EFFICIENT USN MIDDLEWARE FOR ASSET TRACKING

Kwangsoo Kim[1], Minsoo Kim[1], Junghee Jo[1], Cheolsig Pyo[1], Shinyoung Park[2]

[1]Telematics·USN Research Division
Electronics and Telecommunications Research Institute
161, Gajeong-dong, Yuseong-gu, Daejeon 305-806
{enoch,minsoo,dreamer,cspyo}@etri.re.kr

[2]Things to Things Research Center
Motorola Laboratories
215, Yangjae-dong, Seocho-gu, Seoul 137-893
{spark@motorola.com}

ABSTRACT: A small sized device with computing, communicating, sensing capability is changing our life. It will be deployed in the world and acquire a lot of data from the world. It is used for various applications such as military surveillance, environmental monitoring, structure health monitoring, building management, asset tracking, etc. In this paper we focus on USN middleware for asset tracking. A mobile asset is moving here and there within a specific area. The USN middleware tracks the mobile assets in real-time by using sensor nodes and notify their current positions to a user. To achieve the goal, the USN middleware provides some features related to the positions of mobile assets. . They are storing location data by using 3D indexing method, retrieving them by using spatio-temporal query, making trace of an asset, and retrieving the history data of an asset. In the paper, we developed USN middleware to adapt the requirements of asset tracking. It can help users increase the efficiency of their business related to mobile assets and make a valuable decision.

KEY WORDS: USN, Middleware, Asset Tracking

1. INTRODUCTION

A ubiquitous world means that many kinds computing devices are scattered around everywhere we live and captures many data about our world without our recognition. The technologies related to wireless sensor networks have made the ubiquitous idea be true. A small sized sensor node with wireless communication facilities can be installed on anywhere. They produce many data related to real world, and transmit them on their places. USN (Ubiquitous Sensor Network) consists of from only a few wireless sensor nodes to a large number of them according to the requirements of a target application. The coverage of a sensor network also is very various from a house to a country by what it works.

Middleware refers to a software layer that operates between the application and various sensor networks under complex heterogeneous environments to exchange information efficiently. It helps an enterprise to integrate and manage several inner businesses. For example, the u-City consists of various services such as water supply management, sewage management, transportation management, and structure monitoring. Each local government constructing the u-City plans to establish an ICOC (Integrated City Operation Center) to integrate every service into one whole system. It will manage every facility or it will work within the city using wireless sensor networks and wired broad band networks to increase the effectiveness of functionalities and the management of the city. The middleware plays an important role in between integrating several heterogeneous services.

There are so many applications based on USN such as asset tracking, health monitoring, structure health monitoring, environment monitoring, surveillance, supply chain management, military applications, intruder detection, fire detection, building administration, etc.

A mobile device is attached to a target asset to track the asset. Asset tracking can be divided into two categories: indoor and outdoor. One of the outdoor asset tracking is a vehicle tracking. To track the vehicle, we can use two location methods: GPS and CDMA (Hightower, 2001). A GPS receiver is attached to the vehicle and a position received from GPS is sent to a monitoring center. The method has the location accuracy over 10m. However, it can't be used inside of a building. For CDMA, a driver's mobile phone sends the signal to MPC (Mobile Positioning Center) and MPC calculates the position of the mobile phone as the location of the vehicle. The calculated position is sent to a monitoring center. The method has the location accuracy over 300m. The monitoring center captures the positions of all vehicles, displays them on the screen, and provides useful information to the drivers. This service has been applied for LBS (Location Based Services), CNS (Car Navigation Systems), etc.

The target of indoor asset tracking is a moving object existed inside of a building. It might be a person or a valuable asset. The outdoor asset tracking does not require high location accuracy, while indoor asset tracking requires high location accuracy. The indoor service might require the location accuracy within 3 or 5m. If the accuracy is over 3 or 5m, the monitoring center can determine that an asset is located at different wrong room.

This work was supported by the IT R&D program of MIC/IITA. [2007-S-001-01, Development of Distributed Sensor Network System]

In the paper, we want to design and implement an efficient USN middleware that is high reliable for indoor asset tracking.

Contributions In this paper, we design and implement USN middleware and apply it for asset tracking. To implement the middleware, we address the requirements of the middleware and the asset tracking. From the requirement, we discuss how to implement the middleware as well as asset tracking and what functions are required for them. They are helpful for users to manage their business effectively related to mobile assets.

