Innovative Production Machines and Systems
D.T. Pham, E.E. Eldukhri and A.J. Soroka (eds)

Engineering of service-oriented automation systems: a survey

J.M. Mendes[a,b], P.J. Leitão[c], F.J. Restivo[b], A.W. Colombo[a], A. Bepperling[a]

[a] *Schneider-Electric GmbH, P&T H&O Dept., Steinheimer Str. 117, D 63500 Seligenstadt, Germany*
[b] *Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, s/n, 4200-465, Porto, Portugal*
[c] *Polytechnic Institute of Bragança, Quinta Santa Apolónia, Apartado 1134, P-5301-857 Bragança, Portugal*

Abstract

The evolution of manufacturing systems and the emergence of decentralised control require flexibility at various levels of their lifecycle. New emerging methods, such as multi-agent and service-oriented systems are major research topics in the sense of revitalizing the traditional production procedures. This paper takes an overview of the service-oriented approach in terms of platform and engineering tools, from the perspective of automation and production systems. From the basic foundation to the more complex interactions, service-oriented architectures and its implementation in form of web services provide diverse and quality proved features that are welcome to different states of the production systems' life-cycle. Key elements are the concepts of modelling and collaboration, which enhance the automatic binding and synchronisation of individual low-value services to more complex and meaningful structures. Such interactions can be specified by Petri nets, a mathematically well founded tool with features that enhance towards the modelling of systems. The right application of different methodologies together should motivate the development of service-oriented manufacturing systems that embrace the vision of collaborative automation.

Keywords: Service-oriented architecture, Service-oriented engineering, Collaborative automation

1. Introduction

Since the introduction of manufacturing machinery during the Industrial Revolution, the reconfiguration of production systems able to respond to the market needs remains time consuming and expensive. The use of PLC's (Programmable Logic Controller) became a standard in industrial automation and provides a user defined logic to control the inputs and outputs of equipments. From the practical viewpoint, PLC-driven systems are mainly hierarchical and centralized, providing limited reconfiguration capabilities. Currently, to stay in business, a manufacturing enterprise should be able to change promptly and dynamically its product catalogue and react quickly to unexpected disturbances [1]. One of the most significant facts is the emergence of decentralized systems capable of dealing with the rapid changes in the production environment better than the traditional centralized architectures [2]. Decentralized production systems are considered organizational structures able to match agility and efficiency necessary to compete in a global market.

Different approaches had been developed and analysed to cover the requirements of reconfigurable production systems: from the agent-based and holonic systems (see [3] and [4] and the references therein) to the interoperable Web services based on Service-oriented Architectures, SoA (see [5]). This trend is explained by collaborative automation (see [6] and [7]), a set of autonomous and collaborative components and their environmental rules. The rationale paradigm of collaborative automation is explained by three main emerging technologies that are integrated respectively: holonic control systems utilizing agent-based technology, an object oriented approach to control software and mechatronics [7].

For a SoA based collaborative automation systems, there is a gap on mechanisms and engineering tools that provide interaction schemes, protocols and patterns applied for services and corresponding providers and requesters. It is not a technological issue, since Web services are already there and demonstrated by practical use (see the projects of Sirena at http://www.sirena-itea.org and Socrades at http://www.socrades.eu); but merely taking in account the viewpoint of collaborative automation, the strong requirements request further research and applicability of high level methods.

This survey provides an overview of tools that may be used for the benefits towards the idea of a collaborative SoA-based automation, especially the support for successfully interaction of Web service based components. First, section 2 expands the SoA approach, from the viewpoint of e-business to the applicability in automation and production systems. Section 3 explains some trends in form of interaction, coordination and collaboration in SoA engineering and section 4 reveals an approach of intelligent Petri net based engineering of SoA systems. Finally, the paper rounds up with the overall conclusions.

