

# Design and Implementation of Mobile Services Evaluation System

Ruyi My, Jian Liu

School of Computer Science and Engineering, Beihang University, Beijing 100191, China  
Email: maruyi227@yahoo.com.cn

Received July, 2013

## ABSTRACT

With the development of mobile network technology, mobile Web services have also been more and more attention, but standard Web protocols are not applicable in a mobile environment, so people began to design more efficient, compact and consume less mobile Web services protocol. In the course of research services protocol, a convenient and practical evaluation system will undoubtedly bring great help for the protocol, the paper design of mobile service evaluation system using a graphical way to display a variety of mobile services in the overhead data, system design a lightweight service protocol, and the protocol by the system for evaluation compared with a standard Web.

**Keywords:** Mobile Services; Web Services; Evaluation Systems; Light Weight

## 1. Introduction

Currently, web services technology as a service-oriented architecture to achieve good performance is attracting more and more attention. Being characterized with self-contained, self-describing, modular, mobile services can be released and called via the Internet. It also has the characteristics of loosely coupled, and platform independent, very suitable for the Internet environment interoperability between heterogeneous applications and integration.

Web services for heterogeneous resources together to provide effective means. However, in the wireless mobile environment, network bandwidth, memory capacity, processor computing power, power supply and other resources are very limited. Traditional web services can not be well meet heterogeneous devices in such resource-constrained environment interconnected and collaborative requirements. So people began to study web service protocols in a mobile environment on how to allow a service protocol with a smaller communication structure is more compact, smaller storage resource consumption characteristics, without reducing the support of the core functionality, and have achieved some results[1-8].

In the mobile services research, evaluation of the performance of the service protocol is the key to improving service protocol to properly reflect the service protocols and intuitive performance testing tools can help us better refine and improve service protocols. So the design implementation can find and/or invoke services. Mobile network service test system can promote better research

service protocol to work in a mobile environment. In addition, the design of the mobile network service simulation environment is to study large-scale mobile network service protocol in the performance of the important ways.

This design of mobile network service evaluation system is used to evaluate the environmental performance of software services, in software design implements a lightweight web services, and software evaluation can be designed for this service, and traditional service protocols light weight visual comparison. Software evaluation can be automatically configured according to user service calls and service discovery process, the software runs in the evaluation process can be automatically collects service discovery and/or in vocation overhead of various relate draw data and for persistent storage. The original-cost of the collection of data for statistical calculation, the cost of data classification, sorting out the valid data after the completion of the data collection statistics, data can be graphically displayed to the end user and a variety show form at, the software can be individual and comparative approach to the comparative light weigh graphical form and standard service protocols.

## 2. System Function and Structure

### 2.1. Evaluation System Description

Mobile service evaluation system is divided into server-end software evaluation and client-end software evaluation, which are executed separately on the server and client, and jointly complete publish, discover, invoke the

web services. Server-end services of mobile web services complete the task of publishing and providing services. Client-end does the work of service discovery, and there are two kinds of service invocation modes that are designed for lightweight service protocols and standard web protocols, in order to data collection and data display respectively. More details of mobile service evaluation system are described below

- Service invocation: user can configure the service call process, evaluation software is automatically configured according to user service call;
- Service discovery: user can configure the service discovery process, evaluation software automatically according to the configuration service discovery process;
- Collection overhead data: software runs in the evaluation process, the software can automatically collect discovery and/or invocation overhead all relevant raw data, and make persistent storage (file or database schema);
- Statistics overhead data: the collected raw data for statistical computing overhead on the overhead data classification, sorting out the valid data;
- Display overhead data: After the completion of the data collection statistics, data can be graphically displayed to the end user, the custom display form;
- service protocol contrast: the ability to distinguish between lightweight and standard service data can be displayed separately or both comparative data;

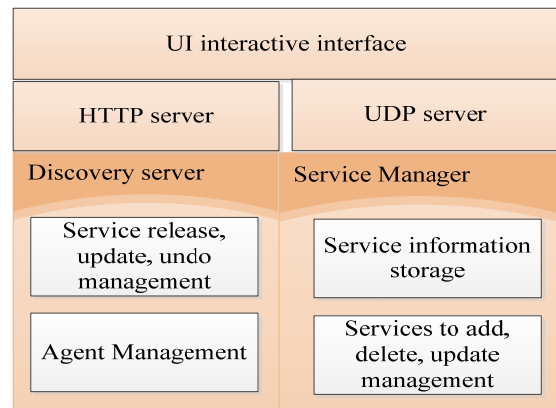
## 2.2. Evaluation System Server-side Architecture

