ISBN: 978-972-8924-30-0 © 2007 IADIS

ENHANCED SENSOR NETWORK: A SPECIALIZED INFRASTRUCTURE FOR CONTEXT-AWARE APPLICATIONS

Hooman Tahayori[1], Elena Pagani[2], Giovanni Degli Antoni[1], Sadegh Astaneh[2]
[1]*Universita degli Studi di Milano, Dipartimento di Scienze dell'Informazione*
[2]*Universita degli Studi di Milano,Dipartimento di Informatica e Comunicazione*
[1,2] *Via Comelico 39/41, 20135, Milano, MI, Italy*

ABSTRACT

Respecting the mobile world, it is about the time to demand for systems to fully take advantage of their environment. In this way, Enhanced Sensor Network is another step toward developing realistic context-aware applications, which is based on the basic infrastructures provided by wireless sensor networks (WSN) and context-aware application development paradigms. In this paper we introduce a framework for integrating WSNs with context-aware application requirements to enhance wireless sensor network as an infrastructure which can provide necessary contextual information for context-aware applications.

KEYWORDS: Context-Awareness, Wireless Sensor Network (WSN), Enhanced Sensor network (ESN).

1. INTRODUCTION

Despite recent advancements in both hardware capabilities and software engineering disciplines, many applications still fail in fulfilling requirements of simplicity, portability, adoptability and productivity. A promising approach in this direction is represented by *context-aware applications*. These applications are able to adapt their behavior basing on implicit inputs from the surrounding environment; they can also affect their environment implicitly. In this work, we discuss how network of sensors could supply applications with context information, and how their architecture should be modified to this purpose.

Context-aware applications can be considered as formed by two parts. The *inner part* is the application itself that can adapt its functionalities depending on the information provided by the *outer part*. The outer part is responsible for information gathering from the environment, processing and presentation to the inner part. In this paper, we deal with the outer part. However, further discussions on the inner part can be found e.g. in [2,9].

WSN, as said to be amongst the twenty one most important technologies in the 21st century [4], potentially can provide a basic infrastructure for gathering contextual data. Anyhow, those data are mostly in a raw form, unsuitable for the needs of the applications, and they only describe directly sensible physical phenomenon. Further context can be extracted from those data. In this paper, we propose the *Enhanced Sensor Network* (ESN) framework, as an integration to WSN in order to gather and process context data according to the requirements of context-aware applications [6].

2. CONTEXT-AWARENESS

In literature, most of the definitions for context are by example which results in non general and comprehensive definitions. The one that is general enough and more applicable is in [5] : "c*ontext is any information that can be used to characterize the situation of an entity. An entity is person, place,* [virtual or real world] *object* [or even piece of software] *that is considered relevant to the interaction between a user and an application, including an application and the user themselves"*. By this definition, contextual information can range from identity, spatial information, temporal information to physiological measurements , social situations , history of interaction, and even happiness, dangerous and critical situations.

Given the above discussions, context awareness is defined as the ability of getting use of contextual information and consequently, context-aware applications are those that can regulate their behavior according to the context changes. This regulation would be divided into two categories, system and user side. On the system side, the application has to maintain its functionality. On the user side, which is mainly related to the adaptability, flexibility and productivity, the system reaction toward the context change would be divided into three sub-categories: *Presenting related information or offering related services to the user according to the context change*, *automatic execution of tasks* and *automatically attaching current context to some information for future uses.* [9]

Some context-aware applications yet exist. The Active Badge [15] is one of the first context-aware systems developed in Olivetti Research Lab. Roughly speaking, the system was to trace the personnel and enable telephone operators to redirect each personnel's telephone call to his/her nearest telephone set, however the system offered some more services like determining who are in the same room, close to each other and what more. The main contextual information used by the system was identity and location. Another useful context-aware application developed at Georgia Institute of Technology, was conference assistance [8]. The system enabled each conference attendee, as he/she get entered a room where a lecture was held, to become aware of the topic, lecturer and related notes. The main contextual information used in this system was identity, location and time. Another outstanding system that was developed at the University of Lancaster was GUIDE [7], which was aimed to give help to tourists visiting the city of Lancaster. The system, with respect to the user's preferences, time of the day, the user's location, showed him/her related information about the part of the city. However, there exist more context-aware applications that almost all of them are developed in research labs.

