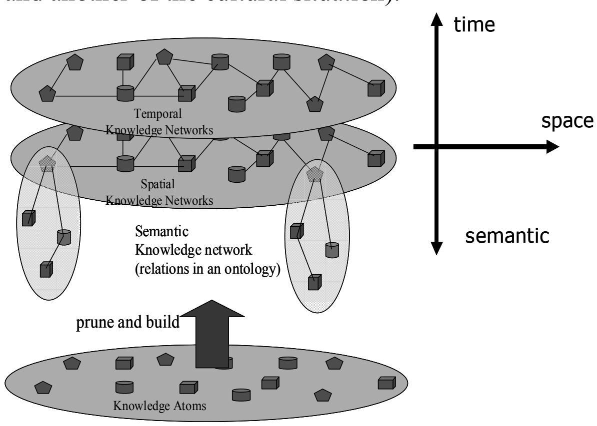


Figure 1: A Conceptual Architecture for KNs

3.2 Implementation

The current prototype implementation of a KN is component-based and tries to turn the abstract reference architecture into a practical tool. The implementation relies on two basic components: knowledge atoms and knowledge containers [1].

A knowledge atom represents the atomic unit of knowledge and encapsulates two concepts: firstly it provides a uniform abstraction to access contextual information independently of its type, size or context; secondly it has relevant descriptions of the knowledge object attached, providing context, system and usage based information, relevant for the creation, and maintenance of the knowledge object itself.

A knowledge container is a structure capable of (virtually) encapsulating knowledge atoms at different levels of granularity. The underlying concept of a knowledge container is similar to that of knowledge atoms (i.e., it encapsulates some sort of contextual information and can relate with other containers). The key point is that knowledge containers make it possible to access, as if it were an atomic information, aggregated contextual knowledge (e.g., the average profile of visitors in an exhibition calculated over a large number of atomic profiles) or to new information generated by analyzing and relating existing knowledge (e.g., the predicted future position of a visitor based on its past movements). By properly relating knowledge atoms with each other and by properly introducing knowledge containers to represent aggregated/derived information, services can access contextual information according to various views, and different services can access knowledge at different levels of granularity. The discussion of the algorithms implemented so far to relate individual knowledge atoms and to produce aggregated information in knowledge containers will be sketched in the next subsection.

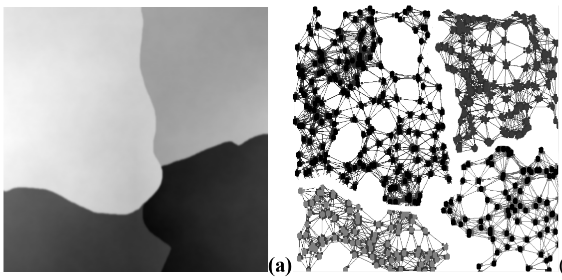
Based on the above two components, the current prototype implementation includes two integrated modules. Firstly, it includes a centralized Web-based KN repository where any needed number of knowledge atoms can be instantiated and connected to e.g. a physical software sensor. Such knowledge atoms make available, on the repository, the contextual information generated by the associated sensor, and knowledge containers can be created in order to access various aggregated/derived information. Secondly, it includes a fully distributed sensor network (CrossBow-Micaz) implementation of knowledge atoms and knowledge containers, to enable relating and aggregating information sensed by different sensors, and enabling services on mobile devices to directly access knowledge atoms/containers allocated to specific sensors. In addition, knowledge atoms on sensors can also be directly connected with a counterpart in the Web-based KN repository and as such make available the same information from two different access points.

A simple querying mechanism based on Linda-like pattern matching is provided to access/query KNs [4]. The definition of a complete API is in progress.

3.3 Algorithms

From an algorithmic point of view, i.e., the self-organizing mechanisms that enable the correlation and aggregation of data, we have so far focused on self-aggregation along the spatial dimension.

