MONITORING FIRE-FIGHTERS OPERATING IN HOSTILE ENVIRONMENTS WITH BODY-AREA WIRELESS SENSOR NETWORKS

Iacono, M.1, Baronti, P.1, Romano, G.2, Amato, G.1, and Chessa, S.1,3
1 Istituto di Scienza e Tecnologie dell'Informazione del CNR, via Moruzzi 1, 56124 Pisa, Italy
3 Vigili del Fuoco di Pisa, via Matteotti 2, 56124 Pisa, Italy
2 Dipartimento di Informatica, University of Pisa Largo B. Pontecorvo 3 - 56127 Pisa, Italy

ABSTRACT

We present a prototype application based on body area wireless sensor network exploiting the MaD-WiSe data management system which allows the operation control staff to remotely monitor the status of the firefighters in the field. Each firefighter is equipped with a set of wireless sensors that acquire both physiological data and data related to the microclimate within the coverall. Sensors self organize into a wireless network and cooperate to issue sensed data and alarms to the control staff.

1.0 INTRODUCTION

The intervention in fields affected by nuclear, chemical or bacteriological pollution is carried out by firefighters equipped by Personal Protective Equipments (PPE), with Totally Encapsulated Chemical Suit (TECS) which completely insulate them from the external environment. Besides being quite uncomfortable, these TECS may make difficult communication between fire-fighters and between the fire-fighters and the incident command system. They also prevent any form of status or health monitoring of the fire-fighters, thus making it very difficult to provide support and help to the men operating in the field.

Recent technological advances enabled the integration of wireless communication technologies, transducers and microcontrollers into tiny, self-powered devices. A number of these devices (also called sensors) deployed in an environment can form a wireless sensor network [1] to provide unattended monitoring and control support to remote users. In particular the user can program the network with data sensing, processing and collecting tasks, and it can receive the results without any physical intervention on the sensors themselves. This usage of the network is enforced by recently introduced database-like paradigms [2], [3] in which data management activity performed in the network is remotely controlled by interactively issuing queries, expressed in a high level language (such as SQL), which specify what data are of interest for a certain task, and how they should be manipulated. Changing the behavior of the data management activity in the network corresponds to execute actions like stopping a query execution and/or formulating a new one.

Along this research trend we have proposed in [4] a new database approach to data management in sensor networks (called MaD-WiSe) that attempts to overtake some of the typical limitations of systems recently proposed in this field. More specifically our approach addresses the following issues:

- in-network distributed processing of queries that relate data acquired by different nodes;
- execution of temporal aggregates;
- offering opportunities for network topology and data statistics aware query optimization;

Following this approach we have designed a prototype application providing remote monitoring of firefighters equipped with TECS. This application enable monitoring of parameters such as temperature, humidity and light within the coverall, physiological parameters such as heartbeat or breadth, and

[1] Work funded in part by MIUR in the framework of the Italian projects named "IS-MANET" and by the European Commission in the framework of the "SatNEx-II" NoE project (contract N. 27393) and the "Intermedia" NoE project (contract N.38419).

movements to detect if the firefighter is still active or if it is fallen. Using these information it is possible to detect alarms to be sent to the team leader in order to undertake the necessary actions, furthermore these information can be recorded to provide a history of the intervention for off-line analyses.

In the rest of the paper we briefly review the MaD-WiSe system and the hardware and software architecture, we then describe the monitoring application and we discuss some of the SQL monitoring queries embedded in the application. Finally we draw the conclusions.

1.1 MaD-WiSe Overview

The MaD-WiSe [4] system allows the interaction with a sensor network as a relational database. It consists of two parts. The first one runs on the sensors and is responsible for implementing the actual data acquisition, data processing and data forwarding activities requested in the network. The second part is an application that runs on a PC acting as network programmer and data collector (or sink). This part leverages on the ability of the sensor software of being dynamically reprogrammable to allow a user to program the sensor network by means of queries, and to collect the corresponding results.

The software running on the sensors employs a data model based on data streams [5] with very fine-grained granularity: data produced by a single transducer can also be modeled as a single data stream. The algebra of our query processor is composed of operators that take data streams as input and produce data streams as output. A query is represented as a combination of operators of the query algebra connected by data streams. Data streams can connect operators executed on different nodes, offering the opportunity of real distributed query processing (a query can be processed across several nodes in which different parts are independently executed). The above model allows us to (in-network) process queries that relate data produced by different nodes, process temporal and spatial aggregates, and opens up new opportunities for more effective query optimization techniques that take into account network topology, transducer statistics, and costs.

