The IEEE International Conference on
Industrial Informatics (INDIN 2008)
DCC, Daejeon, Korea July 13-16, 2008

SOA MIDDLEWARE AND AUTOMATION:
Services, Applications and Architectures

T Kirkham[1], D Savio[2], H Smit[3], R Harrison[1], RP Monfared[1] P Phaithoonbuathong[1]
[1] *MSI Research Institute, Loughborough University, Loughborough, UK*
[2] *SAP Research, Vincenz-Priessnitz-Strasse 1, Karlsruhe, Germany, Kreuzplatz, 20 Zürich, Switzerland*
[3]*Schneider Electric, Corporate R&D, Research Center T3 38050 Grenoble, France*

Abstract: The factory automation community is looking toward the Devices Profile for Web Services (DPWS) set of Web Service standards to provide a Service Orientated Architecture (SOA) enablement of factory elements. This paper examines the issues surrounding the application and future development of DPWS on production line machines and their integration with distributed enterprise applications. Issues raised reflect the need for more development on defining a generic approach to Web Service architecture in automation and the ontologies that support it. The concept of an automation lifecycle is introduced to aid this development from a more holistic perspective.

Keywords: eManufacturing, Automation, Enterprise Integration, Web Services.

1. INTRODUCTION

To date the main demonstrators using DPWS have taken place largely in the home electronics area [1]. Microsoft's support for DPWS in the Windows Vista operating system under the Windows Rally initiative, is focused on connecting electronic consumer goods to the Personal Computer [2,3]. Using the same DPWS toolkits research into use of Web Services on low computational devices in the factory automation field began to develop from major research projects such as SIERENA [4].

However, the application of DPWS in the industrial automation field is very different to the home electronics model and involves different sets of users and services [5]. From an architectural perspective the key differences between the two models is within approaches to application deployment. Inside the home the majority of communications occur on a simple and often horizontal device-to-device level with more complex and often composite vertical application level communication often being linked into specific home PC based applications.

Inside the industrial domain the challenges of device to device communication are more complex and mission critical than in the home arguably depending on greater precision in terms of service choreography, interlock or orchestration. At the "higher" enterprise application level the consumers of factory level DPWS enabled devices are likely to be distributed systems, that are often part of wider SOA based applications. This integration of the devices in wider applications raises issues related to how the factory floor elements are presented to the enterprise services,. for example regarding policies to manage the various types of users and services that may wish to interact with the services or data produced from the element level services.

This paper explores this infrastructure using an industrial automation demonstrator that has implemented element level Web Services linked to higher level enterprise software. The demonstrator was developed in the context of the SODA [10] and SOCRADES [13] projects. It demonstrates an approach for the control of elements by Web Services linked to enterprise level services, and how this control can be both improved upon and supported by extended Web Service infrastructure to support the factory.

2. RELATED WORK

2.1 Motivation

Western manufacturers are increasingly focusing on high variety and highly innovative and complex products [6]. Within the creation of these types of product the provision of agility and flexibility are important elements that must be part of the manufacturing process. Agility, is the ability to provide the capability to manage efficiently and effectively any changes/modifications that are encountered in the manufacturing process. Research into enabling agility in the manufacturing process is the subject to a large amount of current research with improved modularity as the most appealing approach [7, 8].

Realising agility traditionally impacts primarily on automation systems at the manufacturing device level and a modular approach suits the application of SOA at this level. However in SOA environments there is an increased drive to enable the system agility at higher application levels, seamlessly integrating the computational solutions at various levels is essential. to achieve this. These more ubiquitous IT focused solutions for example should enable the business to develop and predict future production trends [9]. The implementation of such a system to support planning involves the linkage of information from the factory floor to higher level management data stored in computation systems associated with applications like Enterprise Resource Planning (ERP). A linkage that needs to be enabled on a SOA level and one which may rely on the use of agents to bridge the management domains of the overall system.

2.2 SOA and Manufacturing

The use of SOA in automated manufacturing is being currently investigated within projects such as the European projects SOCRADES and SODA. The primary objective of the SOCRADES project is to develop a design, execution and management platform for next-generation industrial automation systems, exploiting the Service Oriented Architecture paradigm both at the device and at the application level [11]. Whilst the investigations into the application of the SOA are also focused on enhancing the SOA framework, the SODA project proceeds along similar lines but addresses multiple application domains, including both industrial automation, home automation, telecommunications and automotive electronics.

The central focus of these investigations is on the development of factory automation devices that can host DPWS enabled services. The needs of these devices, the applications that use them and the potential of the technology to support this functionality is important. Thus mirroring the vision of the DPWS devices being able to communicate and configure in a horizontal and vertical fashion. The DPWS toolkit uses the following Web Service Standards: WSDL 1.1, XML Schema, SOAP 1.2, WS-Addressing, WS-MetaDataExchange, WS-Transfer, WS-Policy, WS-Security, WS-Discovery and WS-Eventing [12].

The main focus of these standards is the interaction between services in environments low on computational resource. This aim of the grouping is to enable device to device abilities similar to that of Universal Plug and Play (UPnP™) [13]. This is well supported in DPWS with the inclusion of the main discovery and transfer protocols such as, SOAP [16], WS-Addressing [17] and WS-Transfer [18], supported by WSDL [15]. However, as will be discussed in this paper this model does not support the demands of emerging application models supported by enterprise systems. To support this claim the use of agents in manufacturing and the emerging generation of enterprise systems and their potential support in DPWS will now be discussed.

2.3 The Distributed Enterprise.

The use of Web Services in business in the form of emerging eBusiness technologies along with the push of globalisation is creating the need for enterprise computing environments that span national and organisational boundaries. This innovation has been made possible by the increased pervasiveness of computing via the enablement of legacy systems using SOA and Web Services. This enablement has produced the ability to introduce “just in time” characteristics to supply chain management.

