WEB BASED MODEL FOR DISTRIBUTED AUTOMATION ACROSS THE INTERNET

Ivan STANKOV, Grisha SPASOV

Technical University – Sofia, branch Plovdiv, Faculty of Electronics and Automatics, 61 Sankt Peterburg Blvd., { istankov, gvs }@tu-plovdiv.bg

Abstract. This paper proposes a distributed hierarchically organized model, delegating all functionality among computing devices. This model gives advantage to perform a distributed functionality with induced scalability, flexibility and availability able to resists on fault tolerance. The paper discusses main approaches for realization of such kind of system, including all features and properties for current realization. Proposed model's communication for interconnection exchange is based on XML custom messages. They provide semantic glue for heterogeneous nature of realization in both synchronous and asynchronous communication mode. Real-time and other timing requirements are also presented and the suitable handler task is discussed.

Keywords: distributed embedded systems, hierarchical models, manufacturing automation networks, service-oriented architectures.

INTRODUCTION

In recent years ubiquitous spreading of Internet and concerned technologies forces developing of distributed embedded web based enabled systems. The most touched sphere of this development is distributed automation systems (DAS) [2, 3]. They are a new trend in manufacturing automation networks (MAN) [4] that combine hierarchical automation approach with distributed approach of embedded systems [1] for reaching one interconnected heterogeneous system for distributed automation.

New models of communication and interconnection are proposed for overcoming problems concerning flexibility, scalability, availability, security and reliability of the system [2, 3].

This paper discusses the significant advantages of delegating and separating the functionality of the system among sub-systems with decreased computing capabilities and explicitly defined tasks. The concept of the model avoids some disadvantages by means of using embedded systems and low cost PC compatible machines. Component-based applications are developed for working on each machine, intending to produce a flexible distributed system.

The communication between layers (Figure 1) is based on XML messages. This is done for abstracting cross-platform communication complexities, increasing interoperability between applications and self-testing algorithms for auto configuration capabilities of the system. Communication is based on HTTP as a transport protocol.

RELATED WORK

Systems for monitoring and control exert requirements to increase performance, scalability, flexibility, availability, security, re-configurability, fault tolerance and graceful recovery from failures. These features of any application and models are so called critical challenging factors [6]. The approaches used for realization of such kind of systems are Distributed Embedded Systems, Service Oriented Architecture (SOA), Supervisory Control and Data-Acquisition (SCADA) System and Open Service Gateway Initiative (OSGi).

Distributed Embedded Systems

Embedded Systems nature defines low power consumption, low computing power, low cost and restricts memory usage [12]. Application of the system approach and functional and modular decomposition results in systems' distribution, i.e. consist of a number of physically and/or logically distributed components that constitute the system to reach the common goal [13]. Delegating of the functionality among different embedded systems, each one with particular function, is specific feature for distributed embedded systems.

SOA

SOA is a set of architectural tenets for building autonomous and interoperable system [3]. The nature of these systems is aiming to create separated modules that are independent of each other. They operate independently of their environment as well and provide self-contained functionality. Because of these reasons SAO could be called autonomous. On the other hand, SOA is interoperability by clearly abstracting the interface that a service exposes to its environment, from the implementation of any service. SOA is aiming to produce a service that easily could be joined to a more complex system via high level of interface abstracting. This property induces the granularity of service. That concept for system design distinguishes service-oriented from object-oriented design. The communication pattern is asynchronous, providing high scalability and flexibility of communication process.

SCADA System

SCADA system consists of a supervisory control sub-system, which supports operator control of remote equipment with security features and a data-acquisition sub-system. The sub-system gathers data from various sources such as remote terminal units and shares data with all other sub-systems within the entire system [2, 11].

OSGi

OSGi is used as a framework for delivery of managed services to networked environment through a service gateway. It consists of a service platform and a deployment infrastructure. The service platform mainly supports the interaction among services through a registry that contains service descriptions published by service providers. Once the service description is published it becomes available for other services registered with the framework. The deployment infrastructure provides the execution environment and allows for continuous deployment activities as, for example, installation, start or stop of a bundle, etc. [7].