One of the main burdens in the way of developing context aware application roots at the fact that methods for gathering accurate and reliable contextual information and delivering them to the applications are not well established. A considerable point in existing context-aware applications is that, all of them are based primarily on the existence of mechanisms for gathering contextual information that has limited the applications to the just sensible context that might not be that appropriate and reliable for the application.

If for developing each context-aware application developers must primarily devise and establish related network of sensors as a means to gather contextual information, these types of applications will remain in prototypical versions. On the other hand, if the required infrastructures for developing such application are ready, that is the network of required sensors and equipped with special processors, application developers will be more encouraged, primarily because of the access to the *ever-ready* and *reliable* contextual information and secondly, because of the different levels of abstracted contextual information that would be provided by the network.

3. ENHANCED SENSOR NETWORK

Despite being highly in front of changes and the burdens amongst using WSNs in some environments like deep in the oceans that due to huge propagation delay, floating node mobility and limited acoustic link is yet challenging [10], the distributed and continuum form of gathering information from even hostile environment has made it such attractive that the technology has found its way in many fields, from military, security surveillance, habitat monitoring, building monitoring, environmental applications like flood and forest fire detection to human health care and home automation. By the way, WSNs to directly, accurately and reliably support context-aware applications must satisfy the following requirements demanded by context-aware applications. Equipping WSN with such requirements will result in *Enhanced Sensor Network (ESN)* that is expected to provide a well-defined interface for context-aware applications. The proposed framework will result in a hierarchical network from at least two viewpoints. One, as a network as it is; due to its hierarchical design, protocol stack, etc. and the other one as a hierarchical information provider.

A-Concern Separation

Almost all of the current context-aware applications yet developed, for context sensing rely on one of the two following methods for information gathering. Either, manipulating the sensors is directly hardwired in the main application, which always leads application code being mixed with sensor handling related subcodes. [11] and [13] are the examples of this kind but it must be noticed that their code is rarely reusable. Moreover, reusing the sensors and importantly simultaneously use of them by other applications is burdensome. Some other context-aware applications, on the other hand, are relied on special servers that provide them with the sensed data but force the application to deal with any server in a distinct manner rather the same way. [3] is an implemented example of this approach. Here the application developer has to deal with each server following the rules it imposes.

Context-aware applications are supposed to be used mobile so if they are to be dependent on a special hardware, or following different methods to connect servers, in essence they will not be a real context-aware application. A clear separation between concerns, i.e. inter and outer part, will enable deploying clear interfaces for any context-aware application to be connected to any WSN in the region. However some problems like security, usability of the WSN for the application and required level of accuracy must be considered.

Toward the aim of overcoming the inconsistencies with the aforementioned methods in developing context-aware applications and as the first step, abstracting the contextual information to different levels is required. So sensor networks must be equipped with some more tools, hierarchically. We have proposed either a new layer be devised over application layer in the protocol stack discussed e.g. in [1], named *abstract layer,* simply, *abstractor;* or of course it is also possible to embed it in the application layer. By the way, in any case, abstractor highly uses application layer protocols specially SMP, TADAP and SQDDP.

Abstractor in any sensor node (and even in sink and other nodes) is to provide basic abstraction of the sensed data (higher order nodes are expected to provide higher level of abstraction). According to the hardware specification of a sensor node, complex abstraction is not expected, but a basic. However, it is possible to leave higher level of abstraction to some more capable nodes, like *sinks*. Being more powerful and the gateway of the network, the sink's abstractor must perform much more complex and higher level of abstraction. This is due to the fact that each sensor, senses locally, which is adequate for sensing some phenomenon, whilst some phenomenon require to be sensed globally which is out of the capability of a single sensor node or even nodes in a limited domain, hence collected information from different nodes must be considered to get to a higher level of abstraction. Of course, sometimes it is needed to drive a higher level of abstraction from even a single node's sensed data that due to the limitations of sensor nodes, required complex processing must be performed in sink due to its functionality.

