A Case Study on Message-Oriented Middleware for Heterogeneous Sensor Networks*

Sangim Ahn and Kiwon Chong

Department of Computing
Soongsil University
1 Sangdo-5Dong, Dongjak-Ku, Seoul, Korea 156-743
{siahn69, chong}@ssu.ac.kr

Abstract. Needs of interoperability are getting increased since there have been some issues in heterogeneous wireless sensor networks such as information exchange, network protocol conversion, and integrity among multi-vendor products. We propose message-oriented middleware to make use of an intelligent bridge for information exchange in heterogeneous constraints. It includes middleware architecture, a message format and message types, and rule handling. We present the results of a case study in details to do demonstrate the feasibility of our approach. These are composed of three parts: First, we build a scenario based on an information exchange between wireless sensor networks in home environment so that it shows roles of a bridge, handling sensed events. Second, we define a process of information exchange using UML sequence diagram. Third, we represent soap messages which contain methods and data defined in advance. We confirm effectiveness of information exchange mechanism with extensibility and flexibility through the case study.

Keywords. Intelligent Bridge, Middleware, Heterogeneous Wireless Sensor Networks

1 Introduction

Wireless sensor network is initiatively designed to detect events or phenomena, to collect and process data, and to transmit sensed information to interested users. According as home networks become a practical reality, wireless sensor network is applied to monitor and control environmental conditions at home. Needs of interoperability are getting increased since there have been some issues in heterogeneous wireless sensor networks such as information exchange, network protocol conversion, and integrity among multi-vendor products. The solution of these issues can be the most important part of the many competing standard researches in wireless sensor network [1][2].

We propose message-oriented middleware to make use of an intelligent bridge for information exchange in heterogeneous constraints. It includes middleware architecture, a message format and message types, and rule handling. The architecture makes an intelligent bridge through a home server in home networks, and the home

* This work was supported by the Soongsil University Research Fund.

server can help to exchange information among wireless sensor networks. The intelligent bridge carries out roles of a mediator between middlewares in heterogeneous wireless sensor networks. The rule handling is the most important function of the intelligent bridge which takes charge of possessing information, analyzing input data, searching rules, interpreting selected rule, and controlling statuses of all wireless sensor networks. A message format is composed of message type, destination identification, source identification, the sequence number of message, and contents. The message types are classified by six categories such as request, response, query, notify, control, and acknowledge. The message types are necessary when message elements are analyzed. We present the results of a case study in details to do demonstrate the feasibility of our approach. These are composed of three parts: (1) We build a scenario based on an information exchange between wireless sensor networks in home environment so that it shows the roles of a bridge which handles sensing events. (2) We define a process of information exchange using UML sequence diagram. (3) We represent soap messages which contain methods and data. We confirmed effectiveness of information exchange mechanism with extensibility and flexibility through the case study.

The rest of the paper is organized as follows: In Section 2, we discuss interoperable middleware architecture, including a message format and message types, and rule handling technique. In Section 3, we present the results of a feasibility study based on information exchange between heterogeneous wireless sensor networks. In Section 4, we describe related works with requirements on wireless sensor network middleware, the other bridge researches, and SOAP. Finally, in Section 5, we draw some conclusions and discuss research issues for future work.

2 Middleware Architecture with interoperability

2.1 Middleware Architecture

Middleware provides a mechanism through which application specific code can be injected and triggered inside the network, allowing energy efficiency in data dissemination and increasing the wireless sensor network performance and time life. Middleware also enables the generation and communication of high level tasks, as well as the coordination of such tasks among nodes, even if the nodes have heterogeneous features. In order to suit to the wireless sensor network resource constraint and fault prone, this middleware is designed to be robust and fault tolerant, keeping the messages exchanged as short as possible.

The ultimate objective of our approach is to overcome communication problem related to heterogeneous configuration items such as different network protocol, different embedded operation system and different manufacturers. To solve them through the use of standard protocol in the middleware layer, we propose middleware architecture with interoperability between heterogeneous wireless sensor networks by a mechanism of information exchange. This architecture makes an intelligent bridge using a home server in home networks, and the home server can help to exchange information among wireless sensor networks. A home server generally integrates home appliances, and manages various services of network, security, and remote control in home networks. In our approach, a home server performs the role of an

administrator to manage rules corresponding properly request and response messages that is passed from each sensor nodes, as well as the role of a gateway for information exchange.

