# DISTRIBUTED MEASUREMENT SYSTEM BASED ON JAVA AND WEB TECHNOLOGIES

**Mitko Shopov, Grisha Spasov**

Department of Computer Systems and Technologies, Technical University of Sofia, branch Plovdiv, 61 "St. Petersburg" Blvd. , 4000 Plovdiv, Bulgaria, phone: +359 32 659758, e-mail: {mshopov,gvs}@tu-plovdiv.bg , web: http://net-lab.tu-plovdiv.bg/

## INTRODUCTION

A trend from the recent years is to migrate away from proprietary hardware and software platforms for distributed measurement systems (DMS) in favor of open and standardized approaches. High-level programming languages, object-oriented platforms, Internet technology, and standardized communication interfaces, all influence the development of today's DMS. Additionally, rapidly advancing hardware, provide the market with a plenty of new embedded devices with integrated TCP/IP stack, embedded web server and continuously increasing processing power. This gives the designers of distributed measurement systems the ability to put into practice some of the well-proven architecture models from distributed desktop systems.

This paper describes the design and implementation of a system for distributed measurement that uses popular client-server architectures, Java and Internet technologies. The design is based on three-tier architecture model, with the focus put on the middle tier. The implementation of the system is based on Java Server Pages (JSP) technology.

## DISTRIBUTED MEASUREMENT SYSTEM BASED ON JAVA AND WEB TECHNOLOGIES

The design of a modern, web-based distributed measurement systems have to be carried out in accordance with well-proven architecture models, allowing the system to benefit from various available technologies, thus giving it added flexibility and scalability. It has to be highly abstract, easily extensible and user-friendly. These characteristics have motivated the design of the system for distributed measurement, based on three-tier architecture, Java programming language and Web technologies.

The system architecture is shown on figure 1. From the view point of the three-tier architecture, it consists of standard Web browsers located at the client tier that provides an interface to other applications or operators; Web/application server located in the application tier that realize presentation and application functions; and data producer components – controller networks and database servers – located in data tier.

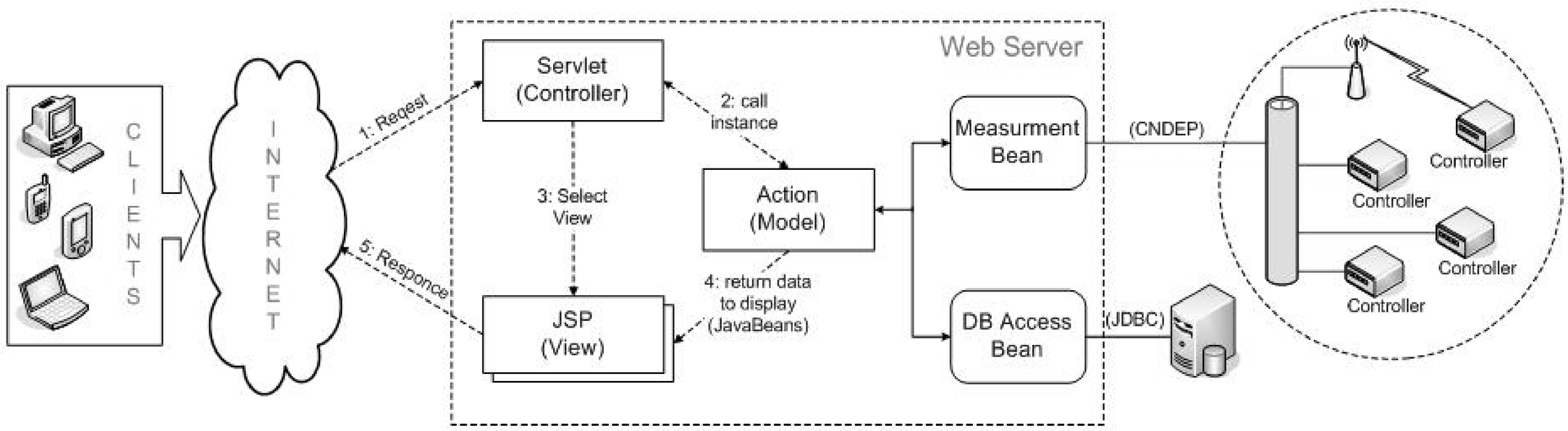


Figure 1. Architecture of a Web-based, three-tier system for distributed measurement.