Server-side of mobile service evaluation system is to complete server control, service publish, services revocation, etc., shown in **Figure 1**, which includes the following modules:

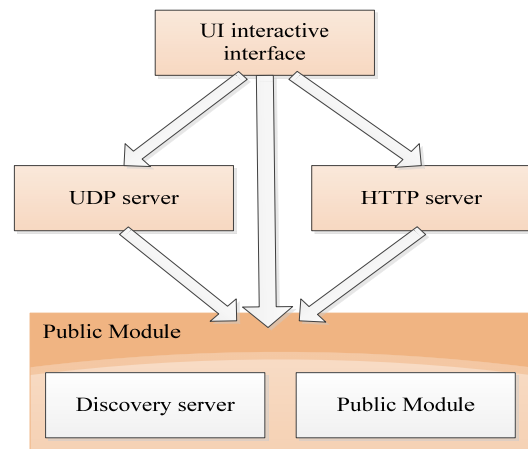
- HTTP / UDP server: As a service container is an essential part of the evaluation of the server, it completes the server on / off control, according to the request call response service and returns the response results.
- Discovery Server: provides a service discovery interface control, mainly to complete the discovery server is turned off / control, service publishing, service revoked, turn on/off proxy service.
- Service Manager: Responsible for managing the published service, mainly to complete the published service information storage, reading, and call to develop services
- UI interactive interface: control services provided to the user side of the interface.

Mobile service evaluation service calls between end module shown in **Figure 2**, a brief description below:

- On the service side, HTTP / UDP protocol is the basis, no need to call any other module;
- Discovery server need to call for service discovery protocol release, revocation and proxy control;



**Figure 1. Evaluation system server-side architecture.**

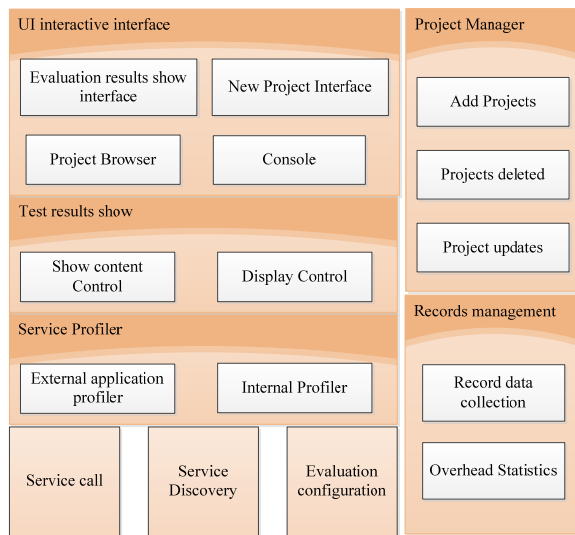


**Figure 2. Servermodule callsRelations.**

- Common module by the service management, discovery server modules, this module calls for all modules;
- UDP / HTTP server modules need to call the public service message parsing, utility service management module for service calls. On the server status changes (on / off) when calling discovery server publishing / revocation service; public module consists of service management, discovery server modules, this module calls for all modules;
- UI interactive interface called UDP / HTTP server module control server status, call the discovery server for discovery control, call the service management module publish / revoke services, and show the published list of services;

## 2.3. Evaluation system Client Structure

Mobile service evaluation system client complete the work of service discovery, service calls, evaluation data collection and evaluation of data management and graphical display of the work, the mobile service evaluation system client structure shown in **Figure 3**, which comprises with the following eight modules:

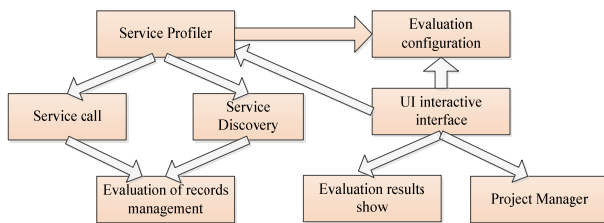


**Figure 3. Servermodule callsRelations.**

- Service call module: encapsulate the underlying service protocol, making the service call details for the upper application and transparent. Module function design based on the need to support the service protocols can be adjusted for different protocol provides different calling interface. And be able to return after the end of the service call service call results;
- Service discovery module: encapsulate service discovery protocol , the upper hide the details of service discovery can be configured according to the user find specific service or find all available current network services has been found for the service management functions;
- Evaluation configuration module: evaluates based client for evaluation, provides a user interface on controlling evaluation;
- Project Manager: User can evaluate the service through the establishment of evaluation projects for evaluation, each project with a unique profile. Thus, user can create multiple projects evaluating a variety of different configurations, or for the same configuration at different angles to be tested. In short, the project manager is designed so users can more easily manage evaluation program;
- Service Evaluation Module: core evaluation environment, Profiler configuration under evaluation, automatic service discovery or call Profiler consists of four sub-modules: Service Call Profiler, the Profiler service discovery, auto-evaluation, external application profiler ;
- Evaluation Records Manager: Responsible for managing the evaluation process in a variety of data services;
- Evaluation Results Display Modules: records manager is responsible for statistical evaluation of the data to be displayed graphically which provides evaluation re-

sults show a variety of ways, choose to display the data, you can monitor real-time data, you can view historical data;

- UI interactions Interface: User Control software behavior only interface, you need to provide a friendly interface;
  - Service call module: encapsulate the underlying service protocol, making the service call details for the upper application and transparent. Module function design based on the need to support the service protocols can be adjusted for different protocol provides different calling interface. And be able to return after the end of the service call service call results;
  - Service discovery module: encapsulate service discovery protocol , the upper hide the details of service discovery can be configured according to the user find specific service or find all available current network services has been found for the service management functions;
  - Evaluation configuration module: evaluates based client for evaluation, provides a user interface on controlling evaluation;
  - Project Manager: User can evaluate the service through the establishment of evaluation projects for evaluation, each project with a unique profile. Thus, user can create multiple projects evaluating a variety of different configurations, or for the same configuration at different angles to be tested. In short, the project manager is designed so users can more easily manage evaluation program;
  - Service Evaluation Module: core evaluation environment, Profiler configuration under evaluation, automatic service discovery or call Profiler consists of four sub-modules: Service Call Profiler, the Profiler service discovery, auto-evaluation, external application profiler ;
  - Evaluation Records Manager: Responsible for managing the evaluation process in a variety of data services;
  - Evaluation Results Display Modules: records manager is responsible for statistical evaluation of the data to be displayed graphically which provides evaluation results show a variety of ways, choose to display the data, you can monitor real-time data, you can view historical data;
  - UI interactions Interface: User Control software behavior only interface, you need to provide a friendly interface;
- Evaluation of calls between the client module shown in **Figure 4**, is briefly described below:
- As can be seen from the figure, evaluation, and evaluation of records management configuration is the entire basis for evaluation of the client, which is also consistent evaluation of environmental design in mind, that the service process for evaluation and monitoring services of the various expenses;



**Figure 4. Client module calling relationships.**

- Service calls, and service discovery module manager by invoking the meter records, records various data in the evaluation process;
- Service evaluation requires a service call and service discovery, and call and find that it is based on the configuration, so you need to call a service call, service discovery and evaluation configuration module;
- Evaluation results show that the module needs to call meter records management module for data statistics, and then obtain statistical data for display;
- Project management module is configured to facilitate user management evaluation established and is therefore not dependent on any other modules;
- UI interactions interface is in the top level of the software architecture. But shielding the low-level operation, only need to call the service evaluation module for service evaluation, call the evaluation configuration module configuration management, project management module called the project management, evaluation results display module call monitoring / evaluation results show, thus ensuring the user on the evaluation of the controls;

### 3. The Key Technologies Used in the System

#### 3.1. Lightweight Service Invocation Protocol

Mobile service evaluation system designed lightweight Web services technology is built on XML, SOAP, WS-Discovery protocol based on criteria such as a distributed application architecture. Lightweight design service protocol's main task is how to reduce the service invocation packet size. In the mobile environment, how to use SOAP-based message format to complete a large number of data transfers, and to ensure efficient transfer process and accuracy, a feasible way is to use compression technology to compress the SOAP message. Since SOAP messages generally use HTTP, FTP and other connection-based standard protocol for data transmission, this transmission is the essence of stream-based data transmission, without compressing the SOAP message itself, so this paper directly to the Web service SOAP messages this layer to achieve the flow compression of the data stream, the input and output streams are encapsulated protocol that is added to the structure lightweight service SOAP message compression and decompression module.