The main internal components of middleware architecture are composed of the SOAP engine, a set of handlers, and user interface module. The SOAP engine acts as the main entry point into the SOAP module. It is responsible for coordinating the SOAP message's flow through the various handlers. Handlers are composed of a transport handler, a message build handler, a logic handler, and a rule handler. First, the transport handler carries out sending and receiving messages through networks. There are two main modules in the transport handler. One is Message Receiver (MR) which takes charge of analyzing request messages transmitted from sender and transferring a message to an actuator if it is necessary. The other is Message Sender (MS) which delivers events to a home server and responds to request messages. Second, the message build handler is building blocks inside the SOAP module. Third, the logic handler carries out processing functional logic. Finally, the rule handler exists only in a home server. It takes charge of possessing information about main roles and functions of each wireless sensor network, analyzing information that is transmitted and controlling wireless sensor networks according to rules.

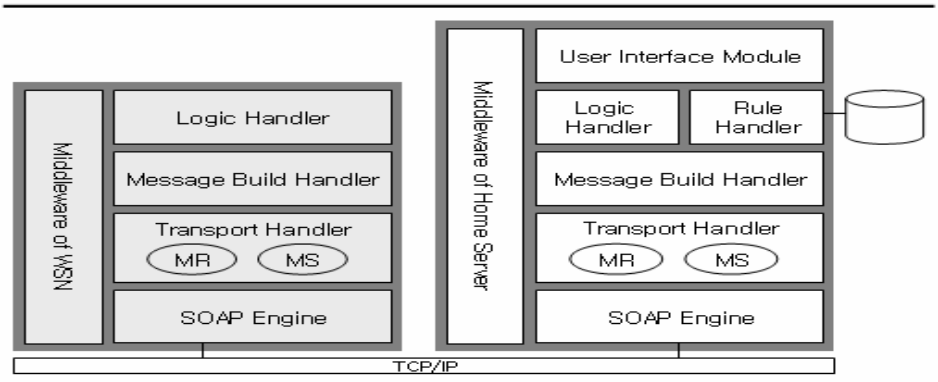


Figure 1. Middleware Architecture

2.2 A Message Format and Message Types

Information exchange of middleware architecture is composed of three factors: a message format, a message style, and a message transmission protocol. The message format is a standard form of messages. Elements of the message format consist of message type, destination identification, source identification, the sequence number of a message, and contents to exchange information between middleware of the home server and middlewares of wireless sensor networks. The message style is a way which information is given, we use a XML document. It allows users to create their own customized tags, definition, transmission, validation, and interpretation of data between applications. Many documents and data have been expressed by XML because they easily and dynamically converted to another type documents. Therefore, XML is very suitable to express information in the network environment participated

in various nodes. As the message transmission protocol, we use SOAP. SOAP is a lightweight XML-based messaging protocol to encode the information for request and response messages. There are a number of merits because SOAP is a text-based protocol using XML. First, it can be easily implemented. Second, Debugging is easy because of text forms that a person can read. Third, it can be used widely because compatibility is high. Hence, we can extend interoperable middleware architecture to web service using mobile equipment since the proposed architecture has adapted XML and SOAP with flexibility and extensity to exchange information.

Table 1. Elements of Message Format

Elements	Tags of Element	Description
Message type	<MessageType>	The type of a message
Destination ID	<To>	The target identification of a message
Source ID	<From>	The source identification of a message
No. of sequence	<MsgNo>	The sequence number of a message
Contents	<content>	The content of information exchange

The message type is classified by six categories such as request, response, query, notify, control, and acknowledge. The message type is necessary when message elements are analyzed. Request and response message type are related to information exchange among middlewares in wireless sensor networks directly. Query and notify message type are related to information exchange between middleware in wireless sensor network and a home server. Especially, an event message is one of notify messages types. The objective of Control and acknowledge message type is to control statuses of wireless sensor network according to rules established in advance. We make a number of methods as the following table 2. The meaning of Content is different according to message types [3][4].

