An Application of Web Services for Distributed Measurement

Nikolay R. Kakanakov
Technical University of Sofia branch Plovdiv
kakanak@tu-plovdiv.bg

Abstract

The paper discusses the application of a state-of-the-market middleware technology of Web Services in the field of distributed measurement. An implementation of a system for distributed measurement in Wide Area Networks using Web Services is proposed. The presented system is built on a multi-tier client/server approach. The implementation presented includes Web Services for temperature and humidity measurements, for data logging (database) and for data presentation (tables, graphics, etc.). The communication with the measuring devices is by means of custom UDP based protocol – CNDEP (Controller Network Data Extracting Protocol). The application is deployed in the "Virtual Laboratory of Computer Networks and Distributed Systems" in Technical University of Sofia, branch Plovdiv.

1 Introduction

Advances in Computer science and technologies over the past few years have led to vast expansion of the field of their application. Technologies for remote and distributed measurements are on the market for over a decade and they evolve trough the growth of communication technologies and Internet. The problem they face is the expensiveness of infrastructure and leak of standards for communication and access. The need for new communication infrastructures for embedded and ubiquitous systems has led to the idea of technology transfer from Business systems to distributed measurement systems. The similarity between these two environments is the need of data organization, visualization and transfer by means of transparent and adaptable middleware technology (Borriello and Want, 2000; Youngblood, 2004).

TCP/IP protocols and Web technologies has already been used for implementation of systems for remote measurement. There are applications based on Java Applets, JSP, CGI, ASP or other dynamic Web technologies (Pianegiani et al., 2003; Topp and Mueller, 2001).

Using Web Services is only the next step, allowing distribution over large distances. It satisfies most of the fundamental requirements of such systems, like: cross-enterprise cooperation, inter-enterprise dynamic integration, scalability and re-configurability, supporting of heterogeneous yet interoperable hardware and software environments (Jammes and Smit, 2005; Kakanakov and Spasov, 2005).

2 Service Oriented Paradigms

As defined in (Jammes and Smit, 2005) a service-oriented architecture (SOA) is a set of architectural tenets for building autonomous yet interoperable systems. The key words in this definition are autonomous and interoperable. Autonomous systems are defined as systems which are created independently of each other, provide independent functionality, and can be administered individually. Interoperability is derived from implementing an abstract interface for inter-systems communication. The SOA systems are design in conjunction with the business process rules – each service definition is derived from a business task (Jammes and Smit, 2005).

As described in (Alonso, 2002) Web Services are a series of specifications for implementation of SOA architectures. They have three main components: service provider, service requester, and a service registry. The main standards defining Web services are: UDDI (Universal Description, Discovery and Integration), WSDL (Web Services Definition Language), SOAP

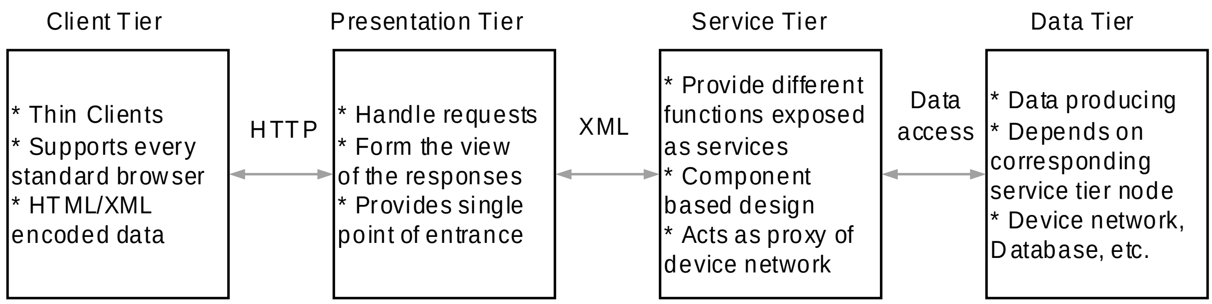

Figure 1: N-tier client/server model.

