

# Naming Service Using an Improved Meta-table for Migration of Mobile Agent in Sensor Network Environment

Yon Sik Lee and Kwang Woo Nam

<sup>1</sup>*Department of Computer Information Engineering, Kunsan National University  
Kunsan, Cheonbuk, 573-701, Korea*

## **Abstract**

*For the activeness and autonomy of a sensor network, a mobile agent's role that enables a code itself to move between sensor nodes or migrate to a server and carry out a given task is important. Since such a migration method of the mobile agent is a factor that affects the overall performance of the entire distributed system, it is necessary to find efficient migration methods of the mobile agent within the sensor network and to collect and store data related to various components (server, sink and sensor node) of the sensor network, thereby providing consistent naming services. For this, in this paper, we design an improved MetaTable that is divided into MetaData where information on the sensor data server is stored and SubMetaData where various types of information on sink nodes and data on sensor nodes connected with the sink nodes is stored. Also, we propose the efficient method of naming service on the sensor network by using the proposed MetaTable. The migration of mobile agent depends on the information of MetaData and SubMetaData referred to, and the reliability of the migration information is determined by mutual cooperation between a naming agent and each sensor server. And, we implement the migration method of mobile agent by using the naming service via meta\_data generation method, migration list and priority. For the verification of validity and guarantee of applicability of the proposed method, we performed experiments on the migrations of the mobile agent and active rule execution through the mutual relations between agents by using meta\_data on the components of the sensor network.*

**Keywords:** *Sensor Network Middleware, Naming Service, Meta\_data, Migration of Mobile Agent*

## **1. Introduction**

An agent middleware system facilitates information sharing and integration in a distributed system environment. The system is composed of message communication, agent management and execution environment to enhance the usability of agent-based applications and mutual cooperation among agents [3, 4, 5]. In the distributed environment, the mobile agent technology reduces network overload and waiting time, while enabling efficient searching and transmission of information. Also, due to its high network adaptability, the technology is more reliable than other systems in case of network error that various applications are possible [1, 2, 4, 6]. In the sensor network environment, an agent system has been used as an alternative solution for efficient data acquisition and network problems such as network delay and bandwidth overhead since a mobile agent executes the given tasks as an agent code itself migrates among sensor nodes or servers. In this case, the migration method of an agent becomes a major factor that affects the overall performance of the distributed system [7, 8,

11, 13, 14]. Thus, a meta\_table that saves, controls, and provides meta\_data of each sensor data server, sink and sensor node and an agent system that can offer such service are necessary for the mobile agent's efficient migration and facilitating the sensing information [4, 5, 6, 13, 17]. Furthermore, problems regarding receiving new sensor servers and adding the meta\_table suitable for a new data format are closely related with the performance of the agent system in the sensor network environment [4, 6, 13, 17].

Considering these issues, a new meta\_table and naming service is essential in today's sensor network environment to efficiently utilize existing meta\_data of sensor data servers and related to sink and sensor nodes for sensor network applications [5, 6, 7, 10]. In this paper, we presents design and implementation of the new meta\_table(MetaTable) that includes locational information and diverse characteristics of sensor data servers, sink nodes and sensor nodes. Then, we propose the migration method of the mobile agent based on the naming method that uses the naming agent's meta\_data. Moreover, to offer a possibility of constructing more efficient sensor network application environment, the method of generating, employing, and updating meta\_data associated with the mutual relationship between each agent and sensor data server is proposed.

The remaining chapters of this paper are organized as follows. Chapter 2 presents the naming agent and proposes design of new extensible meta\_table(MetaTable) that stores information related to the sensor data servers, sink nodes and sensor nodes. Chapter 3 describes the naming service and the migration process of the mobile agent based on newly designed meta\_table. Chapter 4 presents the result of mobile agent migration and active rule execution from the communication process of agent system related with network environment and experiments. Finally, chapter 5 makes brief conclusion of the research and presents further research suggestions.