Table 2. The Kinds of Message Type

Message type	Method	Description
queryWSN	queryWSNStatus	WSN Information Query
notifyWSN	notifyWSNStatus	WSN Information Notify
	notifyWSNEvent	WSN Event Notify
controlWSN	controlWSNTime	Set WSN Time
	controlWSNActuator	Set Actuator Status
	controlWSNSensor	Set Sensor Status
AckWSN	ackWSNStatus	Acknowledge of NotifyWSNStatus Message
	ackWSNEvent	Acknowledge of NotifyWSNEvent Message
	ackWSNTime	Acknowledge of NotifyWSNTime Message
RequestWSN	requestWSNxxx	Direct Message Delivery Request
ResponseWSN	responseWSNxxx	Direct Message Delivery Response

2.3 Rule Handling by an intelligent bridge

The intelligent bridge in a home server carries out the role of a mediator between middlewares in heterogeneous wireless sensor networks. The rule handler is the most important function of the intelligent bridge.

The rule handler takes charge of possessing information about main roles and functions of each wireless sensor networks. The information is used to make rules in rule repository. These rules are updated periodically. There are five main functions by the rule handler: (1) It analyzes input data which are sensed events from wireless sensor networks. (2) It searches rules corresponding with each event. (3) It carries out interpreting the selected rule. (4) It provides proper commands to middleware in wireless sensor network. (5) It also controls statuses of all wireless sensor networks to sustain home environment.

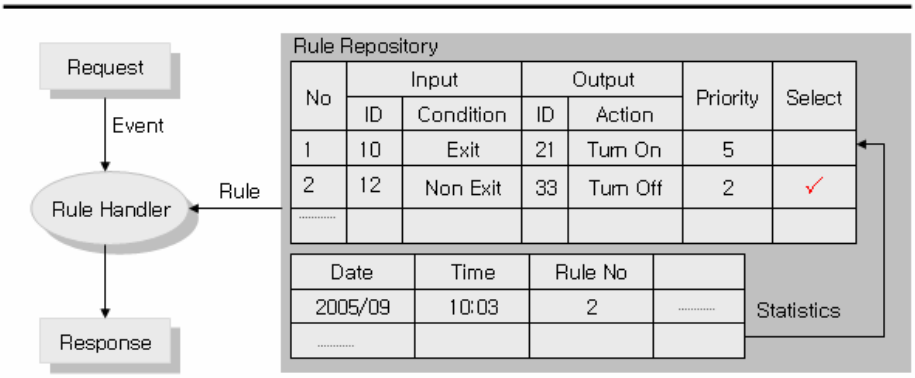


Figure 2. A Process of the Rule Handler

3 A Case Study

3.1 A Scenario for Indoor Environment Control

We performed a case study based on an information exchange between wireless sensor networks with heterogeneous network protocol in order to demonstrate the feasibility of our approach. This section shows a scenario of information exchange which is an indoor environment monitoring with the following characteristics.

First, there are two wireless sensor networks. Each sensor network is formed by a group of sensor nodes that can monitor variables such as temperature and location-based context-aware. Wireless sensor network 1(WSN1) is responsible for monitoring location-based context-aware and wireless sensor network 2(WSN2) is responsible for monitoring temperature. WSN1 uses zigbee for a network protocol and WSN2 uses Bluetooth. Therefore, it isn't possible to communicate directly between WSN1 and WSN2. Second, all sensor nodes are grouped according to function of sending sensed events to own sink node. A base station accomplishes the role of a sink node, and it can afford to communicate with a home server by SOAP because of enough computing resources. Third, a home server was already installed to manage home network. The home server is a bridge between WSN1 and WSN2. It also can control two wireless sensor networks to integrate them. It manages all wireless sensor networks at home.

