Petri net-based Approach for Web Service Automation Resource Coordination

J. Marco Mendes[1], Paulo Leitão[2], Armando W. Colombo[3], Francisco Restivo[1]

[1]*Faculty of Engineering of University of Porto, Portugal*

[2]*Polytechnic Institute of Bragança, Portugal*

[3]*Schneider Electric - HUB & Globalization of Technology, Germany*

Abstract

In industrial automation, control systems and mechatronic devices are from diverse nature, supplied by different manufacturers and made of different technologies. The adoption of web services principles in an automated production system satisfies some requirements, namely the interoperability of such heterogeneous and distributed environments and the basis for flexibility and reconfigurability. Manufacturing processes require to access resources at different precedence levels and time instances, but in the other way resources may also be shared by different processes. A major challenge is then how individual services may interact, coordinating their activities. Petri nets may be used to describe complex system behaviour and therefore also applied to coordinate such systems. The paper introduces a Petri net based approach for the design, analysis and coordination of systems developed using web services to represent individual and autonomous resources. For this purpose, it is presented a Petri nets computational tool to support the design, validation and coordination of web service based automation systems.

1 Introduction

New revolutionary manufacturing concepts and emerging technologies, which take advantage of the newest mechatronics, information and communication technologies and paradigms are being researched and developed since the last decade of the 20th Century (Colombo et al. [1]). The growing of decentralised and distributed control systems is showing a way forward to these new manufacturing processes. The question remains the same: how to efficiently manage such distributed systems? Different solutions arise in the form of multiagent systems, service-oriented architectures, web services, Petri nets and others.

Standard Web technologies and increasing computational power are becoming ever more available even on the smallest devices and also different platforms. Some devices have already natively integrated web servers, supporting standard protocols or even allow user-defined enhancements. The interoperability of service-oriented architectures and the concept of creating a service interface for hiding the service implementation becomes an important feature. Service aggregation allows forming complex and business centric services which are composed by often simpler and elementary services in a "Russian doll" manner (Bepperling et al. [2]).

A major challenge is then how individual services may interact, coordinating their activities. Manufacturing processes require to access resources at different precedence levels and time instances, but in the other way resources may also be shared by different processes. Petri net is a mathematical representation of discrete, dynamic, and distributed systems, being particularly well-suited for systems in which concurrency and parallelism, synchronization, resource sharing and mutual exclusion are important. Requirements of flexibility can be reduced towards to interoperable and reconfigurable manufacturing systems by using web services for resource interoperability and Petri Nets for design, validation and coordination in automation domain. Intricate distributed control systems require well established and proved applications of web services and Petri nets, including the enhanced benefits provided by the combination of both.

The paper introduces a Petri net based approach for the design, analysis and coordination of systems developed using web services to represent individual and autonomous resources. Initially, Section 2 discusses the concepts about service-oriented architectures and web services, referring their applicability in automation domain. In the same section, a brief state of the art on service composition and coordination is also presented. Section 3 discusses the applicability of Petri nets for the web services coordination and aggregation, and Section 4 introduces a Petri nets development tool to support the service coordination and composition. Finally, Section 5 rounds up the paper with conclusions.

2 Service-oriented architecture and Web services

Service-Oriented Architectures (SOA) are nowadays used in design and specification of reconfigurable and distributed systems over the network. In a SOA environment, the distributed resources provide their functionalities in form of services that can be accessed externally by clients without knowing the underlining implementation. In such distributed environments, questions about interoperability arise, not only the independent condition of these resources, but also the individual implementation requires that the system has to operate as a whole. Jammes et al. [3] refer to the challenge of SOA to reconcile the opposing principles of autonomy and interoperability. A service helps in the communication by hiding unnecessary complexity to the outside world and by showing only the necessary information. The service interface describes how internal structure and functionality, represented by a service, is accessed by a service's client. Thus, internal processes can be executed and modified independently and transparent to the client.