## **2. Naming Agent and Meta\_table**

### **2.1. Naming Agent**

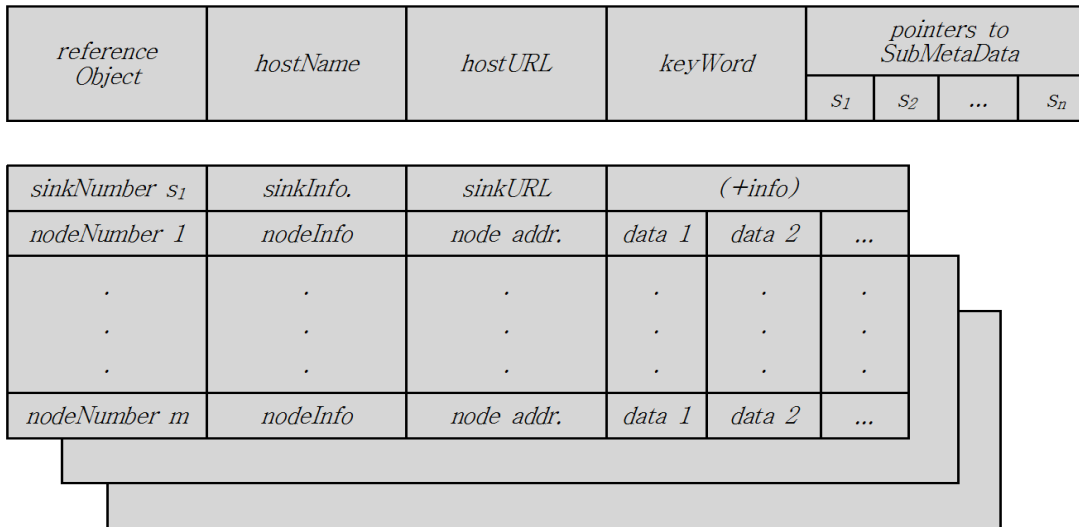
The naming agent collects and manages the information related to System Monitoring Agent, Client Push Agent, Server Push Agent and Sensor Data Server registered in each connected name space and provides the integrated functions of the naming services. Also, the agent maintains and manages the object references by assigning the thread to the naming services [2, 4, 6, 9, 10]. The naming service is offered in the format of thread for rapid response on simultaneous requests. Once a client's request is received, the thread examines whether the Server Push Agent, Sensor Data Server, or Client Push Agent is registered or not, and then returns the relevant object referrer to the client [5, 7, 10]. To manage meta\_data, the naming agent is composed of several classes corresponding to various functions. The relationship among these classes is shown in Figure 1.

In this paper, we design and implement the naming agent based on RMI using the advantage of CORBA obtained from applying J2EE model based RMI-IIOP(Internet Inter-ORB Protocol). Therefore, in new sensor network environment, the proposed method offers faster registration, release and retrieval compared to existing methods.



[/kunsan.ac.kr/engineering/cie/calab01](http://kunsan.ac.kr/engineering/cie/calab01)). However, in this paper, the system is implemented in a way that the address system is mainly used to store each node's address information in the meta\_table.

In this paper, we design and implement new MetaTable which can search the sink and sensor nodes on the sensor network using the proposed address system. Also, this new MetaTable is extensible that it can be adjusted to match the number of nodes. As shown in Figure 2, the MetaTable is divided into the MetaData where information of the Sensor Data Server is stored and the SubMetaData where information of the sink nodes and sensor nodes connected to the sink nodes is stored.



**Figure 2. The Proposed Meta\_table (MetaTable) for Sensor Network Application Environment**

The *MetaData* has the *referenceObject* which registers a remote object of Sensor Data Server, the *hostName* and *hostURL* which indicate the name and address of the corresponding server, the *keyWord* used as a search keyword, and a pointer for the *SubMetaData* that has information of the connected sink nodes. The *SubMetaData* is constructed in a format that can have multiple tables to store information of each sink node individually. The *SubMetaData*, a table to store information of sink and sensor nodes, is composed of *sinkNumber* which is the identification number of a sink node, *sinkInfo* which stores characteristic information and *sinkURL* which indicates the address. To store more specific charactersitic information, new fields(+*info*) can be added to the *SubMetaData*. Also, the *SubMetaData* stores the meta\_data related to the sensor node corresponding to the sink node as the same format in the same table. Various active rule execution through real-time sensor data acquisition and the migration of Mobile Agent is possible using the addresses of the sink node(*sinkURL*) and sensor node(*node addr.*).

The Naming Agent manages each agent's location information, registers agents' names and initializes each related field of the *MetaData*. While performing these tasks, it must prohibit the duplicated registration of names to guarantee the reliability of the location information. In case of the name conflict, the existing agent is deleted and registered again as a new one. On

the registration request, the Naming Agent generates the threads which are mutually synchronized and enables read/write protection. Each agent has its own unique name, is registered in each classified meta\_table and is allowed to interact with other agents. In order to generate the *MetaTable*, we apply the algorithm shown in Figure 3.

