Outlier Detection in Wireless Sensor Networks using Bayesian Belief Networks

D. Janakiram, Adi Mallikarjuna Reddy V and A V U Phani Kumar
Distributed and Object Systems Lab,
Department of CS & E,
Indian Institute of Technology Madras,
Chennai-600036, India.
http://dos.iitm.ac.in
{d.janakiram, adi, phani}@cs.iitm.ernet.in

***Abstract*— Data reliability is an important issue from the user's perspective, in the context of streamed data in wireless sensor networks (WSN). Reliability is affected by the harsh environmental conditions, interferences in wireless medium and usage of low quality sensors. Due to these conditions, the data generated by the sensors may get corrupted resulting in outliers and missing values. Deciding whether an observation is an outlier or not depends on the behavior of the neighbors' readings as well as the readings of the sensor itself. This can be done by capturing the spatio-temporal correlations that exists among the observations of the sensor nodes. By using naïve Bayesian networks for classification, we can estimate whether an observation belongs to a class or not. If it falls beyond the range of the class, then it can be detected as an outlier. However naïve Bayesian networks do not consider the conditional dependencies among the observations of sensor attributes. So, we propose an outlier detection scheme based on Bayesian Belief Networks, which captures the conditional dependencies among the observations of the attributes to detect the outliers in the sensor streamed data. Applicability of this scheme as a plug-in to the Component Oriented Middleware for Sensor Networks (COMiS) of our early research work is also presented.**

I. Introduction

Recent advancements in Micro-Electronic Mechanical Systems (MEMS) and wireless networks have made it possible to perform various tasks with miniature devices called sensors by forming a WSN. These miniature devices have stringent constraints in resources such as processing capability, power, memory, bandwidth etc. They are developed to be used in a wide variety of applications ranging from smart home systems to battlefield applications [1].

According to the survey conducted in mid-2002 by IEEE 1451.5 sensor group [2], data reliability is one of the important requirements of sensor network to be considered. Reliability of the data in WSN is affected by the following reasons [3] :

- Harsh environments in which WSN are deployed
- Interferences in the wireless medium
- Sleeping modes of the sensors
- Cheap and low quality sensors

Due to these conditions in the WSN, the data emanated from sensors may get corrupted resulting in outliers and missing values.

Outliers are data items, which deviate from the general behavior of the data items. These data items are inconsistent with the remaining set of data items [4].

The handling of missing and outlying observations in the sensor data is one of the most important consideration due to the following reasons.

1) Outlier values will predominantly determine the sensor network performance as they enhance the life span of the WSN by avoiding the transmission of the outlier data.
2) Even though some of the outlier values represent faulty sensors, some of them can represent phenomenon of interest, something significant from the viewpoint of the application domain.
3) For many applications, exceptions identified can often lead to the discovery of unexpected knowledge. Missing observations in the sensor data can cause data to be resent resulting in wasting of energy.

Many data mining algorithms try to minimize the influence of outliers or eliminate them all together. This, however, could result in the loss of important hidden information. Thus outlier detection and analysis is an interesting data mining task, referred to as **outlier mining**.

Introduction of outlier detection, one of the data mining technique, in WSN poses many challenges. One has to consider the streaming data at the sensor nodes, which adds up to the challenges. We can not perform outlier detection at sink node due to communication overhead.

We propose a technique for the outlier detection and estimating missing values using Bayesian Belief Networks (BBN) in the sensor stream data. BBNs capture the relationship between the attributes of sensor nodes as well as spatio-temporal correlations that exist among the sensor nodes. Apart from identifying the outliers, we can also estimate the missing values in the sensor network.

The proposed scheme goes through the following phases: training phase, testing phase, and inference phase. Training phase trains the BBN to capture the spatio-temporal correlations among the sensor nodes. In order to test the level of accuracy the network has learned, testing phase is used and the learned parameters are updated if needed. Inference phase infers the missing values and detects whether the stream data generated is an outlier or not.

COMiS [5], is a component oriented middleware for sensor networks. It provides adaptability to add/delete/modify the components at run-time. We are also trying to integrate this mining algorithm as a component of this middleware.

