# Automation Services Orchestration with Function Blocks: Web-service Implementation and Performance Evaluation

Evgenii Demin[1,2], Victor Dubinin[2], Sandeep Patil[1], Valeriy Vyatkin[1,3]

[1]Luleå University of Technology, Luleå, Sweden
[2]Penza State University, Penza, Russia
[3]Aalto University, Helsinki, Finland

{evgenii.demin@ltu.se, victor_n_dubinin@yahoo.com,
sandeep.s.patil@ieee.org, vyatkin@ieee.org}

***Abstract.*** This paper presents service-oriented implementation of distributed automation systems and the results of a practical performance measurement of Web-services deployed on different platforms. In the experiments we used a technique that allows one to separate the characteristics of the Web-service, such as the delays introduced by the medium of communication. It is shown that the technology development and deployment of Web-services significantly affect their performance.

**Keywords:** Service-oriented architecture, Web-service; Cloud; Pick-and-Place manipulator; Web servers; Function blocks

## 1 Introduction

Industrial application of the Internet of Things (IoT) architecture implies embedding intelligence and communication capabilities to machines and parts thereof. Service Oriented Architecture (SOA) [1], initially developed for general purpose computing, is becoming increasingly popular in industrial automation. In the SOA way of thinking, functionalities are encapsulated into services. Services are communicating with others using the message passing mechanism. A service sends a request message, another service receives the message, executes the service invoked and sends a response message if needed.

Cloud computing, that is getting increasingly popular in various IT applications, can provide a very useful complement to IoT and SOA. The use of Cloud-deployed web-services in combination with embedded intelligence is being widely investigated for industrial automation applications. An example of such research activity is Arrowhead project sponsored by ARTEMIS[1]

[1] www.arrowhead.eu

According to [2], “cloud computing is a modern model for ubiquitous and convenient on-demand access to a common pool of configurable remote computing and software resources, and storage devices, which can be promptly provided and released with minimal operating costs and / or calls to the provider”.

Cloud computing is applied in various domains, from research and media to the mail services, corporate computer systems and electronic commerce. Consumers of cloud computing can greatly reduce the cost of maintaining their own information technology infrastructure, and dynamically respond to changing computing needs in peak times, using the property of elasticity of cloud services.

In the development of cloud-based systems, a wide range of programming languages, libraries and technology frameworks can be used, which determine the effectiveness of the software functioning. The task of choosing the adequate development tools is an important stage of the software lifecycle. Thus, the urgent task is to study and perform comparative analysis of software applications productivity, depending on the development tools, as well as its deployment environment. Given the wide spread of distributed information systems such research is of particular interest for the Web-services - applications based on service-oriented model of interaction between providers and consumers of information services. There has been some work in this space, [3, 4] presents a model that helps selecting best end-point for a service (in case of many services with same intention) and this is particularly applicable in a distributed system that we are interested in and what is briefly presented in this paper.

The aim of the paper is to investigate the performance of web-services developed to complement the embedded mechatronic intelligence using different development languages and deployment tools, and identify the various components of the total service time: the transmission delay / service request, and the processing time of the request and response formed by Web-services.

The rest of the paper is structured as follows, section 2 details the IEC 61499 function block implementation of the services, section 3 presents the case study considered to demonstrate our approach, section 4 presents the method of our testing approach and finally section 5 presents the results and evaluation.

## 2 Function block implementation of services

Given the obvious lack of system-level architecture for SOA-based automation systems, some authors of this paper proposed for this purpose in [5-7] the use of IEC 61499 distributed automation reference architecture.

A family of reference examples with increasing level of complexity was considered. The first one in Fig. 1. (a) consists of just one linear motion pusher. Once a workpiece is placed in front of the pusher (that is detected by WPS sensor), the desired service of this system is to push the workpiece to the destination sink and retract the pusher to the initial state. The Figure also shows the hardware architecture of this system that fits to the Internet of Things vision: here all sensor and actuator devices are equipped with embedded microcontrollers and network interfaces.

A design environment based on such a system-level architecture is required for both development and debugging. In that proposal, function block diagrams implement service diagram of SOA-based systems. Each function block refers to a service in the SOA. Connections between function blocks are reflected as messages between services. Thus, entire function block network could be recognized as a service diagram as shown in the Fig. 1. (b). Each event connection inside function block designs is considered as a SOAP message type. Data connections associated with this event connection are placed in the SOAP message content. SOAP messages allow a two way communication: request and response. In the same manner as functions in programming languages, a service provider can provide response messages back to the service requestor after execution is completed. Fig. 2 shows this message interaction.