The rest of the paper is organized as follows. Section 2 presents the requirements of middleware. Section 3 presents the system specification of asset tracking. Section 4 presents the USN middleware. Finally, Section 5 concludes the paper.

2. MIDDLEWARE REQUIREMENT

To support a sensor network, the middleware provide below functionality (Romer, 2002; Kim, 2006).

USN middleware transmits sensing data acquired from the sensor network to the application. To accomplish this role, USN middleware must support a seamless connection between the sensor network and the application by providing openness and interoperability in order to integrate and mange various services.

USN middleware can provide an abstraction for specifications and functions of sensor networks. An application may need to gather different types of data from several heterogeneous sensor networks. As the USN middleware provides an abstraction function to applications, each application can connect to heterogeneous sensor networks without understanding their physical characteristics.

USN middleware can provide filtering, storing and managing data gathered from sensor networks. Data produced by sensor nodes may contain several errors due to resource limitations or environmental factors. Therefore, these errors must be filtered in the data gathering phase.

USN middleware can provide several analysis functions with sensing data. And USN middleware provides fusion functions that produce new information by fusing sensing data from several areas and analysis functions that mine new information from sensed data.

We implement USN middleware that is called MIDSTA (Middleware for Distributed Sensor network Applications) following the requirements and apply MIDSTA for an asset tracking application. The architecture consists of several components which include their own functions. Each component exposes its own functional definitions, methods definitions and attributes definitions. Componentization enhances reusability, easy maintenance, and understanding about the architecture of USN middleware.

3. SYSTEM SPECIFICATION

To design and implement USN middleware, we analyze an operating environment in real world. The system specifications are listed below (Rajendran, 2005).

3.1 Sensor Network

1. A sensor network consists of moderate number of sensor assets to be tracked.
2. There are two types of sensor nodes: fixed nodes and mobile nodes. Mobile nodes are attached to mobile assets and fixed nodes are attached on a wall.
3. The mobile nodes' positions are changed, while the fixed nodes are not moved.
4. The mobile nodes' positions are calculated, while the fixed nodes know their own position.
5. The mobile nodes are operated by a battery, while the fixed nodes are operated by a wired power.
6. The mobile nodes attaches on the mobile assets very strongly.
7. The nodes send the coordinate and the time.

3.2 Middleware

1. A middleware connects to multiple sensor networks because a building may have multiple stories.
2. There are two types of data from the sensor network: asynchronous data and synchronous data.
3. The data produced by the sensor network has sent continuously. Middleware acquires, stores, processes the continuous stream data.
4. Middleware manages location data of assets.
5. Middleware processes a spatio-temporal query for retrieving stored sensing data.
6. Middleware traces the mobile assets.
7. Middleware detects that an abnormal condition has occurred.
8. Middleware provides API that a user can use the middleware to acquire useful information from middleware.
9. Middleware provides a communication method between middleware and an asset tracking application.

4. MIDSTA IMPLEMENTATION

This section presents overall system architecture to satisfy the above requirements of DSN middleware system. Our DSN middleware architecture is composed of multi-tier (Kim, 2006). The multi-tier is physically divided into three parts: (1) information acquisition/collection part, (2) information analysis part, and (3) information service part. As shown in Figure 1, Network Access Management Component (NAMC) plays a role of information acquisition and collection.

Data & Query Management Component (DQMC) and Context Aware Management Component (CAMC) play a role in information analysis. Service Management Component (SVMC) plays a role in information service.