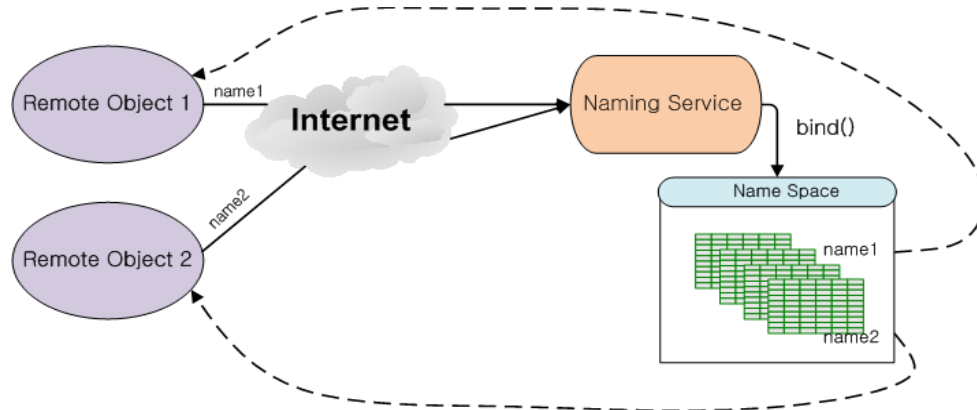
```
MetaTable Generation  
Input : referenceObject, hostName, hostURL, keyWord, pointer to SubMetaData,  
sinkNumber, sinkInfo, sinkURL, (+info)  
Output : MetaTable(MetaData, SubMetaData)  
Process :  
Registry reg = LocateRegistry.createRegistry(port);  
Server Push Agent Registration Request;  
IF (Name == MetaData.Name)  
    THEN  
        Return Re_registrationRequest;  
    ELSE  
        Server referenceObject = (Server)Naming.lookup(hostURL + hostName);  
        LocateRegistry.getRegistry().bind(hostName, referenceObject);  
        MetaData.SIZE();  
        MetaData.INSERT(referenceObject, hostName, hostURL, keyWord);  
        IF (Sensor Data Server == True)  
            THEN  
                GET sinkNumber();  
                SubMetaData.Creation(++sinkNumber);  
                MetaData.INSERT(pointer to SubMetaData);  
                SubMetaData.INSERT(sinkNumber, sinkInfo, sinkURL, (+info));  
                Return MetaData, SubMetaData;  
            ELSE  
                Return MetaData;
```

**Figure 3. MetaTable Generation Algorithm**

### 3. Naming Service and Migration of a Mobile Agent

#### 3.1. Naming Service

The Client Push Agent sends mobile agents to the Server Push Agent or Sensor Data Server using an object referrer. RMI framework provides a simple naming registry to obtain the object referrer. The Naming Agent generates a thread to efficiently manage the *MetaTable* that is extended from the naming registry and responds to various requests. On receiving the user's request, the System Monitoring Agent transmits the return address as a parameter to the Naming Agent to notify the Client Push Agent which collects the user's identity and request. The Naming Agent searches the agent name or keyword in the classified *MetaTable* to find the server list and determines the priority of the migration, if necessary. Figure 4 shows the remote object registration mechanism through naming service.



**Figure 4. Remote Object Registration through Naming Service**

The naming service registers the Server Push Agent or Sensor Data Server on the name space following the order of their requests. The *MetaTable*, *MetaData* and *SubMetaData* storage within the name space, has an extensible structure to accommodate new server or data. The naming service stores and manages information of sink nodes and related sensor nodes using the *SubMetaData*. The client(Client Push Agent) acquires object referrer from the naming space using the naming service and obtains the server list through retrieval using the *hostName*, *hostURL* or *keyWord* in *MetaData*. The following Figure 5 shows an algorithm that generates the server list using the *keyWord* for the migration of the mobile agent.

```

Server List Generation
Input : keyWord
Output : Server List
Process :
NamingSearch (keyWord);
GET MetaDatae;
WHILE(MetaData.HasNext())
    IF (keyWord == MetaData.keyWord)
        THEN
            Server List[][] = hostName, hostURL, keyWord, sinkNumber, sinkURL,
                               sinkInfo;
            Return Server List[][];
        ELSE
            Return Null;
    
```