(Simple Object Access Protocol) and XML (eXtensible Mark-up Language).

The advantages of using Web services are in different directions. First, it is their ubiquitous infrastructure – they can work over different Internet protocols (SMTP, HTTP, TCP, UDP). They also incorporate the proven approaches from other middleware technologies and provide both RPC-based and message-oriented communication. Web services are called "middleware for middleware" (Alonso, 2002; Vinoski, 2002).

3 Multi-tier client/server approach

Data management in complex business entities is not an easy task. Administering and re-configuring such complex systems is very tricky. This has led to layered architectures where administration and work of different layers is done separately and independently. Functions of data storage, data presentation and data processing are to be separated logically and physically. Distributed measurement is not considered to work alone, but to interoperate with other parts of organization's software. The interoperability is achieved by using universal interface and data encoding (Web services), popular communication protocols (HTTP, TCP, IP), and a flexible and scalable architecture (multi-tier client/server) (Kakanakov and Spasov, 2005; Kakanakov et al., 2006b; Youngblood, 2004).

In (Kakanakov and Spasov, 2005) is proposed model for adaptation of Web service architecture in industrial automation and measurements. It is based on the popular three-tier business model and provides distribution abilities. The middle tier is used as a proxy between Web services and embedded devices which provide automation (or measurement) functions. It avoids using expensive embedded devices with enough resources for XML parsing and SOAP processing (Kakanakov and Spasov, 2005).

This idea is expanded to N-tier model for distributed automation system, as shown in (Kakanakov et al., 2006b). The model has four separate tiers which increase distribution abilities and scalability (Figure 1). The middle tier is separated in two tiers – presentation and service. The presentation tier is responsible for the organization of data for best view on different clients. It also provides single entry point for the entire system which makes it simple to access and work on. The service tier incorporates the individual services provided in the plant: e.g., measurements, storage, processing, etc. The communication between these two tiers is expected to be based on middleware technology suitable for distribution of nodes over Internet or Intranets. Each tier can interact only with its direct neighbors, which increases the security – private data on the data tiers cannot be directly accessed from the client tier (Kakanakov et al., 2006b; Youngblood, 2004).

4 Web services for remote measurement

Remote measurement systems based on the Web technologies are a promising trend in computer systems. The main disadvantage in systems based on custom communication infrastructures (Pianegiani et al., 2003) is that they are hard to re-configure and suffer from a lack of common interface for application-to-application interaction. These problems are solved using multi-tier client/server approach (Kakanakov et al., 2006b) and Web services (Jammes and Smit, 2005; Kakanakov and

Spasov, 2005; Topp and Mueller, 2001).

Application of Web services in such complex models can be on different level. Device level services are examined in (Jammes and Smit, 2005) and (Topp and Mueller, 2001). They finger out that Web services can be applied for remote access of embedded device for data extraction and function call. The problem is that most embedded devices have not enough resources (processing power, memory) to run XML parser and SOAP processor. Another problem is that the reconfiguring all nodes at device level is in common cases a hard and expensive task. The model for adaptation of Web services in distributed automation presented in (Kakanakov and Spasov, 2005) offers a proxy based approach that keeps the existing infrastructure at device level but do not suggests model for distribution. The N-tier model proposed in (Kakanakov et al., 2006b) goes one step further. It suggests that physically distributed plants of a business which interacts by means of service-oriented middleware, can have centralized access and administration.

The implementation presented in the paper is based on the N-tier model (Kakanakov et al., 2006b) and service-oriented inter-model communication (Jammes and Smit, 2005; Kakanakov and Spasov, 2005; Topp and Mueller, 2001) in order to achieve distribution and scalability in relatively low price.

5 Sample implementation

The presented application is developed and deployed in “Virtual Laboratory of Computer Networks and Distributed Systems” in Technical University of Sofia, branch Plovdiv (Spasov et al., 2006). It is based on the N-tier model presented in (Kakanakov and Spasov, 2005) but do not implement all of its parts. The implementation covers the Service and Data tiers of the model.

