

## Metering devices remote monitoring system Web-interface developing

Alexey Lagunov  
 Northern (Arctic) Federal University  
 named after M.V. Lomonosov  
 Severnaya Dvina Emb. 17,  
 Arkhangelsk, Russia, 163002  
 Email: a.lagunov@narfu.ru  
<http://www.narfu.ru>

Dmitry Fedin  
 Northern (Arctic) Federal University  
 named after M.V. Lomonosov  
 Severnaya Dvina Emb. 17,  
 Arkhangelsk, Russia, 163002  
 Email: d.fedin@narfu.ru  
<http://www.narfu.ru>

Nikita Gerasimov  
 Northern (Arctic) Federal University  
 named after M.V. Lomonosov  
 Severnaya Dvina Emb. 17,  
 Arkhangelsk, Russia, 163002  
 Email: n.gerasimov@narfu.ru  
<http://www.narfu.ru>

**Abstract**— Recently appeared necessity of enterprises, homes and other objects energy consumption information automatic collection. For this purposes various metering devices by different vendors have been created. While using different metering devices, problem of central information collecting appears. This paper raises the problem of creating a web-interface for metering devices remote monitoring system for data collection from the various sensors and meters.

### I. INTRODUCTION

QUANTITY of sensors by different vendors, collecting different information leads to different meters managing software packages quantity. For this problem solution was required universal hardware-software system, which would can connect to any metering device, get every data, write it in to a database and show in user-friendly format. Common metering devices remote monitoring system scheme presented at “Fig. 1”.

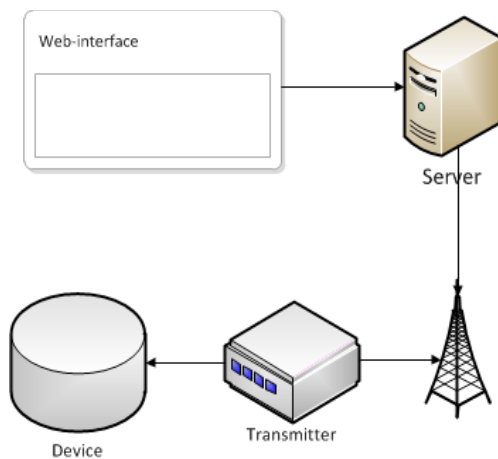


Figure 1. Metering devices remote monitoring system common scheme

Read out by metering device, data is transmitted to the server for following saving to DB. After these actions it can be shown to the customer in user-friendly view. In our case user-friendly format is provided by web-interface. Web-interface is adjusted such way to have ability to show every data types, collected from every sensor. Common web-interface structure is at “Fig.2”.

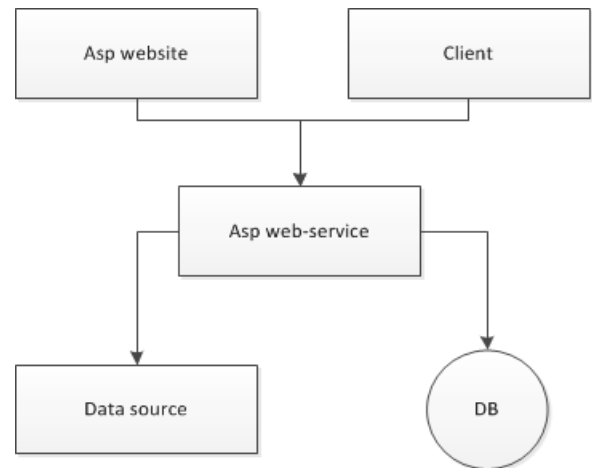


Figure 2. Web interface common model scheme

We decided to use ASP.NET framework because developing simplicity, reliability. ASP.NET framework provides all required tools for creating web-portals: implementation of SOAP REST protocol for web-service and web-site communication, database drivers. Also SAP provides simple reports generation integration. Other variants: PHP or Java requires manual implementation of above-mentioned functionality [1].

### II. DEVICE

Modern equipment installed in the pipe rooms, allows remotely collect data from the meters using wired and wireless communication channels. The majority of the existing systems is focused on a specific manufacturer of the energy counter. Many organizations have counters of different manufacturers and different modifications. Each manufacturer has its own data collection system, often not compatible with the systems of other manufacturer. The device allows to collect the data from the meters. Transfer of the received data is carried out through GPRS. The device includes the printing cost, which contains a power unit, a sim card, slot antennas, signal RS-232 v converter, signal CAN interface converter, on the circuit Board is additionally available programmable GSM module, independent RS-232 and CAN interfaces.

