World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:3, 2011

Efficient and extensible data processing framework in ubiquitous sensor networks

Junghoon Lee, Gyung-Leen Park, Ho-Young Kwak, Cheol Min Kim

Abstract—This paper presents the design and implements the prototype of an intelligent data processing framework in ubiquitous sensor networks. Much focus is put on how to handle the sensor data stream as well as the interoperability between the low-level sensor data and application clients. Our framework first addresses systematic middleware which mitigates the interaction between the application layer and low-level sensors, for the sake of analyzing a great volume of sensor data by filtering and integrating to create value-added context information. Then, an agent-based architecture is proposed for real-time data distribution to efficiently forward a specific event to the appropriate application registered in the directory service via the open interface. The prototype implementation demonstrates that our framework can host a sophisticated application on the ubiquitous sensor network and it can autonomously evolve to new middleware, taking advantages of promising technologies such as software agents, XML, cloud computing, and the like.

Keywords—ubiquitous sensor network, stream data processing, middleware design, real-time data distribution, agent architecture

I. Introduction

Nowadays, wireless sensor networks have been successfully applied to environmental and wildlife habitat monitoring, and still keep extending their application areas [1]. Intelligent and efficient management endowed by the sensor network also improves productivity and profit of the agricultural and livestock farms. Sensor data, inherently quite different from the traditional data records, are inherently created in the form of a real-time, continuous, ordered sequence of sensor readings. Here, the temporal order can be decided either implicitly by arrival time or explicitly by timestamp [2]. Accordingly, a data stream is defined as a continuous sequence of tuples. It is impossible to control the temporal order or to locally store entire data stream [3]. The data item is removed as soon as it is received from the data stream and processed. Structure of data items in a data stream can change in time. Moreover, many data streams can include the spatial tag not just the temporal order, possibly hosting a geographic application on the sensor network.

A sensor network can be viewed as a large database system which responds to the query issued from various applications [4]. After all, queries on the sensor stream must run continuously over a limited period and incrementally return new results as new data arrives. Here, a continuous query is issued once and may remain active for hours and days. The unboundedness of a stream prevents the issuer from getting exact answers. A key to inferring high-level behavior is fusing historic sensor data with general commonsense knowledge of real-world constraints. Practically, most modern sensor node installs TinyOS, which is a free and open source component-based operating system [5]. In this platform, TinyDB is a query processing system for extracting information from a network of TinyOS sensors. TinyDB provides a simple, SQL-like interface to specify the data you want to extract, along with additional parameters, like the rate at which data should be refreshed. Hence, the database view of the sensor network is reasonable also in practice.

In addition to the monitoring operation, the current reliability level of wireless communication allows to run even a process control function on the sensor network [6]. An efficient control scheme based on the embedded sensor technology is essential for the intelligent environment and seamless interaction with users. However, existing control systems, especially in the agricultural and livestock areas, lacks a user-oriented interface, as it is mainly built on top of the noncooperative control devices. Thus, it is necessary to integrate heterogeneous control devices to provide a systematic view to the users. To this end, the system must create the embedded environmental information by combining various low-level values of on/off, temperate, humidity, CO_2, ventilation, wind velocity, and precipitation sensors.

In the mean time, the agent represents an intelligent object automatically interacting with the changing external conditions and reacting to them. For example, a human agent interacts with the physical world by means of body organizations, while a robot agent keeps track of the external change via the camera or infrared sensors. Specifically, a software agent, working inside the computing world, takes binary information as input and decides its reaction also in the binary form. An intelligent agent possibly automates the repeated work, reminds a user of what he may forget, and classifies the complex data. As such, in the recent RFID and sensor network, much effort is put on the agent architecture capable of collecting, transmitting, storing, and distributing real-time sensor data.

After all, for efficient monitoring and intelligent control in the ubiquitous sensor network, or USN in short, it is necessary to design a system-wide framework combining above-mentioned features. In this regard, this paper is to present the design and implement the prototype of an intelligent USN mainly targeting at the agricultural and livestock farms. It

Junghoon Lee and Gyung-Leen Park are with the Department of Computer Science and Statistics, Jeju National University, Republic of Korea. e-mail: {jhlee, glpark}@jejunu.ac.kr

Ho-Young Kwak is with the Department of Computer Engineering, Jeju National University, Republic of Korea. e-mail: kwak@jejunu.ac.kr.

Cheol Min Kim is with the Department of Computer Education, Jeju National University, Republic of Korea. e-mail: cmkim@jejunu.ac.kr.