**Figure 5. Server List Generation Algorithm for the Migration of a Mobile Agent**

### 3.2. Migration of a Mobile Agent

In the mobile agent based sensor network environment, the mobile agent can migrate to the sensor node as well as sink node and selectively collect appropriate data. The mobile agent based sensor network environment has the following advantages. Firstly, the mobile agent can adapt quickly to variation in new application environment as the mobile agent can change data acquisition method and cycle due to dynamic changes. Secondly, it can reduce the network load by eliminating unnecessary and redundant data. And lastly, data acquisition in

converted format that is appropriate for corresponding program is possible without any data conversion or normalization processes [6, 10, 11, 13]. The migration of the mobile agent is executed based on cooperation among agents and provides the reliability of information retrieval and rule execution. The object information registered on the *MetaTable* provides location transparency that enables communication between agents or connection of remote systems, and it supports the mobile agent to execute active rules. Such migration scenario is as follows: Firstly, when the search keyword entered via the client browser is transmitted to the Client Push Agent, it acquires the object referrers of the mobile agent and itself from the Naming Agent. Then, it acquires the migration list which consists of the Sensor Data Server(including sink and sensor nodes) or the Server Push Agent's object referrer from the Naming Agent by comparing the mobile agent's object referrer and search keyword with information of the *MetaTable*. Then, the mobile agent migrates and executes the loaded active rule following the application's request. The execution processes are classified as follows.

**3.2.1. Sequential migration following the order in migration list:** The mobile agent migrates to the Server Push Agent or Sensor Data Server(including sink and sensor nodes) following the order in the migration list, searches the requested data and returns the searched data to the Client Push Agent. Then the agent migrates to next server listed in the migration list. In the sensor network, the mobile agent migrates to the sensor node and acquires current sensor data. If the agent has rules to execute on the sensor nodes, it evaluates the corresponding condition and executes the active rules. After acquiring the data, the mobile agent searches adjacent node following the order in the migration list, repeats migration to other nodes and returns to the sink node after completing migration to all nodes.(Experimental results: Figure 7)

**3.2.2. Migration following the priority:** The agent client determines the priorities of migration targets in the migration list, selects the node with the optimal migration path and migrates the mobile agent to the corresponding node with the priorities of other nodes. The priorities of the migration targets can be adjusted by the loaded rule of the mobile agent. The migrated mobile agent acquires current sensor data, transmits it to the sink node, executes the loaded rule, and then migrates to the highest priority node following the given priorities.(Experimental results: Figure 8)

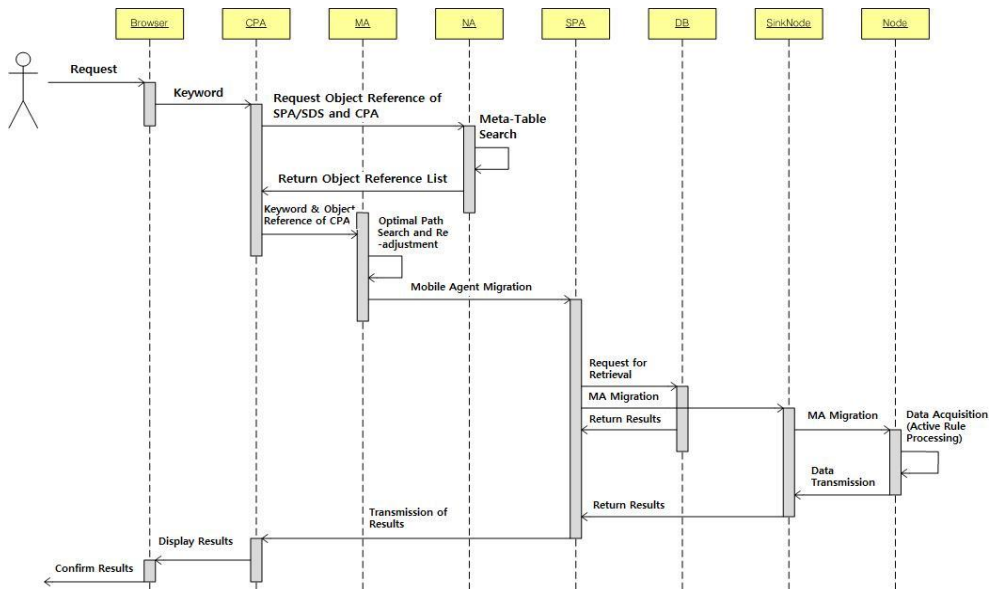
Both methods have the same process of the module execution or loaded rule execution in the Server Push Agent or Sensor Data Server(including sink and sensor nodes). Following the work\_flow of the module execution, the Server Push Agent or Sensor Data Server(including sink and sensor nodes) returns the retrieved results on database or transmits the executed results of the active rules on rule\_base to the Client Push Agent. The Client Push Agent filters the received result data and displays it on the user's browser. In this manner, the mobile agent in the middleware system on the sensor network environment operates in line with the operating system between the sensor nodes or sensor node and sink node, and it provides the data appropriate for the user's request.