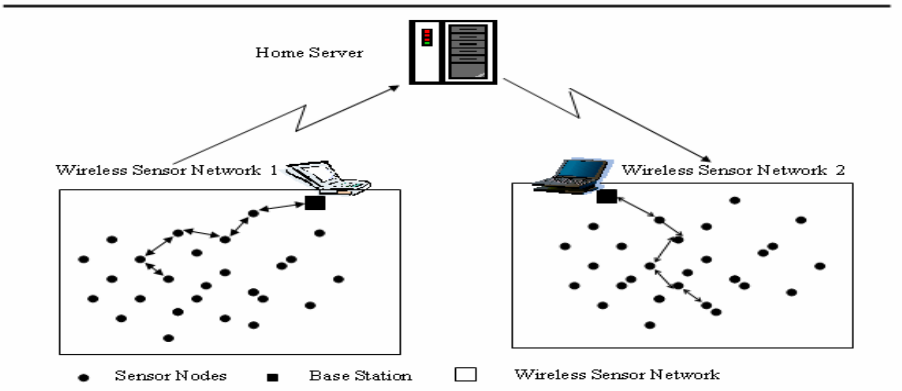
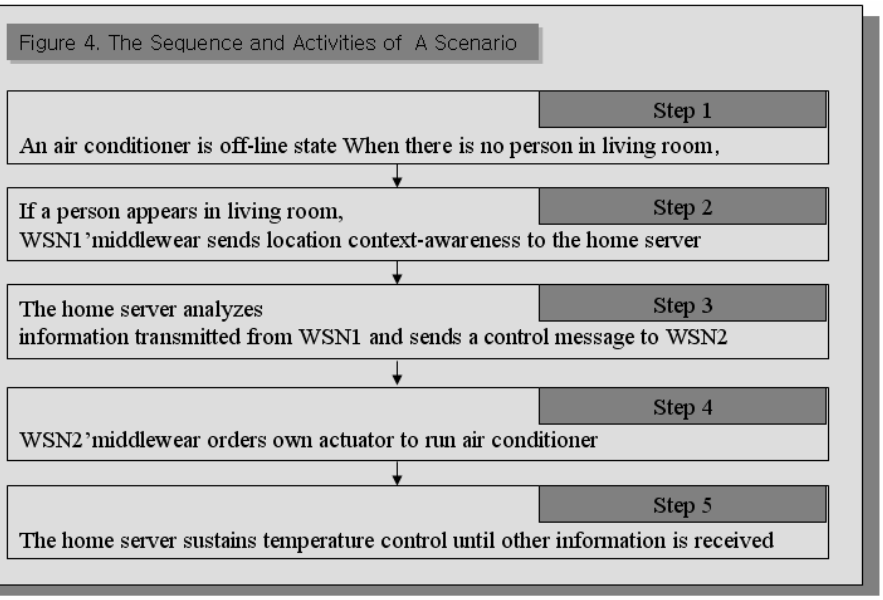


Figure 3. Wireless Sensor Network Configuration in Information exchange Scenario

In this environment, we execute a case study so that temperature can be controlled automatically when a person appears in living room. The sequences of scenario are as followings. At first, there are no people in the living room and an air conditioner is off-line state. After no long time, a person appears in the living room and middleware in WSN1 sends location context-awareness to the home server. The home server analyzes information transmitted from WSN1, and it sends a control message to middleware in WSN2 so that air conditioner is run because a person exists in the living room. Then, middleware in WSN2 orders own actuator to run air conditioner. The home server sustains continuously temperature control until other information is received.

Figure 4. The Sequence and Activities of A Scenario



3.2 A Process of Information Exchange

This section shows a process of information exchange using UML sequence diagram with objects and methods. We divide a process to three cases such as notify status, notify event, and request delivery to present effectively whole process.

Information exchange is begun from that middleware in WSN1 and middleware in WSN2 send own information to the home server. Hence, a MsgSender object of middleware in WSN1 uses firstly a notifyWSNStatus method to send own information to the home server. At this time, the home server updates information of each wireless sensor network in a repository. The information is statuses and these are utilized for a rule handler to control wireless sensor networks.

Next, a sensor node sends the event to a related sink node (is the same of a base station) by defined routing mechanism when a sensor node in WSN1 is sensing the event that someone is in the living room. At this time, the MsgSender object of middleware in WSN1uses a notifyWSNEvent method to send the event to the home server. It also receives an acknowledgment from the home server. After the location context awareness event is arrived from middleware in WSN1, the home server searches rules related to the event message in the repository and makes a decision to control wireless sensor network. These functions are accomplished by a RuleManager object of middleware in the home server. The MsgSender object and the MsgReceiver object in the home server accomplish each role similar to middlewares in WSN1 and WSN2. The home server sends a control message to middleware in WSN2. The MsgSender object in the home server uses a controlWSNActuator method and receives an acknowledgement from middleware in WSN2. If middleware in WSN2 receives a control message, it orders actuator to run air conditioner.

Finally, a request delivery process is carried out in case that middleware in WSN1 wants to send directly to middleware in WSN2. In such case, the rule handler doesn't interfere in the message delivery control between wireless sensor networks. It only helps to bypass the message.