Performance indicators:

- ambient air temperature from -20 to +50°C;

- relative humidity 95% at 35 C;
- power supply 12 Volts +/- 30%, 50 Hz;
- work mode - continuous;
- validity period – 3 years.

### III. WEB-INTERFACE DATABASE DEVELOPING

#### A. Object structure scheme

For simplicity of object geographical managing and structuration ease we decided to allocate all objects, including geographical zones and meters, to different levels with private proprieties and inheritance ability. Common indication storing database diagram is presented at “Fig. 3”.

Such scheme allows creating a root level, representing some geographical object with successor: another geographical object or metering device. First level always represent meter. Every object has its own unique proprieties, user privileges. Such scheme allows describing almost every structure, allows managing it conveniently, allows simply define every device.

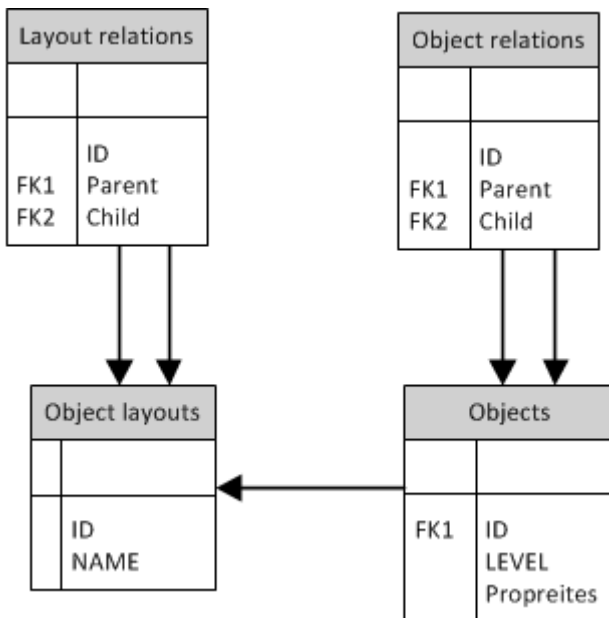


Figure 3. Object information storing database scheme

#### B. Data storing scheme

Collected data is storing in such database structure, presented at “Fig. 4”:

- ID
- Data type ID
- Object ID
- Date and time
- Data relevancy duration in hours
- Applied function
- Value

Thus in the database data with data and relevancy period and applied function are storing. The “Applied function” field describes is the data average or total for some time period. Such structure allows storing collected information and saving user reports calculation results, e.g. average daily re-

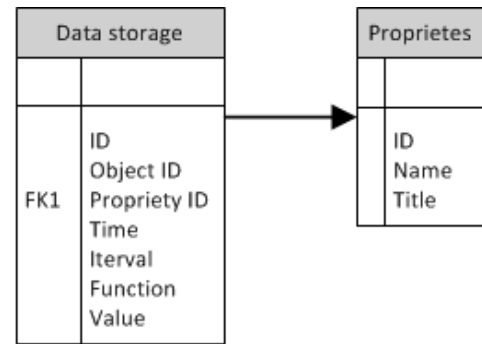


Figure 4. Collected data storing database scheme

sults. After collection data is being processed to get average daily results, is sending to a customer and saving in to the database with such structure:

- Data type ID
- Object Id
- Date with time equal 00:00
- Relevancy duration equal 24 hours
- Applied function - average
- Value

At first time system doesn't calculate any information in such way, database stores only information, transferred from metering device. In future will be realized algorithm, calculating required data

First variant was based on data division on several tables by period, but such way turned out to be inflexible.

Firstly our system contained every object info entry in specialized table field, but after level-oriented structure introduction, all information stores in united table, which containing object identifier, field name and field value: “Fig. 5”.

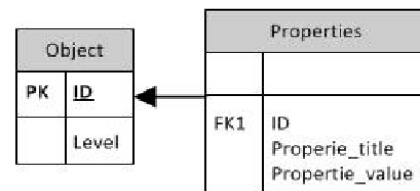


Figure 5. Properties storing.

As alternative to such table, arrays can be used, but .NET PostgreSQL drivers doesn't allow to use this feature.

Other database scheme part is rather typical for web-application: tables for user information and settings storing.

For data storing we choose DBMS PostgreSQL 9.x. Such decision was made for PostgreSQL high efficiency, high reliability and functionality. PostgreSQL allowed us to devide different databases to different schemes, also PostgreSQL allows to easily DB ranging with tables partitioning [2].

### IV. ASP.NET WEB-SERVICE DEVELOPING

Web-site is developing with ASP.NET technologies set.

Web-site schematic view is presented at “Fig. 6”. Prerelease web-site views are at “Fig. 7” and “Fig. 8”.

