Web Based Application for Distributed Remote Measurement Viewing

Ivan Stankov[1] and Grisha Spasov[2]

Abstract – **The paper discusses web based application for distributed automation. Realization is made over three-layer distributed model. XML table driven communication model is used for heterogeneous connection of different parts of the system. Functionality of the model is delegated and distributed among servers and embedded systems. Majority features of realization, concerning scalability, flexibility, distribution, collecting and delegating of functionality, reliability and security of the applications over heterogeneous entities are discussed.**

Keywords – ***distribution, automation, web application, delegation, scalability, heterogeneous systems.***

I. Introduction

Recent years automation systems became more complex and wide spread with various applications due to ubiquitous using of communication technologies and especially Internet [3]. This led IT market to really huge growth and increasing familiarity with devices as pocket PCs, PDAs, Laptops, 3G Mobile phones, as well as PCs and PC compatible machines and concerned technologies. As a consequence of increased functionality, because of purposes laid in front of web based systems, is their complexity. For simplifying the process of monitoring and observation, new models for communication with system layer separation is assumed.

A realization of such kind of system takes into consideration with the particular aims of the end user. The most appropriate way for fast and accurate apprehension of environment's parameters is table driven and diagram approach. That way of data representation allows user for precisely observation of watch data.

In distributed information systems, based on World Wide Web, two technologies ASP (Active Server Pages) [11] and JSP (Java Server Pages) [10] are mostly used. These are current trends in web pages used over cyber space powered relevantly by *Microsoft* and *Sun*. Each of them has its features and advantages. The model of proposed realization is based on *ASP.NET*.

The entire automated process is distributed among embedded systems increasing in that way availability. This approach provides scalability, flexibility and avoids a crush of the whole system.

II. Related Work

Distributed Automation Systems [1,2,4,6,7,8] exert requirements to increase performance, scalability, availability, security, reconfigurability, fault tolerance and graceful recovery from failures. These features of any application and discussed models are so called critical challenging factors [8].

Each realization owes some ways to manage and escape these weaknesses [8, 9]. Ubiquitous widespread automation models are based on hierarchical separation, while distributed models use delegation and separation of functions in one multi-layer model leading to Distributed Automation Systems (DAS). Each layer has a particular role in the overall functionality. Most commonly the system is build software and hardware heterogeneous, and communication model makes the whole system to work as an interconnected logic not separating individual parts working away from each other. Because of that reason communication is a crucial issue [2]. In our realization the employed model combines the advantages of both – distributed systems and automated systems [4, 7].

Communication model includes servers, which purpose is to connect one homogenous sub-system with others. The exchange of data and messages is based on classical client-server communication. Applications are working on each server to provide interconnections into entire system. Applications could be analyser [8] or transaction server, for data processing and getting complete distribution of information system. As a result of delegating functionality the model becomes light-weighted, with improved level of defeating DoS (Deny of Service). Even declining of any layer, or just a part of a sub-system, the rest functionality does not break down and still is working. If some self-testing algorithms are available, it is possible the information system to become self-reconfigurable and to continue working properly [6].

Separating the communication entry point to the distributed system by accounts or groups with equal privileges is crucial for requests analyzing and

[1] Ivan Stankov – is with the Faculty of Electronics and Automatics in Technical University Sofia, branch Plovdiv, 4000 Plovdiv, Bulgaria, e-mail: istankov@tu-plovdiv.bg

[2] Grisha Spasov – is deputy dean of Faculty of Electronics and Automatics, Technical University Sofia, branch Plovdiv, 4000 Plovdiv, Bulgaria, e-mail: gvs@tu-plovdiv.bg

predicting data flows of each respond. Using data analyzer makes system capable to extract statistics and make point for improvement. It is possible to control data exchanges' route into data network.

Communication model is a significant part of each proposed concept closely connected with the model of system architecture. It is possible to avoid many of the problems by appropriate design of inner customized transaction messages. Well packed messages allow overcoming of heterogeneous character of the system, closely related with the design of communication model [14].

Recent years XML is demanding as a standard for customizable protocol for individual purposes [16]. This way for communication is very scalable and flexible, and easy for implementation.