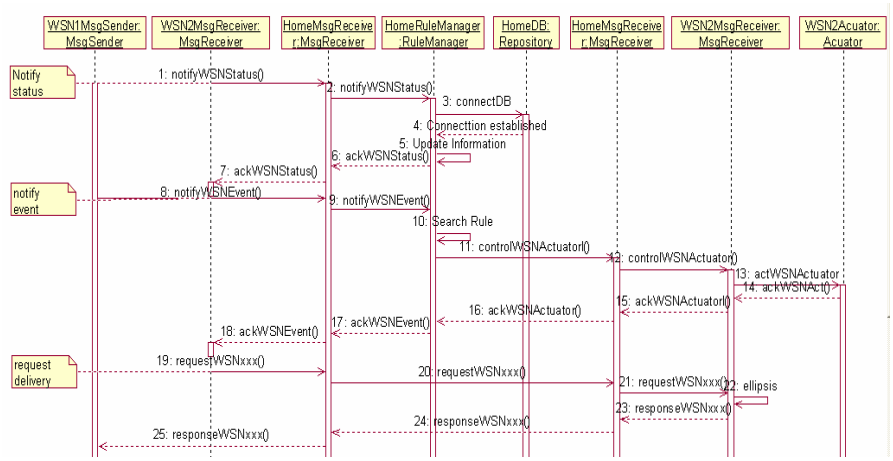


Figure 5. A Process of information exchange in UML Sequence Diagram

3. 3 SOAP Messages

This section shows an example of SOAP message for information query. The SOAP message contains envelope, body, and header element. Methods and data of a message which we have already defined, are in the body part. Figure 6 represents a queryWSNStatus method that is used when the home server sends a query to update information of middleware in the repository. Figure 7 represents a notifyWSNStatus method that is used to response for a query message.

```
<?xml version="1.0" encoding="UTF-8"?>
< SOAP-ENV:Envelope
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    < SOAP-ENV:Body>
        <msg:queryWSNStatus xmlns:nsg="http://hocalhost/simple">
            <MessageType>queryWSN</MessageType>
            <To> wsn2</To>
            <From> home server</From>
            <No>1024</No>
            <Content></Content>
        </msg:queryWSNStatus>
    </ SOAP-ENV:Body>
```

Method

Data

Figure 6. A SOAP message for server query about information in WSN2

```
public void queryWSNStatus(SOAPEnvelope env, String stMsgType, String
stDest, String stSrc, String stMsgNo, String stContent) throws
javax.xml.soap.SOAPException        {
            StatusImpl statusImpl = new StatusImpl();
            StatusBean statusBean = null;
            String msgtypeOfStatus = null;
            String destOfStatus = null;
            String srcOfstatus = null;
            String noOfstatus = null;
            String contenOfstatus = null;
            try
            {       msgtypeOfStatus = statusImpl.getStatusInfo();
                    msgtypeOfStatus = statusBean.getMsgType();
                    destOfStatus = statusBean.getDest();
                    srcOfStatus = statusBean.getSrc();
                    noOfStatus = statusBean.getNo();
                    contenOfStatus = statusBean.getContent();          }
            catch (Exception e)              {
                    msgtypeOfStatus = e.toString();              }
            SOAPBody body = env.getBody();
            Name bodyName =    env.createName("notifyWSNStatus", "msg", "
http://hocalhost/simple ");
            SOAPElement bodyElement = body.addBodyElement(bodyName);
            Name childName = env.createName("MessageType");
            SOAPElement state = bodyElement.addChildElement(childName);
            msgType.addTextNode(msgtypeOfStatus);
            Name childName = env.createName("To");
            SOAPElement state = bodyElement.addChildElement(childName);
            dest.addTextNode(destOfStatus);
.....
```

Figure 7. A notify method corresponding with a query method

4 Related Works

Requirements on Wireless Sensor Network Middleware Middleware in wireless sensor network should support all sorts of services that need in various application system development based on heterogeneous sensor nodes, maintenance, installation, and accomplishment. In order to support these purposes, middleware should provide a lot of functions which make suitable information for high level application by mixing sensing data, conduct data fusion, transfer them, and finally report data to user. Middleware should also accommodate needs of sensor network with efficient utilization of limited energy, network steadiness, and network extensibility [5][6]. We concentrated on abstraction of sensor node's heterogeneity among various roles of wireless sensor network. We build a bridge as a mediator in our architecture and integrate these middlewares.