For widget system creation we use Droptings framework, which requires only quick widget creation for web-portal creation.

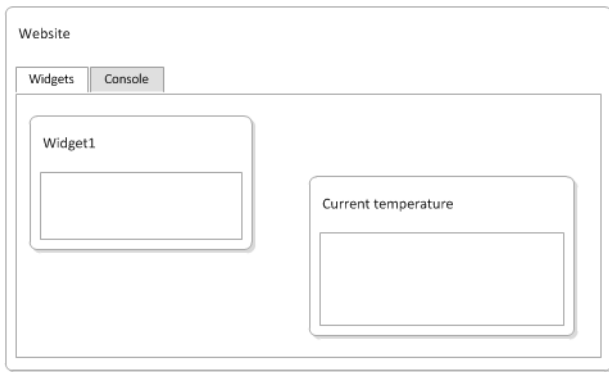


Figure 6. Common web-site scheme

“Widgets” tab at common scheme contents set of various configured and arbitrary placed widgets. Widgets quantity is limited only by self sizes and website workspace size. Every widget can be created several times at one tab workspace with individual settings [3].

Widgets examples:

- Last data, collected from specified device.
- Maps with geographical and/or meters location.
- Indicator of deviation of specified metering device collected information by type and period

Console allows to:

- Create typical reports for different devices or device groups energy consumption
- Manage data collection time intervals
- Etc.

Console is also developed as widget, despite the widget usually executes tiny procedure.

While web-site is developed with Droptings framework, web-service is developed using common MVC pattern. Scheme is presented at “Fig. 8”. United Service class is created for more simple SOAP client using at the web-site side, also it is “View” component. When user asks data for object

(city, street or metering device), controller creates this instance of “ObjectInfo” class, that provides self-calculation. After object instance determinates: is it high-level object or counter, it creates instances of child-objects or instance of value provider. If level is high, instance asks children to self-calculate else asks specified value provider to self-calculate. After all data is got, instance calculates it, gets its reliability and returns to controller or high-executed instance. Better to say, it is recursion. Reliability is created for data integrity mark, it is also calculating in value and object provider.

For reporting system we decided to use CrystalReports for its functionality and free MS Visual Studio version. As CrystalReports requires previously defined database structure, but our targets requires to create new reports without recompilation we developed several ways to make datasource for reporting system. Both based on one DataSet table, containing one field. First using parameter-based Crystal Report and dynamic DataSet. Second using XML document for creation Crystal Document and also dynamic Dataset. For simplicity we using second variant. Using report sample we decide needed values, then form special XML document, containing DataSet structure. Using this XML document Crystal Report is creating and registering at the system. Then, when user asks system for the report, web-site generates DataSet with structure, similar to XML file, fills it with data and executes CrystalReports to generate the report.

While using reports system, we met one more trouble: using Droptings and CrystalReports with advanced documents led to JavaScript errors. This was solved with blank template.

## V. FIGURES AND TABLES ASP.NET WEBSITE AND ASP.NET WEBSERVICE COMMUNICATION

For website and web-service communication is used SOAP protocol, which is fully supported by MS .NET framework.

Information about users, user’s sessions, user’s success and failed logins is stored at database. While authentication process client sends “login” and “password” pair to the web