2. The service-oriented approach: main concepts

The proliferation of the internet in the '90s, due to the possibility of sharing information to other people and organisations, guided to discover of common interests and to search for new market possibilities. In order to survive the massive competition created by the new online economy, many organizations are rushing to put their core business competencies on the Internet [8]. In consequence, the emergence of Service-oriented Architecture (SoA) and one of its technological standard, Web Services (WS), became notorious. The ability to efficiently and effectively share services on the Web is a critical step towards the development of the new online economy driven by the Business-to-Business (B2B) e-commerce [8].

The idea of SoA is that distributed resources and organisations should provide their functionalities in form of services that requesters can have access to them. In heterogeneous environments made of different kind of participants, questions about interoperability arise. For this case, standard specification of Web service protocols can handle this and thus specify a set of interaction and technology rules that should be followed by all involved partners to successfully permit the conversation. A Web service itself is software that can process a received XML document through some combination of transport and application protocols [9]. Formal definition of WS and additional algebra is given in [8] and [10].

A typical Web services architecture consists of three entities [11]: i) *service providers* who create Web services and publish them to the outside world by registering the services with service registry; ii) *service registry or brokers* who maintain a registry of published services; and iii) *service requesters* who find required services by searching the service broker's registry. Requesters then bind their applications to the service provider to use particular services.

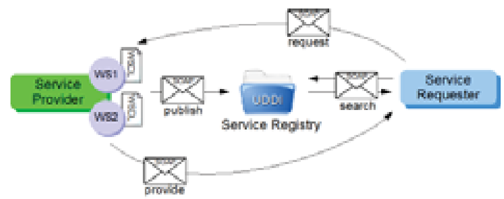

Fig. 1. Web service platform with participants and technological protocols.

Fig. 1 shows not only the three entities, but also refers to the technology side, from which Web services are build-on three essential protocols (known as the core Web services technologies [11]): Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI). Useful information and description about these and other protocols related to Web services can be found in [12].

2.1. SoA in automation

Service-oriented architecture and Web services were originally projected to electronic commerce and business, but progressively adopted by other means. The idea behind SoA fits well with collaborative automation, in sense of autonomous, reusable and loosely-coupled distributed components. The use of the SoA paradigm implemented through Web services technologies enables the adoption of a unifying technology for all levels of the enterprise, from sensors and actuators to enterprise business processes [7, 13].

Moreover, a current challenge in production control is to approach Service-oriented architectures with emergent technologies, such as multi-agent systems. In [14] the objective of the authors is to integrate software agents and Web services technologies into an agent-based serviced-oriented integration framework for cooperative distributed systems, in particular, to develop an Agent-based Web Services (AWS) solution for collaborative manufacturing systems.

From the viewpoint of collaborative automation, each autonomous component or device should provide one or more web services that match its functionalities. For example, a robot gripper (see Fig. 2) could include a service for management (device configuration), an operational service (e.g. to move an iron plate from one place to another) and an event service (e.g. to alert when the iron plate is too heavy to move precisely). To participate in collaborative activities, it should also include an interaction entity (agent) that can communicate to others and request different services for its own needs.

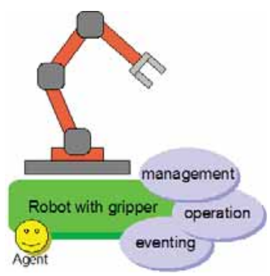

Fig. 2. Web service enabled collaborative robot.

Web services have already been implemented and demonstrated in devices. The Sirena project (see [15]) has extended the SoA paradigm into the realm of low-level embedded devices, such as sensors and actuators. 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. 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 events.

3. Connecting "loosed-coupled" entities together: the SoA engineering

When an architecture is being specified and developed with Web services technologies and thus, had made available different kind of distributed services, it is clear that the basic structure is not sufficient to automate the interactions and to permit the application of diverse functionalities. For example, we need to describe a transaction that should depend on a well establish sequence of services and also a way to validate the transaction model to discover possible deadlocks or conflicts. Thus, there are distinct tools that offer particular features for Web services. Aoyama et al. [16] has classified Web service technologies in two layers (see Fig. 3): Web services platforms to describe, discover and execute Web services, and Web service engineering to develop applications with Web services.