Lightweight call protocol module structure includes the following five modules:

- Applications call interface: provides a convenient interface to application calls;
- Message Packaging: As in the message encapsulation process, only basic types (int, boolean, string, etc.) can be directly encapsulated into XML. As for complex objects, such as the Date object, as well as custom objects, you cannot directly encapsulate them to XML. In order to solve complex objects and custom objects encapsulation issues, as well as reduce equipment burden and the amount of data transmitted as much as possible, the paper design a unified object encapsulation mechanism, which creates a lightweight service message and Soap Object object, fills the message content into Soap Object, then convertes Soap Object to XML.

- Message serialization:

After completing the message encapsulation, Soap Object object cannot be transferred, so you need to convert Soap Object into XML, then serialize it into byte code, finally send it out. This module just finishes this job.

- Compression / decompression: This module's function is to reduce the service data packet, and compress the XML serialized data stream before sending it to the client. The client then uses the corresponding decompression algorithm to decompress.

- Transmission module:

Since the overhead of TCP protocol is relatively large, plus there are many additional HTTP protocol overhead and it is difficult to establish a stable connection in mobile networks. Therefore, this article uses UDP transmission in the mobile environment. On one hand, eliminate the need for managing TCP link state control overhead. On the other hand, save HTTP data header and effectively reduce the packet size.

#### 3.2. Lightweight Service Discovery Protocol

Traditional mode of service calls are on the premise that clients know in advance the design target address of the service, and create a client endpoint for the service call based on this address. The service discovery protocol in our mobile service evaluation system is the client does not know in advance the target service address and dynamically invoke the detected available services. Service Discovery lifts the dependencies between client and server, which allows the server dynamically change its address. The client and server can also dynamically join and exit.

Lightweight service discovery protocol defines two basic service discovery mechanisms to achieve the operating mode: Ad-Hoc mode and Managed mode.

Ad-Hoc Mode:

In Ad-Hoc mode, the client in a certain form of net-

work-wide broadcast sends probe (Probe) messages to search for the target service. The probe message contains the appropriate search criteria. After receiving the probe message, the target service which complies with this condition will sent back self-related information (including address) as a broadcast message to the source client. Client is access to services based on information, and selects the appropriate service calls.

#### Managed Mode:

In Managed mode, you can maintain a number of available target service discovery agents (Discovery Proxy). Clients only need to send a message to the detection agent can be found in the corresponding target service information. As in Ad-Hoc mode, broadcast mode detection mechanism Managed be transformed into unicast form, the benefits that greatly reduce the network load.

## 4. Mobile Service Evaluation System

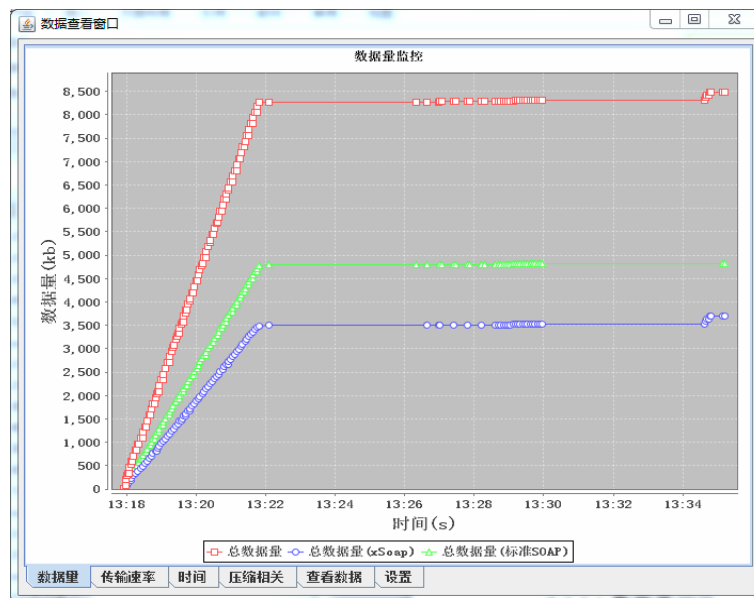
Next, by using this mobile service evaluation system, wetest system lightweight web protocols and standards, in order to better display this mobile service evaluation function. Five mobile devices are used for testing, two of them run evaluation procedures, a client computer runs evaluation, a computer runs service agent, and the other one as a server. Then two computers are running on mobile service evaluation client and server software. The server provides the client with MP3 music playback service running for some time to gather around 40k record data, pause the program to run, and then restart the process, only open system designed lightweight service to show evaluation system's various test methods (single service protocol testing, multiple service comparison

test), followed by shows two services protocol performance's graphical comparison of data using mobile service evaluation system.