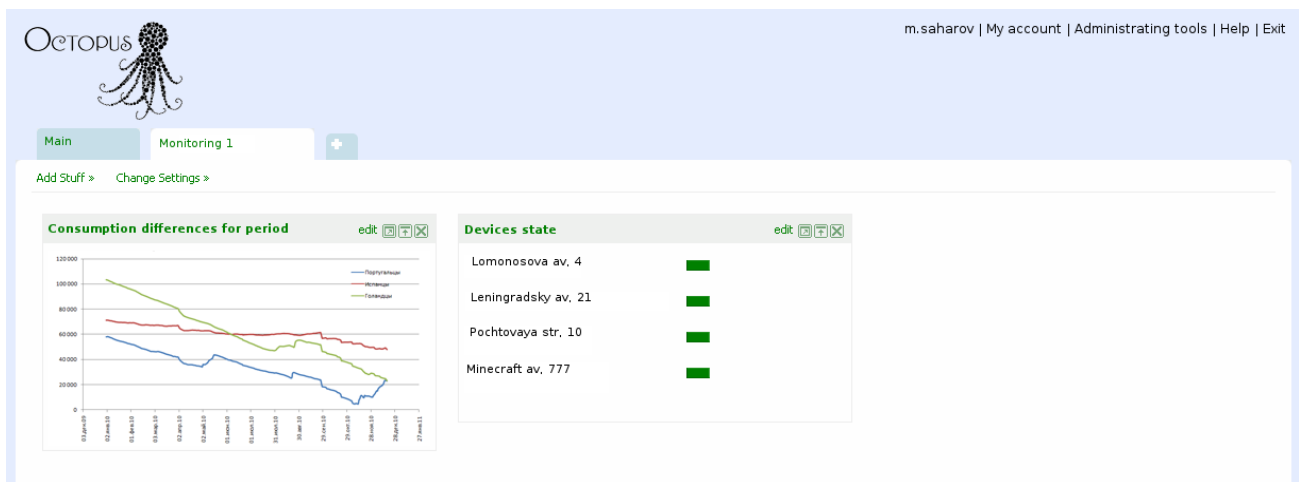


Figure 7. Widgets

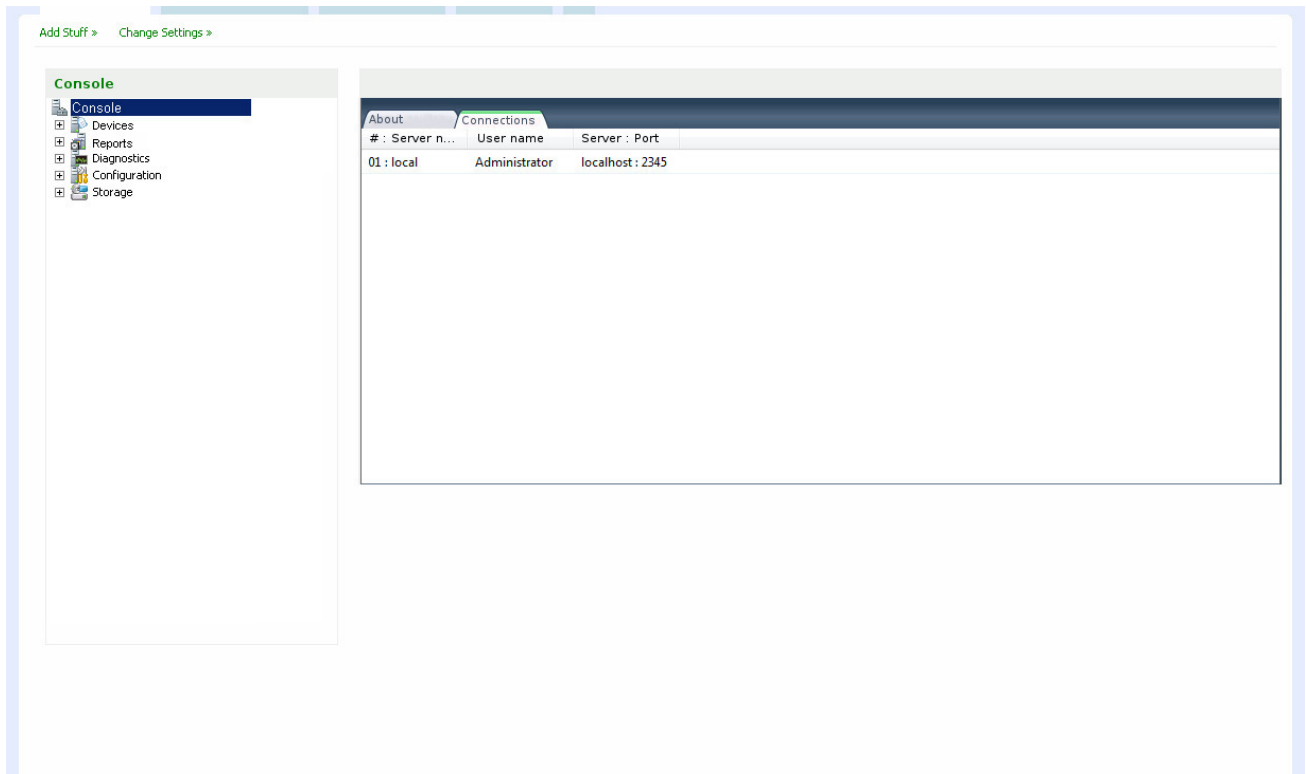


Figure 8. Console

service, who writes session information, session duration and session key in to the DB. Session key is sending with reply to the website. In case of invalid login-password pair client gets “SOAP Exception” message and web services writes information about failed authentication attempt.

After successful authentication, while new .NET session is starting, client sets saved session key, using special web method. Actual session key sets one time per .NET session.

In case of every fail execution client gets SOAP Exception message with details.