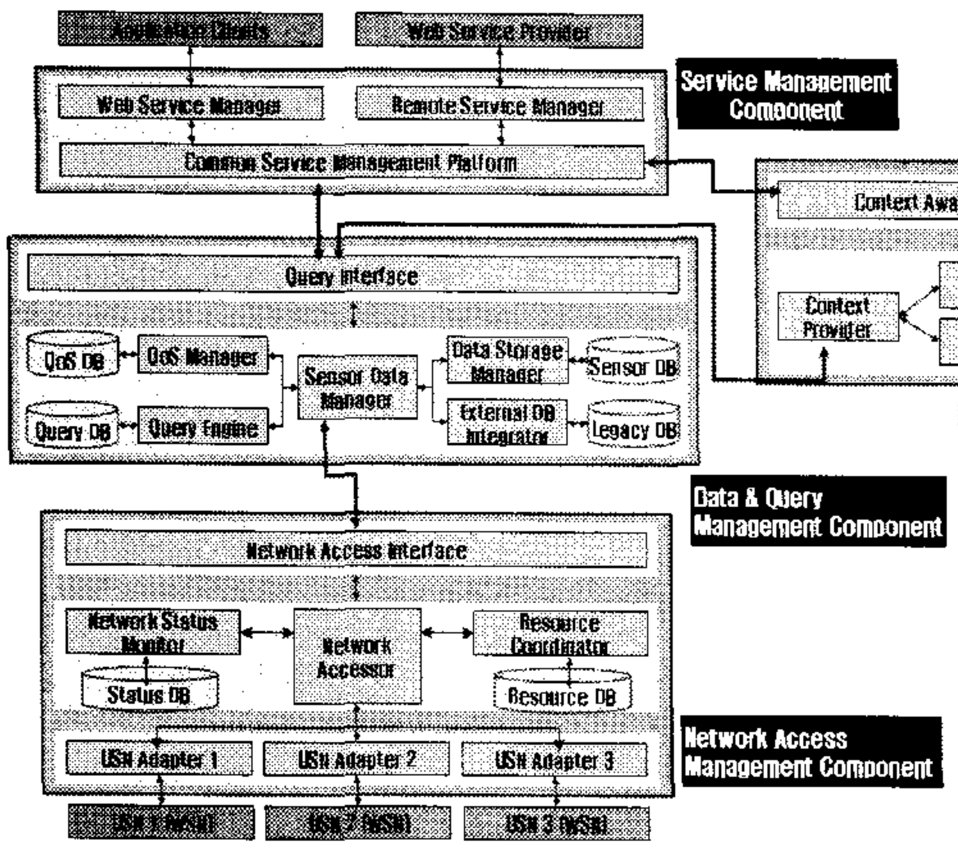

Figure 1. System architecture of MIDSTA

When many complicated queries are requested at the same time, this architecture can assure efficiency because it can separately process query types on information retrieval or information reasoning. In concrete, a query type on information retrieval is processed at the DQMC and a query type on information reasoning is processed at the CAMC. When handling service requests, DQMC and CAMC first process the requests using their own sensor database, legacy database, and context database. If the requests require newly sensing information, they activate the NAMC to get an access to the sensor networks infrastructure. NAMC efficiently acquires newly sensed information.

4.1 Network Access Management Component

This component plays a key role in efficient management of various kinds of sensor networks infrastructures. This component is consisted of USN adapter, device manager, health monitor, resource coordinator, and device manager interface. USN adapters access various kinds of USN infrastructures, device manager makes an abstraction for the USN adapters, health monitor periodically checks the status of sensor networks and nodes, resource coordinator efficiently allocates resources of sensor networks and nodes on user requests, and device manager interface interprets requests of Data & Query management component.

In the first phase, we developed a connection establishment module to a sensor network, a network configuration management module, an information retrieval module that acquires sensing data from a sensor network, protocol transfer module that coverts user level requests into network level commands, and asynchronous interface that is possible to send or receive irregular request and response. NAMC supports two types of data structure: binary format and XML format. Figure 2 shows the running example of NAMC. NAMC established a connection to a sensor network work and received database information of the network.

