A Web Service for Data Visualization in Distributed Automation and Information Systems

Mitko Shopov, Nikolay Kakanakov, Grisha Spasov
Technical University of Sofia, branch Plovdiv, Plovdiv, Bulgaria

Abstract: *This paper is an attempt to continue development of the multi-tier model for distributed automation systems proposed in previous work. It suggests the use of an open-source Web service application for generation of graphical plots from given data series. The implementation of the service uses the popular open-source tool Gnuplot with capability to generate 2-D and 3-D graphics. The suggested Web service – WebPlotter, is intended to work on the Service tier of the multi-tier model and to offer its services to the applications working on the same tier or the tier above (Presentation tier). The data series should come from the controller networks working on the tier below – Data tier. On the upper tiers a Web based interface for configuration of the various Gnuplot parameters is working. The WebPlotter service is supposed to provide a more convenient and user-friendly view of the various collected measurement parameters from the controller networks.*
Keywords: *Multi-tier Architecture, Web Services, XML encoded binary, Charting, Attachments, Data Acquisition.*

1. INTRODUCTION

Nowadays, the Web Service Architecture has become the most adoptable choice for building highly interoperable and scalable distributed systems, especially for application that require interoperation of different web based blocks of the entire system. Development of complex measurement and data acquisition systems based on the principles of Information systems need to be easily expandable and architecture independent. The data collected by these systems should be presented to the user in tables, graphics, and charts so it is more readable. These functions can be realized as web services because web service architecture is a proven concept in application-to-application interactions across the Internet. These services should be expandable and implementation independent as long as easily accessible and discoverable [1, 2, 9, 10].

The paper presents a realization of a Web service that receives a set of data series and as a result produces an image – a chart representation of the data series. It also supports different configuration parameters for the modification of the chart view and type.

2. BACKGROUND AND RELATED WORK

2.1. Multi-tier architecture

Web-based multi-tier architectures are traditionally used for database applications. In recent years, with the attempts to integrate controller/plant network to the enterprise network and Internet they become a popular solution for building distributed automation systems. Multi-tier architecture provides many benefits over traditional client/server architecture by allowing designers to achieve greater flexibility and finer granularity of the systems and its components and to benefit from the inherited advantages of Web-based interface. 3-Tier architecture is the most commonly used for building Web applications. It allows separation of business logic from display and data, but, it does not focus on specifics of functional layers. The application server still has too many functions grouped together. This reduces flexibility and scalability. N-Tier Architectures provide finer granularity, which provides more modules to choose from as the application is separated into smaller functions [6, 12].

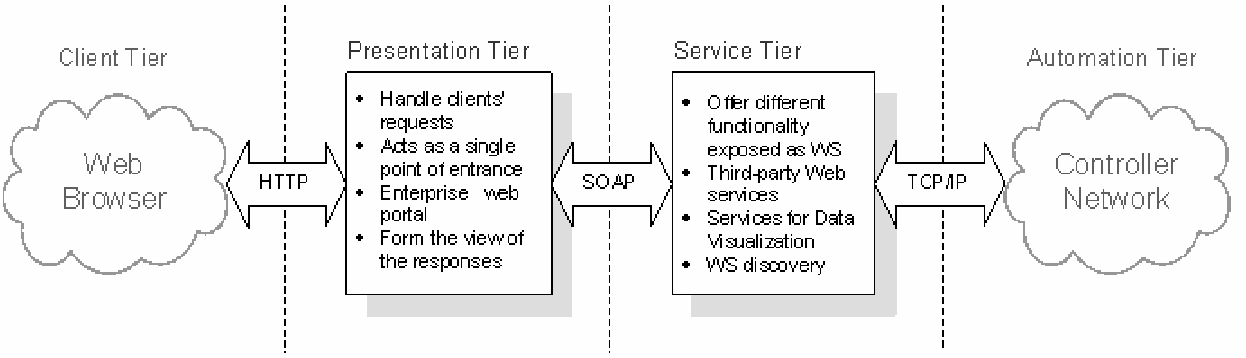

Fig. 1: N-Tier model for Distributed Automation – data visualization.

One example of an N-Tier model for distributed automation system is proposed in [7] – figure 1. This model generally consists of four tiers, separated in their functionality and administration. The client tier works on the top of the model. Clients request services from the system using regular Internet browser or via web services. Different responses can be constructed depending on the client's platform (PC, PDA, cell phone). The presentation tier is responsible for handling the requests and forming the responses. The requests are analyzed and dispatched to the appropriate service on the next tier. The server working on this tier is called enterprise Web Portal and is based on the portal technology. On the services tier the main functionality of the model is placed. Different services may work on different servers, so failure of a single server would affect only the corresponding service. This modular approach increases the flexibility and reliability of the system. The data tier role is to produce and/or store data. It depends on the corresponding upper tier server. It can be a Database or a network of controllers.