Service-oriented architectures are commonly related to web services, being its preferred implementation. Web Services (WS) provide a language-neutral, loosely-coupled, and platform independent way for linking applications over the network (Fu et al. [4]). These services are made available from a web server for web users or other web-connected programs. Web Services are totally platform-agnostic and can communicate with and/or be aggregated with other web services. Besides the standardization and wide availability of the Internet itself, web services are also enabled by the ubiquitous use of XML as a means of standardizing data formats and exchanging data (Jammes and Smit [5]). Formal definition of WS and additional algebra is given by Hamadi and Benatallah [6] and Bing and Huaping [7].

In practical use, the real meaning of web services is provided by the wide use of the following core protocols: Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). The first one is a platform and language independent communication protocol used between applications, using a simple and extensible XML based format for sending messages over the internet. WSDL provides a XML based language for describing web services and how to access them. The last one, UDDI, manages a service directory where applications can register and search for web services in form of interfaces described by WSDL.

In a web service environment, the service provider exhibits its service described in WSDL, see Figure 1.

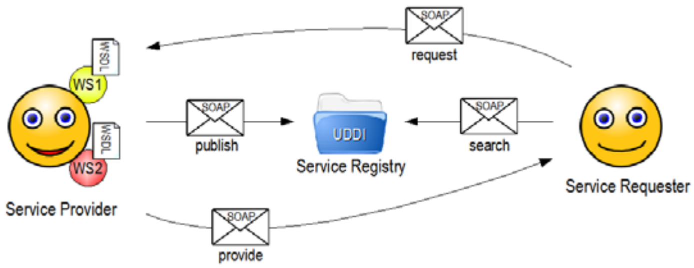

Figure 1: Web service platform with involved elements and protocols.

The external accesses to the service provider resource are via the available services, which can be published in a service registry, and indirectly discovered by the service requester. For this reason, a service discovery feature is required in which services can be added, removed and located. The service registry is not strictly necessary in situations where the locations of services are known from the beginning or when the discovery is made in a decentralised way. A service requester may access some service functionality by sending corresponding SOAP messages.

The SIRENA project (see Jammes and Smit [5]) has extended the SOA paradigm into the realm of low-level embedded devices, such as sensors and actuators (Colombo et al. [8]). The feasibility of this approach has been demonstrated through a proof-of-concept implementation based on the Devices Profile for Web Services (DPWS), a device-oriented subset of the Web Services protocols. By virtue of using DPWS, devices are able to automatically discover each other's presence, and in simple cases, they can thus start to communicate once connected. The DPWS specification is intended to foreshadow the next major upgrade of UPnP (Jammes and Smit [5]). With DPWS, all messaging, whether related to discovery, control or event notification, is based on the use of SOAP. In addition to its core protocols, it adds WS-Discovery and WS-Eventing to provide plug-and-play discovery of network-connected resources and publish-subscribe eventing.

2.1 Service Interaction

Complex web services may be created by aggregating the functionality provided by simpler ones. This is referred as service composition and the aggregated web service becomes a composite web service (Chafle et al. [9]). The aggregation of services requires mechanisms for coordination and synchronisation. Web service composition problem shares many common features with workflow systems. However, web service composition requires additional functionalities for discovery and checking interoperability of the web services (Karakoc et al. [10]).

Other terms, such as service orchestration and choreography, important to the concepts of coordination and composition, are used. Orchestration is the practice of sequencing and synchronising the execution of services, which encapsulate business or manufacturing processes (Jammes et al. [3]). An orchestration engine implements the required logic application for workflow-oriented execution and sequencing of atomic services, and provides a high-level interface for the composed process. The choreography level considers the rules that define the messages and interaction sequences that must occur in order to execute a given process through a particular service interface. Additionally, choreography can be used independently in a collaborative system without a centralised approach.