An application can requests to specific sensors the execution of operators of the query algebra and the management of data streams by sending appropriate command messages, or it can request that the currently running query be stopped and replaced with another. In this way it is possible to change the behavior of the sensors on the basis of user needs without physically acting on them (i.e. without uploading a new firmware on the sensors). After starting a certain distributed query and observing the data coming back from the sensors, the user can decide to replace the current query with a new one. To this purpose he can define the new query and send the appropriate messages to the sensor network.

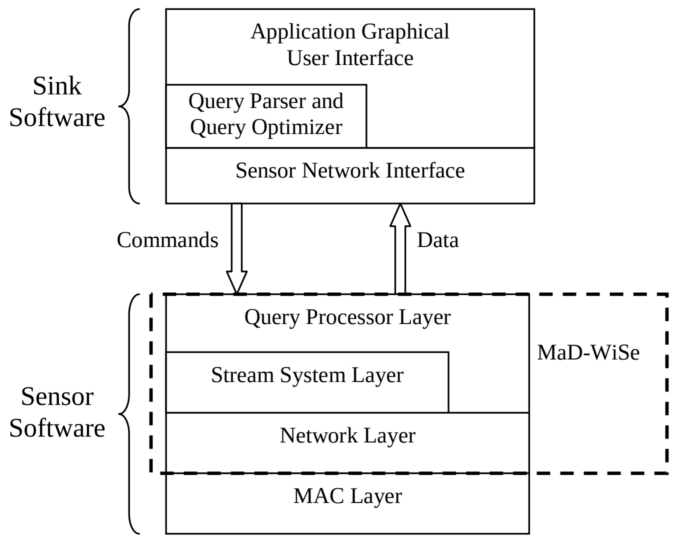

Figure 1. Software layers.

1.2 Hardware and Software Architecture

We implemented our system on the micaz motes [6]. These sensors were developed at UC Berkeley and are widely used in academic environments. They are equipped with an 8 MHz microcontroller, 4 KB of data memory, 128 KB of instruction memory, an IEEE 802.15.4 radio interface, and several transducers for light, temperature, acceleration (along X and Y axes), humidity, noise level and others. The operating system is TinyOs [7], [8] which is simple and offers basic hardware abstraction functionalities. The sensors can communicate with each other using the wireless interface to the purpose of in-network data processing, and communicate with the sink node (which can be a laptop or a hand-held device).

The sensor side of the MaD-WiSe system is organized into three layers, as depicted in Figure 1. The arrows indicate use relations among the layers. The layers interact through well defined interfaces and are autonomous with respect to each other. Each layer can be replaced with a new (different) implementation provided it complies with the existing interfaces.

The Network layer sits on top of the standard IEEE 802.15.4 MAC layer of TinyOS. It provides 2 types of communication services to the above layers. It offers both a connectionless and a connection-oriented service. At network startup a distributed protocol assigns a set of three of virtual coordinates to each sensor which is used by a multi-hop geographic routing protocol [9]. The network layer also implements an application-driven energy efficiency protocol for the connection-oriented service [10].

The Stream System Layer offers abstraction mechanisms for data access by means of data streams. It can be thought of as the equivalent of a file system on a sensor network, the main difference being that, in the latter, data is continuously produced as a consequence of acquisition from transducers or processing. The basic concept is the stream: a unidirectional data channel implemented over the connection-oriented service of the Network layer. It carries a flow of records (in the simplest case each record contains a sensed data or a combination of data sensed by different transducers). A stream is implemented as a finite size queue that holds recently acquired records. The Stream System offers functionalities to create/remove streams as well as read and write records from/to existing streams.

There are three types of streams: local, remote, and sensor streams. A local stream is local to a sensor in the sense that writing to and reading from the stream can only be requested by code running on the sensor. A remote stream is a data channel between two distinct sensors: writing to a remote stream happens on one sensor while reading from the stream happens on the other sensor. A sensor stream is directly connected to a transducer to carries-out data originated from it. Readings are only possible on this type of streams, given that the stream is automatically fed by the associated transducer.

The Query Processor Layer implements the query processor of a distributed database over the Stream System. It can be programmed remotely to take part in a distributed query execution. Queries are defined in terms of operators connected by streams. Streams play the role of relations (tables) and operators manipulate them similarly to relational algebra operators. However there are some fundamental differences. Tables are (more or less) static collections of records while streams are flowing records. Correspondingly, operators do not act on static relations but process records on-the-fly when they arrive. We have adapted some of the relational algebra operators to fit our data model.