Data visualization and presentation to the users occur on the two middle tiers of the N-tier model (figure 1). Upon request from a client, the presentation tier contacts a data collecting service from the Services tier to extract the data needed. It also prepares the appropriate settings for the data visualization Web services and invokes it to plot appropriate graphics and charts. The presentation tier than uses the result to generate the response to the client.

2.2. Web service attachments

As long as graphics are stored as pictures, it is a challenge to decide the way they are enveloped in the SOAP message. There are several approaches for doing it: e.g., SwA (SOAP with Attachments), WS-Attachments, embedding data in XML. In SwA and WS-Attachments data is transmitted outside the SOAP message using MIME or DIME specifications. Embedding the data in XML is done by writing the binary data as a sequence of bytes. The usage of binary data embedded in XML fit with the SOAP specification but leads to significant increase of data size. There are cases where serializing data into XML is unwise because an efficient binary compression scheme is already required to insure that the data is not too large. Another option for transmitting binary data is by URI. That option is rarely used because the actual transfer of data is not encapsulated in SOAP message [5, 8].

The usage of SwA and WS-Attachments leads to the presence of two data models which is a security leak. The industry is rapidly adopting XML-based security mechanisms designed for use with the XML data model (and in the case of WS-Security, the SOAP data model). When a second data model is present (e.g., multipart MIME, DIME), additional (and yet to be specified) measures must be taken to ensure the integrity and confidentiality of the non-XML data. For example, a digital signature over a SOAP envelope does not necessarily protect any data referenced by embedded URI [3].

There are experiments on evaluation of web services implementation in different platforms. Davis and Parashar [4] provide results for latency performance of MS SOAP tookit, Apache SOAP and Apache Axis. The Apache Axis implementation provides shortest response times and deviations for services with string or array data (at the range of 20 miliseconds). Ying et al [11] compared the latency of data transmission of arrays of different sizes using XML encoded binary, SwA and DIME. For small size arrays (less than 50x50 matrices) the first method work fastest, for medium size arrays DIME is the fastest method (the first method works 5-6 times slower), and for big size arrays SwA and DIME shows same latency. These results show that performance of SOAP could be significantly improved by the introduction of SwA. Moreover, the SwA-DIME is proved to perform faster and more efficiently than SwA-MIME, but SwA-MIME has the advantage to be supported by Web Service Definition Language (WSDL) specification.

3. USE CASE SCENARIO

3.1. Gnuplot utility

Gnuplot [13] is a portable command-line driven interactive data and function plotting utility available for many platforms. It is freely distributed open-source software that was originally created for the needs of academia to visualize mathematical functions and data. It has grown to support many non-interactive uses, including web scripting and integration as a plotting engine for third-party applications. Gnuplot supports many types of plots in both 2D and 3D. It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text. It also supports various specialized plot types. Gnuplot also supports many different types of output: interactive screen terminals (with mouse and hotkey functionality), direct output to pen plotters or modern printers (including postscript and many color devices), and output to many types of file formats (jpeg, LaTeX, pdf, png, postscript, and etc.). It is easily extensible to include new devices. Because of these characteristics the Gnuplot utility is chosen as the plotting engine of the data visualization Web service.

3.2. Functional design

The use case scenario presented in figure 2 is realized in Distributed Systems and Networking – Virtual Laboratory [14]. Some preliminary experiments are taken out as prove-of-concept of the implementation of visualization and charting service. The experiments are made to test functionality, ease of use, reliability and interoperability of this particular part of the multi-tier architecture and its use in distributed data acquisition and measurement systems.