Section 3 presents a complete and a more complex use case and the rest of the paper deals with this case study.

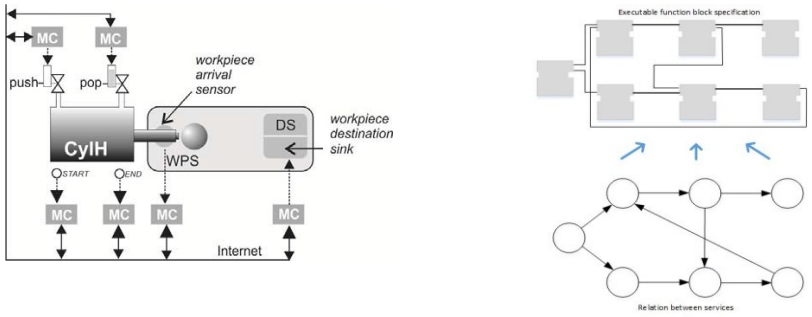


**Fig. 1**. (a) Workpiece transfer system with one linear motion pusher. (b) A function block application generated to implement requirements specified in the form of services.

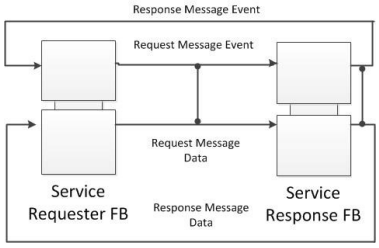


**Fig. 2.** SOAP response message support in IEC 61499.

## 3 Case Study: Pick and Place Manipulator

This study was performed using a simulated model of pick and place (PnP) manipulator presented in Fig. 3. The manipulator, consisting of two axes of pneumatic cylinders and a suction device, performs the function of moving items (work pieces) from one place to another. This manipulator has a fully decentralized control that is

based on collaboration of controllers embedded into each cylinder. This architecture facilitates Plug & Play composition of mechatronic devices. One approach to totally decentralised control of the manipulator implemented using the IEC 61499 standard is described in detail in [8, 9].

PnP- manipulator is an automated system consisting of intelligent mechatronic components, e.g. pneumatic cylinders. Several configurations of the manipulator are described in [10, 11]. In this paper, we use a configuration with 6 cylinders (3 vertical and 3 horizontal). Each cylinder can be moved in and out by the appropriate control signals issued by its embedded controller.

The decentralized system of planning is used to plan the movements of cylinders in the PnP-manipulator. The result of the planning process is to determine the combination of cylinders, which is required for the delivery of parts from a designated place.

The logic of the PnP-manipulator was designed and implemented using the IEC 61499 in the nxtStudio 2.1 development environment. This platform is intended for the design and deployment of applications based on IEC 61499.

The intelligent mechatronic architecture demonstrated in the PnP-manipulator enables use any of mechatronic modules and extensions. This offers significant business benefits in terms of flexibility and maintainability of products.

For the experiment we chose a configuration of the PnP-manipulator with a separate module that implemented Web-based service. This configuration has been developed and described previously in [12]. In this configuration, the cylinder movement planning (scheduling of which cylinders participate in the given job of picking and placing of the workpiece) is performed using an external Web-Service.

The scheme of interaction of the PnP-manipulator system components is shown in Fig. 3.

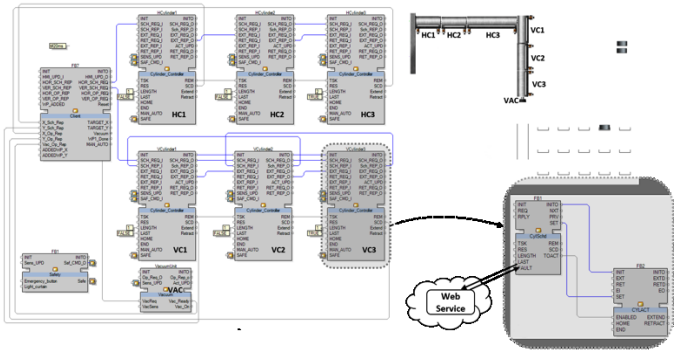


**Fig. 3.** Interaction scheme of the PnP-manipulator components.

## 4 Methods of testing

In order to study the effect of different platforms deploying Web-services on the performance, we used a technique that allows separation of the transmission delay request and results from service request' processing time for each component of the system [13]. For this operation, four timestamps were recorded:

1. $T_1$ — the time of sending a request to the Web-service by the client embedded in mechatronic component; 2. $T_2$ — the time of receipt of the client's request by the Web-service; 3. $T_3$ — the time of sending of response by Web-service; 4. $T_4$ — time the response is received.