The remainder of the paper is organized as follows. Section II describes BBN in general. Section III discusses how to use BBN to detect outliers and estimate missing values. View of integration with middleware is discussed in section IV. A real time example of Great Duck Island is discussed in section V. Section VI discusses the importance and the trade-offs involved in using BBNs for outlier detection and missing values. Applications of detecting outliers and estimating missing values is discussed in section VII. Section VIII describes the related work on detection of outliers and estimating missing values in the WSN. Section IX concludes the paper with some future directions.

II. Bayesian Belief Networks

A BBN is a directed graph, together with an associated set of probability tables. The graph consists of nodes and arcs [15]. The nodes represent variables, which can be discrete or continuous [16]. The arcs in BBN represent causal/influential relationships among variables. Here we are considering continuous variables, because a sensor reading can take a range of values in their domain.

The key feature of BBN is that they enable us to model and reason about *uncertainty*. In BBN we model the dependency between uncertain variables by filling a *node probability table* (NPT). The NPT captures the conditional probabilities of a node given the state of its parent nodes.

BBNs are a way of describing complex probabilistic reasoning. The main use of BBNs arise in situations that require statistical reference in addition to statements about the probabilities of events. The user in this case has some evidence and wishes to infer the probabilities of other events, which have not yet been observed. Using probability theory and Bayes theorem, it is possible to update the values of all other probabilities in the BBN.

BBNs on their own enable us to model uncertain events and arguments about them. However, the real power of BBNs comes when we apply the rules of Bayesian probability to propagate consistently the impact of evidence on the probabilities of uncertain outcomes.

The whole process can be divided into three phases:

1) Constructing Bayesian Belief Networks
2) Learning Bayesian Belief Networks
3) Inferring from Bayesian Belief Networks

Constructing Bayesian Belief Networks :

BBNs make explicit dependencies between different variables. In general there may be relatively few direct dependencies (modeled by arcs between nodes of the network), which means many of the variables are conditionally independent (there is no arc linking them). In general, all the probabilities can be computed from the joint probability distribution.

For example, joint probability distribution P(A,B,C,D,E) can be written over several number of variables using the chain rule of probability as

$$P(A,B,C,D,E)= P(A)P(B|A)P(C|B,A)P(D|C,B,A)P(E|D,C,B,A)$$

But probability distribution is not so general; which means that there are conditional dependencies that may arise. For example, if we know A and C might no longer depend on what value B takes. Other such conditional dependencies could occur which enable us to write the probability in a simpler form is given below

$$P(A,B,C,D,E)= P(A)\ P(B|A)\ P(C|A)\ P(D|B)\ P(E|D,B)$$

From this probability distribution we can build the BBN by considering the dependencies among the attributes.

Learning Bayesian Belief Networks : It introduces three different cases:

1) If the network structure is given in advance and all the variables are fully observable in the training examples. Trivial Case: just estimating the conditional probabilities is enough.
2) If the network structure is given in advance but only some of the variables are observable in the training data. It is similar to learning the weights for the hidden units of a Neural Net using Gradient Ascent Procedure [17].
3) If the network structure is not known in advance, then we should use a heuristic search or constraint-based technique to search through potential structures.

Inferring from Bayesian Belief Networks:

We can infer the probability distribution of certain attributes, given the fact that we know what values other attributes can take. One situation where belief network inference is straightforward is in singly connected networks. A singly connected network is a graph where there is only one undirected path from any node to any other node. In this situation, the simple approach to solve the problem is to use belief propagation for inference.

III. Outlier Detection and Estimating Missing Values

Belief Network is used to represent conditional dependencies among the attributes that are measured by the sensor node. The parameters to be learned by the BBN are the probabilities of sensor current readings, previous reading of sensor, neighbour readings falling in one of the classified ranges. These probabilities can be learned using the training algorithms of the belief network like [18]. The learned parameters can be tested for the accuracy specified by the user or the application. Thereafter the inference phase starts with the estimation of missing values and detection of outliers using learned parameters.

To formally describe our model, we assume that the sensor readings are continuous values which falls in the range [x,y]. We divide this range (x-y), into a finite set of non-overlapping subintervals R= $\{r_1, r_2, \ldots, r_m\}$. These can be of any length, which are mutually exclusive and are called *classes*. The parameters to be learned are: P(S), P(His|S) and P(N|S).

where

P(S) - probability of sensor reading falling in one of the class

P(His|S) - probability of history reading falling in some class, given sensor reading falling in some other class

and P(N|S) - probability of neighbor readings falling in some class, given sensor reading falling in some other class

Estimating the probability values of each attribute can be calculated by training the BBN for some period. This can be done by considering the Markov chain assumption for spatial data [19] and Markov Random Fields for time-series data [20] [21]. After the training phase, we can perform test on the BBN, whether they have learned the parameters correctly or not by estimating the observations of the sensors and checking the correctness of these observations from the original readings.