The platform technologies of Web services is generally overviewed in section 2, mostly related to the core protocols. As a new domain of software engineering, Web services engineering concerns every aspect from development, deployment, use, to evolution of Web services, such as analysis, architectures, development methodologies, descriptions, testing, development environments, management, and applications [16]. Relatively new are collaborations, dynamic provision and evolution of Web services.

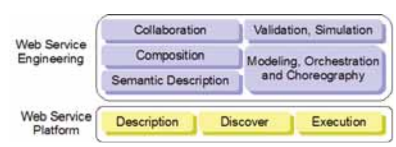

Fig. 3. Web service platform and engineering levels.

In the next sub-sections follows a description and important works inside each main SoA engineering field.

3.1. Semantic Web services and ontology

Current SoA implementations require some manual reconfiguration and for automatic processing there should be present some description that can be interpreted by software components. A potential solution and support for other techniques is to enrich the service descriptions with machine-interpretable semantics using the semantic Web services upper ontology (see [17] and [18] for more information).

3.2. Modelling, orchestration and choreography

Synchronization and parallel processes are very common in distributed computing. The case of SoA is not different and special needs require that services are invoked in a specific order and precedence, in such a way that it can be described by models (or more commonly known as work flow models). The terms orchestration and choreography are commonly used when describing interaction patterns along services from different perspective or "owner" of the conversation. The practice of sequencing and synchronized execution of services, which encapsulate business or manufacturing processes, is denominated orchestration [19]. Differently, 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 instance.

Some protocols already deal with service modelling 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) form Karakoc [20]. The authors of [21] present an executable web service architecture model (SO-SAM) that incorporates predicate transition nets with the style and understanding of component-based concepts. To overcome the centralized nature of these processes, in [22] is given a technique to partition a composite web service written as a single BPEL program into an equivalent set of decentralized processes.

3.3. Service composition

Web service composition is today a very active topic of research. It involves the combination of a number of existing Web services to produce a more complex and useful service [23]. Semantic enriched Web services (semantic Web services) provide the possibility to enhance the automatic tying and corresponding composition of Web services. Composition can be viewed as a modelling technique, but the emphasis on creating new services from old ones and the problem of efficiency and effectiveness of the process, may have different motivations and methods behind. A central challenge is the development of modelling techniques and tools for enabling the semi-automatic composition and analysis of these services, taking into account their semantic and behavioural properties [24]. In [25] is given an approach to specify and orchestrate flexible and reliable Web services compositions based on the concept of transactional patterns. Hamadi and Benatallah [8] describe a Petri net-based algebra, used to model control flows, as a necessary constituent of reliable Web service composition process. An agent-based and context-oriented approach is used in [26] to support the composition of Web services.

3.4. Analysis and simulation

In the case where work flow models are presented (for example in form of a Petri net), it is possible to analyse it and/or simulate it before actually proceed to its orchestration. A deep analysis may raise some performance costs and inconsistency when models are distributed, but may also uncover useful properties and evaluation parameters. Petri nets and Coloured Petri nets are well known tools for system analysis (see [27] for information about property specification in Coloured Petri nets).

3.5. Collaboration

Many SoA may enhance in functionality and autonomy when they incorporate ideas from multi-agent systems, such as collaboration. In other hand, a multi-agent system can use Web service as its communication platform and thus benefit of some of its standards. The collaboration among independent entities can result in a work plan that describes which services are to be requested and in which order. The approach of [28] views service orchestration as a conversation among intelligent agents, each one responsible for delivering the services of a participating organization. In this context, an agent is essentially a layer wrapping each peer organization.

4. Intelligent Petri net-based engineering of SoA

The use of Petri nets to model SoA activities have been studied in some research projects due to its unique characteristics. Petri Nets (PN) and Coloured Petri Nets (CPN) provide strong mathematical principles and are commonly used for system design and validation. The possibility of graphical representation, mathematical evaluation, validation, simulation and system tests are some of the major benefits introduced by the PNs. CPNs are an elaboration of ordinary Petri nets. In a CPN, each place is associated with a 'colour', which is a type [29].