Some protocols are currently available for dealing with service aggregation and coordination, such as Web Services Business Process Execution Language (WSBPEL), Web Services Flow Language (WSFL), Web Service Conversation Language (WSCL), Web Services Choreography Description Language (WS - CDL) and Composite Web Service Language (CWSL) from Karakoc et al. [10].

According to Moldt and Ortmann [11], web services might be composed to accomplish arbitrary complex tasks. Agents can compose these Web services as long as they know their semantics. Here process ontology offers a way to give agents an understanding of the services offered. Elfatatry and Layzell [12] write about complex forms of interaction, such as negotiation, that will become dominant towards a service-oriented model of development. In automation domain, the vision of using service-oriented architectures is to support the lifecycle needs in the context of agile and flexible manufacturing. The request for easy reconfigurable manufacturing systems composed of standard components that may be remotely supported by geographically distributed engineering partners to suit changing and unpredictable business needs is presented by Colombo et al. [8].

The adoption of web services principles in an automated production system satisfies some requirements, namely the interoperability between equipments and the basis for flexibility and reconfigurability. The higher level of abstraction and transparency introduced by the composition provides the access only to the necessary features. Consequently, the autonomous resources can control their own environment and reconfigure itself when necessary, without the knowledge of the external requesters (or at least sending the corresponding information of reconfiguration). From the viewpoint of interaction, service providers and clients (that may represent devices), should control the interaction based on specific communication patterns and rules.

3 Petri nets for service coordination and composition

Distributed systems, such as those based on web services, require coordination schemes to manage complex and aggregated tasks. When handling with services, the interaction and synchronization between clients and providers can be described and directed by workflow structures. These structures may have a dedicated engine that manages the environment according to the task relations of the workflow or, in a distributed control system, collaboration among service providers and clients should follow the established work plan.

The organizational processes among service providers and clients, when established according to a structured management, oblige special attention. For instance, the coordination of web services may bring undesirable and erroneous behaviours, if not planned properly. On the other hand, when handling with composition of services, the simplified and transparent service that aggregates a more complex structure of services must synchronize the individual operations and its information flow. Such considerations demand a powerful and flexible modelling tool; therefore we consider Petri nets for this reason.

The graph-based Petri nets are used to describe complex task synchronization, not only for validation purpose, but also to control the interaction and conditional rules. The distributed nature of web services and the requirement for coordination fits well with the benefits provided by Petri nets.

3.1 Petri nets Formalism

Petri nets (PN) formalism is a graphical oriented language for design, specification, simulation and verification of systems, designed by Carl Adam Petri in 1962. It is in particular well-suited for systems in which communication, synchronization and resource sharing are important. On one hand, as a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. On the other hand, as a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behaviour of systems (Murata [13]).

Figure 2 exemplifies the execution steps of a product manufacturing process with sequential and parallel tasks, modelled by a Petri net, with transitions representing the execution of the tasks. The actual state of the Petri net (in which p_2 has a token) activates transition t_2. After the firing of transition t_2, the token of p_2 is consumed (removed) and the places p_3 and p_4 hold each one a token. In this case, two parallel processes are activated and are again synchronized by transition t_5.

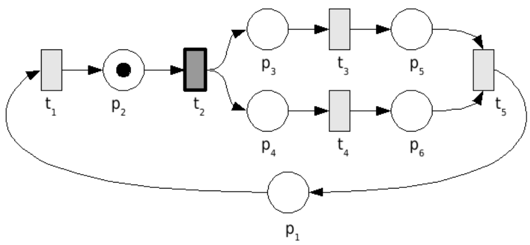

Figure 2: Example of a Petri Net with sequential and parallel processes.

The powerful analysis and modelling features provides also the ability of control of complex manufacturing systems. Due to their capability in modelling the dynamics of the systems, Petri nets have been combined with fault tree analysis techniques to determine the average rate of occurrence of system failures (Adamyan and He [14]). A novel approach to the development life-cycle of agent-based production control applications, from the design to the operation, based in a catalogue of High-level Petri nets is given by Leitão et al. [15]. The High-level Petri net-based approach facilitates the conception, definition and formal specification of an "encapsulation process" in industrial production systems.