Bayes classifier is a model for probabilistic inference; the target class r_{NB}, output by the classifier is inferred using maximum a posteriori (MAP) [22] [19] [23].

$$r_{MAP} = argmax_{r_i \epsilon R} P(r_i|h, n)$$

where h,n are the values of His, N, respectively. By using Bayes rule yields,

$$r_{MAP} = argmax_{r_i \epsilon R} \frac{P(h,n|r_i)P(r_i)}{P(h,n)}$$

i.e $r_{MAP} \approx argmax_{r_i \epsilon R} P(h|r_i)P(n|r_i)P(r_i)$

where P(r_i| h,n)- the probability that sensor reading falling in r_i

This model inherently uses the naïve Bayesian classification method proposed in [14] for capturing the spatio-temporal relationship among the sensor nodes.

IV. Integrating with Middleware (COMiS)

COMiS [5] is component oriented middleware for sensor networks developed using component technology. The main advantage with this is, components are loaded as and when required by the application [24]. Components are loaded into memory as dictated by application semantics.

It provides services such as finding components within distance ‘d’, registration, component updation, power management etc. The middleware has six components namely *Listener*, *Discovery*, *Send*, *Register*, *Update* and *PowerManagement*.

Apart from the core components listed above, the data mining component which detects the outliers and missing values can be integrated into this middleware. The deployment of this component in the WSN solely depends on the interest

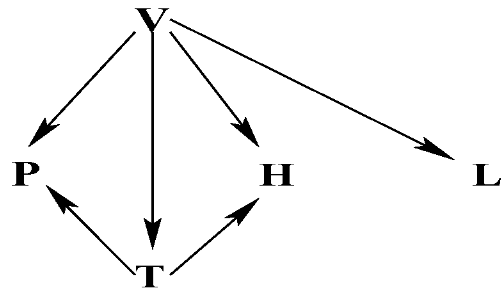

Fig. 1. Conditional Probability Dependencies among the attributes

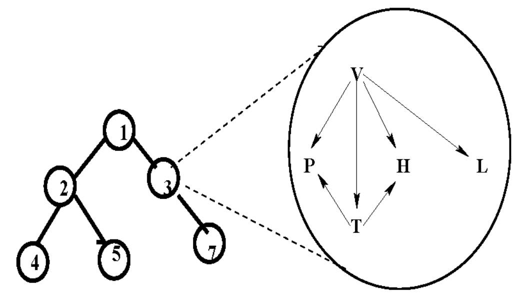

Fig. 2. View of WSN and the nodes with BBN

of the application user and applicability of this component to the application.

V. Example

This section describes the applicability of this method to a real world project used for habitat monitoring on Great duck island [26] using WSN. They measured two kinds of attributes viz habitat monitoring attributes, burrow monitoring attributes. We are considering habitat monitoring attributes. They are:

1) Temperature (T)
2) Relative Humidity (H)
3) Barometric Pressure (P)
4) Light intensity (L)
5) Mote Voltage (V)

Use of BBN to identify the outliers and missing values in the generated stream data involves constructing the BBN first, then obtaining the conditional probability table for each node in the BBN and finally inferring the required information from the BBN.

Relationships among the attributes:
Relative Humidity and Barometric Pressure readings are related to the temperature sensor readings. Voltage of mote determines the accuracy of the value that is sensed. So, it determines all the attributes in the BBN.

Figure 1 depicts possible relationship between the variables. Here the relationship between the voltage and the other variables is probability of affecting the other variables by the voltage levels. This is same in case of temperature, humidity, and pressure. We can infer these relationships among the attributes while training the BBN.

TABLE I
CONDITIONAL PROBABILITY TABLE

S	P(S_t)	P(S_v)
r_1	0.9	0.6
r_2	0.1	0.4

TABLE II
CONDITIONAL PROBABILITY TABLE

S_t	S_v	P(S_p)		P(S_h)	
		r_1	r_2	r_1	r_2
r_1	r_1	0.5	0.5	0.6	0.4
r_1	r_2	0.9	0.1	0.3	0.7
r_2	r_1	0.2	0.8	0.2	0.8
r_2	r_2	0.7	0.3	0.6	0.4

Figure 1 can be considered as a BBN for the attributes at the sensor node in WSN. We are assuming here that nodes in the WSN are homogeneous in their sensing capabilities.