This research was supported by the MKE (The Ministry of Knowledge Economy), through the project of Region technical renovation, Republic of Korea.

focuses on how to handle the sensor data stream and user-generated queries for a variety of applications. Our framework first addresses a USN middleware which mitigates the interaction between the application layer and low-level sensors, defining the respective functions of service, server-side, and in-network middleware layers. Then, an agent-based architecture is proposed for real-time data distribution to efficiently forward a specific event to the appropriate application which may decide the control action based on the well-defined logic and knowledge.

II. Background and Related Work

As for continuous query processing, [7] proposed the SyncQuery language that expresses composable queries over streams, pointing out that composition of queries, and hence supporting views is not possible in the append-only stream model. This language employs the tagged stream model in which a data stream is treated as a sequence of modifications over the given relation. Particularly, the sliding-window approach is generalized by introducing the synchronization principle that empowers SyncSQL with a formal mechanism to express queries with arbitrary refresh condition. Besides, this work includes an algebraic framework for SyncSQL queries, couple of equivalances and transformation rules, and a query-matching algorithm.

Specifically, XML data stream processing is also of interest, as XML becomes common part of information systems, including RFID (Radio Frequency IDentifier), ad-hoc sensor data collection, network traffic management, and so-called service-oriented architecture [8]. Generally, XML streams are created as a second-hand product obtained from information exchange in XML systems, rather than from the raw sensor values. XML data streams can be viewed as a sequence of XML documents, and each data item in the stream is a valid standalone XML document, which is independent of other items in the stream. Moreover, queries on data stream can support data mining and filtering. While the first evaluates queries that span over a long time period, processing a great deal of time-sequenced data, the second takes the data items from the stream matching the filtering condition. Anyway, processing XML has an attractive real-world motivation.

Sometimes, each tuple can include a spatial tag such as GPS reading in addition to the temporal tag, especially when the system includes a mobile component. *Project Lachesis* has proposed a number of rigorously defined data structures and algorithms for analyzing and generating location histories [9]. In their approach, *stays* are instances where a vehicle has spent some time at a single location, while *destinations* are clusters of stays. Based on this classification, this system investigated two probabilistic models, both with and without first-order Markovian conditioning. In addition, the records of a same GPS tag can be used to fix the positioning error in stream data manipulation [10]. The location field is also important in a fixed sensor node, when the node is dependent on the terrain effect and this must be integrated in the analysis model [11].

III. USN Middleware

Middleware can work between the sensor layer and the diverse monitor and control applications. It can analyze the great volume of sensor data by filtering and integrating to create, store, manage, and retrieve the value-added context information. It is also possible to dispatch the information to the industrial process control, and even synthesize the multiple services on the middleware platform. Our framework is illustrated in Figure 1, which consists of 3 layers of in-network middleware, server-side middleware, and service middleware.

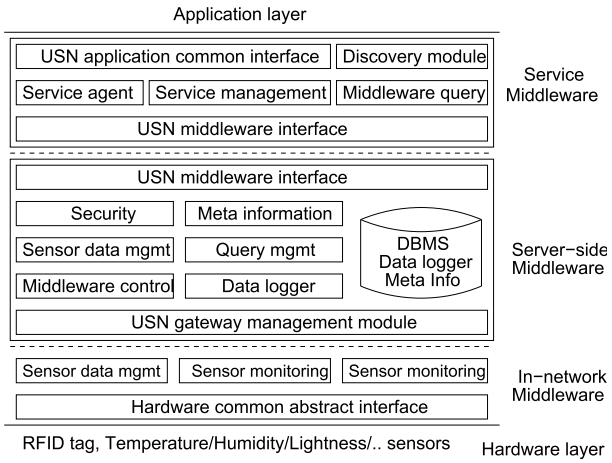

Fig. 1. USN middleware architecture

To begin with, the in-network layer embeds in the sensor node, providing a sensor network monitor and the common sensor network interface. For sensor monitoring, our system differentiates raw sensor values from computed values, and respectively represents their temporal dynamics. The network configuration is given as a form of metadata specification to match the sensor and actuator identifier in the network location. By this, along with basic network monitoring, it is possible to cope with the heterogeneity of sensor networks and network topology changes. Additionally, sensor network abstraction is the fundamental of the common sensor network interface. It can provide a consistent middleware interface, sensor network adapter management, message format standardization, and the like. The network abstraction covers the underlying communication mechanism and actual network devices. In addition, the message format is defined for data sensing, data storage, data processing, and report generation to the upper layer module.

