

## Example of Communication between Distributed Network Systems Using Web Services

Martin Tsenov

**Abstract:** *This paper explains decision for communication between distributed network systems based on the Web Services standards and protocols - SOAP and WSDL. Software solution based on the proposed architecture, developed using SOAP extension for PHP5 is presented and explained.*

**Key words:** *PHP5, SOAP, Web Services, WSDL*

### INTRODUCTION

The use of data exchange and need to define resource on the World Wide Web is expanding rapidly with the application-to-application communication and interoperability grows. These services provide a standard means of network communication between different software applications involved in presenting dynamic context-driven information to the user. In order to promote interoperability and extensibility among these applications, as well as to allow them to be combined in order to perform more complex operations, Web services architecture is needed. In this paper the author present an example, which is developed to illustrate the role and the functionalities of the Web Services based communication. The example demonstrates the implementation of network services, which can be invoked both from a workflow orchestration engine and through graphical user interface. The role of the proposed Web Services implementation is related with the automatic interfacing and invocation of network information services, but the triggering of these services is performed from an object, belonging to workflow execution architecture. The example is developed for receiving and monitoring data from microcontroller that work in the local network.

### WEB SERVICES ARCHITECTURE FOR RECEIVING AND MONITORING DATA BETWEEN DISTRIBUTED NETWORK SYSTEMS

#### *A. Used standards and protocols*

- SOAP - SOAP (Simple Object Access Protocol) [1] is a lightweight XML-based protocol for exchanging structured information between distributed applications over native web protocols, such as HTTP. SOAP specifies the formats that XML messages should use, the way in which they should be processed, a set of encoding rules for standard and application-defined data types, and a convention for representing remote procedure calls and responses.

SOAP protocol consists of three parts:

1. An envelope which describes the contents of the message and how to use it
2. A set of rules for serializing data exchanged between applications
3. A procedure to represent remote procedure calls, that is, the way in which queries and the resulting responses to the procedure are represented.

- WSDL - WSDL (Web Service Description Language) [2] is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.

*B. Description of the architecture*

The proposed Web Services communication [3] is developed for receiving and monitoring temperature and humidity data from microcontrollers DS TINI and IPC@CHIP, which is maintenance by Virtual Laboratory of Computer Networks and Distributed Systems-TU Plovdiv.

The sequence of operation and the invocation of the Web Services for the receiving and monitoring data from microcontroller is explained as follows:

- A Client invokes a Web services. The Web Services are formalized in two ways: by WSDL description and by PHP5 Web Services application. The invocation of the web Services could be triggered by:
  - Graphic User Interface (GUI), which performs direct communication with the Web Services.
  - Graphic User Interface (GUI), which performs WSDL communication with the Web Services.
  - GUI, which influence the sequence of operations of a workflow engine (for example the BPEL defined functionality of the workflow engine). Thus the invocation of the Web Services is performed by the workflow engine.
  - An ERP, which performs automatic business and operational support services. The ERP system participates in an integrated workflow system, where the engine automatically communicates with the Web Services.
- Client of the GUI applications sends SOAP Request messages to the Apache Axis based Web services for direct communication with two controllers - DS TINI and IPC@CHIP.
- Apache Axis Web services receive requested data, prepared and send SOAP response for the Client Request.
- The client (GUI) receives SOAP Response. Currently the propose example of Web Services support two ways for data integration:
  - To store received data into MySql DB for further using.
  - Only to represent received data.
- After the implementation of the Web Services operations, the results can be used for further operation like statistics chart draw for received data.

Figure 1 and figure 2 presented two ways for used of the presented architecture.