Petri nets are commonly used to specify multi-agent systems or, more precisely, its social behaviour. A formal specification methodology for developing multi-agent-based intelligent control systems based on CPN is given in [27]. In [29] agents and Web services are integrated into grid service work flow system based on Coloured Petri Nets. This combination of agents and Web services enhances the adaptability and dynamics of the framework.

4.1. Decision and conflict detection/resolution

A conflict can be viewed as a resource or state that is to be taken by more entities than its capacity or transitions that activate from the same state leading to different paths. Both of the two situations requires mechanisms that should pro-actively detect conflicts and resolve them [30]. The existence of conflicts does not strictly mean that there are design problems in our system, but should be also understood as an opportunity of applying decision to a more flexible system. Situations in conflict may be present both in multi-agent and service-oriented systems, due to the existence of shared resources and activities.

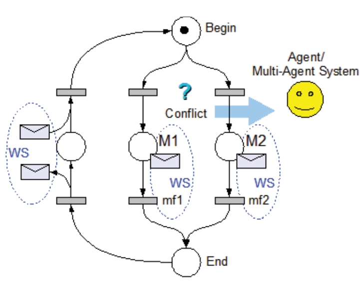

Fig. 4. Petri net based orchestration of Web services with intelligent agent for conflict resolution.

In the example of Fig. 4, a Petri net model is used to describe the relation between two machines (*M1* and *M2*) that are activated when the corresponding place has a token. Both machines have Web services and when activated, a message is sent to the corresponding machine. An event occurs when a machine finishes its operation (*mf1* and *mf2*), which is communicated also via a message. The logic here is that only one machine can operate for one request at time. The decision point is translated in the Petri net model as a conflict, but requires that someone (in this case an agent system) resolves the conflict, i.e. choose one of the machines depending on various criteria.

Decision points, alternative ways and other conflicts can be modelled in Petri nets, the same way that web services related functionalities are able to figure out as transitions and places in Petri nets. Agents itself can be technologically specified within web services and so be easily integrated in the system to provide indirection, control decision and conflict resolution. Their tasks can be extended into the reconfiguration of the domain in any case where the model does not exhibit the actual possibilities.

5. Conclusions and future work

This survey takes a trip along the service-oriented paradigm concerned with automation systems, specifically the trend of collaborative automation. The basic concepts and platform of Web services are conceptually insufficient, but the modular foundation is able to evolve and incorporate diverse methodologies and technologies to resolve high degree problems that are common in automation systems. A key point is the interaction among individual services and its providers/requesters that motivate the research from different frontlines.

Upcoming efforts are to specify and develop a basis for model representation and execution in SoA manufacturing that fits with collaborative automation and provides a flexible but stable platform for inclusion of different supporting methodologies, such as multi-agent systems for decision and conflict resolution, automatic binding of services and dynamic reconfiguration. This vision should contribute to the overall flexibility that is required by modern automation and production systems.

Acknowledgements

The authors would like to thank the European Commission and the partners of Network of Excellence "Innovative Production Machines and Systems" (http://www.iproms.org/) and the SOCRADES project (http://www.socrades.eu) for their support.

References

[1] Leitão P and Colombo AW. Petri net based methodology for the development of collaborative