Next, the server-side layer mainly includes intelligent manipulation, context information management, and metadata interpretation. To filter the meaningful value from the set of data which keep being accumulated over a long time interval, the sensor mining module conducts data classification and time series analysis. Here, the abnormal situation is detected based on the predefined pattern and registered knowledge. The intelligent manipulator takes both the real-time sensor value and the history data to define and interpret a specific event. Based on the metadata given by an application to specify the

respective event detection condition, the middleware can catch the event and notify to the application. The context information is stored in the middleware and forwarded to the server in the form of the filtered information, summary values, or low-level sensor data. metadata specification is registered by the application server in the middleware in the form of a XML document. To this end, the middleware layer provides the interface to register, retrieve, and update the metadata.

The service middleware layer exports an open USN application interface, internally consisting of a service discovery module, application agents, and middleware query handlers. This API makes it easy to program an application through the USN middleware, taking advantage of object abstraction and various intelligence techniques. Particularly, each functional module is designed separately and implemented in the DLL (Dynamic Link Library) components. Moreover, it possible to extend API set according to the addition of a new service.

IV. Agent Module

One of the most common applications in USN is necessarily event detection and reaction [12]. The agent-based architecture requires not just a database system which can simply store and retrieve the collected data. Instead, a distribution function is essential to monitor the real-time sensor data, detect an event, and select an appropriate application that can handle it as depicted in Figure 2. Add, modify, and delete operations are 3 major event types taking place in the database. Any external entity that invokes those operations makes a data item to proceed to the forwarding module via trigger utility. Then, the forwarding module converts the data into the XML message and sends to the corresponding remote handler. The real-time data distribution server receives a message from the message queue and parses it. After filtering, the data item is transferred to and buffered at the thread pool belonging to the service associated with the data [13]. Each data item is forwarded to the remote application via the XML interface for better interoperability.

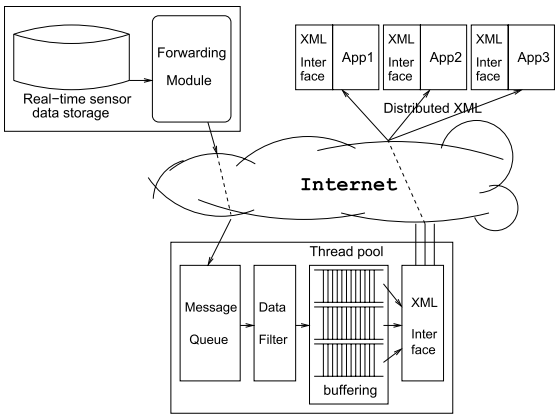

Fig. 2. Real-time data distribution scheme

The real-time data distribution server can monitor the server status as well as control the server application. It can initiate, pause, resume, and stop the respective service component selectively. The user interface also provides CPU and memory utilization graph, current message queue status, per-thread statistics, and other useful information as shown in Figure 3.

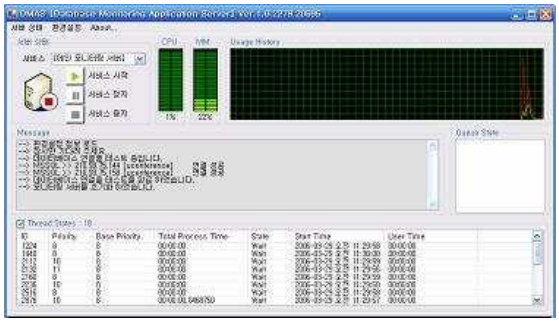

Fig. 3. Real-time data distribution

As a pilot application, the temperature monitor tracks the current temperature of a specific position selected via the geographic map. According to the initiation command, the server module begins to collect and store sensor readings. During the lifetime of this operation, the event detection is carried out based on the criteria specified in the query. The client also retrieves the current temperature value to monitor the up-to-date temperature change. Figure 4 and 5 show the UI implemented in this application. First of all, they display the map, location of sensors along with the current status of them. In addition, the series of sensor values are scrolled in the listbox, while a graph is created to plot the temperature change. Figure 4 indicates the normal status where no sensor value deviates fromthe given bound, and all nodes are marked blue. Whereas, in Figure 5, one sensor node detects the value out of the normal range and this node turns red.