## VI. CONCLUSION

System which can provide to the customer previously processed, user friendly consumption information, can save processing results was created at result. Such system allows

working with various device types by various vendors and with various data types. This system has a low threshold of entrance, it is easily set up, it is informative for all categories of customers. It doesn't require any installation and setup.

## REFERENCES

- [1] Matthew MacDonald, Mario Szpuszta, “Pro ASP.NET 3.5 in C# 2008”, ISBN: 1590598938, November 15, 2007
- [2] Hannu Krosing, Kirk Roybal, “PostgreSQL Server Programming” ISBN: 1849516987, June 2013
- [3] Bill Scott, Theresa Neil, “Designing Web Interfaces”, ISBN: 978-0-596-51625-3, 2009

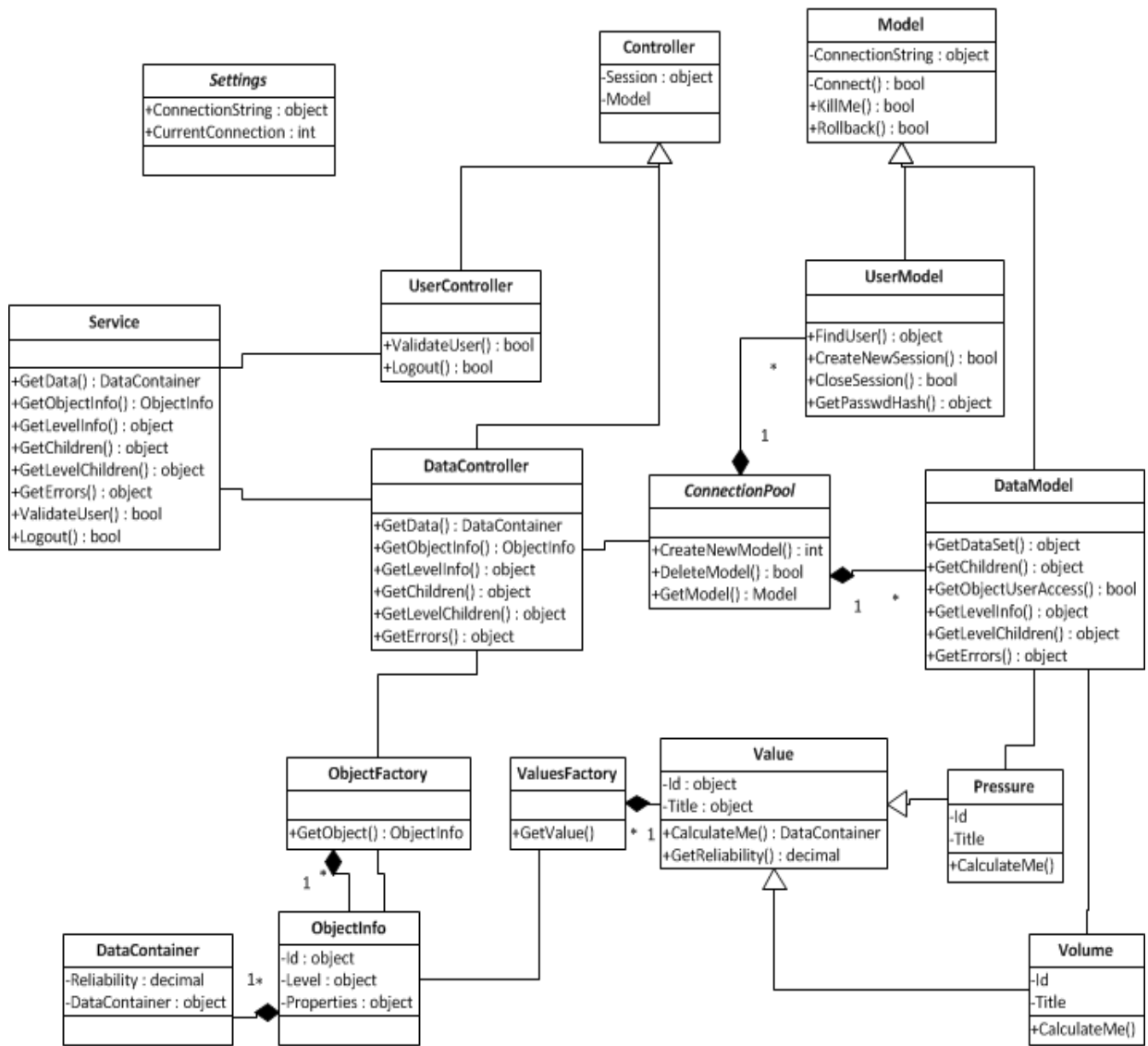


Figure 9. Web-service scheme