Most of proposed models and realized solutions are based on distribution model, hierarchy organized, leading to simplicity in asynchronous interaction and information exchange into the inner network consisting of embedded systems and servers.

III. Discussed Model

This paper discusses realization of a distributed three-layer model, described in details in [14] (Figure 1). This model presents a simple distributed system for overcoming a lot of fore-mentioned problems. They are solved by realizing a few small applications, distributed on different machines with appropriate functionality and delegated privileges, to provide reliable system for data transaction with web interface. Distribution allows increasing availability of the system. Communication model is based on XML transaction messages. Two main points of the system are communication model and architecture model.

Proposed Communication Model overcomes more of the problems connected with web application; employing customizable XML transaction messages. The model is based on Local Area Network, which allows isolation and separation of different levels. That way increases security policy. Entire network is hidden behind a Web Server which connects the system to Internet. Internet supplies a connection to the system all over the world. The server accepts the client requests and sends to the lower level of the model a suitable transaction message. In three-layer-modelling, lower level applications are for data processing. Data processing includes parsing the request and working out suitable transactions to data layer for extracting the appropriate data. Data layer collects the data from embedded systems (in our realization these are values for measured temperature and humidity).

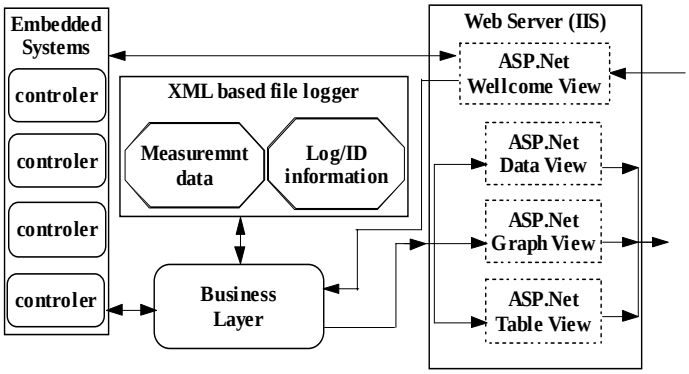

Figure 1. Model Functional Scheme

In three-layer-model architecture the following layers are outlined:

- Client Processing Layer – this layer is used for interaction with end user of the system. The consumer could be a manager for controlling or monitoring of any parameters; this layer is represented by IIS (Internet Information Service), which is Microsoft technology for web server;
- Request/Response Processing and Data Management Layer are represented by application with one of the most important function in entire model. Its purpose is managing, collecting and distributing of entire data flow. This layer is centralized one. In most models this layer is called business layer. It makes the system homogenous because of its coordinating function. This part of communication is very flexible and could be expanded easily because of Object Oriented Thread programming.
- Data Processing Layer – its realization is the most distributed one among the system. Depending on its role in the model, the layer is separated in several tiers for data collecting, data storing, data logging and data extracting.

Each of these tiers interacts with upper layer of the model (Request/Response Processing and Data Management Layer). The aiming reason for distribution as much as possible is for increasing scalability and flexibility and decreasing the level of unavailability of entire system.

IV. REALIZATION

The development tool used for the realization of the model is Visual Studio .NET. Web interface's realization is done with ASP.NET web pages. The interaction mechanism and code's connection is shown on Figure 2. So called Code Behind is doing all functionality, while View part visualize the web interface of the website. Cache is used for data transfer between Codes Behind and View's controls initializations fitted in web application.

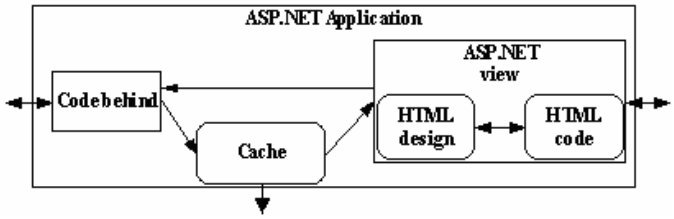

Figure 2. Conceptual Scheme of ASP.NET

The realization of the model is three-layer: web page for acceptance of user's requests and displaying of the respond; Business Layer (Transaction Server) – it is for data processing layer; the last layer is data producer layer. Data processing layer is divided in two tiers: extract data and write it to Database (DB) and the second one for request acceptance from web server and read data from DB (Figure 3). Each tier is processed by an appropriate application. Its realization is done by building a separate thread for a single functionality. In this case Read and Write threads shares general resource – access to an XML filed driven Database. Synchronization is needed to be done over these two threads. It is a classic case of reader writer (producer/consumer) synchronization.