This paper proposes a distributed model delegating entire functionality of service oriented architecture among embedded systems, PCs and PC compatible machines aiming to low cost and very scalable design concept. Communication model is a crucial issue for each model. It is XML based and provides synchronous as well as asynchronous messages' exchange.

PROPOSED MODEL

The level of integration determines the effectiveness of modern technical systems [8]. In integration of one heterogeneous system, consisting of individual modules, communication model is the flexible glue for connecting different components of the systems.

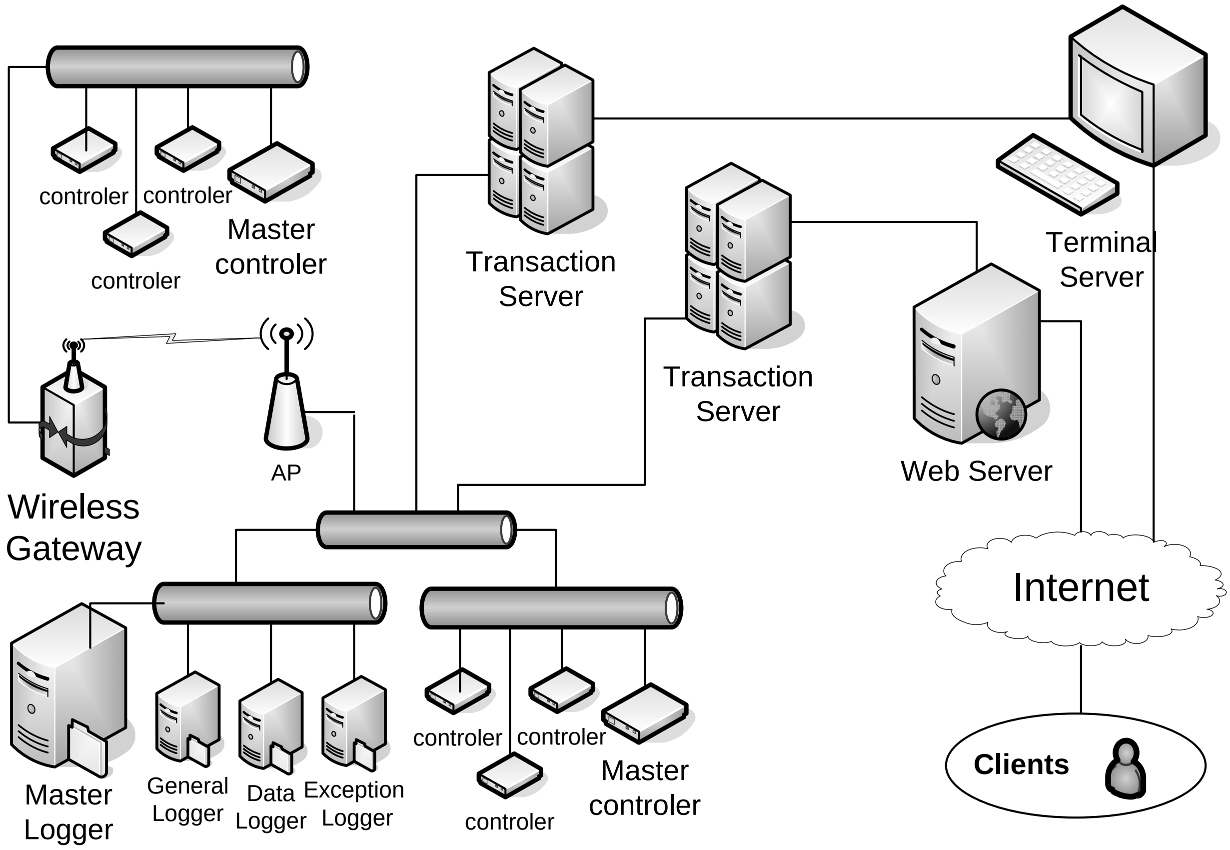

Figure 1: Functional scheme of distributed model for monitoring and control system

All efforts of proposed web-based model are aimed towards tasks distributing among embedded systems, PCs and PC compatible machines for providing high level of security, scalability, easily sizing, monitoring and setting a huge set of parameters for achieving control over the environment. However, the design provides communication flexibility and scalability, management and arrangement. New tasks and functionality easily can be added due to the hierarchy and design of delegating modules nature. The conceptual overview of the proposed model is shown on Figure1.