With separated concerns and a clear interface for applications to connect with ESN, different scenarios are possible for a context-aware application using a ESN. Applications can be run either in the sensor field, i.e. where the sensors are deployed, far from it or even both e.g. due to the needs to multidisciplinary contextual information, accordingly ESN can function actively, passively or both.

We define ESN to be passive while there is no sink node in it, otherwise it is active, however from one perspective it would be active and from other would be passive. In fact physical existence of sink node is not important here. What is important is their ability to provide required data at the desired abstract level. Any context-aware application, while entering a ESN, searches for a sink, if the discovered sink could not support the application’s needs or any sink node could not be found, the machine running the application may take the role of the sink in the ESN temporarily, as is present in the field. In the former case the basic abstractions provided by each node would enable new sink nodes to start interacting with the ESN.

B-Transparent and Distributed Communication

Essentially all contextual information needed by a context-aware application would not be found in the same field, but rather must be gathered from different sensor fields and even of different types. In this case the machine running the application can be present just in one place. More generally, it is imaginable that an application, e.g. the one that is for monitoring, is executed on a machine far from all its required contextual information source fields. It is necessary to be able to get access to all its required information. What we have proposed is a special node, named *granulator* that is responsible to connect different sensor networks. In effect, any granulator is connected to some ESNs through their sinks. This node is able to perform higher level of abstractions due to higher abstracted level of data and more comprehensive information it receives.

Consequently, it would be a legitimated expectation that by the existence of granulators, ESN may provide even not-directly sensible information. It is worth mentioning that in an ESN where different types of sensors are deployed, the sink node itself would be a restricted granulator too.

C-Constant Availability of Information

ESN provides contextual information constantly; however, this is due to the application how to use them. For example a context-aware application can follow the changes continuously or reacting according to a special change in the environment. To do the latter and in order to reduce the network and the machine running the application load, the application must be able to register its needs in related granulators or/and sinks depending on which level it needs to work. While any changes happen in the environment interesting for the application, sink or granulator should inform the application.

D-Context Storage and History

ESN must be able to record all it has sensed which can be used either by applications or even by the network. Network, for example can use it to predict its future context. In this case, the network can even estimate its error rate or its functionality. In the case of a great error, ESN can announce being unreliable to applications relying on it and ask the applications to switch to other ESNs, if possible, or demands for rebuilt.

E-Resource Discovery

Since context-aware applications are not aware of all resources available, there must be a road map available for them to find their required ESNs and resources. While in a field, by being aware of the existence of sensors, the application must try to find the sink or granulator. In the sink all available resources must be registered and be offered to the application on its request. This can be the same but at a higher level for granulators too.

F-Information Reliability

Information reliability is a significant factor for context-aware applications, since they have to set their behavior due to this information, any inconsistencies in the gathered data will result in unexpected application reaction. Nature of WSNs makes them reliable, since the network would resume its functionality with the failure of even some sensor nodes and links, but this much is not enough for context-aware applications.

With each sensor node failure, being able to continue its functionality, the quality of information prepared by the network degrades. The ESN's sink node must trace the behavior of the network and try to reduce the effect of nodes’ failure for example by sensor fusion, that is using data from multiple sensor nodes to deduce more accurate data. However sink node, while being unique in a ESN, would be bottleneck. Its failure will result in the ESN to become passive and looses its higher level abstractions. We recommend deploying multiple sinks for any enhanced sensor network, specially those whose information are vital. Sink fusion, that is combining data from different sinks in a given enhanced sensor network will result in enhanced information quality, whilst the network remains stable and yet reliable enough in the case of not its all sink nodes failure. However with the existence of multiple sinks, grouping the nodes as mentioned in [12] is easily feasible that will help more reusability of the network, even concurrently and also from different views.

In any ESN a mechanism for network calibration is required. Since calibrating each sensor node is almost not feasible, the calibration must be done passively in higher level nodes, here mainly sink, that in the case of calibration error in the network, each of its sink nodes should compensate for the error. *MacroCalibration* is a method for calibrating WSNs introduced in [14] that is also applicable here.