Fixing these timestamps allows us to estimate the following performance of the Web-service:

1. Total service time ($T_{RT}$) — the time difference between sending the request by the client and the time of receipt of the reply from the Web-service:

$$T_{RT} = T_4 - T_1; \quad (1)$$

2. Service delay by the web-service ($T_{RPT}$) — time web-service takes to process the request:

$$T_{RPT} = T_3 - T_2; \quad (2)$$

3. Transmission delay introduced by remote execution of the service ($T_{RTT}$) — the time spent on data transfer of request / response between the client and Web-based services:

$$T_{RTT} = T_{RT} - T_{RPT} = (T_4 - T_1) - (T_3 - T_2). \quad (3)$$

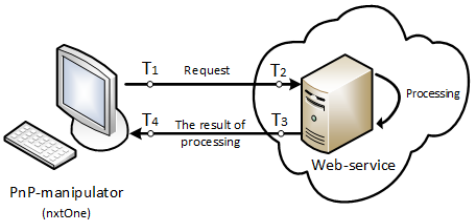


**Fig. 4.** Methods of measuring the duration at different points in a single Web-service request/response

To analyse the changes in the performance characteristics of deployment platforms the computer with following characteristics was used: CPU: Intel Core i5-2500 3.30 GHz, RAM: 8,0 GB, OS: Windows 8.1. It was used for establishing and running the Web-server and Web-deployed service for planning and execution (scheduling) of the cylinders [14]. The length of the system testing was one hour. The interval between requests was 2 minutes [15].

## 5 Performance evaluation

The experimental results show that the time of service request of the Web-service greatly depends on the implementation technology and platform on which it is deployed. Based on the data obtained during the experiments, as well as performing

statistical analysis, one can make the following practical conclusions about the performance of deployment platforms from different manufacturers:

1. Despite the fact that the products of Oracle and IBM provide more flexibility when designing Web-services, their use significantly reduces the performance of such Web-based applications.
2. When deploying Web-service on one such platform with Microsoft Internet Information Services (MS IIS) the observed productivity of Web-service is twice as much as of the IBM WebSphere Application Server.
3. The Borland's Web-platform has shown significant volatility, in contrast from the Microsoft's one. At the same time there is a steady, albeit small, deviation of the average processing time for $T_{RPT}$ for a Web-service deployed on a local server.
4. All Web-platforms use different technology to optimize Web-services.
5. Network latency has a significant effect on the performance characteristics of Web-services. In addition, the timing of network connections is characterized by significant levels of volatility compared to the time of service request for a Web-service.
6. The instability of the network environment interaction of the consumer and the provider of Web-services has a significant impact on the uncertainty of performance characteristics. The uncertainty can be characterized by the coefficient of variation, which determines the ratio of the standard deviation and the expectation of service time. In some cases, this option is too high, which indicates substantial uncertainty of Web-platform performance characteristics measured experimentally.

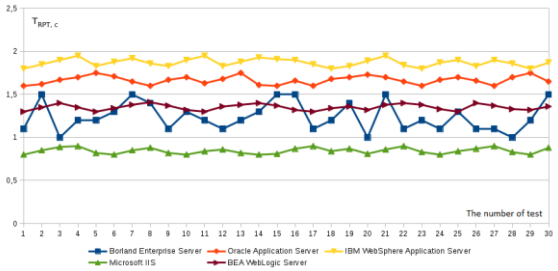


**Fig. 5.** Statistics change the time of service request Web-service $T_{RPT}$.

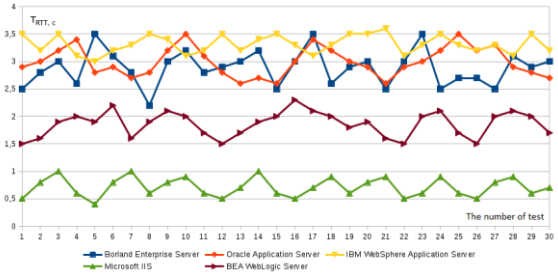


**Fig. 6.** Statistics changing of network delay $T_{RTT}$.

The above chart shows the results of testing of Web-based platforms from different vendors. Fig. 5 shows the change of service time Web-service statistics. Fig. 6 shows the statistics of change in network latency.