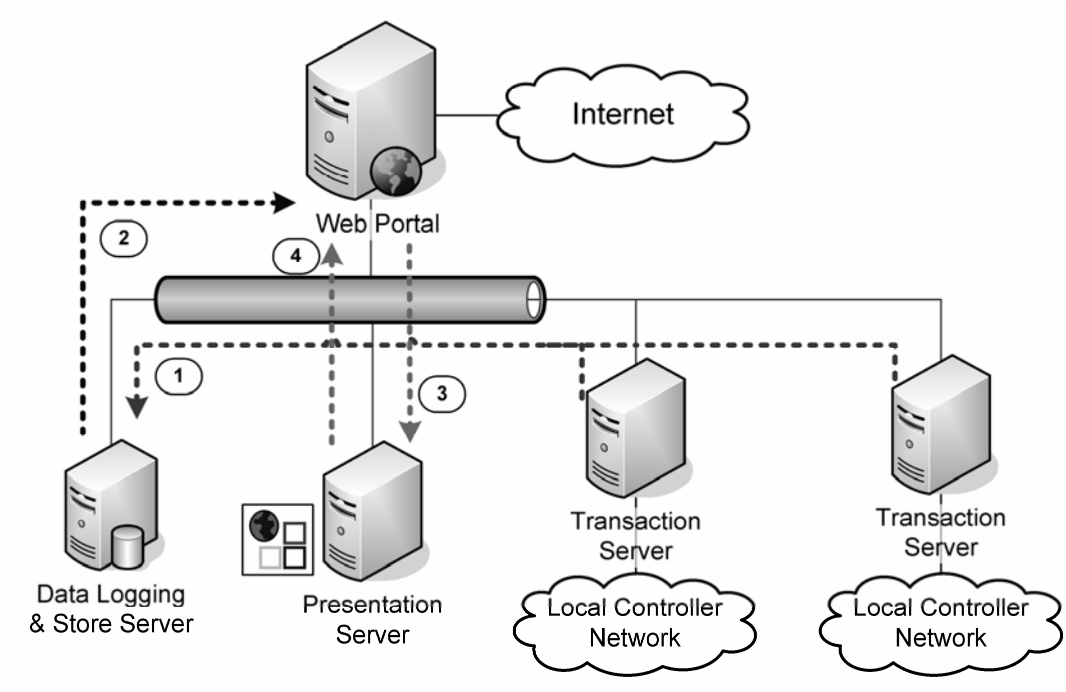

Fig. 2: Use case scenario architecture.

The presented use case scenario implements a distributed data acquisition system for measuring of environmental parameters of remote objects. The implementation is based on the multi-tier architecture presented in section 2.1. Different controllers are sensing the parameters of the environment and sent the sensor measurements periodically to a transaction server that collects them. A storage server collects the data from all transaction servers and stores it. The server on the presentation tier gets series of data from the storage server and calls the WebPlotter service with them. The produced image is used for presentation of the measurements to the clients. Clients can choose what data, for what periods to view and can configure the properties of the resulting chart (lines, points, scale, dimensions, etc.) (figure 3).

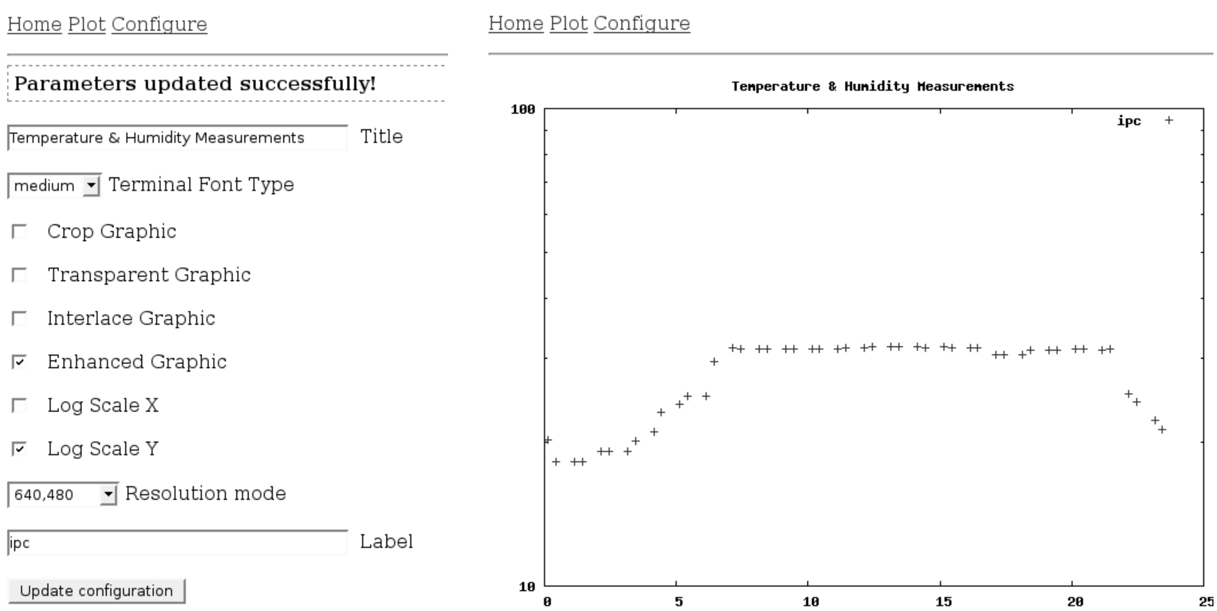

Fig. 3: WebPlotter client interface.

4. PRELIMINARY EXPERIMENTAL RESULTS

The preliminary experimental results provide some data on the latency performance and message size of the WebPlotter service in the context of the use case scenario presented in the previous section. The transfer of graphics uses two methods: binary encoded XML and SOAP with attachments. The delay is measured with the use of High performance timer – available in .NET framework. This timer provides a precision of less than a millisecond. To exclude the time needed for graphic generation a predefined graphics are used with size of 150 Kbytes and 7 Mbytes. The transfer of graphics with SOAP attachments took about 100 milliseconds for the small graphic and about 400 milliseconds for the bigger one. The transfer with binary encoded XML took about 200 milliseconds for the smaller graphic and 4000 milliseconds for the . All values are taken in local area network and communication delay for network paths and delays in Internet is not included.