**Figure 5** shows the change of the total data amount during evaluation process, it is obvious that for the mobile services lightweight design in evaluation system services protocol (hereinafter referred to XSOAP), the amount of data is smaller than the standard data amount SOAP. Let's analyze the change of the curves.

- 13:18-13:22 period, the rapid increase in the amount of data, corresponding to the period of continuous service during a call. Although during this process, the SOAP's and XSOAP's data are rapidly increased, but the standard SOAP increased significantly faster.
- Then 13:22-13:26 period, no data points, corresponding to the service is not available, waiting for a response.
- Then the period 13:26-13:30, the data increases slowly, which corresponds to the process of the discovery service, the data volume is created in a service discovery and withdrawn before the service call result is in vain.
- At 13:34 and later find a service, but this time only opens XSOAP service, so XSOAP services and the total amount of data is on the rise, but the data amount of the standard SOAP has not risen.

**Figure 6** shows a graph of the response time change. Observed on the whole, the response time of XSoap is less than of the standard SOAP. It can be seen, XSoap's response time is better than the standard SOAP, and the volatile of the response time is less, more stable.



**Figure 5.** Comparison of the amount of data (The blue line represents XSOAP, The Green line represents the standard SOAP).

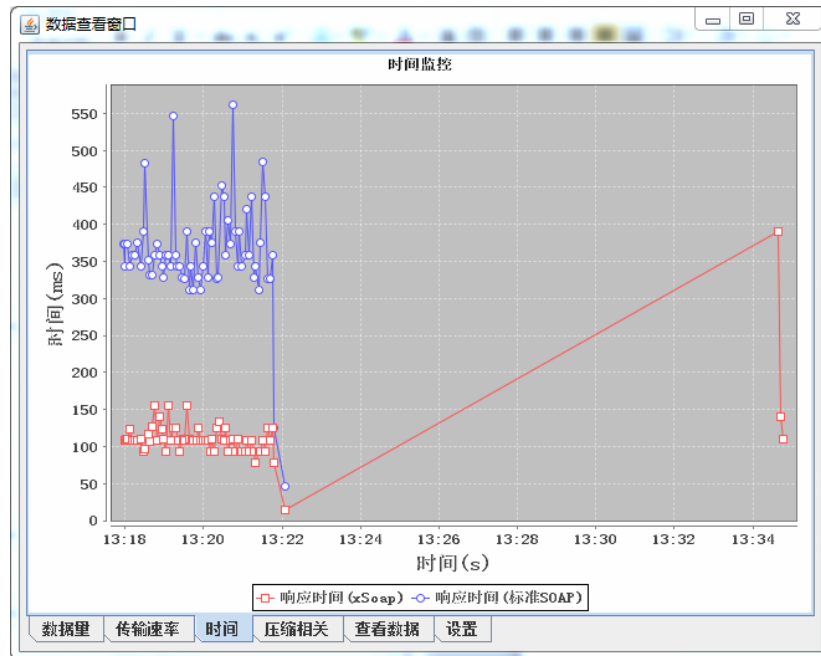


Figure 6. Comparison of response time (The red line represents XSOAP, The blue line represents the standard SOAP).

