



**HAL**  
open science

# Petri Net Based Engineering and Software Methodology for Service-Oriented Industrial Automation

J. Marco Marco Mendes, Francisco Restivo, Paulo Leitão, Armando W.  
Colombo

► **To cite this version:**

J. Marco Marco Mendes, Francisco Restivo, Paulo Leitão, Armando W. Colombo. Petri Net Based Engineering and Software Methodology for Service-Oriented Industrial Automation. First IFIP WG 5.5/SOCOLNET Doctoral Conference on Computing, Electrical and Industrial Systems (DoCEIS), Feb 2010, Costa de Caparica, Portugal. pp.231-238, 10.1007/978-3-642-11628-5\_25 . hal-01060807

**HAL Id: hal-01060807**

**<https://inria.hal.science/hal-01060807v1>**

Submitted on 17 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# Petri Net based Engineering and Software Methodology for Service-oriented Industrial Automation

J. Marco Mendes<sup>1</sup>, Francisco Restivo<sup>1</sup>, Paulo Leitão<sup>2</sup> and Armando W. Colombo<sup>3</sup>

<sup>1</sup> Faculty of Engineering of the University of Porto, Rua Dr. Roberto Frias s/n, 4200-465  
Porto, Portugal  
{marco.mendes, fjr}@fe.up.pt

<sup>2</sup> Polytechnic Institute of Bragança, Quinta Sta Apolónia, Apartado 134, 5301-857  
Bragança, Portugal  
pleitao@ipb.pt

<sup>3</sup> Schneider Electric Automation GmbH, Steinheimer Str. 117, D-63500 Seligenstadt,  
Germany  
armando.colombo@de.schneider-electric.com

**Abstract.** Collaborative industrial systems are becoming an emergent paradigm towards flexibility. One promising solution are service-oriented industrial automation systems, but integrated software methodologies and major frameworks for the engineering are still missing. This paper presents an overview of the current results on a unified and integrated methodology based on intrinsic and novel features of Petri nets. These nets are applied to the modeling, analysis, service management, embedded software controllers, decision support system and monitoring, to improve the fundamentals in the engineering of service-oriented automation systems. The solution may contribute to the reduction of the design, operational and reconfiguration phases in the life-cycle of novel automation environments. Results were obtained and discussed from simulations and real industrial demonstrators.

**Keywords:** Service-oriented Architecture, Service-oriented Computing, Industrial Automation, Software engineering, Petri Nets.

## 1 Introduction

In Service-oriented Architectures (SoA) for industrial automation [1] several research directions and solutions have been presented, but overwhelmingly directed to a specific part of the whole engineering problem that SoA reflects in industrial automation. Thus, an inroad is required for integrated solutions of engineering, respecting users, developers, available hardware and software. Petri nets (PN) were identified as being part of the solution by presenting a set of useful characteristics supporting the life-cycle of service-oriented systems. With the choice over Petri net based formalism, the research question and motivation of this work follows as: “How does Petri Nets based methodology's features contribute as a unified tool to the integration and collaboration of autonomous service-oriented automation control components and support their life-cycle management?”

This work focuses on the actual results in terms of research, development and application of the integrated methodology for the engineering of service-oriented automation systems using Petri nets. This paper is organized in the following sections: Section 2 describes the contribution to technological innovation, section 3 presents an overview on related literature and state of the art, section 4 explains the overall engineering approach and software for service-oriented automation systems using as core methodology Petri nets, the application is resumed and discussed in section 5 and finally conclusions and future work are exposed in section 6.

## **2 Contribution to Technological Innovation**

The present research work by using Petri nets for the engineering of service-oriented automation systems offer several novel aspects that already were reviewed by international committees (see e.g. the SOCRADES project at <http://www.socrades.eu>) and recent disseminations. Note that the aspects discussed here are part of the individual research work used in the context of the project. From the SoA side, it is well known that it represent a future trend in the automation and production world, but from the software engineering and also end-user perspective, integrated solutions are missing. As such, the research work focus on a Petri net based solution for those systems, for the design, analysis and also operation. Contributions are in a form of an open methodology for developing custom and feature-full Petri net applications and control solutions, an engineering process based on Petri nets and several runtime features such as distributed orchestration, composition and active conflict resolution.

## **3 Overview on Related Literature and State of the Art**

In SoA, the main objective is the interoperability between different participants in the system by providing and using resources in form of services via an open protocol based communication platform (e.g. Web Services). In automation and manufacturing, service providers and requesters include also industrial devices, manufacturing equipment and products (besides the typical software agents that run on normal computers and servers). Several efforts are being done in integrating and managing these new systems. The first visible application resulted from the SIRENA project [1] with the objective to develop a service infrastructure for real-time embedded networked applications. Other projects in the same research area are currently in advance research and development state, such as the SOCRADES project.

