

OSGI Implementation on a Mobile Grid Middleware for Body Sensor Network

*Rossi Kamal, **Choong Seon Hong

Kyung Hee Univ.

*rossi@networking.khu.ac.kr, **cshong@khu.ac.kr

Abstract

This paper presents how OSGI (Open Service Gateway Initiative) can be implemented on a mobile grid middleware for a body sensor network. We have proposed the architecture, performance analysis of this OSGI implementation on this middleware.

I. Introduction

In a body sensor network (BSN)[1], a number of sensor nodes are attached to a patient's body. Each sensor is connected to a small processor, wireless transmitter, battery- all of these form together a BSN node complex. A BSN node captures accurate data from sensors, carries-out low level processing of data and wirelessly transmits information to LPU (local processing unit). Data collected from all sensor nodes are processed by LPU and sent to central monitoring server via wireless LAN or mobile phone (GPRS or 3G).

Body sensor nodes consist of heterogeneous hardware from different vendors. These nodes are deployed for a service or a set of service. Often we need to add, delete or edit any service. But, we have to consider the energy constraints like limited memory, power of nodes. Mobile middleware can solve some problems by hiding underlying hardware, operating system from the application layer. A mobile middleware can use grid[2] approach to overcome energy limitations of those nodes.

OSGI[3] with its service-oriented design has become ideal for grid computing. Problems involved in Jini[4] (lack of platform in-dependency in RMI based implementation, management-overhead for central look-up server) and UPnP[5] (type mapping overhead, high verbosity of XML and decreased performance with SOAP) have created a

stage for OSGI based design for a grid middleware on BSN. For any service, loosely –coupled components of different OSGI instances (on a mobile middleware of different sensor nodes) can interact as in a grid environment. In this paper, we have proposed the OSGI architecture for mobile grid middleware for BSN and the performance analysis of this OSGI design in the distributed environment.

II. Related work

Jini is a software infrastructure that provides distributed computing environment offering 'network plug and play'. Clients and services communicate with RMI in many implementations. It is not possible to publish any service under any specific interface. Every service object is accessible under all its interfaces. The requirement of central lookup server costs management overhead. UPnP is a service oriented standard to communicate with consumer devices. It is based on communication standard like HTTP, XML and SOAP. High verbosity of XML and SOAP lead to decreased performance in implementations. UPnP also does not allow subscribing to specific events; clients need to subscribe to all events. So, Jini and UPnP are invasive for the design of a grid middleware on BSN.

III. Description

A. Proposed Architecture: Proposed mobile grid middleware consists of several layers. Application layer is in the uppermost position on the architecture. Services like blood pressure, temperature calculation can be implemented as OSGI bundles in this layer. To add a new

"This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)" (NIPA-2010-(C1090-1021-0003)) Dr. CS Hong is the corresponding author.

node to the system, OSGi bundle implementing node's service should be installed in this layer. Inside the middleware, installation involves invoking of loadNode from ServiceManager.

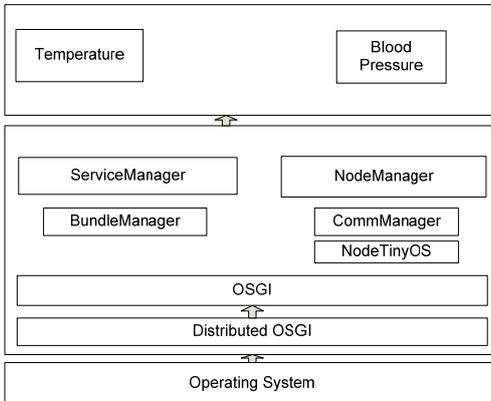


Figure 1: Proposed OSGi based architecture for grid mobile middleware for BSN

The service location of node's bundle and serial identifier are important parameters in this invocation. Next comes the middleware layer consisting of layers namely: ServiceManager, CommManager, NodeTinyOS, OSGi and DistributedOSGI. ServiceManager is in the uppermost and Distributed OSGi is the lowermost layer of all middleware layers. ServiceManager allows application bundles to get node objects by using services. This layer allows upper application layer to install and uninstall nodes. It sends beacon message to other nodes to update their status and forwards incoming message to application bundle so that they can process information. NodeManager layer allows nodes to send and receive 802.15.4 message by using services from CommManager layer. A set of open interface can be defined in this layer for each node, allowing interoperability between devices of different hardware vendors. CommManager layer sends/receives message service to upper layer through serial port. This layer uses the service provided by NodeTinyOS layer made of TinyOS OSGi packages. This layer helps in registering received-message. Application bundle registers incoming message with NodeManager. Message comes to CommManager, NodeTinyOS one after another. It is then sent to ServiceManager that checks message's source to forward to application bundle. The lowermost layer is the DistributedOSGI layer. It allows that multiple peers with

OSGi framework can communicate in a way that makes them like in a large OSGi framework.

B. Performance Analysis: In this middleware, distributed feature of local OSGi framework is masked and remote services are accessible as if they are present in framework (transparency). Although JVM is deducted from this OSGi architecture, there are no other restrictions on OSGi specification and all existing bundles are re-usable (non-invasiveness). This architecture shows that remote node for a service the same as that of local framework (consistency). This architecture allows OSGi framework to run on a large range of body sensor nodes and distribution does not limit configurations where OSGi can be used (generality). It is also avoided here that a BSN node is overwhelmed by number of available services (statement of supply and demand).

IV. Conclusion

Mobile grid middleware for BSN has to be lightweight, considering energy constraints of body sensors. Existing sensor nodes utilize component programming of nesC on TinyOS, most popular OS for BSN. So, to design the lightweight middleware, we cannot but adopt this development environment. OSGi design paradigm uses JVM (Java Virtual Machine) over TinyOS. In this middleware, we use OSGi paradigm with nesC over TinyOS. So, OSGi instances on different body sensor nodes use component programming of TinyOS to communicate in the grid environment.

Reference

- [1] Guang-Zhong Yang, Magdi Yacoub, "Body Sensor Networks", Springer-Verleg London Limited 2006,
- [2] Foster, I., "The Anatomy of a Grid: Enabling Scalable Virtual Organizations", Euro-Par 2001, LNCS 2150, pp. 1-4, 2001.
- [3] OSGi Alliance, "OSGi Service Platform Core Specification", Release 4, Version 4.2, June 2009
- [4] Jini Specification, (<http://www.jini.org>)
- [5] UPnP Forum, "Universal Plug and Play Device Architecture", (<http://www.unp.org>).