The sink side of MaD-WiSe comprises a query parser and optimizer which receives high level SQL-like queries and produce an optimized query execution plan. This plan, which specifies individual sub queries to be executed on individual sensors, is distributed to the sensors using the Sensor Network Interface layer. The query results are in turn received by the Sensor Network Interface layer and forwarded to the application.

Defining a query for the query processor means defining what activities must be carried out by each sensor in the network. Among the activities we distinguish: data acquisition from local transducers, data processing and data forwarding. All these activities are expressed through streams and operators. Operators are active entities that take some inputs and produce an output. Inputs and output take the form of streams (we can think of streams as the means to connect operators). In more concrete terms an operator reads a record from its input stream(s), performs calculations/tests on the basis of its defining properties and the input record(s) and writes a (possibly) modified version of the inputs to the output stream.

1.3 The Monitoring Application

The sink side of the application is designed to run on a portable computer. It is developed in java, so it can easily be migrated to any platform. It interacts with the Body-Area Wireless Sensor Network composed by the sensors worn by the fire-fighters to detect alarms and to monitor the status of the people involved in an operation. Specifically we assume that five sensors are applied on the body of a fire-fighter: one on the chest, and one on each limb. Sensors on the limbs can be used to infer the dynamic activity of the person by using the accelerometers. It is possible to detect if a person is running, walking slowly, weaving the arms, etc. The sensor on the chest can be used to detect the position of the person (standing, laying, etc.), to monitor the temperature in the coverall.

From a remote location, it is possible to ask information to the wireless sensor network and to set alarm thresholds. Figure 1 shows the interaction paradigm between the remote computer and the wireless sensor network. The user interface is able to give a quick overview of the status of a person using graphical representations for the various measures acquired. In addition the user interface can also save in a file and show a time series with data acquired during the operation.

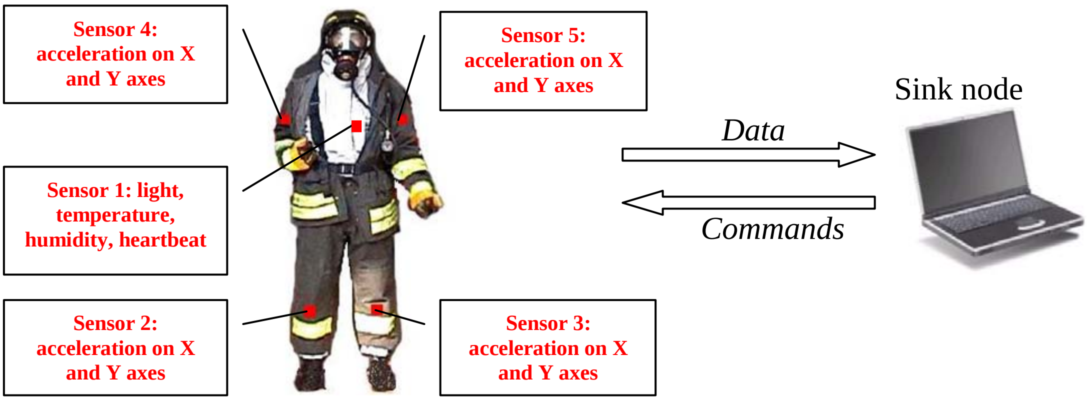

Figure 2. The body area wireless sensor network model.

A. Measured parameters

Various parameters can be measured by the wireless sensor network. In the prototype that we have built so fare we are able to monitor parameters such as temperature, humidity and light within the coverall, movements of legs and arms, and position (laying/standing).

Temperature, light and humidity and position (laying/standing) are measured periodically by a sensor placed on the chest of the fire-fighter. These data are acquired with a low sampling rate (once per second). We plan in the next application versions to measure the number of heartbeats and breaths per minute. The movements are detected by four sensors placed at each limb which acquire data from the X and Y acceleration transducers (in this case the sampling rate is higher, once per 100 milliseconds). This information can be used to detect if, within a time window, the fire-fighter did not move significantly so that an alarm can be raised. Figure 2 shows an example of deployment of the sensors on the body of the fire-fighter.

The combination of all the sampled data and the operation time could provide an estimation about the fatigue to which the firefighter has undergone. Secondly these data may be useful to estimate the level of comfort provided by the NBCR coveralls (for example combining temperature, humidity and heartbeats). These estimations are currently part of undergoing work.