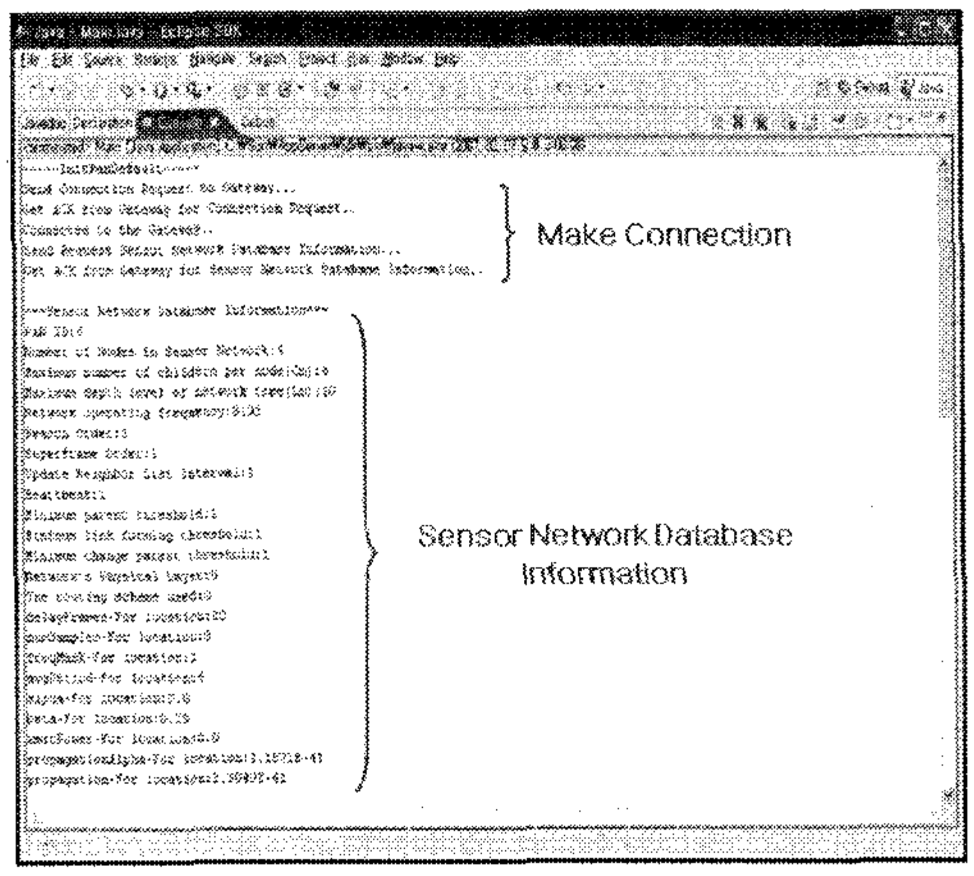

Figure 2. An example of NAMC

4.2 Data & Query Management Component

This component plays a key role in efficient analysis of sensing information for complicated user requests. This component is consisted of query interface, QoS manager, query engine, data storage manager, external DB integrator, and sensor data manager. Query interface receives and understands user requests, QoS manager processes user QoS requirements, query engine actually processes various kinds of complicated user requests, data storage manager store the acquired data to the secondary memory, external DB integrator integrates several external legacy databases, and sensor data manager wholly schedules and manages user requests. This component meets requirements of 'QoS for DSN service', 'Automatic Update of Node S/W', 'Location Determination of Sensor Node', 'Sensor Data Management', 'Service Request Processing', and 'Integration of Legacy DB'.

In the year, we developed a query analyzer module, a query executer module, and a storage manager module. DQMC can manage the position with time and geographic coordinate that moving objects produce and support a spatio-temporal query such as "Search assets located at room 412 between 10am and 11am". They are very helpful for user to retrieve history data of mobile assets with time and place as well as to make any valuable decision such as an optimal number of assets at a specific area. Figure 3 shows the running example of DQMC.

4.3 Context Aware Management Component

This component plays a key role in efficient generation of new context-aware information. This component is consisted of context-aware manager interface, context data manager, and data mining manager. Context-aware manager receives and analyzes context-aware requests from user, context data manager stores and implements rules that can be extracted from requests using a rule engine, and data mining manager predicts and reasons new context-aware information by fusing all kinds of existing information. This component meets the requirement of 'Sensor Data Mining'.

In the year, we developed rule manager module, a rule processor module, a service manager module, and a repository of rule. It is applied for extracting a specific condition while the system is running. CAMC defines a rule consisted of event, condition and action. If a condition is satisfied, then an event will be occurred. Because an event is related to an action, CAMC returns the action to a user. If the battery status of a node turns into a low status, CAMC detects the low battery status and reports the status to a user. Figure 4 shows the running example of CAMC.