5.1 Functional design of the system

The system is designed to examine the ability to use Web services as building block for complex remote measurement systems. It consists of three separate services, interoperating between each other in order to store, execute and preview measurements. The services are implemented as a part of distributed system for remote measurement and control. They run as services on the Service tier of N-tier client/server model for distributed automation (Kakanakov and Spasov, 2005). Each of them is deployed on separate machine and machines are installed in different segments of TCP/IP network for testing the interaction in Internet environment. The functional scheme of the system is shown of Figure 2.

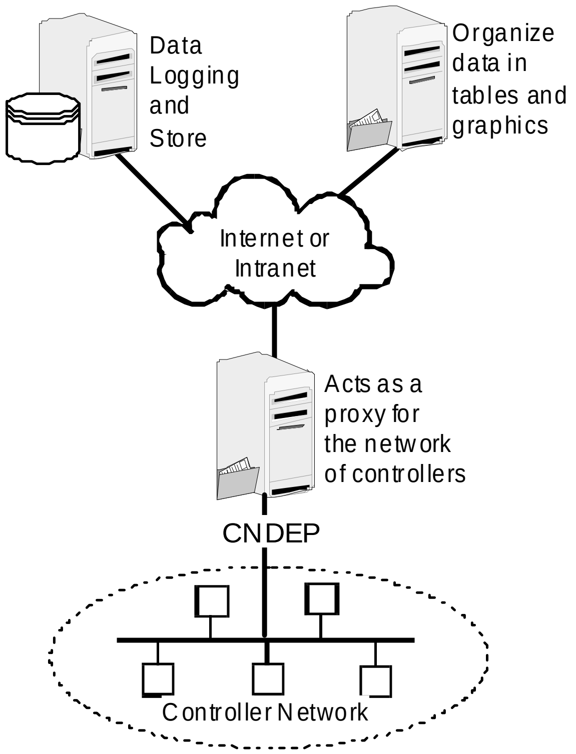

Figure 2: Functional scheme of the system.

5.2 Operation of the services

As it was stated in previous section, the system consists of three Web services – data logging, data organization and execution of measurements. All services are implemented using Microsoft .NET Framework version 1.1 with Web Services Enhancements 2.0 and are deployed on PC, running Windows XP Professional and Internet Information Services 5.1. Web Services Enhancements are needed for using SwA and WS-Attachments. The Web services libraries used are integrated in the .NET Framework 1.1 and Web Services Enhancements 2.0 (Microsoft.Web.Services2 v. 2.0.3.0, System.Web.Services v. 1.0.5000.0, System.Runtime.Serialization.Formatters.Soap version 1.0.5000.0).

Data logging service accepts data from measuring service. The data includes measured value for temperature and humidity from one controller at a time. It stores measured values with the controller name (or id) and the time when the measurement was executed. The database management system is Microsoft SQL Server 2000 and the access to it is done by ADO.NET components.

Data organizing service receives series of data from data logging service and organizes them in tables or graphics. Tables are stored in Comma Separated Values format (.csv) and then returned to service requester. Graphics are stored as pictures (bmp format). Building the graphics from data is done by means of Graphic Server .NET 3.0 component from Graphic Server Technologies. Graphics can be made for different data that can be received from logging service (temperature versus time; humidity versus time). Example graphic for temperature and humidity measured from one controller is shown on Figure 3 and corresponding data is shown on Table 1.

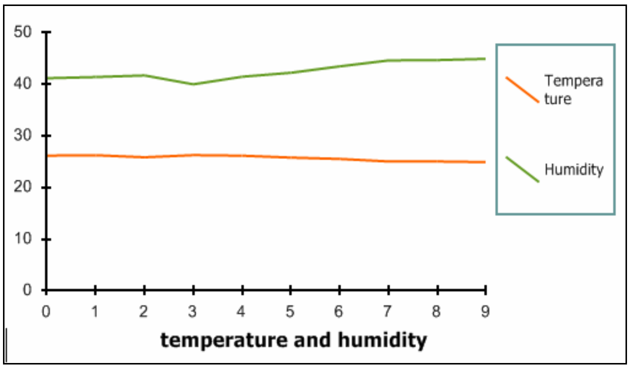

Figure 3: Temperature and humidity organized in graphics.