B. The graphical Interface

The purpose of the Application graphical interface is to let the user interact with the sensor network, issuing commands and viewing query results.

Figure 3 illustrates a snapshot of the main application frame. As can be seen, three areas are identifiable: the sensor canvas (the largest, white, area), a tool bar on the left and a tool bar on the bottom.

The toolbar on the left contains four buttons corresponding to four different monitoring commands. At the moment the commands are:

- Sample temperature, light, humidity and position from the sensor placed on the chest.
- Sample movements of the arms
- Sample movements of the legs
- Sample the only the temperature.

The toolbar at the bottom contains five buttons to:

- Activate the monitoring task (button Start).
- Stop the monitoring task (button Stop).
- Activate recording of all the sampled data (button Rec).
- View sampled data in a graph (button Data).
- Terminate (button Exit).

The central canvas is used to display the position of the five sensors on the body of the fire-fighter (each sensor is represented by a red square) and, for each sensor, a graphical indicator of the sampled data. It should be observed that the sensor 1 is associated to 3 indicators corresponding to temperature, humidity, light and position, while the other sensors are only associated to movements indicators.

C. The queries implementing the commands

In the following we show some sample query embedded in the application.

1. Monitoring the status of the operator using the sensor placed on the chest every one second:
```
select *
from Chest.Light, Chest.Humidity, Chest.Temperature, Chest.AccelerationX
every 1000
```
2. The following query is used to monitor the movements of the legs:
```
select *
from LeftLeg.AccellerationX, RightLeg.AccelerationX
every 100
```
3. The following is an alarm for temperature exceeding 35 degrees.
```
select *
from Chest.Temperature
where Chest.Temperature>35
every 1000
```

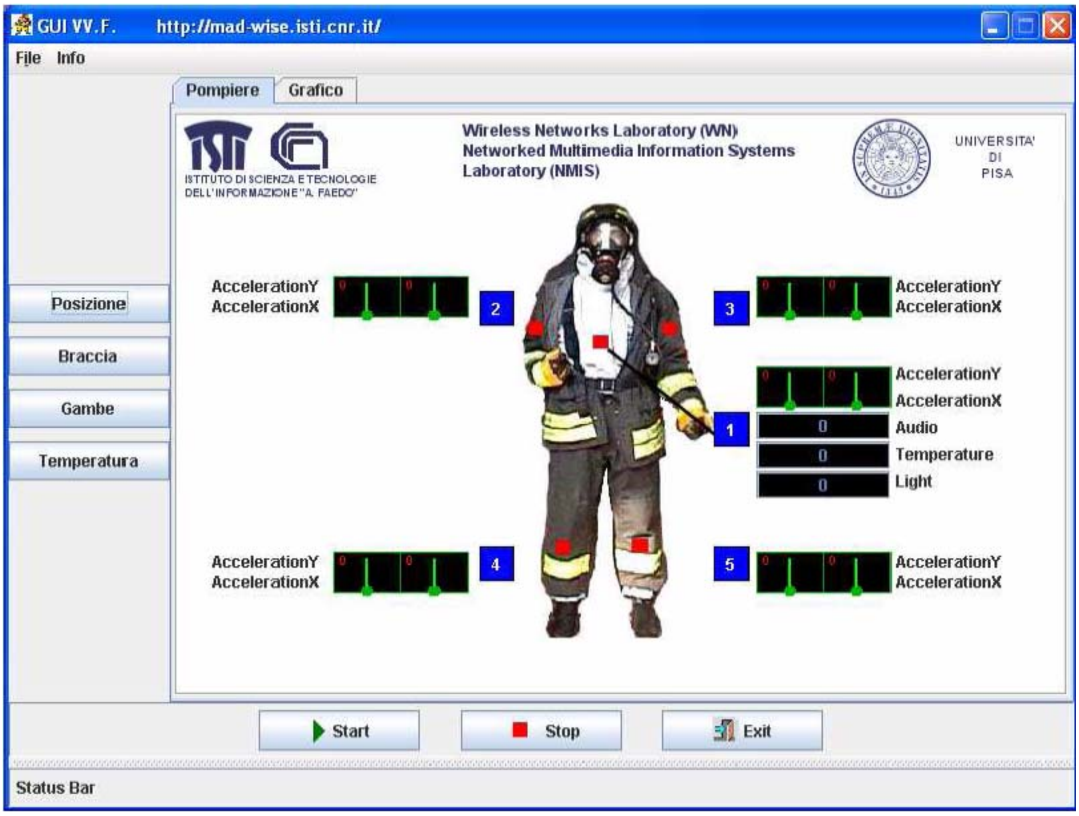

Figure 3. Application main frame.