Read Thread accepts request from web server in a queue (FIFO). For each element in the queue the XML file is read from data server. If the file is not changed, since the last read, there is no need to re-read the same information, hence new reading is not initialized and the last read data is transferred. Write Thread works out a suitable transaction messages for extracting the necessary data from data producing layer (embedded systems). CNDEP protocol is used for data extraction [13]. When the data is extracted, it is written to an XML file.

Business layer's realization is done with daemon application, using TCP sockets for connection with web application, and UDP sockets for embedded systems connection [13]. XML request/respond messages are transferred between business layer and web application. Business layer works out all necessary messages, packing them for producing an appropriate communication.

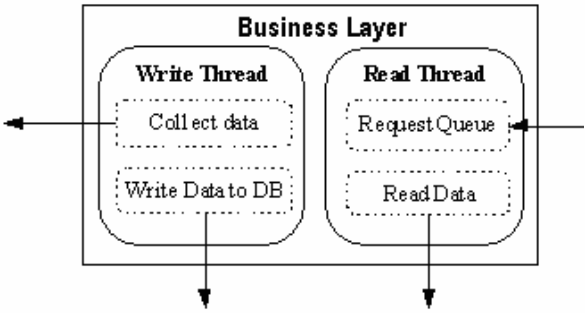

Figure 3. Scheme of Business layer

This realization is very scalable, because a new functionality could be very easily added to Business layer, and this part of the model reasonably is the most significant one. Business layer is the application which makes significant requirements to power calculation of the machine it is working on.

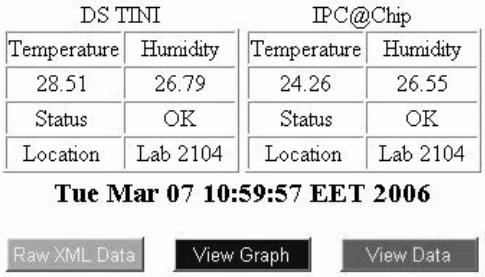

Figure 4. General View (entry point for the system)

Web presentation part of the model (Figure 4) displays current values for measured temperature and humidity, as well as status of embedded system and the location of measurements. It also presents three functions; Raw XML data – no parsing of XML file is available except self-parsing capabilities of Internet Browser; View Graph displays in datagram view last ten measurements in the system (Figure 5). This representation of data is very suitable for fast data analysing. When elaboration is available in business layer the displaying data will be more accurate and adopted for the purpose of integration. The last function is for table view of already parsed data (Figure 6). The model represents the main approaches used for viewing and presenting data – in table and graphical view. The data, presented in such way, is possible to be analysed quickly, accurately and consistently with needed requests, done by the end user.

XML processing is the stage when data is received by an application and data is needed to be

parsed. This stage is for data extracting. Many XML parsers exist and additional elaboration and modifications for each of them are available. Most often the parameters for comparison are reliability,

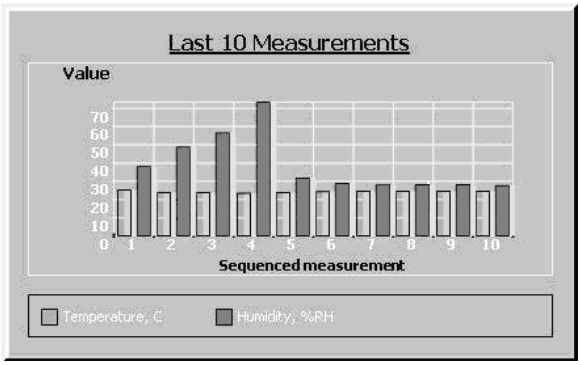

Figure 5. Graph view representation

data processing quickness, accuracy of parsed data and the scheme of parsing. In proposed realization a modification of SAX parser [12] is used, which is implemented in .NET. This parser is a "push" type. The other alternative is DOM parser. These two parsers are the most widely spread and commonly used ones.