During the experiments all packets are captured and analyzed. The server receives the request and respond with HTTP code 100 (Continue) for about 3-4 milliseconds. The measured latencies are taken only in the cases when no retransmission of TCP segment occurs. All other latencies are accumulated from the transfer of the graphic. For very small sized graphics (640x480, 3Kbytes) the transfer is within 3 TCP segments and the latency is small. For bigger graphics more segments are needed and it took some time for reordering and reassembling of the application data. The generation of the graphics on the server with the Gnuplot utility depends on the amount of data in the series. For bigger data files some latency is added from their transfer and generation of the graphics.

5. CONCLUSIONS AND FUTURE WORK

The paper presents a Web service for visualization of various data extracted from automation and information systems – Webplotter. It is working in distributed environment and can be integrated into systems with variety of applications like industrial process monitoring and automation, weather forecasting, agriculture, safety and fire alarms. The main advantage of the Webplotter service is its ability to be easily distributed and reconfigured and its independence from the specifics of the systems application due to the use of web service middleware and component-based approach.

The preliminary test and experiments prove that the service is functional and is applicable in the chosen use case scenario. SOAP with attachments has better performance especially in the transfer of big sized binary data. But these approach has some inherited problems with the standardization and security. Using of attachments has some problems with the interoperability of applications implemented in different platforms and program languages (Microsoft relies on DIME, Java relies on the DataHandler data type). In the presented application of the WebPlotter service the size of the graphics is not expected to be more than 100Kbytes and the use of XML encoded binary data (e.g. HexBinary, base64Binary imbedded in the SOAP body) is the prefered choice, despite of the bigger latencies.

Some future work includes analysis of the data flows in the context of the multi-tiered architecture. These include the communication between different services working on the Service tier and the flow of the measured data from the source to its final destination – another web service or a web client. Another direction for future work includes further analysis of the delay performance of WebPlotter service and how this delay will influence the rest of the system and its time requirements.

6. ACKNOWLEDGMENTS

The presented work is supported by National Science Fund of Bulgaria project – "ВУ-966/2005", entitled "Web Services and Data Integration in Distributed Automation and Information Systems in Internet Environment", under the contract "ВУ-МИ-108/2005".

7. REFERENCES

[1] Alonso, G., 2002, "Myths around Web Services," *IEEE Data Engineering Bulletin*, Vol. 25, no 4.
[2] Borriello, G. and R. Want, 2000, "Embedded Computation Meets the World Wide Web", *Communications of ACM*, Vol. 43 no 5, pp. 59-66.
[3] Bosworth, A., D. Box, M. Gudgin, M. Nottingham, D. Orchard, J. Schlimmer, 2003, "iXML, SOAP and Binary Data," Website: http://www.xml.com/pub/a/2003/02/26/binaryxml.html, [May, 2008].
[4] Davis, D., M. Parashar, "Latency performance of SOAP implementations," In 2nd IEEE International Symposium on Cluster Computing and the Grid, pp. 407-412, 2002.
[5] Deem, M., 2002, "WSDL Extension for SOAP in DIME," Website: http://www.gotdotnet.com/team/xml_wsspecs/dime/WSDLextension-for-DIME.htm, [May, 2008].
[6] Jammes, F., H. Smit., "Service-Oriented Paradigms in Industrial Automation, Industrial Informatics," *IEEE Transactions* on Vol.1, Issue 1, Feb. 2005 pp. 62 - 70.
[7] Kakanakov, N., M. Shopov, G. Spasov, A New Web-based Multi-tier Model for Distributed Automation Systems, Information Technology and Control, (in press – March 2006).
[8] Powell, M., 2002, "Understanding DIME and WS-Attachments," Microsoft Corp. Website: http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/dimewsattch.asp, [May, 2008].
[9] Topp, U., P. Müller. Web based service for embedded devices. 2001.
[10] Vinoski, S. Where is Middleware, IEEE Internet Computing, March/April 2002, vol. 6, no. 2, pp. 83-85.
[11] Ying, Y., Y. Huang, and D. W. Walker, "A Performance Evaluation of Using SOAP with Attachments for e-Science," In Proc. of the UK e-Science All Hands Meeting, pp 796–803, 2005.
[12] Youngblood, G. M., Smart Environments, Ch. 5: "Middleware", pp. 101-127, 2004, ISBN: 0-471-54448-5.
[13] Gnuplot official Web site – http://www.gnuplot.info
[14] Virtual Laboratory of Computer Networks and Distributed Systems, website: http://dsnet.tu-plovdiv.bg/.