## 4. Experiment and Evaluation

Figure 6 shows the communication process between each agent including the migration of the mobile agent and active rule execution, for the experiment. This process is based on the role of Naming Agent that uses the *MetaTable* proposed in this paper.

Initially, the *ClientPushAgentImpl* class receives a keyword as a parameter and executes the *tableSerach(String keyword)* method. Then, it transmits the keyword to the

*NamingSearch(String keyword)* method of the *NamingAgent* class and requests the corresponding meta\_data. The Naming Agent compares the meta\_data with the data in the *MetaTable*, searches the corresponding server list and returns the list in the table format(*String[][]*). The Client Push Agent transmits the server list, keyword and its own information(if necessary, including information for migration priority selection) to the mobile agent. The mobile agent configures the migration sequence or alternative route following the order of priority and migrates to the corresponding server. The Server Push Agent or Sensor Data Server(including sink and sensor nodes) which received the keyword from the mobile agent and the request from the Client Push Agent executes the corresponding operation module or searches the DB using the *access(String keyword)* method of the *DBAccess* class (if the middleware system links with the active rule system, the active rules are executed according to the Event-Condition-Action rule coupling mode). After the search (or rule execution), the obtained data is stored in the *List<String>* format and transmitted via the *setTitle()* and *setContent()* methods of the *ClientPushAgent* class. Moreover, when the mobile agent arrives at the Sensor Data Server or sensor node, the agent can execute the loaded active rule as well as acquire data. Thus, in the future, various sensor network application using dynamically acquired sensor information, circumstances information and active rule can be developed. For this, separate communication and execution methods are required for the operation at sink node or sensor node connected to the Sensor Data Server.

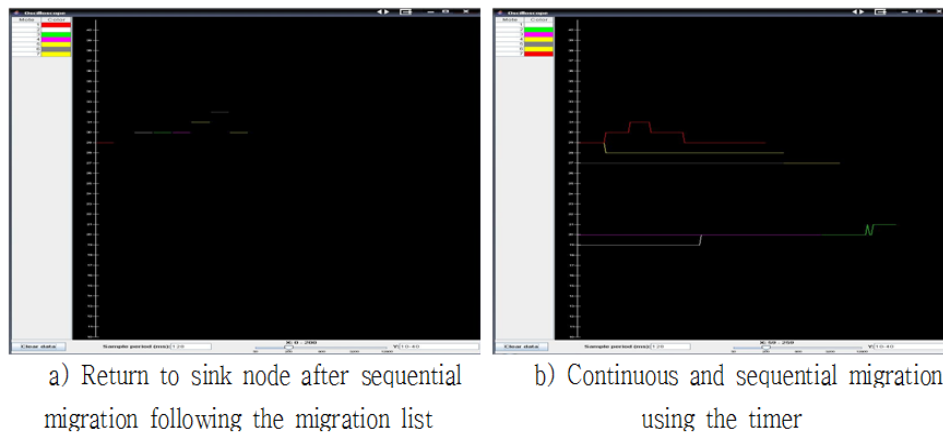


**Figure 6. Communication Process between each Agent in a Sensor Network**

The experiment of the mobile agent middleware system using the proposed *MetaTable* was performed with Hmote2420 sensor nodes. The MCU of Hmote2420 was MSP430F1611 from TI Co. and RF chip was CC2420. The communication frequency band was 2405MHZ and the RF power was [Output Power 0 dBm, Current Consumption 17.4mA]. The operating System used was TinyOS-2.x and Cygwin tool was used for the development of TinyOS on Windows. The Experimental data is temperature and the total of 8 nodes, including the sink node, were used for the experiment.