3.2 Service coordination and aggregation with Petri nets

Compositional aspects and coordination of Petri nets and workflows in general have been studied, such as described by Anisimov et al. [16] and Pankratius and Stucky [17], to provide an approach to distributed systems. The workflow management system can be orchestrated in a centralized viewpoint (master/slave) in which the workflow model should be interpreted and coordinated by the corresponding manager. In other hand, distributed management requires additional care because the global execution and state identification is not synchronized and supervised by one element.

Web service behaviour is basically a partially ordered set of operations. Therefore, it is straight-forward to map it into a Petri net. Operations are modelled by transitions and the state of the service is modelled by places. The arrows between places and transitions are used to specify causal relations (Hamadi and Benatallah [6]).

Figure 3 exemplifies how individual web services can be coordinated and aggregated to form a new service. This model is based on some concepts of the SO-SAM architecture, described by Fu et al. [4]. The first web service manages a transport system with two lines and the second is a simple manufacturing machine. The composition of both services requires coordination between the inputs and outputs of the services. The aggregated composed service manages its internal structure and shows externally only the necessary functionality (input, output). A token in the input place (result of requesting the input port of the service) selects an available transport line. To activate the machine service, both transport lines must be activated and transportation concluded. After the machine finished its activity, the output place gets a token that can be used to notify the ending of a specific activity.

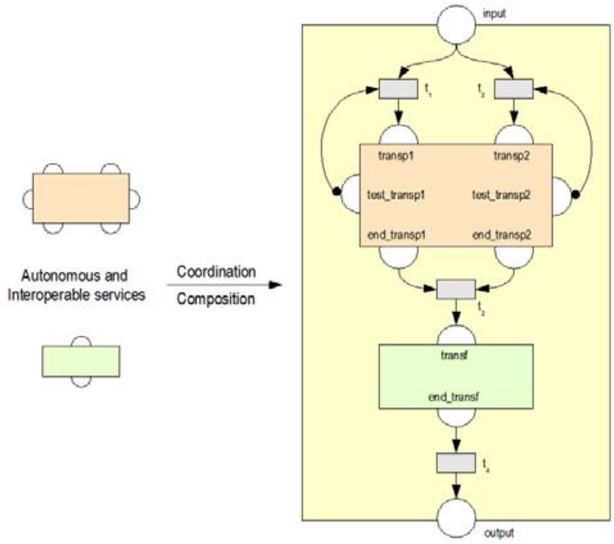

Figure 3: Example of service coordination and composition using Petri nets

Diversified research has been done by using Petri nets in web service environment. Hamadi and Benatallah [6] propose a Petri net-based algebra for modelling Web services control flows, and Zhai et al. [18] integrate agents and Web Services into Grid Service Workflow System based on Coloured Petri Nets. The combination of agents and Web Services enhances the adaptability and dynamics of the framework. The verification of Web services composition by using Coloured Petri nets is presented by Yang et al. [19]. In Bing and Huaping [7], a Petri net-based algebra is used to capture the semantics of complex Web service combinations. Chatain and Jard [20] reveal the interest in the questions of supervision and diagnosis, by explaining how to use unfolding of dynamic nets for the diagnosis application. To facilitating web services integration and verification, Zhang et al. [21] introduces WS-Net as an executable architectural description language incorporating the semantics of Coloured Petri-net with the style of object-oriented concepts.

In automation, where the requirements of flexibility and reconfigurability are high, the combination of web service for interoperability and Petri nets for distributed control and management appears as an emerging and promise application. The environment-specific features of industrial beds must be taken in account when specifying these technologies together. The challenge is, in our perspective, the research and development of Petri net based service coordination for industrial automation scenarios, and thus serving the concept of a higher flexibility of these systems.