### Implementation of a Distributed System for Temperature and Humidity Measurements

An application of the architecture from figure 1 has been developed for measurement of temperature and humidity. Most of the implementation issues concern the middle tier. Its implementation is based on Sun Microsystems's Java Server Page (JSP) technology.

The JSP technology uses the MVC architecture. Controller functions are handled by a servlet (figure 1). It processes all HTTP requests and determines the appropriate object from the model and appropriate view. The servlet also offers authentication and validation services.

The model consists of two components – Measurement Bean and DB Access Bean (figure 1). The Measurement Bean component is a permanent object for the application. It is instantiated once at the first request received. Its functions are to contact the remote controllers that measure temperature and humidity and to collect these values in internal variables. This is done periodically in intervals of 10 seconds. This interval is chosen because decrease of the interval will not gain more informativeness to the users. On the other hand, with the decrease of the interval a disruption of the controllers' performance is observed. Traditional sockets and CNDEP protocol are used for connection with the controllers.

```
<jsp:useBean scope="application" id="Measurment"
             class="temperature.MeasurmentBean"/>
...
<table>
  ...
  <TR align=\"center\">
    <TD><jsp:getProperty name="Measurment" property="tempTini"/></TD>
    <TD><jsp:getProperty name="Measurment" property="humTini" /></TD>
  </TR>
  ...
  <TR align=\"center\">
    <TD><jsp:getProperty name="Measurment" property="tempIpc"/></TD>
    <TD><jsp:getProperty name="Measurment" property="humIpc"/></TD>
  </TR>
  ...
```

Figure 2. Formation of the response using a JavaBean.

The view consists of various JSP pages – for observing of measurement values and for statistical data, for HTML and WML clients and etc. The formation of a HTTP response with temperature and humidity data from the Measurement Java Bean component is shown on figure 2 and the view of the response in the client's browser is shown on Figure 3.

A JSP page may contains all standard elements of an HTML page (HTML tags, content text, etc.) among with some Java code incorporated as action tags and scriplets. Figure 2 exposes the use of the *jsp* tag library for extracting dynamic data from a Java bean. The *jsp:useBean* directive is used to initialize the Measurement bean. The scope *application* means that the bean will be shared for every Servlet and JSP in the web application, thus the same bean will be used every time. The directives *jsp:getProperty* are used for extracting of temperature and humidity values from the Measurement bean.

DS TINI

| Temperature | Humidity |
|---|---|
| 21.98 | 40.12 |
| Status | OK |
| Location | lab 2104 |

IPC@Chip

| Temperature | Humidity |
|---|---|
| 20.42 | 37.46 |
| Status | OK |
| Location | lab 2104 |

**Sat Apr 29 17:36:43 EEST 2006**

Figure 3. User Interface.

For the Web server an Apache Tomcat 5.5.14 servlet container from Apache Software Foundation is used. It is run on a 333MHz Pentium II machine, with 256 MB operational memory, and OS Debian Linux 2.4.27-2-386. The system is tested and it proves to operate correctly with PC, PDA, and Cellphone clients.

## CONCLUSION AND FUTURE WORK

The presented system shows the possibility for adaptation of open and standardized approaches, well-proven architectures, high level abstraction, and standard system protocols in distributed measurements. The use of off-the-shelf solutions, like internet browsers and web servers, brings various advantages and gives the system added flexibility and scalability.

Web servers have a build-in support to high loads, and authentication mechanisms. They are well tested and supported and various development tools exist. The standard user interface used – web browser, allows access to the system from various clients.

Some possibilities for future work include adaptation of web service architecture and employing rich client applications. Web services can be adopted by the controllers offering services like measurement of temperature and humidity. The web server will use these services basing communication on SOAP/HTTP. Rich client applications can be employed for clients having enough resources.