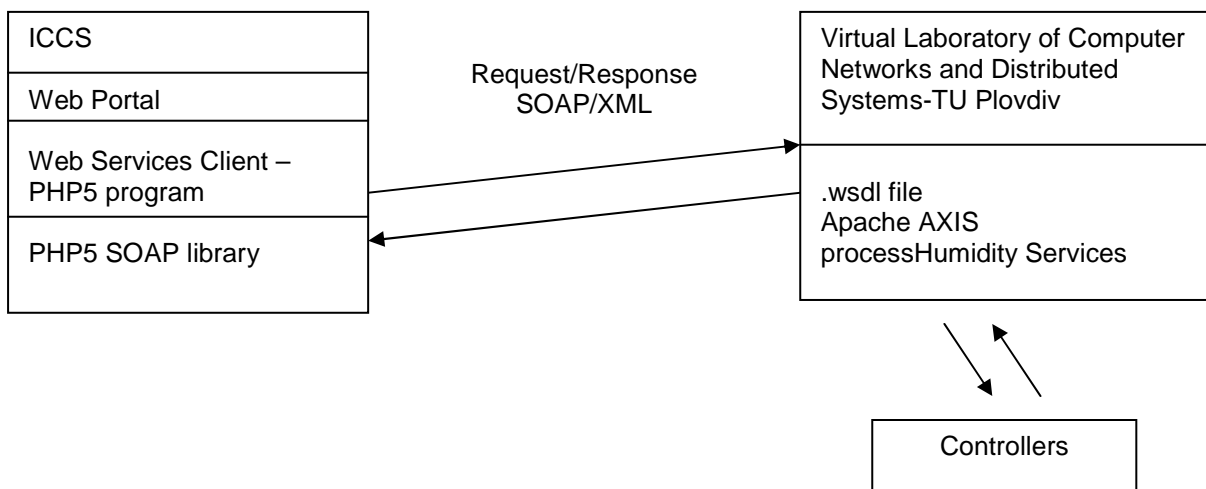


Fig. 1 Web Services client architecture for the Temperature/Humidity services

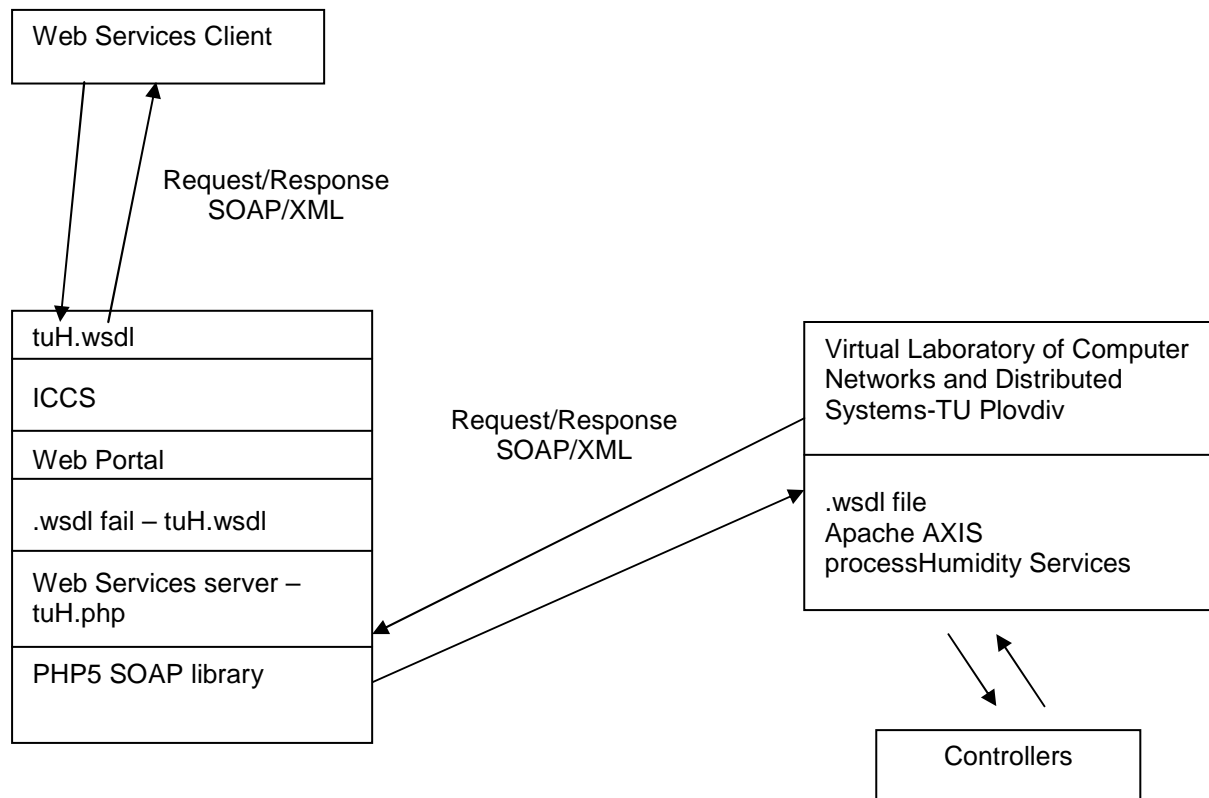


Fig. 2 WSDL file client architecture for the Temperature/Humidity services

### C. Software Implementation

The software implementation for receiving and monitoring data from microcontrollers has been developed and tested in companion of a suite of software tools, which description is given below. Each software component is presented according to the general functional architecture, related to the fig.1 and fig.2. The software implementation for the proposed architecture is based on:

- Client side technology: Software implementation is tested with two clients:
  - A client, based on PHP4 and NuSOAP library [4]. It is used for the invocation of network services, formalized as Web Services by appropriate WSDL file;
  - Eclipse SDK (v.2.0) Web Service Explorer tool.
- Software implementation applies Server side technology. It is implemented with the following software modules:
  - Apache web server;
  - PHP5;
  - PHP5 SOAP library [5];
- MySql DB for data storage.
- Zend Studio or Eclipse SDK for development of WSDL description.

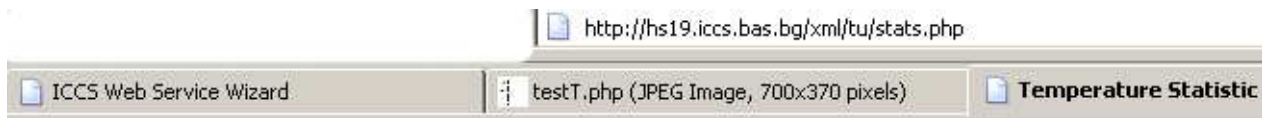