4 Petri nets development toolKit (PndK)

The practical usage of Petri nets is limited by the lack of computer tools which would allow handling large and complex nets in a comfortable way (Suraj et al. [22]). A good editor, simulator and powerful analysis engine are essentials for modelling and analysing Petri. Moreover, for distributed coordination in a service-oriented environment, these tools should provide orchestration mechanisms based on Petri nets and additionally permit the design of composed services.

The Petri net development toolKit (PndK) is a Petri net design tool with analysis and discrete simulation, which is being developed for service coordination and aggregation in automated production systems. The simple but effective GUI makes possible the design/configuration of Petri nets and the Petri net library/plug-ins adds analysis environment to the framework. The built-in orchestration engine is able to coordinate and synchronize the workflow described by the Petri net. Despite of its early development stage, the main objective of PndK remains to be a development tool for distributed, web service and agent based applications in automation systems, filling the emptiness in this area. Figure 4 shows a screenshot of the PndK application, demonstrating the modelling and analysis phases of the Petri net illustrated in Figure 2.

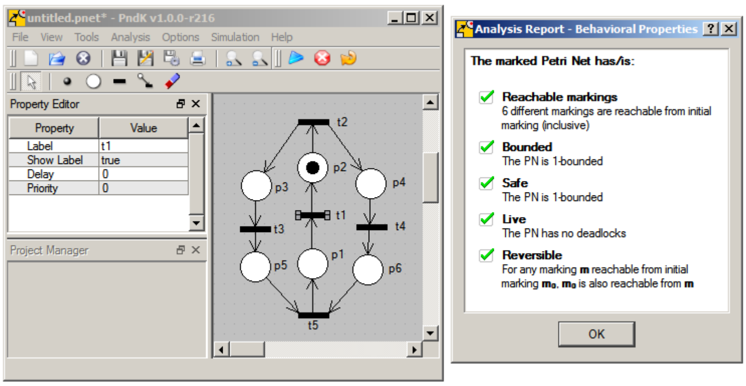

Figure 4: PndK in action showing behavioural properties of a Petri net.

At this development stage, PndK tool has already many solid features, such as an user friendly graphical editor for Petri nets design, extensible through plug-in technology, Petri net kernel with some High-level Petri Net characteristics (timed, priority based transition), validation of Petri nets and analysis methods (reachability tree, invariants and discrete simulation) to extract different properties.

In the near future, diverse additions and enhancements are planned, such as transition explosion (hierarchical transitions), coordination between Petri nets and web services, XML/code generation, handling of Petri net extensions (e.g. Coloured Petri Net) and the integration in automation and control environment for modelling, analysis and orchestration.

5 Conclusions

The authors' vision is to approach the modelling, control and analysis features of the Petri nets to the interoperability concepts of the web service technology in the automation field. Several researches have been done in the area of business processes and e-commerce, but in automation and manufacturing environments it is still a major challenge. Not only the intrinsically complex structure of distributed automation systems, but also the need for mature applications from both sides (web services and Petri nets) is requested for further research on how to get the best of both technologies.

This paper introduces compositional aspects and coordination of Petri nets in web service environment applied to automation environment. The research and work is on progress, demonstrated by the emerging of Devices Profile for Web Services (DPWS), from the side of web services applied at the device level, and Petri nets development toolKit (PndK) for Petri net modelling of automation systems. A step forward should be extending the PndK to integrate web service capabilities to support the design and control of distributed automation systems.

References

[1] Colombo, A., Schoop, R., Leitao, P. and Restivo, F., A collaborative automation approach to distributed production systems, 2nd IEEE International Conference on Industrial Informatics, pp. 27-32, 2004.

[2] Bepperling, A., Mendes, J., Colombo, A., Schoop, R. and Aspragathos, A., A framework for development and implementation of web service-based intelligent autonomous mechatronics components, IEEE International Conference on Industrial Informatics, pp. 341-347, 2006.