Just to make an example of the kind of algorithms we are studying, we have explored the possibility of extracting high-level knowledge about the structure of an environment as sensed by a sensor network [2]. The basic idea is to have components acting as knowledge atoms in the nodes of a sensor network executing distributed gossip-based algorithms and periodically exchange data with knowledge atoms on neighbor nodes. A logical link between two neighboring knowledge atoms is re-enforced if the environmental characteristics sensed by the two nodes are similar, weakened otherwise. Eventually, the network of knowledge atoms self-organizes into a set of distinct partitions each 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 2). Then, a set of knowledge containers can be instantiated to average data on a per region basis, with the result that the possibly large amount of sensorial data generated by the sensor network is no longer perceived as a multiplicity of unrelated information. Rather, the algorithm makes it possible to perceive the sensor network as if it were made up of a more limited number of “macro sensors”, each associated to a well-characterized region of the physical environment. To some extent, the algorithm provides for the automatic construction of a KN by aggregating data to represent the overall “situation” of a region of the environment for facilitated usage by services.



(a) (b)

Figure 2: (a) 4 recognizable regions of an environment as identified by a specific property. (b) Self-organized network of sensors reflecting the environmental characteristics.

In the case study, one could think of running a similar aggregation algorithm that firstly partitions the cloud of all user profiles into clusters of users that are characterized by similar interests, and then averages data over each cluster. This can enable services to get a synthetic clue of what the overall preferences of users are and to reach a quick decision on what advertisement to show on a screen. Also, due to the fact that KNs may be realized as a fully distributed deployment (in sensor networks) and as a centralized one (in the Web-based repository), services can flexibly decide how to access information.

4. Research Directions

Applications and services need to take advantage from knowledge organization along the semantic and temporal dimensions, or along additional application-specific dimensions. This calls for more advanced algorithms that, ideally, can operate on multiple dimensions thus providing situational knowledge derived form multiple contextual levels.

From the semantic viewpoint, it is necessary to integrate self-organization algorithms that enable to discover and enact relations among initially uncorrelated knowledge atoms. From the emerging network of such relations, it will then be possible to acquire new knowledge about facts and situations, which could be made available via knowledge containers. With reference to the application example, one can consider analyzing the activities of a visitor as they can be detected by various sensors and discover, by properly relating them, more information than available in his/her personal profile. For instance, by relating the fact that a user walks very slowly, has been to the pharmacy, and has been to the toilet several times, one can detect he is an elder person. Similarly, by analyzing the patterns of social relations of a visitor (e.g., by regarding the Bluetooth connections of its PDA) one can understand whether this person is accompanied by children or by friends. Such newly generated knowledge can then be used to tune the advertisements displayed in his presence accordingly. Clearly, to make this possible, KNs should rely on reference ontologies to relate knowledge atoms with each other and should be supported by proper semantically-oriented self-organization algorithms which autonomously generate new knowledge. Two key aspects are focus of our current research work.

From the temporal viewpoint, the basic idea is that the analysis (both spatial and semantic) of contextual information about the past can be used to infer information about the future. For instance, the analysis of the fact that a visitor at the exhibition has already visited specific sections of an exhibition can be used not only to increase the accuracy of its profile but also to reasonably predict what sections/events in the exhibition he is most likely to visit next. Accordingly, one can tune the information displayed on the screens close to her/him. Such predictive knowledge mechanisms – to be grounded on a large body of existing research work on predictive technologies – will soon be included in the scope of this research.

Specific solutions that have to be integrated into the KN toolset itself form another research direction. From the discussion so far, it is clear that the amount of information relevant for situational knowledge and reasoning thereof are characterized to be rather large, unstructured, unrelated and possible redundant. This calls for advanced knowledge lifecycle mechanisms and for mechanisms to ensure the consistency of sensed and newly derived knowledge. With respect to the former, the issue is to evaluate how long the information should be held and how much of its history should be stored for future use and for predictive features. Simplified, it is the possibility of a system to “forget” things which is proven to be often more difficult than making a system learn. With respect to the latter, the issue is to evaluate discrepancies for inconsistent and incomplete knowledge (e.g., different tools providing different information about the location of a visitor), and how to measure reliability and accuracy of information.