**Table 1.** Statistics estimating the total service time TRT

| | *Borland Enterprise Server* | *Oracle Application Server* | *IBM WebSphere Application Server* | *Microsoft IIS* | *BEA WebLogic Server* |
|---|---|---|---|---|---|
| ***The minimum value, s*** | 1,0 | 1,6 | 1,8 | 0,8 | 1,3 |
| ***The maximum value, s*** | 1,50 | 1,75 | 1,95 | 0,90 | 1,41 |
| ***Expected value*** | 1,24 | 1,66 | 1,87 | 0,84 | 1,35 |
| ***Standard deviation*** | 0,1673 | 0,0484 | 0,0460 | 0,0352 | 0,0351 |
| ***Coefficient of variation, %*** | 0,6548 | 0,0408 | 0,0328 | 0,0425 | 0,0264 |

## 6 Conclusion

The experimental performance evaluation of Web-services is important for selecting a platform for a particular application. Platform manufacturers do not provide such a comparative assessment of the performance of the development platform. Therefore, the end user cannot predict the cost of the equipment required for the deployment of distributed or cloud computing applications. As a general limitation of this study, it should be noted that this method of testing does not account for the impact of delays in the communication environment of Web-services and Cloud applications themselves. The difference in the implementation of modern technologies of Web-based applications, as well as the instability of the characteristics of the Internet environment interaction, have a significant impact on the performance of Web-based software and cause a significant degree of uncertainty in their practical dimension.At the same time, experimental studies, such as those described in this paper are important because they provide developers and users of Web-based applications to choose between information technology implementation and deployment methods, as well as the prediction of non-functional characteristics.

## Acknowledgments

This work was partially supported by the program "Fundamental research and exploratory research involving young researchers" (2015-2017) of the Russian Science Foundation (project number 15-11-10010), and by Luleå Tekniska Universitet, grants 381119, 381940 and 381121.

## References


[1] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*: Prentice Hall PTR, 2005.

[2] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges," in *Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on*, 2012, pp. 877-880.

[3] V. N. Serbanescu, F. Pop, V. Cristea, and O. M. Achim, "Web Services Allocation Guided by Reputation in Distributed SOA-Based Environments," in *Parallel and Distributed Computing (ISPDC), 2012 11th International Symposium on*, 2012, pp. 127-134.

[4] O. M. Achim, F. Pop, and V. Cristea, "Reputation Based Selection for Services in Cloud Environments," in *Network-Based Information Systems (NBiS), 2011 14th International Conference on*, 2011, pp. 268-273.

[5] W. Dai, V. Vyatkin, and J. H. Christensen, "The application of service-oriented architectures in distributed automation systems," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 252-257.

[6] W. Dai, J. H. Christensen, V. Vyatkin, and V. Dubinin, "Function block implementation of service oriented architecture: Case study," in *Industrial Informatics (INDIN), 2014 12th IEEE International Conference on*, 2014, pp. 112-117.

[7] W. Dai, L. Riliskis, V. Vyatkin, E. Osipov, and J. Delsing, "A Configurable Cloud-Based Testing Infrastructure for Interoperable Distributed Automation Systems," in *IEEE International Conference on Industrial Electronics IECON 2014*, Dallas, 2014.

[8] V. Vyatkin, "IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review," *IEEE Transactions on Industrial Informatics,* vol. 7(4), pp. 768-781, 2011.

[9] M. Sorouri, S. Patil, V. Vyatkin, and Z. Salcic, "Software Composition and Distributed Operation Scheduling in Modular Automated Machines," *Industrial Informatics, IEEE Transactions on,* vol. PP(99), pp. 1-1, 2015.

[10] S. Patil, J. Yan, V. Vyatkin, and C. Pang, "On composition of mechatronic components enabled by interoperability and portability provisions of IEC 61499: A case study," in *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, 2013, pp. 1-4.

[11] V. Vyatkin, "Intelligent mechatronic components: control system engineering using an open distributed architecture," in *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, 2003, pp. 277-284 vol.2.

[12] E. Demin, S. Patil, V. Dubinin, and V. Vyatkin, "IEC 61499 Distributed Control Enhanced with Cloud-based Web-Services," in *IEEE Conference on Industrial Electronics and Applications*, Auckland, New Zealand, 2015.

[13] L. Feng, W. Gesan, L. Li, and C. Wu, "Web Service for Distributed Communication Systems," in *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on*, 2006, pp. 1030-1035.

[14] L. Cheung, L. Golubchik, and S. Fei, "A Study of Web Services Performance Prediction: A Client's Perspective," in *Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, 2011, pp. 75-84.

[15] G. Velkoski, M. Simjanoska, S. Ristov, and M. Gusev, "CPU utilization in a multitenant cloud," in *EUROCON, 2013 IEEE*, 2013, pp. 242-249.