Table 1: Measured data

No	0	1	2	3	4	5	6	7	8	9
T	26	26	25	26	26	25	25	25	25	24
H	41	41	42	40	42	42	44	45	45	45

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 (Deem, 2002; Powell, 2002).

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 (Bosworth et al., 2003).

Measurement service is the most complex of the implemented services. It returns the temperature or humidity (or both), measured from particular controller. Calling it, you can tell which controller to execute the measurement. Otherwise the data returned is from default controller (it is applicable if two or more controllers measure same parameters for reliability). This service is actually a proxy between Web services middleware and network of controllers. The communication between controller and proxy server are done by means of custom protocol – CNDEP (Kakanakov et al., 2006a). The work of this protocol is proposed in the following section.

5.3 CNDEP (Controller Network Data Extracting Protocol)

CNDEP is custom protocol for data extraction from micro-controllers in local area network, design and implemented in “Virtual Laboratory of Computer Networks and Distributed Systems” in Technical University of

Sofia, branch Plovdiv. It works on the data tier of N-tier client/server model. It is designed for local area networks and thus, UDP communication is chosen. It works on the request/reply model and fits for data extracting and for sending commands to embedded devices. The common message format for the protocol is shown on Figure 4 (Kakanakov et al., 2006a).

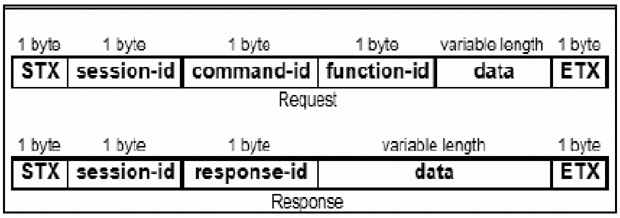

Figure 4: CNDEP message format.

5.4 Preliminary experimental results

The implemented system is deployed for testing the functionality. Each of the three Web services is started on separate machine and machines are installed in different segments of TCP/IP network for testing the interaction in Internet environment. The transfer of graphics is implemented as binary data embedded in the XML document and as attachment using SwA. Times are calculated for transfer of graphics in both ways using High performance timer provided in .NET framework. This timer provides precision of less than millisecond. Transfer via SwA is accomplished for about 300ms and the one via embedded binary data – about 200ms in local area networks. The corresponding times in Internet environment should be corrected with communication delay that depends on the network paths and connections. The reason for this is that using SwA needs more processing for forming the DIME package but is smaller in size which reflects in transmission latency. The size of SwA binary data is the same as the size of the stored graphic – 986KB. The size of the binary data in the other approach depends on the encoding used. In case of UTF-8 encoding the binary data is increased 1.33 times for xs:Base64Binary and 2 times for xs:hexBinary. When using UTF-16 encoding, this values are doubled.

The executed experiments show the functionality of the system and provide information for simple comparison between the two most popular approaches for transmitting binary data in SOAP. The comparison and analysis shows that choosing the right model for binary data transfer is a little tricky. For optimizing the transfer size and keeping the existing compression of the object the DIME approach is the best one. Nevertheless, using binary data embedded in XML is best for keeping the recommendation of WS world – “all data should be in XML format”.

The differences of the measured times for the two approaches 100ms for an image with size of about 1MB in size. For a binary file with larger size the difference will increase and become significant. In most applications this difference of about 100ms is not a problem but in the case of the presented system timing is important.

6 Conclusions

The proposed system demonstrates an application of Web services and N-tier client/server model in remote measurement systems. It can be applied in the field of agriculture, weather forecasting, industrial process monitoring, safety and fire alarms. The main advantage of the system is its ability to be distributed over large distances and the ease of re-configuration. These advantages are derived from the use of Web service middleware and N-tier approach. The system is deployed for testing its functionality in laboratory circumstances.