Researches related to Bridge Technique Researches related to bridge techniques have started in home networks. Existing home applications are based on a large number of different APIs that are often proprietary, incompatible, and normally just address subsets of the devices that exist in a home. This leads to a demand for a general API giving standardized access to all home devices independent of the network type used. Open Service Gateway Initiative(OSGi) [7] specifications define a standardized, component oriented, computing environment for networked services. Adding an OSGi Service Platform to a networked device, enables the capability to manage the life cycle of the software components in the device from anywhere in the network. TinyLime[8] presented bridging the gap between the applications and the hardware. TinyLime uses middleware as solution. TinyLime must be specialized to the qualities of sensor networks, especially energy consumption. TinyLime presents both the model and the implementation of middleware incorporated with the Crossbow Mote sensor platform. FMS[9] proposed a service-based middleware system as comprising of heterogeneous devices assisting to a large range of applications. FMS addresses the specific requirements of wireless sensor network and FMS is based on the concept of services. Services are defined as the data provided by the sensor nodes and the applications to be executed on data. Our approach is motivated by the fact that current works on wireless sensor network don’t consider such a technology in the design of heterogeneous wireless sensor network despite of the advantages related to information exchange. We propose an interoperability layer of middleware for heterogeneous wireless sensor networks. It can be more extensible by an intelligent bridge. It is also generic and flexible, providing the basic functional message required for any wireless sensor network.

SOAP, A Lightweight XML-based messaging protocol Simple Object Access Protocol is a lightweight XML-based messaging protocol used to encode the information in Web Service request and response messages before sending them over a network. The SOAP protocol extends XML so that computer programs can easily pass parameters to server applications and then receive and understand the returned semi-structured XML data document [10]. The SOAP protocol is responsible for defining exchanging rules and messages format in our architecture. The SOAP module in our architecture is composed of two main components: the SOAP engine

and a set of handles. The SOAP engine acts as the main entry point into the SOAP module. It is responsible for coordinate the SOAP message's flow through the various handles. Handles are the basic building blocks inside the SOAP module and they represent the messages processing logic.

5 Conclusion and Future Works

We propose message-oriented middleware to make use of an intelligent bridge for information exchange in heterogeneous wireless sensor network. We present the results of a case study in details to do demonstrate the feasibility of our approach. Our approach offered two contributions to the study of information exchange in heterogeneous constraints. First, we described a rule management technique with interoperability in an intelligent bridge. Second, we defined general message exchange mechanism. We confirmed effectiveness of information exchange mechanism with extensibility and flexibility through the case study. Hence, it can be possible to extend home environment to ubiquitous environment as any control server carries out the same role of the home server.

We are working on identifying general message types and methods in details. We will make lots of efforts to extend this architecture to mobile equipments. We will also add QOS management into interoperable middleware architecture as described in this paper. We hope that the future works' results prove the total energy spent in transmission and processing do not overcome the values found in current wireless sensor network.

References

[1] Ilyas, M., Mahgoub, I.,: Handbook of Sensor Networks:Compact Wireless and Wired Sensing Systems. CRC Press (2005)

[2] Hac, A.: Wireless Sensor Network Designs. Jon Wiley & sons (2003)

[3] Lee, T., Jeong, C.: Service-Oriented Home Network Middleware Based on OGSA. Lecture Notes in Computer Science, Vol. 3480 (2005), 601–608

[4] Park, J.: The technique of interoperable middleware between electronic appliances. TTA journal (2004)

[5] R¨omer, K., Kasten, O., Mattern, F.: Issues in Designing Middleware for Wireless Sensor Networks.IEEE Network (2004)

[6] R¨omer, K., Kasten, O., Mattern, F.: Middleware Challenges for Wireless Sensor Networks. ACM Mobile Computing and Communications Review, Vol. 6 (2002)

[7] http://www.osgi.org/osgi_technology

[8] Curino, C., Giani, M., Giorgetta, M., Giusti, A.: TinyLIME: Bridging Mobile and Sensor Networks through Middleware. The 3rd IEEE International Conference on Pervasive Computing and Communications (2005)

[9] Delicato, F.: A Flexible Middleware System for Wireless Sensor Networks. IFIP International Federation for Information Processing (2003)

[10] Monson, R.: J2EE Web Service. Addison Wesley (2003)