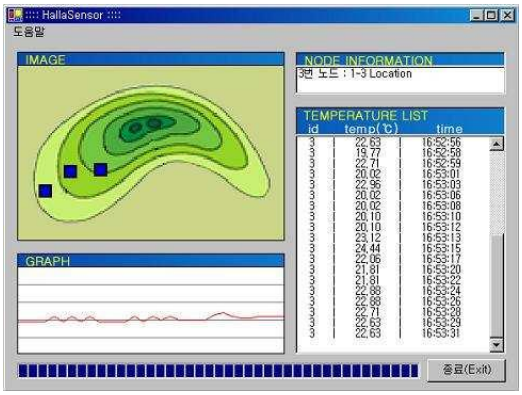

Fig. 4. Normal status indication

V. Concluding Remarks

This paper has presented the design and implemented the prototype of an intelligent USN data processing framework, based on the result of a research and technical project named as *Development of convergence techniques for agriculture, fisheries, and livestock industries based on the ubiquitous sensor networks*. Our system mainly focuses on how to handle the great volume of the sensor data stream as well as the interoperability between the low-level sensor data and application

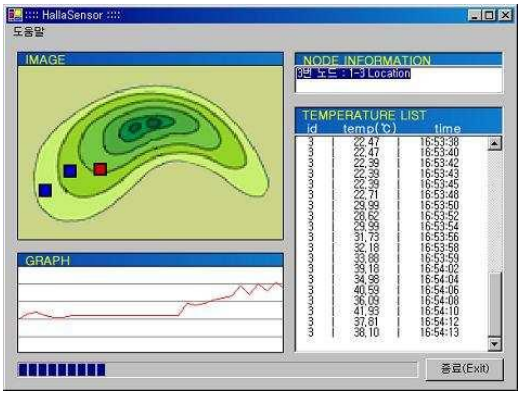

Fig. 5. Abnormal status detection

clients. USN middleware was first addressed to mitigate the interaction between the application layer and low-level sensors, defining the respective functions of service, server-side, and in-network middleware layers. Then, an agent-based architecture is proposed for real-time data distribution to efficiently detect and forward a specific event to the appropriate application which may decide the control action based on the well-defined logic and knowledge.

The prototype implementation demonstrates that our framework can host a sophisticated application on the ubiquitous sensor network and it can autonomously evolve to a new middleware, taking advantages of new technologies such as software agents, XML, reliable communication protocols, cloud computing, and the like.

As future work, we are planning to design an inference engine capable of achieving context information, analyzing information using the well-defined knowledge, and providing a user with accurate target status to decide the prompt and correct reaction. In addition, clear logic processing can create a new descriptive context information by merging multiple context information sets.

REFERENCES

[1] L. Golab, M. Oszu, "Issues in data stream management," *ACM SIGMOD Record*, Vol. 32, Issue 2, 2003, pp. 5-14.
[2] U. Sricastava, J. Widom, "Flexible time management in data stream systems," *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 2004, pp. 263-274.
[3] S. Madden, M. Franklin, J. Hellerstein, W. Hong, "The design of an acquisitional query processor for sensor networks." *ACM SINGMOD* 2003.
[4] S. Madden and M. J. Franklin, "Fjording the stream: An architecture for queries over streaming sensor data," *Proc. of the 2002 Intl. Conf. on Data Engineering*, 2002.
[5] *http://www.tinyos.net*
[6] J. Lee, H. Song, A. K. Mok, "Design of a reliable communication system for grid-style traffic control networks," *The 16th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2010, pp. 133-142,
[7] T. Ghanem, A. Elmagarmid, P. Larson, W. Aref, "Supporting views in data stream management systems," *ACM Transactions on Database Systems*, Vol. 35, No. 1, 2010.
[8] J. Ulrych, "Processing XML data streams: A survey," *WDS Proc. Contributed Papers*, 2008, pp.218-223.
[9] R. Hariharan and K. Toyama, "Project Lachesis: Parsing and modeling location histories", *Lecture Notes in Computer Science*, Springer Verlag, Vol. 3234, 2004, pp.106–124.
[10] J. Lee, G. Park, C. Sung, H. Choi, "A Cooperative Position Fix Scheme Based on a Group Management on the Vehicular Network," *Journal of Information Science and Engineering*, Vol. 26, No. 1, 2010, pp. 15-26.
[11] J. Lee, J. Hong, "Design and implementation of a spatial data processing engine for the telematics network," *Applied Computing and Computational Science*, 2008, pp.39-43.
[12] C. Lee, A. K. Mok, P. Konana, "Monitoring of Timing Constraints with Confidence Threshold Requirements," *IEEE Trans. Computers*, Vol. 65, No. 7, 2007, pp. 977-991.
[13] H. Woo and A. K. Mok, "Real-time monitoring of uncertain data streams using probabilistic similarity," *Proc. of IEEE Real-Time Systems Symposium*, 2007, pp. 288–300.