The resources used for the software implementation for receiving and monitoring data from microcontrollers test architecture are described with the appropriate URLs:

- Client side technology: -
  - WSDL client:[http://hs19.iccs.bas.bg/work/nusoap/ws\\_wizard.php](http://hs19.iccs.bas.bg/work/nusoap/ws_wizard.php)
  - Eclipse SDK (v.2.0) Web Service Explorer tool.

- Web Services technology:
  - WSDL (temperature) - <http://hs19.iccs.bas.bg/xml/tu/tuT.wsdl>
  - PHP5 (temperature) - <http://hs19.iccs.bas.bg/xml/tu/temp.php>
  - WSDL (humidity) - <http://hs19.iccs.bas.bg/xml/tu/tuH.wsdl>
  - PHP5 (humidity) - <http://hs19.iccs.bas.bg/xml/tu/hum.php>

### TESTS AND RESULTS

Figure 1 and figure 2 presented different views for data results, which is receive using Web Services application for receiving and monitoring temperature and humidity data from microcontrollers. The application, documentation and available sources can be reached at [http://hs19.iccs.bas.bg/work/nusoap/ws\\_wizard.php](http://hs19.iccs.bas.bg/work/nusoap/ws_wizard.php).



#### Results:

Temperature statistics (last 10 results) from controller **ipc**

Temperature: (in °C)	Time:	Date:
16.44	13:42:48	March 28
16.48	13:42:16	March 28
16.32	13:41:5	March 28
16.08	13:39:55	March 28
15.24	18:51:2	March 26
16.16	15:8:35	March 23
15.64	14:40:3	March 15
16.32	14:53:7	March 6
16.36	14:10:4	March 6
16.32	14:6:9	March 6