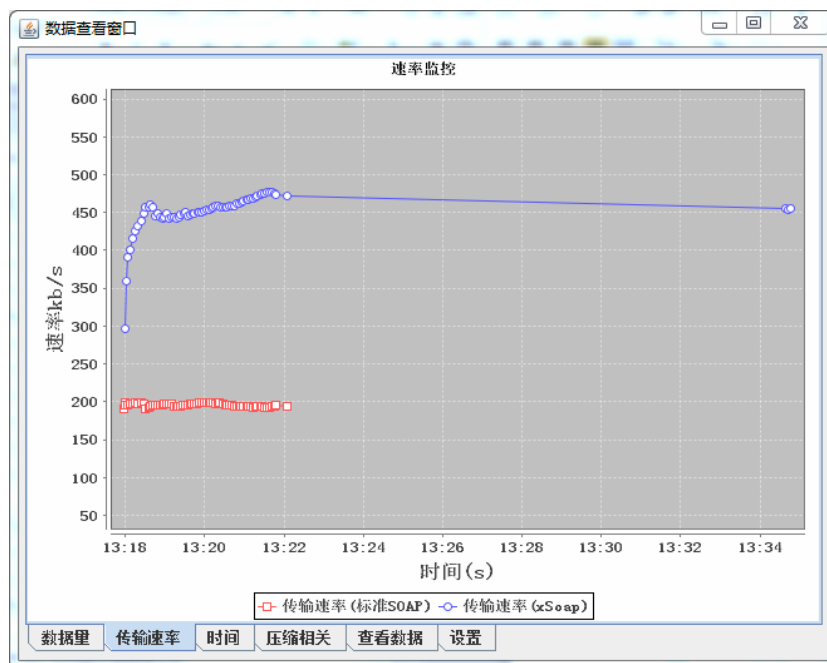


Figure 7. Comparison of transmission rate (the red line represents the standard SOAP, The blue line represents XSOAP).

Figure 7 contrasts the transmission rate of x Soap and standard SOAP. SOAP’s transmission rate can be seen that is about basic mark 200kb / s, not much fluctuation. As for x Soap, there was a significant increase in process, and then stabilizes at about 460 kb / s, probably 2.5 times of the standard SOAP. This increase process is mainly because of the MP3 file. The MP3 file header at the beginning of a data is text content, so the compression ratio

becomes bigger that results less amount of data to transmit and a lower transmission rate. Then, as the text content transmission is completed, the subsequent contents are binary. The amount of data becomes large, thereby increasing the rate and then be stabilized.

### 5. Conclusions

With the development of network technology and mobile

applications, mobile Web services will receive more attention and the research about Web services suitable for the mobile environment will become a hot spot. The service test software which is reasonable and practical mobile plays a very big role on studying in the service protocol the process. This design of mobile services uses a variety of test systems, graphically demonstrating the performance of the Service Protocol. And in the system we also design a lightweight mobile service protocol. It can be seen that in the evaluation system test process, the service protocol is more suitable for mobile network environments than the standard service.

### REFERENCES

- [1] R. A. Van Engelen, K. A. Gallivan, "The g SOAP Toolkit for Web Services and Peer. to-peer Computing Networks// Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid, *New York*, IEEE Computer Society Press, 2002, 128.
- [2] R. Pellerin, "The Moo DS Protocol; AJ2ME Object-oriented Communication Protocol//Proceedings of the International Conference on Mobile Technology, Applications and Systems 2007. *New York*: ACM Press, 2007, pp. 8-15.
- [3] S. Cuenca-Asensi, H. Ramos-Morillo, H. Llorena-Martinez and F. Macia-Perez, Reconfigurable Architecture for Embedding Web Services//Proceedings of 4th Southern Conference on Programmable Logic. *New York*; IEEE Computer Society Press, 2008, pp. 119-124.
- [4] F. Maci'a-p'erez, D. Marcos-Jorquera and V. Gilart-Iglesias, "Industrial TCP/IP Services Monitoring through Embedded Webservices," *EURASIP Journal on Embedded Systems*, 2008, Vol. 1, pp. 7-17
- [5] N. Apte, K. Deutsch and R. Jain, Wireless SOAP: Optimizationsfor Mobile Wireless Web Services//Proceedings of the 14<sup>th</sup> International Conference on World Wide Web. *New York*; ACM Press, 2005:1178—1179
- [6] S. De Deugd, R. Carroll, K. E. Kelly, "SODA: Service-oriented Device Architecture," *IEEE Pervasive Computing*, 2006, Vol. 5, No. 3, pp. 94-96. [doi:10.1109/MPRV.2006.59](https://doi.org/10.1109/MPRV.2006.59)
- [7] R. Van Engelen, "Code Generation Techniques for Developing Light-weight XML Web services for embedded devices," Proceedings of the 2004 ACM Symposium on Applied Computing, *New York*: ACM Press, 2004, pp. 854-861.
- [8] J. S. Rellermeyer and G. Alonso, Concierge: A Service-platform for Resource-constrained Devices, *ACM SIGOPS Operating Systems Review*, Vol. 41, No. 3, 2007, pp. 1245-1258. [doi:10.1145/1272998.1273022](https://doi.org/10.1145/1272998.1273022)