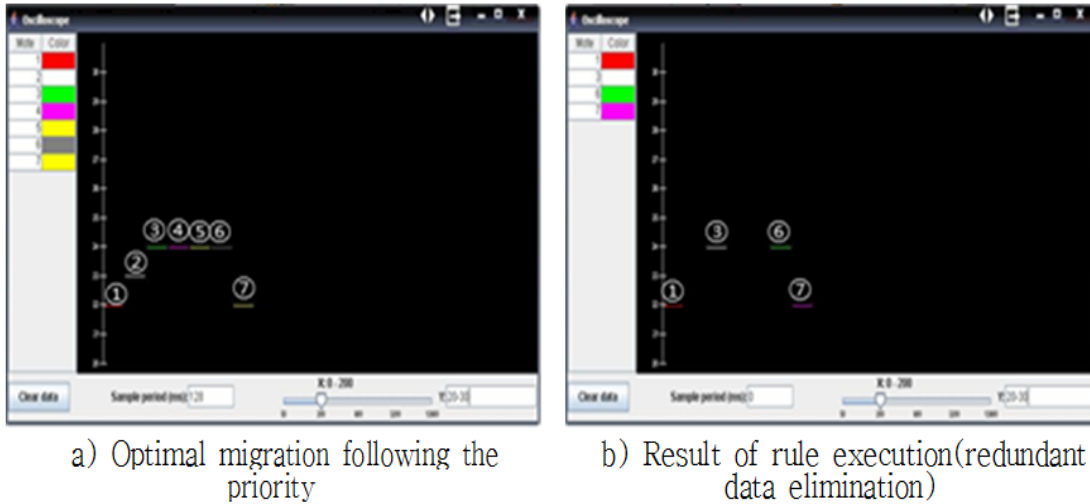


Figure 7 shows the experiment results of the sequential migration of the mobile agent following the order in the migration list. a) shows the result from the experiment where the mobile agent migrates to the Sensor Data Server following the order in the migration list and acquires and transmits the sensor data using the pointers to the sink nodes and the *nodeInfo* in the *MetaData*. If there are loaded active rules to be executed on the acquisition of data, the mobile agent executes the rules after evaluating the corresponding conditions. After the data acquisition and rule execution, the mobile agent searches adjacent node following the order in the migration list, repeats migration to other nodes and returns to the sink node after completing migration to all nodes. b) shows the result of the experiment where the mobile agent continuously acquires and transmits the sensing data using the timer. All other experimental environment is same as the environment of a).



**Figure 4. Sequential Migration of a Mobile Agent following the Order in Migration List**

Figure 8 shows the experimental results of the optimal migration following the priority and rule execution. a) shows the result of the experiment where the mobile agent selects the optimal Sensor Data Server from of the acquired object referrers using the priorities and acquires and transmits the sensor data using the pointers to the sink nodes and the *nodeInfo* in the *MetaData*. After the acquisition of all sensor's data connected with the Sensor Data Server, the mobile agent selects the next optimal Sensor Data Server from the acquired object referrers using the priorities again, and then the agent migrates to the server and repeats the same processes. b) shows the experimental result of the rule execution that eliminates the redundant sensing data at the corresponding sensor node using the *Redundant Data Elimination Algorithm* [12]. In our experiment, we defined the elimination of sensor data as the difference between the sensing data(*localData*) with two previous sensed data(*pData*, *ppData*) is less than the given threshold(*THRESHOLD*). The *Redundant Data Elimination Algorithm* can be executed more efficiently by the optimization of the *THRESHOLD* and *MAX\_NOSENDING* values according to the distribution degree of sensors, variation of sensor data and characteristics of the corresponding area [12, 14].



**Figure 8. Optimal Migration following the Priority and Rule Execution**

Author names and affiliations are to be centered beneath the title and printed in Times New Roman 12-point, non-boldface type. Multiple authors may be shown in a two or three-column format, with their affiliations below their respective names. Affiliations are centered below each author name, italicized, not bold. Include e-mail addresses if possible. Follow the author information by two blank lines before main text.

## 5. Conclusion and Further Researches

The multi-agent system is used for more efficient data acquisition and resolving network delay and bandwidth overhead in the network system. The naming agent, a component of the multi-agent system, manages and provides the meta\_data such as name, location information, keyword and registered implemented object of each agent for consistent naming service. However, in the sensor network environment, new meta\_data management method and naming service are required to efficiently use the sensor data for sensor data application. Thus, in this paper, we designed and implemented new meta\_table(*MetaTable*) that includes the location information and various characteristics of the Sensor Data Servers, sink and sensor nodes in the sensor network environment as well as existing meta\_data. The implemented meta\_table(*MetaTable*) has the *MetaData* section that stores the Sensor Data Server information and the *SubMetaData* section that stores the information of sink and sensor nodes connected to the sink nodes. Thus, it provides extensibility for appropriate naming and addressing methods. Moreover, the paper also presented the implementation of rule execution methods of the mobile agent applying the naming method based on RMI that used the sensor network related meta\_data and proposed the possibility of constructing efficient sensor network application environment.