production systems. IEEE Conf. on Emerging Technologies and Factory Automation, 2006.
[2] Harrison R, Colombo AW, West AA and Lee SM. Reconfigurable modular automation systems for automotive power-train manufacture. International Journal of Flexible Manufacturing Systems, Springer-Verlag, London, 2007.
[3] Deen S. Agent-based manufacturing: advances in the holonic approach. Springer Verlag Berlin Heidelberg, 2003.
[4] Leitão P. An agile and adaptive holonic architecture for manufacturing control", PhD Thesis, University of Porto, Portugal, 2004.
[5] Colombo A and Jammes F. Collaborative automation and service-oriented architectures in the industry. Tutorial at the IEEE IECON´06, November 6th, 2006, Paris, France. (See http://iecon06.iut-amiens.fr)
[6] Harrison R and Colombo AW. Collaborative automation: from rigid coupling towards dynamic reconfigurable production systems. Proc. of the 16th IFAC World Congress, Prague, July 2004.
[7] 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, 2006, pp 341-347.
[8] 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., 2003, pp. 191-200.
[9] Vogels W. Web services are not distributed objects. IEEE Internet Computing, vol 7, 2003, pp 59-66.
[10] Bing L and Huaping C. Web service composition and analysis: a Petri-net based approach. First International Conference on Semantics, Knowledge and Grid, 2005.
[11] Roy J. and Ramanujan A. Understanding Web services. IT Professional, vol 3, 2001, pp 69-73.
[12] Ma K. Web services: what's real and what's not? IT Professional, vol 7, 2005, pp 4-21.
[13] Lastra J and Colombo AW. SoA middleware form Manufacturing. Section: Applications in manufacturing and industrial monitoring and industrial surveilance. European Union Info-day IST Program, May 23rd, 2007, Brusseles, Belgium.
[14] Shen W, Li Y, Hao Q, Wang S and Ghenniwa H. Implementing collaborative manufacturing with intelligent Web services. 5th International Conference on Computer and Information Technology, 2005.
[15] Jammes F and Smit H. Service-oriented paradigms in industrial automation, IEEE Transactions on Industrial Informatics, 2005, vol 1, pp 62-70.
[16] Aoyama M, Weerawarana S, Maruyama H, Szyperski C, Sullivan K and Lea D. Web services engineering: promises and challenges. Proceedings of the 24th International Conference on Software Engineering, ACM Press, 2002, pp 647-648.
[17] Delamer I and Lastra J. Ontology modeling of assembly processes and systems using semantic web services. IEEE International Conference on Industrial Informatics, 2006, pp 611-617.
[18] Milanovic N and Malek M. Current solutions for Web service composition. IEEE Internet Computing, vol 8, 2004, pp 51-59.
[19] Delamer I and Lastra J. Self-orchestration and choreography: towards architecture-agnostic manufacturing systems. 20th International Conference on Advanced Information Networking and Applications, vol 2, 2006, 5pp.
[20] Karakoc E, Kardas K and Senkul PA. Workflow-based web service composition system. International Conf. on Web Intelligence and Intelligent Agent Technology, IEEE Computer Society, 2006, pp 113-116.
[21] 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, 2006, pp 81-87.
[22] Nanda MG, Chandra S and Sarkar V. Decentralizing execution of composite web services. Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, ACM Press, 2004, pp 170-187.
[23] Tang Y, Chen L, He K and Jing N. SRN: an extended Petri-net-based workflow model for Web service composition. IEEE International Conference on Web Services, 2004, pp 591-599.
[24] Hull R and Su J. Tools for composite web services: a short overview. SIGMOD Rec, ACM Press, vol 34, 2005, pp 86-95.
[25] Bhiri S, Perrin O and Godart C. Extending workflow patterns with transactional dependencies to define reliable composite Web services. International Conf. on Internet and Web Applications, 2006, pp 145-145.
[26] Maamar Z, Mostefaoui S and Yahyaoui H. Toward an agent-based and context-oriented approach for Web services composition. IEEE Transactions on Knowledge and Data Engineering, vol 17, 2005, pp 686-697.
[27] Colombo AW. Development and implementation of hierarchical control structures of flexible production systems using high-level Petri nets. In Manufacturing Automation Series. Klaus Feldmann / PhD-Thesis. 216 pages, 86 figures, MeisenbachVerlag Bamberg. 1998.
[28] Blanchet W, Stroulia E and Elio R. Supporting adaptive Web-service orchestration with an agent conversation framework. IEEE International Conference on Web Services, 2005.
[29] Zhai Z, Yang Y, Guo W and Tian Z. Integrating agent and Web service into grid service workflow system. 6th Inter. Conf. on Parallel and Distributed Computing, Applications and Technologies, 2005, pp 407-410.
[30] Feldmann K, Schnur C, Colombo AW. Modularised, distributed real-time control of flexible production cells, using Petri nets. Control Engineering Practice, Int. Journal of IFAC, pp. 1067-1078. August 1996.