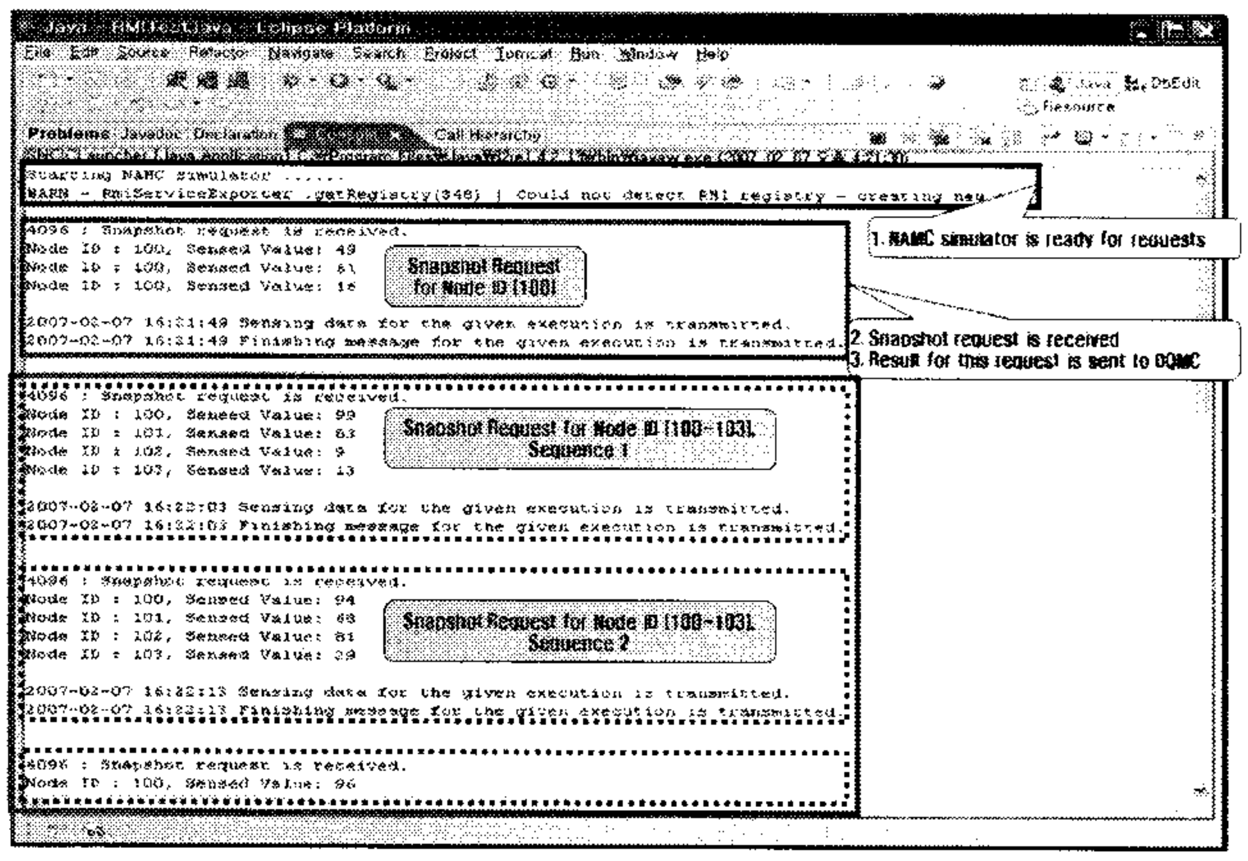

Figure 3. Running example of DQMC

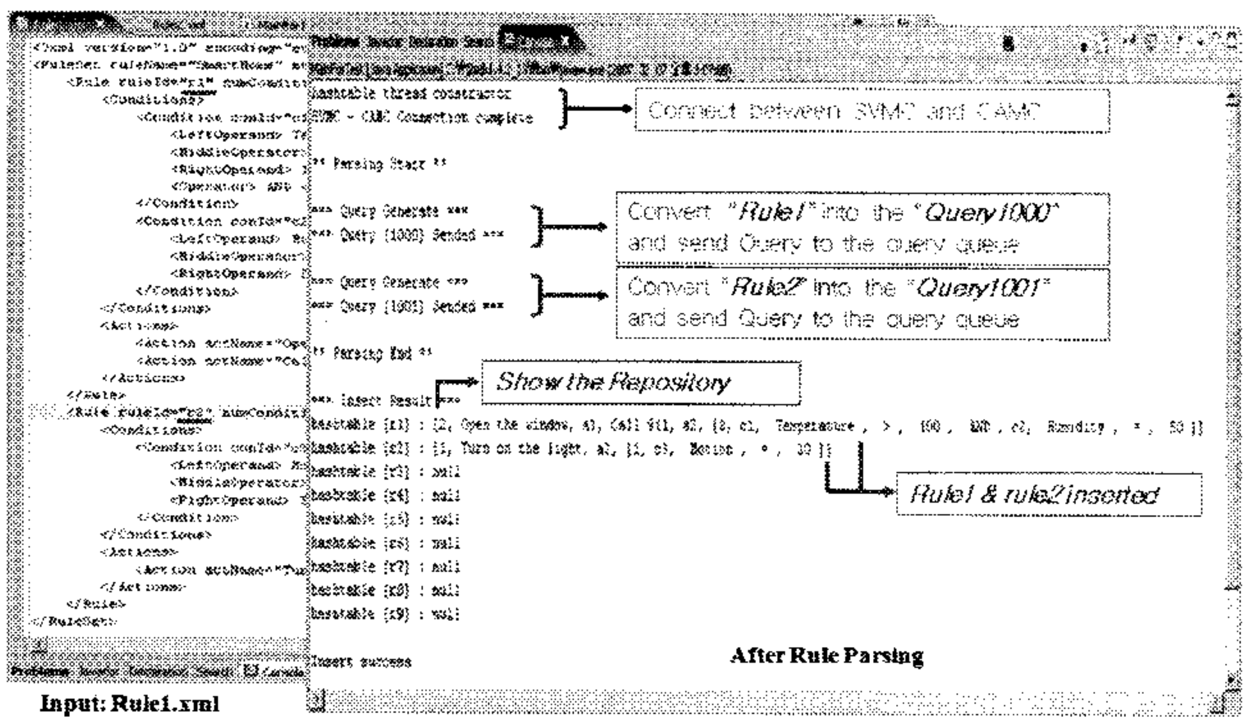

Figure 4. An example of CAMC

4.4 Service Management Component

This component plays a key role in the interface between MIDSTA kernel and application services. This component is consisted of web service manager, remote service manager, and common service management platform. Web service manager mainly provides standard mechanism for accessing data collected and managed by MIDSTA. Remote service manager invokes external services or delivers messages to remote service providers whenever pre-defined event are happened. Common service management platform manages and maintains the USN application services efficiently. This component meets a requirement of 'USN service management'.

5. CONCLUSION

This paper presents the features of USN middleware, MIDSTA, and the requirements of an asset tracking application based on USN. MIDSTA supports the development, maintenance, and execution of sensor network following by requirements of applications. MIDSTA might be considered as a software infrastructure that glues together sensor networks and applications. MIDSTA has been a key technology in supporting distributed sensor network applications by providing commonly used network access mechanisms, data acquisition mechanisms, query processing mechanisms, and context awareness mechanisms.