Fig. 3 Result from testing Temperature Web Services – html table

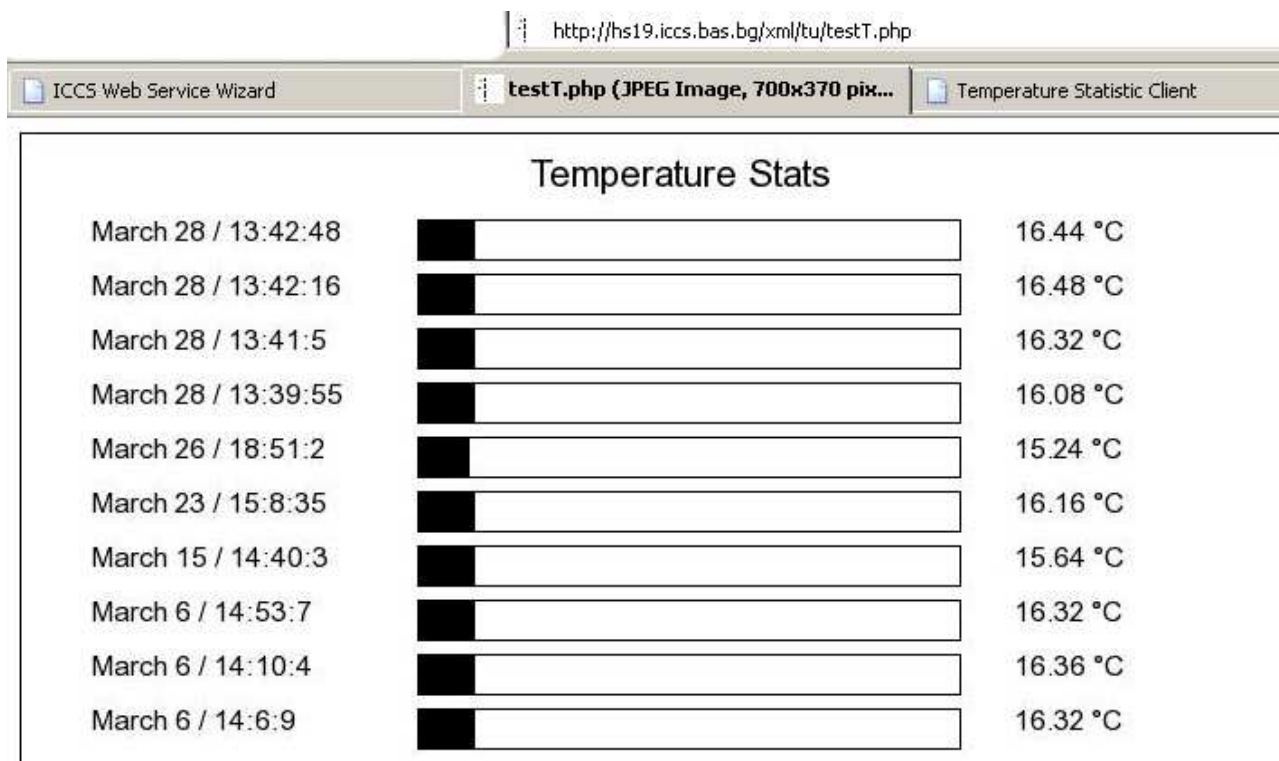


Fig. 4 Result from testing Temperature Web Services – chart

### CONCLUSIONS AND FUTURE WORK

This paper explained decision for Web Services architecture between distributed network systems. Software solution based on the proposed architecture, for receiving and monitoring temperature and humidity data from microcontrollers is presented, explained and tested. The example revolves around the open source standards, which allows clients to access and exchange data and resources between distributed systems.

The future work is to store propose Web Services into UDDI Registry.

### REFERENCES

- [1] <http://www.w3.org/TR/2003/REC-soap12-part0-20030624/>
- [2] <http://www.developer.com/services/article.php/1602051>
- [3] <http://www.developer.com/services/article.php/1550461>
- [4] <http://dietrich.ganx4.com/nusoup/>
- [5] <http://devzone.zend.com/node/view/id/689>

### ABOUT THE AUTHOR

Martin Tsenov, Department Hierarchical Systems, Institute of Computer and Communication Systems - Bulgarian Academy of Sciences, Sofia, Phone +359 2 979 2774, E-mail: [mcenov@hsi.iccs.bas.bg](mailto:mcenov@hsi.iccs.bas.bg), [mcenov@hsh.iccs.bas.bg](mailto:mcenov@hsh.iccs.bas.bg)

### ACKNOWLEDGES

The development, maintenance and technical support for the functionalities of the microcontrollers DS TINI and IPC@CHIP are provided by Virtual Laboratory of Computer Networks and Distributed Systems-TU Plovdiv: <http://net-lab.tu-plovdiv.bg/website/>.

This research is partly supported by: 1. European Commission; Project VISP; Project no. FP6-027178, 2. Project no. VU-966/2005 - Bulgarian Scientific Fund.