Important subjects are still related to engineering solutions of service-oriented systems from the software/hardware and user point of view, and also to stimulate the industrial adoption. In service-oriented automation systems, a formal and unified method is required to provide design facilities, with the ability to validate models and to be also used as an integration middleware at runtime with enough flexibility and features. Previous works in the domain report some advances with engineering methodologies. Main research directions are related to SoA device integration and Device Profile for Web Services (DPWS) [1] [2] [3], orchestration and semantics of

devices and systems [1] [4], the use of Business Process Execution Language (BPEL) and other formalisms [5] [6], process optimization [7], business integration (for example [8], integration of SoA and Multi-agent Systems (MAS) [9] and virtual SoA enabled production environments [10].

Petri net is a language for design, specification, simulation and verification of systems. Petri nets are a graphical and mathematical modeling tool applicable to any systems. They are a promising tool for describing and studying information processing systems that are characterized as being concurrent, asynchronous, distributed, parallel, nondeterministic, and/or stochastic [11]. Petri Nets can be applied to many areas, including the industrial automation and control. Since the formal Petri nets are very basic in the application point of view, numerous extensions nets have been introduced over the years, normally designated by high-level Petri net. Logic controllers of Petri nets have also been developed (see the sequence controllers developed in the early 80's by Hitachi Ltd. [12] and the application of fuzzy Petri net controllers [13]) or converted into other formalisms (such as the industrial standard languages of IEC 61131) that can be understood by the devices [14].

The traditional application of Petri nets in SoA environments seems to be orchestration and choreography, to define sequences, conditions, interactions and compositions of services. Petri nets and high-level Petri nets are used in orchestration and choreography for modeling processes/composition [15] [16], analysis purposes [17] [18] and negotiation [19]. However, the OASIS standard of Web Services Business Process Execution Language (WS-BPEL) is the most prominent standard for orchestration of services. The application of BPEL directly in automation environment can be discussed. From one side, it has already a well defined syntax in XML that can be used directly with Web services. From the other side, BPEL is a specification mainly targeting business requirements, it is unknown to automation system engineers, too complex and descriptive to be interpreted by resource constrained devices (typically used in automation), not suitable for internal service process description based on device/software capabilities, technology dependent on Web services and missing direct analysis and validation support. However, some efforts have been done in terms of the application of BPEL and orchestration in general for industrial automation with results (see for example [5]).

#### **4 Research Contribution and Innovation**

Especially for service-oriented industrial automation, the application of Petri nets must be open to several specifications, requirements and methodologies for engineering that will possibly come in future. Considering the several aspects, an approach was specified for a concrete Petri nets methodology that is feasible and customized for the studied needs. The resulting open methodology for the definition of Petri net based applications permits the construction of custom software using a ground basis made of several elementary packages. This includes the Petri net formalism (according to [11]), analysis routines and conflict management. Besides the required openness, another requirement is the introduction of the time factor when evolving the Petri net. Consequently, the basis of the Petri nets is guaranteed, except

that the doors are open in terms of delays that can be used for customized operations. Over this, a modular property system was specified and a novel approach for the creation of token games (an “engine” that runs a Petri net) based on a customized template. Please consult the documentation of [20] for more information.

Having the basis for Petri nets applications and engineering, the next step would be identify what are the characteristics that can be extracted from them to be used in service-oriented automation systems. Most of these required or welcome features were already researched and documented. The remaining ones can be considered as future research work and possibly will also generate new characteristics. Table 1 lists the features and applications of Petri nets (and extensions) that are considered and also references to some publications with more detail over them.

**Table 1.** Features and applications of Petri nets and reference publications.

Features and applications	Ref.	Features and applications	Ref.
Open methods for PN token games	[20]	Migration and virtual environments	[23]
Property system	[20]	Knowledge extraction from PN	-
Analysis and validation	[21]	PN interchange format (XML)	-
Petri net modularization/interfaces	[21]	Petri net-based engineering	[24]
Petri net service association	[20]	Application software tools for PC	[24]
Composition of Petri nets	[21]	Petri net controller for devices	[24]
Decision support system for PN	[22]		

The use of Petri nets can be presented in the life-cycle of processes as modeling, analysis (simulation) and execution (control). For the development of the required software applications, there are several needs since an integrated engineering approach and resulting software. Based on the requirements of physical hardware and users, it was decided that the basic building blocks that compose the distributed system should be configurable software components assuming different tasks. Therefore, the software components were designated as “bots” (that have the so called “orchestration engine” embedded inside) and are able (in a service-oriented fashion) coordinate their activity and proceed also to collaboration processes with other components in the system. The central part of Fig. 1 has represented the concept, where the domain of autonomy of the bot is, for example, a transfer unit (conveyor). Communication to other software components in the system (e.g. other bots) is done via the exposition of services as well as the requesting of necessary services from others. To design, configure and maintain bots, there is a need of specific tools, that are user-friendly and speed-up the development, using a high-level programming approach (visual languages, such as Petri nets).