1.4 Field Experiments

We tested our application prototype during a fire-fighters training day held on April 6th 2006 in the countryside of S.Piero a Grado - Pisa. The fire-fighters simulated a rescue mission whereby they first isolated a chemically contaminated area and then they proceeded to inspect the terrain, decontaminate it and rescue injured people.

One of the fire-fighters was equipped with five sensors inside his TECS (one on the chest and four on the limbs) and was monitored during the entire operation. Figure 4 shows a detail of the sensor on the left arm of the firefighter before wearing the TECS. We covered the drill area with some sensors in order to implement multi-hop communication and track the fire-fighter during motion.

We collected traces of temperature level inside the coverall as well as limb activity. Figure 5 shows the graphical interface plotting the temperature inside the TECS during the firefighter undressing. From the plot it can be seen the difference between the temperature inside the TECS before and after the undressing. Figure 6 shows the graphical interface plotting the movements of the left arm of the firefighter.

We did not experience radio interferences due to the coverall or other instrumentation. On the other hand we lost contact with the fire-fighter in a few occasions due to the limited radio range of our sensor nodes, their limited number, the large extent of the operation area and the presence of obstacles. Deploying a larger number of sensors and/or using more powerful radios would solve these problems.

1.5 Conclusions

We have presented an application to support monitoring of firefighters equipped with NBCR coveralls and operating in polluted fields. The application, which exploits a body-area wireless sensor network to sample different parameters, is built on the top of the MaD-WiSe database management system for sensor networks which support complex queries comprising temporal and spatial aggregates. This allowed for a simple and effective development of the application which allows for real-time monitoring of position, movements, comfort, and fatigue of the firefighters.

Figure 4. A detail of the sensor on the left arm of the firefighter.

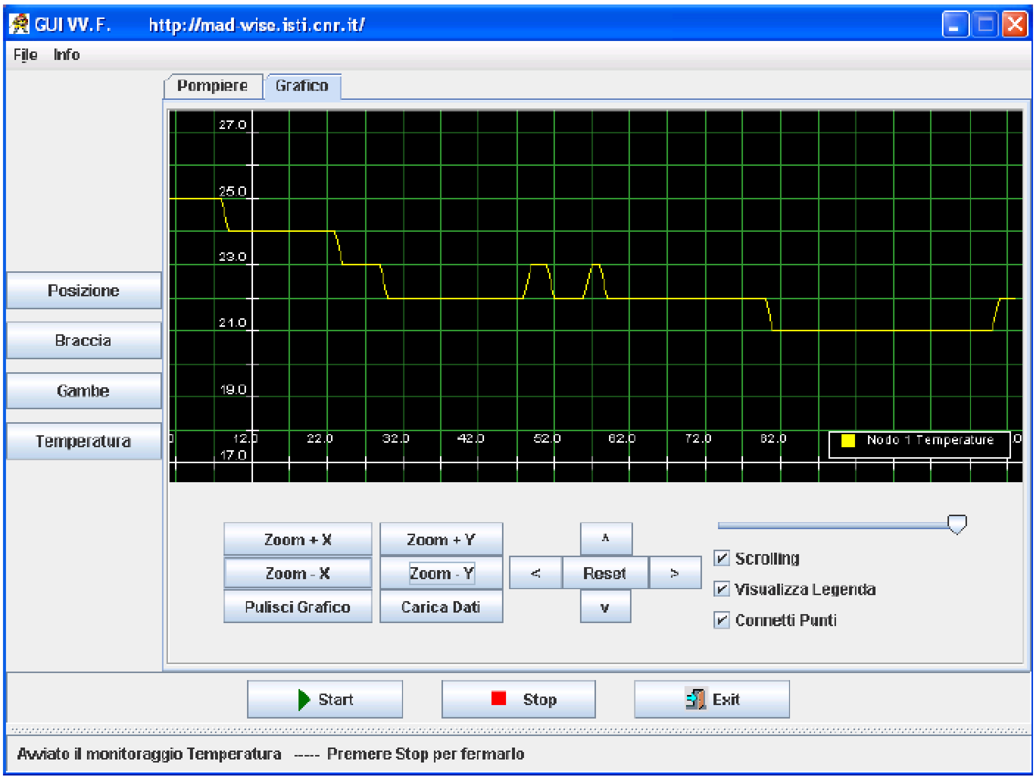

Figure 5. The graphical interface plotting the temperature inside the coverall.