Estimating the probability values of each attribute that it takes can be calculated by training the BBN for some period by considering the First order Markov models [19] for time-series data and Markov Random Fields [20], [21] for spatial data. After the training phase, we can perform test on the BBN, whether they have learned the parameters correctly or not by estimating the observations of the sensors and checking the correctness of these observations from the original readings.

The Conditional Probability Tables (CPT) at every node in the BBN shows how the readings of one sensor are dependent on the readings of other sensors.

The CPT of temperature shows the effect on other variables such as humidity and pressure. In the same way the CPT at the voltage observation shows the effect of voltage value at the node with the readings of all the sensors.

Figure 2 depicts the broader view of the WSN with the BBN at every node.

The joint probability distribution of the attributes of a sensor node can be calculated using the formula:

$$P(V,T,H,P,L)=P(V)\ P(L|V)\ P(T|V)\ P(H|V,T)\ P(P|V,T)$$

The conditional probability tables maintained at every node in the WSN are shown in tables I II and III.

Assume that the attribute values are divided into two classes r_1 and r_2. The tables depicted give the values of the probabilities of temperature, voltage and pressure. Table I shows the probability of voltage and temperature readings falling in the ranges r_1 and r_2. The probability of falling pressure and humidity given that temperature and voltage falling in different classes is shown in table II. The probability of history values of each individual sensor nodes falling in different ranges is given in table III. Table IV shows the conditional probabilities of neighbors falling in different ranges while the sensor node readings falling in different range.

For example, suppose we know that a sensor node humidity is in range r_2, and voltage is in range r_1. we might

TABLE V
CALCULATIONS

T	P(H|V,T)P(T)P(V)	P(T |V=r_1,H=r_2)
r_1	0.4*0.9*0.6 =0.216	0.4
r_2	0.2*0.9*0.6 =0.108	0.8

want to know whether the temperature reading falls in range r_1 or not. We are interested in the posterior probability P(T|V=r_1,H=r_2). To calculate this probability we first calculate P(T |V=r_1,H=r_2). This is proportional to P(H|V,T)P(T)P(V) (by using Bayes rule).

From the table V we can see that the probability of temperature falling in r_1 goes down (from 0.9 to 0.4) in light of the information.

Detecting Outliers: Assume that the generated data from the temperature sensor is X, then we will calculate the probability of this value falling in the range of specified classes. The range values with the highest probability is compared with the current reading. If it does not belong to that range then we can detect the reading as an outlier and can be reported to the base station and if it is needed, then we can correct the value and send the possible reading at that instant.

Let us assume that if the range of temperature [40,60] is divided into r_1=[40,50] and r_2=[51,60]. If the temperature reading is 55 , His= r_2 and N =(r_1,r_2). We compute P(r_1 | h=r_2,n=(r_1,r_2))≈ 0.23*0.6*0.9= 0.1242, while P(r_2 | h=r_2,n=(r_1,r_2))≈ 0.3*0.35*0.1=0.0105. The first class is therefore more likely. This indicates that the reading of temperature is outlier.

Estimating Missing Values: If the value to be generated at some epoch is missing, then based on the CPTs available at that particular attribute node in the BBN, we can predict the class it can fall. Median of that range gives us the estimated value for that attribute. But, the accuracy here depends on the length of the subintervals chosen.

VI. DISCUSSION

The consideration of BBNs in estimating the missing values and detecting outliers improves the accuracy. There are two types of dependencies exists at each sensor node.
They are:

1) Dependencies among the attributes of the sensor node
2) Dependency of sensor node readings on history and neighbor node readings

We have discussed capturing of both dependencies in this paper, however the latter one alone is explored in [14]. The conditional probability tables here captures the above mentioned dependencies. It is necessary to consider the correlations among the attributes for accuracy (if they exist). There are certain trade-offs exist in capturing the above mentioned dependencies viz choosing the number of neighbors and the approximation of probability values while training the BBN. Training BBNs would be difficult if the number of variables considered are large in number. The basic assumption is that the WSN deployed is dense in nature. It is realistic in the monitoring applications.