[3] Jammes, F. , Smit, H., Lastra, J.L. and Delamer, I., Orchestration of service-oriented manufacturing processes, In 10th IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, pp. 8-, 2005.

[4] Fu, Y., Dong, Z. and He, X., An approach to web services oriented modeling and validation, Proceedings of the 2006 international workshop on Service-oriented software engineering, ACM Press, pp. 81-87, 2006.

[5] Jammes, F. and Smit, H., Service-oriented paradigms in industrial automation, IEEE Transactions on Industrial Informatics, vol. 1, pp. 62-70, 2005.

[6] Hamadi, R. and Benatallah, B., A Petri net-based model for web service composition, Proceedings of the 14th Australasian database conference, Australian Computer Society, Inc., pp. 191-200, 2003.

[7] Bing, L. and Huaping, C., Web service composition and analysis: a Petri-net based approach, First International Conference on Semantics, Knowledge and Grid, 2005.

[8] Colombo, A., Jammes, F., Smit, H., Harrison, R., Lastra, J. and Delamer, I., Service-oriented architectures for collaborative automation, 32nd Annual Conference of IEEE Industrial Electronics Society, 2005.

[9] Chafle, G. B., Chandra, S., Mann, V. and Nanda, M. G., Decentralized orchestration of composite web services, Proceedings of the 13th International World Wide Web Conference on Alternate track papers & posters, ACM Press, pp. 134-143, 2004.

[10] Karakoc, E., Kardas, K. and Senkul, P. A, Workflow-based web service composition system, Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, pp. 113-116, 2006.

[11] Moldt, D. and Ortmann, J., A conceptual and practical framework for web-based processes in multi-agent systems, Third International Joint Conference on Autonomous Agents and Multiagent Systems, volume 3, pp. 1464-1465, 2004.

[12] Elfatatry, A. and Layzell, P., A negotiation description language, Software: Practice and Experience, volume 35, pp. 323-343, 2005.

[13] Murata, T., Petri nets: Properties, analysis and applications, Proceedings of the IEEE, volume 77, pp. 541-580, 1989.

[14] Adamyan, A. and He, D., System failure analysis through counters of Petri net models, Quality and Reliability Engineering International, volume 20, pp. 317-335, 2004.

[15] Leitão, P. and Colombo, A., An approach towards the development life-cycle of agent-based production control applications, 10th IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, pp. 8-, 2005.

[16] Anisimov, N. A., Golenkov, E. A. and Kharitonov, D. I., Compositional Petri net approach to the development of concurrent and distributed systems, Program Computer Software, volume 27, pp. 309-319, 2001.

[17] Pankratius, V. and Stucky, W., A formal foundation for workflow composition, workflow view definition, and workflow normalization based on Petri nets, Proceedings of the 2nd Asia-Pacific Conference on Conceptual Modelling, Australian Computer Society, pp. 79-88, 2005.

[18] Zhai, Z., Yang, Y., Guo, W. and Tian, Z., Integrating agent and web service into grid service workflow system, Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies, pp. 407-410, 2005.

[19] Yang, Y., Tan, Q. and Xiao, Y., Verifying web services composition based on hierarchical colored petri nets, Proc. of the 1st Int. Workshop on Interoperability of heterogeneous information systems, ACM Press, pp. 47-54, 2005.

[20] Chatain, T. and Jard, C., Models for the supervision of web services orchestration with dynamic changes, Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ E-Learning on Telecommunications Workshop, pp. 446-451, 2005.

[21] Zhang, J., Chung, J., Chang, C. K. and Kim, S. W., WS-Net: A Petri-net based specification model for web services, Proceedings of the IEEE Internat. Conference on Web Services, IEEE Computer Society, 2004.

[22] Suraj, Z., Fryc, B., Matusiewicz, Z. and Pancerz, K, A Petri net system - an overview, Fundamenta Informaticae, volume 71, pp. 101-120, 2006.