Model is based on comprising the features of manufacturing automation systems and distributed embedded systems. Dynamically changing environment is an appropriate place for applying distributed and real time approaches. Model aims real time reaction for any changing beyond pre-defined boundaries of any system's parameters. The model is figuring out the following features:

• The systems comprise physically and/or logically distributed components (entities);

• The entities are essentially heterogeneous (different architecture, hardware, networking, operating systems and software) – system works as one despite the difference in the entire model;

• Cross-communication and co-operation between the entities and their environment are key features – these features directly touch the communication process;

• The entities act as a unity to achieve a common goal – each entity has function and implements sensors and/or actuators.

In distributed model design and communication model (Figure 2) could be distinguished fast changing environment with high requirements to response time as well as control timing, and a location with no real time requirements. For real time requirements embedded systems with RTOS working on them are used. If any real time requirements appear on upper layer QNX (a microkernel-based operating system for real time tasks) and Real time Linux are suitable solutions. Application working on RTOS platform handles all *sensors'* and *actuators'* timing diagrams. CNDEP [14] is used for upper communication layer. It is especially worked out for data extraction of data from embedded systems. The next communication layer uses XML based custom protocol. XML stands for data describing and it is suitable for such kind of messages exchange.

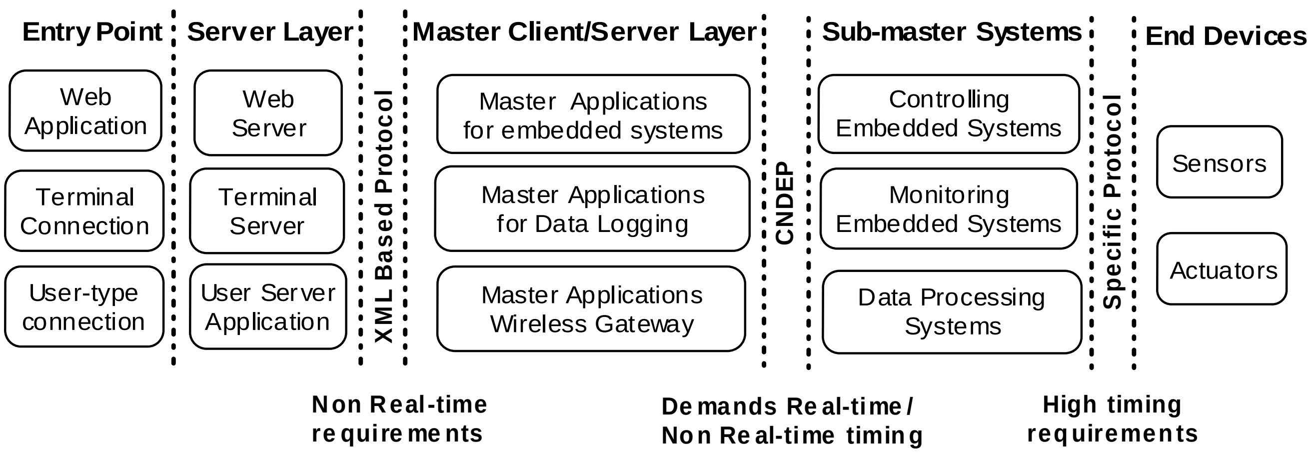

Figure 2. Functional Layers and Communication model

The hierarchical nature of discussed model is adopted from manufacturing automation networks. These networks distinguish two types of devices: fast and slow. First of them have real-time requirements, while the second one are used for interface to an operator, who controls the system, through a communication environment. Communication environment for current model is Internet. Everything behind server layer is hidden from the outer surroundings and has an autonomous management.