Since SoA-based automation systems are lacking in dedicated software applications and methodologies, effort where done in the development of several software packages for the demonstrator. In any case, system engineering and associated tools are required to facilitate the developer's intervention. From the Petri nets side, the practical usage is limited by the lack of computer tools which would allow handling large and complex nets in a comfortable way. The project was baptized *Continuum Development Tools* (CDT), named after the continuum concept used in physics and philosophy. First developments were started by integrating already developed software components, in special the old editor called *PndK* (Petri nets development toolKit), under the same umbrella. Along with the integration, it

was identified that several software packages are needed, namely: a framework for developing bots, engineering tools for the design and managing of bots and several utilities (mainly libraries) for supporting activities (e.g. such as communication and interface for devices). The main component is the Continuum Bot Framework (CBF) for the development of bots and their functional modules, inspired in the anatomy of living beings. Another component, the Continuum Development Studio (CDS) that is based on an extensible Document/View framework, provides an engineering tool for service-oriented bots, for example, supporting the visual description, analysis and simulation of their behavior (for now, in Petri nets formalism). The utilities package includes several reused software libraries and tools, some developed internally others adopted from the outside, such as the SOA4D DPWS library (available at <https://forge.soa4d.org>) providing facilities for the development of Web services and the Qt toolkit (see <http://qt.nokia.com/products/>), used mainly as a graphical toolkit for human interaction in the CDS.

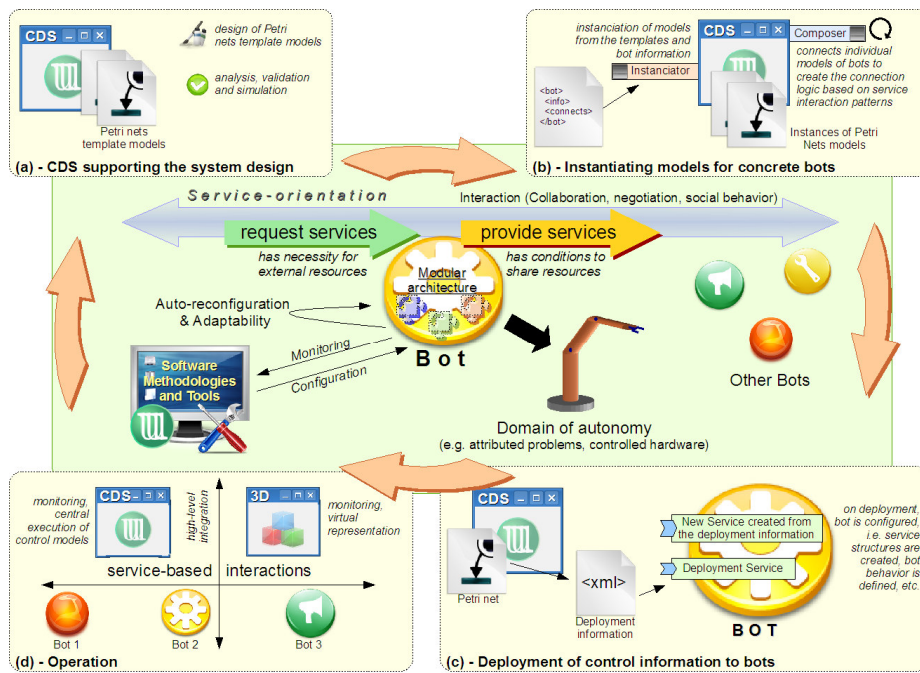


Fig. 1. Concept of the automation bot and engineering process (adapted from [24]).