TABLE III
CONDITIONAL PROBABILITY TABLE

His	S_t	P(His|S_t)	S_v	P(His|S_v)	P(His|S_p)		P(His|S_h)	
					r_1	r_2	r_1	r_2
r_1	r_1	0.4	r_1	0.3	0.32	0.68	0.4	0.6
r_1	r_2	0.23	r_2	0.56	0.43	0.57	0.2	0.8
r_2	r_1	0.5	r_1	0.34	0.4	0.6	0.65	0.35
r_2	r_2	0.3	r_2	0.42	0.81	0.19	0.3	0.7

TABLE IV
CONDITIONAL PROBABILITY TABLE

N	S_t	P(N|S_t)	S_v	P(N|S_v)	P(N |S_p)		P(N |S_h)	
					r_1	r_2	r_1	r_2
(r_1,r_1)	r_1	0.4	r_1	0.2	0.3	0.7	0.6	0.4
(r_1,r_1)	r_2	0.65	r_2	0.3	0.34	0.66	0.82	0.18
(r_1,r_2)	r_1	0.6	r_1	0.8	0.4	0.6	0.7	0.3
(r_1,r_2)	r_2	0.35	r_2	0.7	0.5	0.5	0.35	0.65
(r_2,r_1)	r_1	0.32	r_1	0.43	0.2	0.8	0.4	0.6
(r_2,r_1)	r_2	0.4	r_2	0.82	0.61	0.39	0.66	0.34
(r_2,r_2)	r_1	0.6	r_1	0.5	0.83	0.17	0.32	0.68
(r_2,r_2)	r_2	0.2	r_2	0.6	0.53	0.47	0.19	0.81

VII. APPLICATIONS

This section describes the possible applications of detecting outliers and estimating missing values in the WSN.

1) **Detecting faulty/malicious sensors:** If a node keeps on generating a outlier value when compared to the neighbor nodes, then it can be considered as a faulty or malicious sensor.
2) **Query Evaluation:** While the query is executing in the WSN, if it encounters a missing value or outlier value then execution of query on such data items leads to inconsistencies in the results of queries and in some cases it may lead to trigger the actuators or alarms.
3) **Energy Conservation:** Detecting the outliers in the generated sensor streamed data eliminates the spurious values from sending to other nodes and/or base station saving energy.
4) **True Outliers:** True outliers or the phenomenon of interest can be easily identified with the data generated from sensors, because the training set constitutes the spatial as well as temporal data. We can capture the environmental changes in the probability tables, so that when the data generated from the sensor node in WSN can be informed as the phenomenon of interest.

VIII. RELATED WORK

There has been much research done in the areas of WSN, outliers, missing values, streamed data, but not much work has been done in the conjunction of WSN and the rest.

Literature in the area of WSN includes wide range of applications ranging from smart home to battle-field applications, protocols, programming languages and the middleware architectures [1] [6].

Several algorithms have been proposed in statistics community for the problem of finding outliers [7], missing values [8] [9], deviations in large datasets [10] [11]. However, none of the above approaches are directly applicable to WSN.

The outlier detection in WSN using kernel density estimators in a distributed manner is described in [12]. This considers the streamed data values of window size N, and updates the probability distribution model of the sensor node. It does not consider the spatial relationships among the sensor nodes.

The paper [13] uses association rules of data mining in order to find the association among the attributes and to estimate the missing values at the base station. It is applicable to the case where the whole data of sensors is stored in centralized database at the base station and posing the queries on this centralized database.

The naïve Bayesian classification technique to detect outliers and missing values in the WSN is proposed in [14]. Here spatio-temporal correlations are only considered but not correlations among the attributes. Correlations are very important in detecting outliers and estimating missing values for greater accuracy.

IX. CONCLUSIONS AND FUTURE WORK

In the recent years, sensor networks have become increasingly popular. Efforts have been devoted to this area, demonstrating their functionality and wide applicability to variety of applications.

In this paper, we have proposed Bayesian Belief Network method for capturing the correlations among the attributes and how to use it to detect outliers and estimating missing values from the stream data emanated from the sensors in a distributed and on-line fashion. It improves the accuracy in detecting the outliers and missing values.

We would like to integrate this with the middleware [5], and test the performance and accuracy of this methods over the sensor testbed. Possible ways of applying the identified spatio-temporal correlations in the WSN applications is an interesting research problem.