From the viewpoint of services, a key problem is to understand how they can access information in KNs and to decide when access should be denied. Furthermore, based on the fact that specific services may require the construction of specific KNs and the access to specific views on knowledge, one must provide the possibility for services to somehow access the inside of knowledge networks for re-configuration and dynamic instantiation of specific algorithms within. Finally, considering the distributed aspect of KNs, their components and their tools, the issues of how to distribute, replicate and manage knowledge atoms and knowledge containers arise. We are planning to use P2P approaches to manage distribution, the issue is still to be investigated.

5. Conclusions

Self-organizing KNs promise to become a very useful tool for future generations of pervasive services. 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 situation-awareness.

Despite preliminary encouraging results, there is still a lot of R&D to be done to make KNs an efficient and usable tool such as: study and experimentation of further knowledge aggregation algorithms, integration in the prototype of different mechanisms for knowledge organization and distribution and for knowledge consistency, and flexible APIs to allow services to access, analyze, and customize, KNs.

Acknowledgments: Work supported by the project CASCADAS (IST-027807), FET Program of the European Commission.

References

1. M. Baumgarten N. Bicocchi, M. Mulvenna, F. Zambonelli , “Self-organizing Knowledge Networks for Smart World Infrastructures”, International Conference on Self-organization in Multiagent Systems, Erfurt (D), 2006.
2. N. Bicocchi, M. Mamei, F. Zambonelli, “Self-organizing Spatial Regions for Sensor Network Infrastructures”, 2nd IEEE Symposium on Pervasive and Ad-Hoc Communications, Niagara Falls (ON), May 2007.
3. T. Buchholz, A. Kupper, S. Schiffers, “Quality of Context Information: What is it is and Why We Need It”, 10th HP-OVUA Workshop, Geneva (CH), July 2003.
4. G. Castelli, A. Rosi, M. Mamei, F. Zambonelli, “A Simple Model and Infrastructure for Context-Aware Browsing of the World”, 6th IEEE Conference on Pervasive Computing and Communication, New York (NY), March 2007.
5. D. Clark, C. Partridge, C. Ramming, J. Wroclawski, “A Knowledge Plane for the Internet”, 2003 ACM SIGCOMM Conference, Karlsruhe (D), 2003.
6. C.-Y. Chong, S. Kumar, “Sensor Networks: Evolution, Opportunities, Challenges”, Proc. of the IEEE, 91(8), 2003.
7. J. Crowcroft, S. Hand, R. Mortier, T. Roscoe, A. Warfield, “Plutarch: An Argument for Network Pluralism”, ACM SIGCOMM 2003 Workshops, August 2003.
8. A. K. Dey, G. D. Abowd, “Towards a Better Understanding of Context and Context Awareness”, Workshop on the What, Who, Where, When and How of Context-Awareness, 2000 ACM Conference on Human Factors in Computer Systems, April 2000.
9. D. Estrin, D. Culler, K. Pister, G. Sukjatme, “Connecting the Physical World with Pervasive Networks”, IEEE Pervasive Computing, 1(1), 2002.
10. J. Hightower, G. Borriello, “Location Systems for Ubiquitous Computing”, IEEE Computer, 34(8), 2001.
11. A. Manzalini, F. Zambonelli, “Towards Autonomic and Situation-Aware Communication Services: the CASCADAS Vision”, 1st IEEE Workshop on Distributed Intelligent Systems, Prague (CZ), June 2006.
12. M. Philipose, K. Fishkin, M. Perkowitz, D. Patterson, D. Fox, H. Kautz, “Inferring Activities from Interactions with Objects”, IEEE Pervasive Computing, 3(4), 2004.
13. J. Serrat, J. Serrano, J. Justo, R. Marín, A. Galis, K. Yang, D. Raz, Efstathios D. Sykas, “An Approach to Context Aware Services”, NOMS2004, Seoul (KR), April 2004.
14. L. Tummolini, C. Castelfranchi, A. Ricci, A. Omicini, “Exhibitionists and Voyeurs do it Better: Coordination with Tacit Messages", Environments for MultiAgent Systems, LNAI 3374, Springer-Verlag, Jan. 2005.
15. R. Want, “An Introduction to RFID Technology”, IEEE Pervasive Computing, 5(1):25-33, 2006.