In this paper, for faster registration, release, retrieval than traditional methods in new sensor network environment, we designed and implemented the naming agent by applying J2EE model based on RMI-IIOP(Internet Inter-ORB Protocol) technique with the advantage of CORBA.

In the future, further research related to the naming system and name resolve method of sink and sensor nodes is necessary. Moreover, new meta\_data expression, storage, management and application methods are required as data generalization method to store and manage sensor data with heterogeneous data types and protocol is needed.

## Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 2009-0074891 and No. 2009-0067958).

*This paper is a revised and expanded version of a paper entitled [Design and Implementation of Meta-table for Naming Service in Sensor Network Environment] presented at [CCA2012, Seoul Korea, March 30, 2012].*

## References

- [1] J. G. Hwang and C. S. Pyo, Journal of KIEES, vol. 19, no. 6, (2008), pp. 51.
- [2] S. I. Choi, Proceedings of KISS, vol. 33, no. 2B, (2006), pp. 138.
- [3] C. L. Fok, G. C. Roman and C. Lu, 4th International Conference on Information Processing in Sensor Networks (IPSN'05), (2005), pp. 382.
- [4] K. J. Kim, H. Ko and Y. S. Lee, KSCI Journal, vol. 12, no. 3, (2007), pp. 165.
- [5] E. H. Jung, KSCI Journal, vol. 12, no. 4, (2007), pp. 95.
- [6] Y. S. Lee and K. J. Kim, Proc. of 4th International Conference on Networked Computing and Advanced Information Management(NCM2008), vol. 2, (2008).
- [7] M. S. Shin and M. J. Lee, Journal of KSII, vol. 10, no. 1, (2009), pp. 19.
- [8] C. Konstantopoulos, IEEE Transactions on Knowledge and Data Engineering, vol. 22, (2010), pp. 1679.
- [9] J. S. Lee, Y. C. Choi and Y. S. Lee, Proceedings of KIC2009, (2009), pp. 147.
- [10] Y. S. Lee and J. S. Lee, Proceedings of KIPS, vol. 17, no. 1, (2010), pp. 1070.
- [11] T. Heimfarth, 2010 24th IEEE International Conference on Advanced Information Networking and Applications, (2010), pp. 820.
- [12] Y. S. Lee and J. S. Lee, Journal of KSII, vol. 12, no. 3, (2011), pp. 27.
- [13] H. Y. Hu and Hua Hu, International Journal of Multimedia and Ubiquitous Engineering, vol. 5, no. 4, (2011), pp. 31.
- [14] A. Salatian, International Journal of Smart Home, vol. 4, no. 4, (2011), pp. 27.
- [15] M. Moshirpour, S. M. Sherif, A. Mousavi and B. H. Far, International Journal of Energy, Information and Communications, vol. 2, no. 4, (2011), pp. 101.
- [16] H. Chen, International Journal of Security and Its Applications, vol. 3, no. 2, (2009), pp. 21.
- [17] H. Y. Hu, Hua Hu, Y. Zhuang and L. Yang, International Journal of Software Engineering and Its Applications, vol. 2, no. 4, (2008), pp. 11.

## Authors



**Yonsik Lee**

He is currently a tenured professor at Department of Computer Information Engineering of Kunsan National University in Korea. He received his B.S. (1982) and M.S. (1984) degrees in Computer Science from Chonnam National University, Korea. He received the Ph. D. degree in Computer Application Engineering from Chonbuk National University, Korea, in 1994. His current research interests are in the area of sensor network middleware, active and mobile agent system, and computer application software development based on USN.



**Kwang Woo Nam**

He is currently an associate professor at Department of Computer Information Engineering of Kunsan National University in Korea. He received his B.S. (1995), M.S. (1997), and Ph. D. degree in Computer Science from Chungbuk National University, Korea. His current research interests are in the area of spatial and temporal databases, location-based services, sensor databases, and mobile convergence applications.