Table View of Data -- Web Page Dialog

Controllers/No	0	1	2	3	4	5	6	7	8	9
TINI temperature	29,35	29,26	29,18	29,14	29,15	29,12	29,18	29,21	29,19	29,15
TINI humidity	25,32	25,39	25,46	25,53	25,53	25,49	25,46	25,46	25,46	25,53
IPC temperature	24,85	24,76	24,77	24,76	24,78	24,78	24,69	24,73	24,74	24,82
IPC humidity	25,08	25,22	25,22	25,22	25,22	25,22	25,32	25,29	25,22	25,15

http://net-lab.tu-plovdiv.bg:8080/WebClient/WebForm4.aspx Internet

Figure 6. Viewing Parsed XML data in table

XML files represent storing function of the DB. XML Reader class, used for XML parsing, is elaborated version of SAX parser, validation is skipped and no DTD files are required.

VI. Conclusion

Development and improvement of models for distributed embedded systems aim to overcome significant problems connected with scalability, flexibility, availability, easily expansion and modifications. All consequences from developing embedded systems and implementing such models are oriented to significant decrease the level of DoS. For distributed automation this factor is really important. If self diagnostic tools are available it would be one really completed automation system. The realization of the proposed model conforms that most of the problems could be decided by delegating and separating the functionality on different levels and the whole system to be organized combining hierarchical and distributed approaches. Realization of hierarchical or distributed models is employed in systems, in which ideology is fitted multiple-point entries for requests and responds data flows, organized with data producing layers working independently from each other.

V. Future Work

A self-diagnostic system as well as developing of selective driven realization, based on web services, is required for accurate comparison of two possible approaches of system architecture. This approach allows comparing the proposed application to other realizations that uses HTTP as a transport protocol for communication.

VI. Acknowledgements

The work in this paper is supported by National Science Fund of Bulgaria project – "BY-966"/2005, entitled "Web Services and Data Integration in Distributed Automation and Information Systems in Internet Environment".

Reference

[1] Britton, C., *IT Architectures And Middleware*, Addison-Wesley, 2001, 296 p.
[2] Burmakin, M., Krassi, B. "Distributed Automation and Control System"
[3] Duo Li, Serizawa, Y., Mai Kiuchi.. Concept design for a Web-based supervisory control and data-acquisition (SCADA) system. Proceeding of Transmission and Distribution Conference and Exhibition 2002: Asia Pacific. IEEE/PES , Volume: 1 , 6-10 Oct. 2002 Pages:32 – 36 vol.1
[4] Kakanakov, N., G. Spasov, "Adaptation of Web service architecture in distributed embedded systems", Proceedings on the International Conference – CompSysTech'05, 16-17 June 2005, Varna, Bulgaria, pp IIIB.10-1 – IIIB.10-6.
[5] Kakanakov,N., "Web Based Models For Distributed Automation", Proceeding Automatic and Informatic;
[6] Li, G., Sun, L., Li, H., "A Study of the self-organization Modeling of Distributed Manufacturing Network", Proceeding of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business, 0-7695-2206-8/04
[7] Spasov, G. , N. Kakanakov, "CGI-based applications for distributed embedded systems for monitoring temperature and humidity", Proceedings on the International Conference – CompSysTech'2004,, Rousse, Bulgaria, pp. I.6-1–I.6-6.
[8] http://www.foundrynet.com/products/webswitches/serveriron/PDFs/XML%20Switching%20on%20ServerIron.pdf
[9] http://hum-molgen.org/meetings/courses/0689.html
[10] http://java.sun.com/products/jsp/
[11] http://msdn.microsoft.com/asp.net/
[12] http://msdn.microsoft.com/library/default.asp?url=/library/en-us/cpguide/html/cpconcomparingxmlreadertosaxreader.asp
[13] http://net-lab.tu-plovdiv.bg/CNDEP/
[14] http://net-lab.tu-plovdiv.bg/ASPThreeLayerModel.html
[15] http://www.microsoft.com/mind/1196/dcom.asp
[16] http://www.w3.org/XML/