The engineering is overviewed in the external boxes of Fig. 1. The CDS is employed for designing and analyzing the Petri nets template models for describing the behavior of the bots (Fig. 1.a). When importing the device/connection information, several steps are done by the CDS: 1) instantiate Petri nets models for each bot based on the designed template models, 2) create the necessary properties of the Petri nets models so that several parameters of the given device/connection information and 3) based on the “enrichment” of the Petri nets models, composition of models can be done for creating connection logic and for the overall system

analysis (Fig. 1.b). After the analysis and simulation (that can be done with the CDS), bots must now be configured. The process of deploying a service that encapsulates its logic as a Petri nets model to a bot that provides an embedded Petri Nets Kernel Module is depicted in Fig. 1.c. The deployment functionality is a standard feature of the DPWS and is exposed as a dynamic deployment service. The target and the deployment service can be discovered by stacks built-in discovery service. After deployment, a new service endpoint has been added and the execution of the services logic has been initiated. Deployment information includes the Petri nets behavior model, connection information of neighbors (required services), provided services by the bot and also extra configuration information for the other modules of the bot. The bot will configure itself (and its modules) and is then ready for operation (Fig. 1.d).

## 5 Application and Discussion

The implementation and evaluation of the current methodology is done using a virtual and real hardware demonstrator. The virtual one was modeled using the DELMIA solution to provide a simulation environment based on the real system. A service framework was added to DELMIA so that the environment can be controlled by the exposition and request of services, in the same way as pretended by the physical counterpart. The assembly equipment comprises a set of conveying modules, two lifters that make the interface between the two levels and two assembly workstations. The objective is to route the incoming pallets to the correct workstation, according to their production plan. The equipment is controlled by automation devices where the bots (including the Petri net engine, DPWS framework and other modules) run. These devices are then connected to the network, so that they can be configured, and their services can be used.

First results show the applicability and correct behavior of the system and reduced efforts in development. However and since it represents a first prototype application, there are some tasks of improvement on both methodology and application. One requirement is the previous knowledge about Petri nets, SoA and the engineering approach. This information has to be transmitted to the users and developers that are normally used to the traditional automation systems. The software part has to be improved in terms of stability, performance (e.g. improving response form embedded devices) and user-friendliness (mostly the engineering procedures using the CDS). Not less important is the automatic integration into higher level IT-systems, especially dealing with production orders and business needs.

This approach is used actively in the SOCRADES project for the specified demonstrator and has been evaluated by experts in the field. More information can be seen on the public available video at YouTube (“IP SOCRADES Demonstration of Service-Oriented Architecture integrating Real and Virtual Devices in the Electronic Assembly Scenario”) that shows some of the application results of this approach.

## 6 Conclusions and Further Work

The current contributions of this work can be resumed by the specification of unified service-oriented engineering method that can be used in a collaborative automation environment and prove features of Petri net-based solution in service-oriented automation and production systems as a unified methodology and integration middleware. In the end, it should contribute to the reduction of the design, operational and reconfiguration phases in service-oriented automation systems. New features of this work can be addressed by the combination of the methodology itself, especially the token game template for customized implementations, conflict management, property system and also some aspects of engineering in service-oriented industrial automation. Worth of mention is that it concludes also in a new form of engineering that is different from the traditional used in automation systems.

**Acknowledgements.** The authors would like to thank the European Commission and the partners of the EU IST FP6 project “Service-Oriented Cross-layer infrastructure for Distributed smart Embedded devices” (SOCRADES), the EU FP6 “Network of Excellence for Innovative Production Machines and Systems” (I\*PROMS), and the European ICT FP7 project “Cooperating Objects Network of Excellence” (CONET) for their support.

## References

1. Jammes, F., Smit, H.: Service-oriented paradigms in industrial automation. In: IEEE Transactions on Industrial Informatics, vol. 1, n. 1, pp. 62--70 (2005)
2. Bobek, A., Zeeb, E., Bohn, H., Golasowski, F., Timmermann, D.: Device and service templates for the Devices Profile for Web Services, In: Proceedings of the 6th IEEE International Conference on Industrial Informatics, pp. 797--801 (2008)
3. Li, Q., Shu, Y., Tao, C., Peng, X., Shi, H.: Service-Oriented Embedded Device Model in Industrial Automation. In : Proceedings of the Workshop on Intelligent Information Technology Applications, pp. 525--529. IEEE Computer Society (2007)
4. Delamer, I.M., Lastra, J.L.M.: Self-orchestration and choreography: towards architecture-agnostic manufacturing systems. In: Proceedings of the 20th International Conference on Advanced Information Networking and Applications, vol. 2, pp. 5 (2006)
5. Puttonen, J. Lobov, A., Lastra, J.L.M.: An application of BPEL for service orchestration in an industrial environment. In: Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, pp. 530--537 (2008)
6. Lobov, A., Puttonen, J., Villasenor, V., Andiappan, R., Lastra, J.L.M.: Service oriented architecture in developing of loosely-coupled manufacturing systems. Proceedings of the 6th IEEE International Conference on Industrial Informatics, pp. 791--796 (2008)
7. Rahman, M., Ahmad, H.F., Suguri, H., Sadik, S., Longe, H.O.D., Ojo, A.K.: Supply chain optimization towards personalizing web services. In: 4<sup>th</sup> International IEEE Conference on Intelligent Systems, vol. 3, pp. 17--22 (2008)
8. Nguyen, D., Savio, D.: Exploiting SOA for adaptive and distributed manufacturing with cross enterprise shop floor commerce. In: Proceedings of the 10<sup>th</sup> International Conference on Information Integration and Web-based Applications & Services, pp. 318--323. ACM New York, NY, USA (2008)