Although adaptation of Web services in distributed automation is directed to the device level in most of the present applications, the current work is directed to one level upwards. It is chosen to prevent the need of re-configuration of the devices and for increasing the security which is crucial in such systems. The all needed reconfiguration and adaptation is by means of adaptive middleware techniques.

7 Future work

The future work is directed in simulation and experimental analysis of the system’s components. This analysis will take place on every level of system's implementation – from client interface, through the middleware, to data producing components. The experimental results will be used for optimization of the system and finding the right trade-off between complexity and scalability.

Together with experimental evaluation of the system, an investigation of possible security and reliability leaks must be made. This includes using of encrypted connections for HTTP communication (TLS, SSL), using Web Services security specification (WS-Security), inter-connection of distributed nodes via Business-to-Business Virtual Private Networks.

Acknowledgement

The author's participation at the 2006 John Vincent Atanasoff Information Day was supported by the project BIS-21++, funded by the European Commission in FP6 INCO via grant 016639/2005.

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".

References

Alonso, G., 2002, "Myths around Web Services," IEEE Data Engineering Bulletin, Vol 25, no 4.

Borriello, G. and R. Want, 2000, "Embedded Computation Meets the World Wide Web", Communications of ACM, Vol. 43 no 5, pp. 59-66.

Bosworth, A., D. Box, M. Gudgin, M. Nottingham, D. Orchard, J. Schlimmer, 2003, "XML, SOAP and Binary Data", Website:http://www.xml.com/pub/a/2003/02/26/binaryxml.html, [June, 2006].

Deem, M., 2002, "WSDL Extension for SOAP in DIME", Website:http://www.gotdotnet.com/team/xml_wsspecs/dime/WSDL-extension-for-DIME.htm, [June, 2006].

Jammes, F., H. Smit., Service-Oriented Paradigms in Industrial Automation, Industrial Informatics, IEEE Trans-actions on Volume 1, Issue 1, Feb. 2005 pp. 62 – 70.

Kakanakov, N., G. Spasov, 2005, "Adaptation of Web service architecture in distributed embedded systems," Proc. on the International Conference – CompSysTech'05, pp. IIIB.10-1 – IIIB.10-6.

Kakanakov, N., I. Stankov, M. Shopov, and G. Spasov, 2006a, "Controller Network Data Extracting Protocol – design and implementation," Proc. on the International Conference CompSysTech'06.

N. Kakanakov, M. Shopov, and G. Spasov, 2006b, "Distributed automation systems based on java and web services," Proc. on the International Conference CompSysTech'06.

Pianegiani, F., D. Macii, P. Carbone, 2003, "An Open Distributed Control and Measurement System Based on Abstract Client-Server Architecture", IEEE Trans on Instrumentation and Measurement, Vol 52, Iss 3, pp 686-692, ISSN:0018-9456.

Powell, M., 2002, "Understanding DIME and WS-Attachments", Microsoft Corporation, Website: http://msdn.microsoft.com/archive/default.asp?url=/archive/en-us/dnarxml/html/dimewsattch.asp, [June, 2006].

Spasov, G., I. Stankov, M. Shopov, and N. Kakanakov, 2006, "Virtual Laboratory of Computer Networks and Distributed Systems," Website: http://net-lab.tu-plovdiv.bg/.

Topp, U., P. Mueller, 2003, "Web based service for embedded devices," Lecture Notes in Computer Science, Vol 2593 / 2003, pp. 141-153, ISSN: 0302-9743.

Vinoski, S., 2002, "Where is Middleware," IEEE Internet Computing, vol. 6, no. 2, pp. 83-85.

Youngblood, G. M., 2004, Book: "Smart Environments", Ch. 5: "Middleware", pp. 101-127, ISBN: 0-471-54448-5.