MIDSTA has special features related to the location data produced by mobile assets for an asset tracking application. They are storing location data by using 3D indexing method, retrieving them by using spatio-temporal query, .making trace of an asset, and retrieving the history data of an asset. They are indispensable mechanisms to effectively manage the position of a mobile asset.

We are planning to extend the functionality of MIDSTA. MIDSTA exposes some synchronous functions, but they ought to be changed into asynchronous functions to overcome the latency of a sensor network. It had better multiple query processing to adapt multiple users and multiple sensor networks.

Acknowledgements

This study was one of the results that the ETRI-Motorola cooperation project produced. The authors would like to thank Matt Perkins, Loren J. Rittle, Byung Y. Sung, and other Motorola members for their valuable reviews and comments.

REFERENCES

Hightower J., et el, 2001. Location Systems for Ubiquitous Computing, *IEEE Computer*, Aug, pp.57-66

Romer K., et el, 2002. Middleware Challenges for Wireless Sensor Networks, *ACM Mobile Computing and Communication Review*, Oct

Rajendran N., et el., 2005. WATS-SN: A wireless Asset Tracking System using Sensor Networks. In: *Proceedings of IEEE International Conference On Personal Wireless Communications*, New Delhi, India, pp.237-243

Kim K.S., et el, 2006. MIDSTA: Middleware for Distributed Sensor network Applications, In: Proceedings of *International Ubiquitous Convergence Conference*, Jeju, Korea, pp359-362