References

[1] I.F. Akyildiz, et al., "Wireless Sensor Networks: A Survey", *Computer Networks*, Elsevier Science, Vol. 38, No. 4, 2002, pp. 393-422.

[2] http://grouper.ieee.org/groups/1451/5/.

[3] E. Elnahrawy and B. Nath, "Cleaning and Querying Noisy Sensors",In:*Proc. WSNA03*, San Diego, CA, 2003.

[4] Hawkins. D, *Identification of Outliers*, Chapman and Hall, 1980.

[5] D. Janakiram, R. Venkateswarlu, S. Nitin, "COMiS: Component Oriented Middleware for Sensor Networks", To appear in the proceedings of 14th IEEE Workshop on Local Area and Metropolitan Networks (LANMAN), September 2005, Chania, Crete, Greece.

[6] D. Janakiram, R. Venkateswarlu and S.Nitin, "A Survey on Programming Languages, Middleware and Applications in Wireless Sensor Networks", Tech. Rep. IITM-CSE-DOS-2005-04

[7] V. Barnet and T. Lewis, *Outliers in Statistical Data*, John Wiley and Sons, Inc., 1994.

[8] G. McLachlan and K. Thriyambakam, *The EM Algorithm and Extensions*, John wiley and Sons, 1997.

[9] D. Rubin, "Multiple Imputations after 18 years", *Journal of American Statistical Association*, No. 91, 1996, pp. 473-478.

[10] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jorg Sander. "LOF: Identifying Density-Based Local Outliers", In: *ACM SIGMOD International Conference*, Dallas, TX, USA, 2000, pp. 2132.

[11] Andreas Arning, Rakesh Agrawal, and Prabhakar Raghavan, "A Linear Method for Deviation Detection in Large Databases", In: *International Conference on Knowledge Discovery and Data Mining*, Portland, OR, USA, August 1996, pp. 164169.

[12] T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Distributed Deviation Detection in Sensor Networks", In: *SIGMOD Record*, Vol. 32, No. 4, December 2003, pp. 77-82.

[13] Mihail Halatchev, Le Gruenwald, "Estimating Missing Values in Related Sensor Data Streams", In: *Proc. of COMAD*, 2005 Goa, India, pp. 83-94.

[14] E. Elnahrawy and B. Nath, "Context-Aware Sensors", In: *Proc. European Workshop on Wireless Sensor Networks*, 2003, pp. 77-93.

[15] Pearl, J., Fusion, "Propagation and Structuring in Belief Networks", *Artificial Intelligence* No. 29, 1986, pp. 241-288.

[16] Shachter, R.D. and Kenley, C.R., "Gaussian influence diagrams", *Management Science*, No. 35, 1989, pp. 589-604.

[17] D.Heckerman, D.Geiger, and D.M. Chickering, "Learning Bayesian networks:The combination of knowledge and statistical data", *Machine Learning*, Vol. 20, 1995, pp. 197-243.

[18] J Cheng, R Greiner, "Learning Bayesian Belief Network Classifi ers: Algorithms and System", In *Canadian Conference on AI*,2001,

[19] D.Hand, H.Mannila, and P.Smyth, *Principles of Data Mining*, The MIT Press, 2001.

[20] P.Smyth, "Belief Networks, hidden markov models, and markov random fi elds: a unifying view", *Pattern Recognition Letters*, 1998.

[21] J.S. Yedidia, W.T. Freeman and Y. Weiss, "Understanding belief propagation and its generalizations", Tech.Rep. TR2001-22, MERL, November 2001.

[22] T. Mitchell, *Machine Learning*, McGraw Hill, NewYork, 1997.

[23] I. H. Witten and E. Frank, "Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations", *Morgan Kaufmann*, 2000.

[24] N Parlavantzas, G Coulson, M Clarke, G Blair, "Towards a Reflective Component Based Middleware Architecture" ,In: *Workshop on Reflection and Metalevel Architectures*, June, 2000.

[25] D. Janakiram and R. Venkateswarlu, "A Distributed Compositional Language for Wireless Sensor Networks", In: *Proc. of IEEE Conference on Enabling Technologies for Smart Appliances (ETSA)*, Hyderabad, India, January 2005.

[26] www.greatduckisland.net, "Habitat monitoring on Great duck island".