Communication inside the model could be realized via local Web services, defining local UDDI register used only by inner consumers.

Autonomous management of the model is achieved by self-diagnostic algorithms. Two processes are present: discovery of failure and detect plug/unplug of a device or so called hot spot re-configurability.

After any request is accepted by the server layer it forms a customized XML message addressed to appropriate master device on lower layer. The communication on that layer with upper or lower one is accomplished by the Master device. It is possible to communicate one lower layer with another one exactly through these master devices, by passing the server layer. Master devices coordinate data flows within and outside the management system's segment. Data flows are characterized by logging data, concerned security, access to the system, measured data, etc.

The aim of the model is to set apart independent modules for producing an autonomous system. Most embedded systems are web enabled, which means that they can resist to all requirements of the model. The reason all embedded systems to be hidden behind some kind of server is to induce the security level and to provide higher scalability. That approach separates all functionality not only in different layers, but also restricts the failure spreading in the entire system. Self-diagnostically algorithms are situated in the master devices. An additional functionality could be very easily added by modifying only appropriate part of the layer. For example, a case when a master device is functioning as a Wireless gateway is shown on Figure 1.

CONCLUSIONS AND FUTURE WORK

Representation of such distributed hierarchical model is going to produce an appropriate, light-weighted, low-cost, and distributed and fault of tolerance model. Main features of such kind of model are customizable nature of model design. Distributed nature of the system decreases necessary data process time among the sub-systems. These features of the model make it highly effective distributed model for automation with prospective features for development.

According to the unification wide spread standards this model would bring additional system interconnectivity if the abstraction level of different layers induces. This means, instead of using local web services, the different modules to be situated in different locations which is the real value of using that kind of technology. The master devices should be the appropriate gateway for embedded systems hidden behind it. The concept of the model is not to define the count of different layers but to induce the level abstraction for interconnection.

ACKNOWLEDGMENT

The work in this paper 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**”..

REFERENCE

[1] An Introduction To SCADA Systems, http://members.iinet.net.au/~ianw/primer.html

[2] Britton, C.. *IT Architectures And Middleware*, Addison-Wesley, 2001, 296 p.

[3] Burmakin, E.M., Krassi, B.A., Distributed Automation and Control Systems // Preprints of the 9th International Student Olympiad on Automatic Control (Baltic Olympiad), St. Petersburg, Russia, June 3-5, 2002, pp. 25-28.

[4] Burmakin, E., and Krassi, B.. Distributed automation and control systems. Proceeding of the 9th International Student Olympiad on Automatic Control (Baltic Olympiad), St. Petersburg, Russia, June 3-5, 2002, pp. 25-28

[5] CNDEP – Controller Network Data Extracting Protocol http://net-lab.tu-plovdiv.bg/CNDEP/

[6] Coulouris, G., Dollimore, J., Kindberg, T., *Distributed Systems: Concepts And Design*, 3rd edition, Addison-Wesley, 2001.

[7] Ditze, M., Kämper, G., Jahnich, I., Grisson, R.. Service-based access to distributed embedded devices through the open service gateway. Proceeding of the 2nd Conference in Industrial Informatics, Berlin, 2004.

[8] 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

[9] Francois, J., Smit, H.. Service-Oriented Architectures for Devices – the SIRENA View. Proceeding of Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference, 0-7803-9094-6, pp 140- 147

[10] 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, 0-7695-2206-8/0, 114- 117 pp.

[11] Mäntylä, M., Andersin, H.,. Enterprise systems integration, Helsinki, HUT, 1998, 313 p.

[12] 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.

[13] Wu, Q., Wang, F.-Y., Lin, Y.. A mobile-agent based distributed intelligent control system architecture for home automation, Proceedings of the 2001 IEEE International Conference on Systems, Man, & Cybernetics, Tucson, Arizona, October, 2001.

[14] http://www.foundrynet.com/products/webswitches/serveriron/PDFs/XML%20Switching%20on%20ServerIron.pdf