9. Ribeiro, L., Barata, J., Mendes, P.: MAS and SOA: Complementary Automation Paradigms. In: IFIP International Federation for Information Processing, Innovation in Manufacturing Networks, vol. 266, pp. 259--268. Springer Boston (2008)
10. Cachapa, D., Colombo, A., Feike, M., Bepperling, A.: An approach for integrating real and virtual production automation devices applying the service-oriented architecture paradigm. In: Proceedings of the IEEE Conference on Emerging Technologies & Factory Automation, pp. 309--314 (2007)
11. Murata, T.: Petri nets: Properties, analysis and applications. In: Proceedings of the IEEE, vol. 77, pp. 541--580 (1989)
12. Murata, T., Komoda, N., Matsumoto, K., Haruna, K.: A Petri Net-Based Controller for Flexible and Maintainable Sequence Control and its Applications in Factory Automation. In: IEEE Transactions on Industrial Electronics, vol. 33, n. 1, pp. 1--8 (1986)
13. Gomes, L., Steiger-Garção, A.: Petri net based Programmable Fuzzy Controller targeted for distributed control environments. In: Proceedings of the the International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and the Second International Fuzzy Engineering Symposium, pp. 1427--1434 (1995)
14. Uzam, M., Jones, A.H., Ajlouni, N.: Conversion of Petri net controllers for manufacturing systems into ladder logic diagrams. In: Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, vol. 2, pp. 649--655 (1996)
15. Hamadi, R., Benatallah, B.: A Petri net-based model for web service composition. In: Proceedings of the 14<sup>th</sup> Australasian database conference, pp. 191--200. Australian Computer Society, Inc. (2003)
16. Bing, L., Huaping, C.: Web Service Composition and Analysis: A Petri-net Based Approach. In: Proceedings of the First International Conference on Semantics, Knowledge and Grid (2005)
17. Yang, Y., Tan, Q., Xiao, Y.: Verifying web services composition based on hierarchical colored petri nets. In: Proceedings of the first international workshop on Interoperability of heterogeneous information systems, pp. 47--54. ACM Press, New York, NY, USA (2005)
18. Martens, A., Moser, S., Gerhardt, A., Funk, K.: Analyzing Compatibility of BPEL Processes. In: Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services, pp. 147--147 (2006)
19. Jiang, H., Gu, J., Yu, Q.: Modeling of Web-based collaborative negotiation systems using colored Petri net. In: Proceedings of the 12<sup>th</sup> International Multi-Media Modelling Conference, 8pp. (2006)
20. Mendes, J.M., Restivo, F., Leitão, P., Colombo, A.W.: Customizable Service-oriented Petri Net Controllers. Accepted for the 35<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society (2009)
21. Mendes, J.M., Leitão, P., Colombo, A.W., Restivo, F.: High-Level Petri Nets Control Modules for Service-Oriented Devices: A Case Study. In: Proceedings of the 34<sup>th</sup> Annual Conference of the IEEE Industrial Electronics Society, pp. 1487--1492 (2008)
22. Leitão, P., Mendes, J.M., Colombo, A.W.: Decision Support System in a Service-oriented Control Architecture for Industrial Automation. In: Proceedings of the 13<sup>th</sup> IEEE International Conference on Emerging Technologies and Factory Automation, pp. 1228--1235 (2008)
23. Leitao, P., Mendes, J.M., Colombo, A.W.: Smooth Migration from the Virtual Design to the Real Manufacturing Control. In: Proceedings of the 7<sup>th</sup> IEEE International Conference on Industrial Informatics (2009)
24. Mendes, J.M., Bepperling, A., Pinto, J., Leitão, P., Restivo, F., Colombo, A.W.: Software Methodologies for the Engineering of Service-Oriented Industrial Automation: The Continuum Project. In: Proceedings of the 33<sup>rd</sup> Annual IEEE International Computer Software and Applications Conference (2009)