By the way ESN can deal with inexact information in at least three ways. Passing the erroneous information to the application by indicating that the information is not that exact, or attempting to remove the error for example by the above methods, or ask the user to handle the case manually.

4. CONCLUSION

Context-aware applications do need infrastructures to provide them with reliable contextual information. Current sensor networks provide raw data, sensed from physical environment however context-aware application may need more abstract information even gathered from different origins. Hence a specialized entities responsible for abstracting data to desired level must be added to WSNs and be available for context-aware applications through well defined interfaces. On the other hand contextual information gathered must be reliable enough moreover the network reliability must be testable and correctable. This leads to the introduction of Enhanced Sensor Network as a specialized sensor network that provides access to required contextual information with a uniform interface for any context-aware application that is running actually or virtually in the environment. Enhanced Sensor Network, due to its nature can get use of not only hardware sensors but also software sensors equally, and can be installed in any kind of environment, actual or virtual.

REFERENCES

Akyildiz I.F., Su, W., Sanakarasubramaniam Y., Cayirci E., (2002), “Wireless sensor networks: a survey”, Computer Networks 38, 393-422.

Brown, P.J., Bovey, J.D., Chen X., (1997), “Context-Aware Applications: from the Laboratory to the Marketplace”, IEEE Personal Communications, 4(5), pp. 58-64.

Bauer M., Heiber T., Kortuem G. ,Segall, Z., (1998), “A collaborative wearable system with remote sensing”. Proceedings of the 2nd International Symposium on WearableComputers (ISWC'98), 10-17. Los Alamitos, CA: IEEE.

Chong, C., Kumar, S. P., (2003), “Sensor Networks: Evolution, Opportunities, and challenges”, Proceedings of IEEE, Vol. 91, No. 8.

Dey, A.K. and Abowd, G.D., (1999), “Toward a better understanding of context and context-awareness.” GVU Technical Report GIT-GVU-99-22, College of Computing, Georgia Institute of Technology.

Dey A., Abowed G., Salber D., (2001), “A conceptual framework and toolkit for supporting the rapid prototyping of context aware applications”, Human Computer Interaction (HCI) Journal, Vol. 16(2-4), pp 97-166.

Davies, N., Mitchell, K., Cheverest, K., Blair, G. (1998). “Developing a Context Sensitive Tourist Guide”, First Workshop on Human Computer Interaction with Mobile Devices, GIST Technical Report G98-1.

Dey, A.K., Salber, D., Abowd, G.D., Futakawa, M., (1999), “The Conference Assistant: Combining context-awareness with wearable computing”. 3rd International Symposium on Wearable Computers, San Francisco, California, 18-19, pp. 21-28.

Degli Antoni G., Tahayori, H., (2006), “Multi paradigmatic development of context aware applications”, IADIS, Applied Computing, pp. 442-446.

Kong J, Cui J., Wu D., Gerla M., (2005), “Building underwater *Ad Hoc* networks and sensor networks for large scale real-time aquatic applications”, Military Communications Conference (MILCOM'05), Atlantic City, New Jersey, USA.

Harrison B. L., Fishkin, K. P., Gujar, A., Mochon, C., Want, R. (1998), “Squeeze me, hold me,tilt me! An exploration of manipulative user interfaces.” Proceedings of the CHI'98 Conference on Human Factors in Computer Systems, New York, NY: ACM Press.

Liu L., Chu M., Liu J., Reich J., Zhao F., (2003), “State-Centric programming for sensor-actuator network systems”, Pervasive Computing, IEEE, Vol. 2, No. 4, pp. 50-62

Rekimoto J., (1996), “Tilting operations for small screen interfaces”, Proceedings of the 9th AnnualACM Symposium on User Interface Software and Technology (UIST'96), 167-168. NewYork, NY: ACM Press.

Whitehouse K., Culler D., (2003), “Macro-calibration in Sensor/Actuator Networks”, Mobile Networks and Applications (MONET), Special Issue on Wireless Sensor Networks., 8, 4, 463-472.

Want, R., Hopper, A., Falcao, V., Gibbons, J. (1992), “The Active Badge Location System”. ACM Transactions on Information Systems 10(1) pp. 91-102.