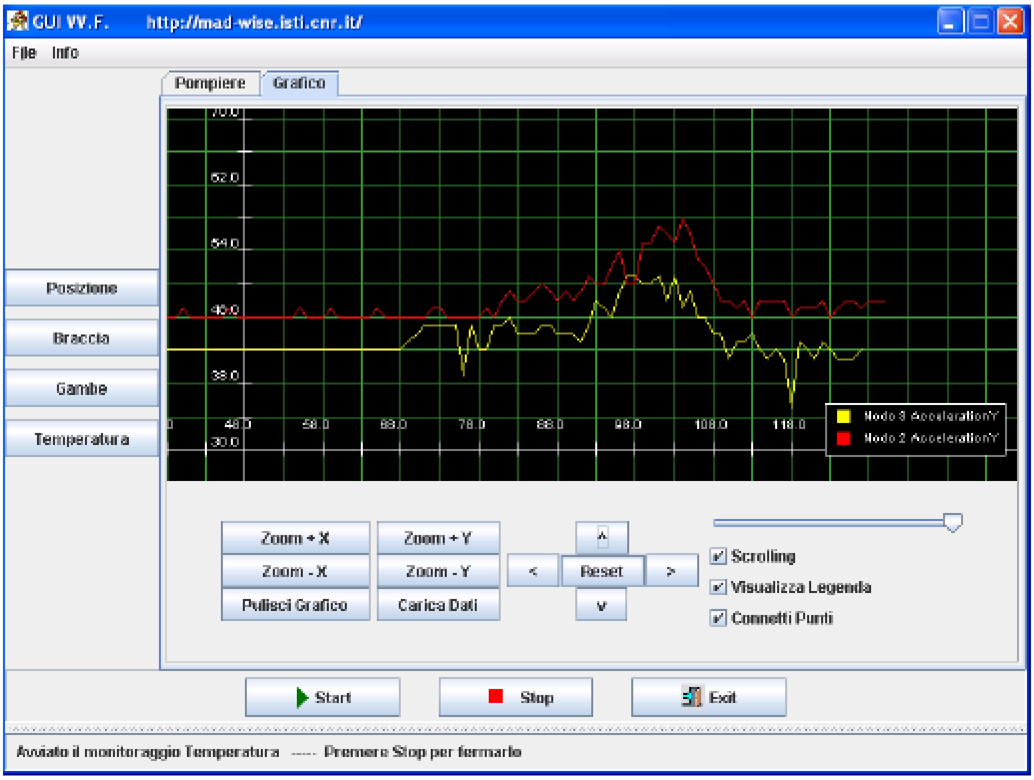

Figure 6. The graphical interface plotting the left arm movements.

REFERENCES

1. Akyildiz, F., Su, W., Sankarasubramaniam, Y., and Cayirci, E., Wireless Sensor Networks: a Survey, Computer Networks, 38, No. 4, 2002, pp. 393–422.
2. S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “The Design of an Acquisitional Query Processor For Sensor Networks”, In 2003 Proc. SIGMOD Conf., pp. 491.502.
3. Yao, Y. and Gehrke, J., E., The Cougar Approach to In-Network Query Processing in Sensor Networks, Sigmod Record, 31, No. 3., 2002.
4. Amato, G., Baronti, P., and Chessa, S., MaD-WiSe: Programming and Accessing Data in a Wireless Sensor Networks, Proceedings of the IEEE Eurocon, Belgrado, Serbia & Montenegro, 2005.
5. Abadi, D. ,J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, S., B., Aurora: a New Model and Architecture for Data Stream Management, VLDB Journal, 12, No. 2, 2003, pp. 120–139.
6. Crossbow Technology Inc. Mica2 website. http://www.xbow.com/products.
7. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D., E., and Pister, K., S., J., System Architecture Directions for Networked Sensors, Proceedings of the Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), 2000, pp. 93–104.
8. TinyOs Community Forum, http://www.tinyos.net.
9. Caruso, A., Chessa, S., De, S., and Urpi, A., GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks, Proceedings of the IEEE INFOCOM, Miami USA, June 2005.
10. Amato, G., Baronti, P., and Chessa, S., Connection-Oriented Communication Protocol in Wireless Sensor Networks, Istituto di Scienza e Tecnologie dell'Informazione – CNR, Report TR-10, Pisa, Italy, 2005.