The development of Web Services at factory element level has the potential to further develop the pervasiveness of the distributed computing applications for the enterprise. Through this approach a new generation of manufacturing intelligence applications is emerging. In this paper SAP MII is joined to manufacturing device-level Web Services developed on an industrial machine demonstrator.

3. WEB SERVICES AT ELEMENT LEVEL.

3.1 Service Implementation

Web Services were implemented on Schneider Electric supplied on modified Field Terminal Blocks (FTBs). These ruggedized devices support industry standard 24V DC inputs and outputs and an Ethernet communications. The hardware environment supporting these interfaces and software is based on an ARM 9 core.

These devices were installed on a Ford specified machine test rig from Festo. The mechanisms on this rig presents control problems typical in engine assembly and handling machines. The machine is separated into four sections and on each section a FTB controls the I/O for that section. Web Service interfaces to this control are presented from the FTBs that support multiple services linked to an Ethernet network by which the enterprise level communication is achieved. The FTBs and the rig can be seen in Figure 1.

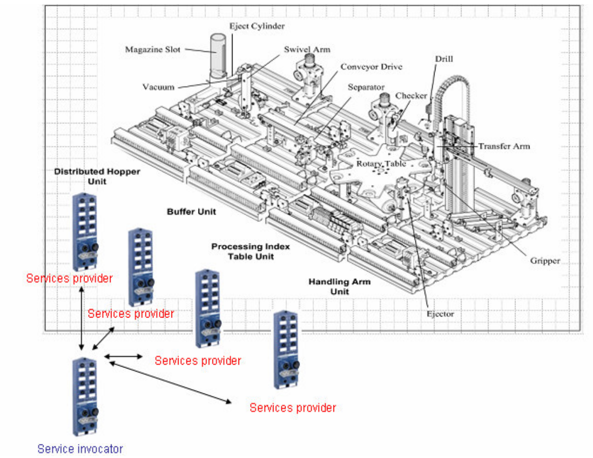

Figure 1: FTB and Rig layout

The control of the devices in order to achieve the successful completion of a machine execution is managed by a service orchestrator, incorporated in a separate FTB. This orchestrator is the point by which the services are invoked in a specific order to ensure a specific control of the elements on the test rig.

3.2 Application Execution

The basic execution of the machine involves the movement of a workpiece from one end of the machine to the other. During this process the workpiece is passed through various mechanical elements. In Web Service terms these elements have been represented by seven services, these are listed in Table 1.

TABLE I Services on the Rig.

Service Name	Function
Hopper	Receive new workpiece and eject into path of TransferArm
TransferArm	Move workpiece from Hopper to Conveyor.
Conveyor	Control flow and movement of workpieces to the Table.
Table	Control and position the movement of workpieces between Drill and Checker components.
Checker	Check if the workpiece has a hole in it.
Drill	Drill a hole in the workpiece.
HandlingArm	Move the workpiece from the Table and place into a workpiece out bay.

The functionality of each one of these services depends on the combined application specific functionality of the sensors and actuators that it consists of. The execution of the services reflect this interlock at a higher level and has been modelled using Business Process Execution Language (BPEL) [22]. This has allowed workflows to model various application execution scenarios. Using an orchestration engine such as a BPEL interpreter on a server or on the FTB it is possible to synchronise the execution of the machine using its seven main services.

3.3. Improving Execution.

The orchestration approach currently implemented is one possible solution to SOA based machine execution. One of its major advantages is that the orchestration logic is concentrated in one place and can therefore easily be changed should the machine need to be extended or reconfigured. Furthermore, the absence of direct communication between the individual devices greatly facilitates the machine set-up process as well as the replacement of devices. On the other hand, more time is spent in exchanging messages between the devices and the orchestrator than in case of direct device-to-device communication. This may or may not be an issue, depending on the dynamics of the production machine. In some production environments real-time execution and control of actuators and sensors has to be at a speed that makes the use of an orchestrator uneconomical.

In order to achieve a distributed service model without a centralised orchestrator the services will need to configure and manage the machine execution using their own embedded logic. In the SOCRADES project this approach is referred to as service choreography. Standards exist such as WS-CDL [19] to express how choreography between services can be achieved. To enhance execution speed using choreography three phases of execution of the machine can be visualised, configuration time, run time and evaluation time.

The configuration time is given to the deployment of the services on the machine and configuration of execution logic upon them. Run time is the time given to the execution of the machines and the services on it and the evaluation time is the period when the machine in the line are evaluated prior to a new execution. The process can be seen in Figure 2.

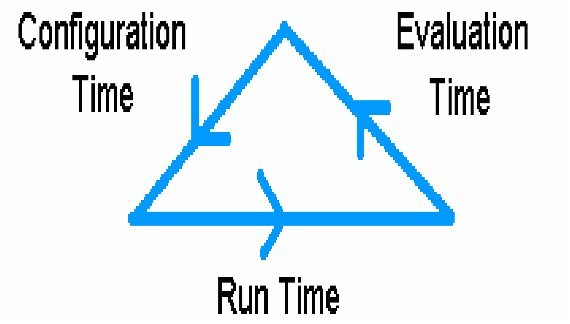

Figure 2: Machine Lifecycle.

The three phases together ensure that the maximum execution potential of the machine is achieved. Implementing using WS-Eventing publish-subscribe mechanism to service messaging allows the services to react on other service states during the Run Time phase in the model without the need for an orchestrator service.

The model also encapsulates the services together as machines. The behaviour of machines previously expressed as workflows can be grouped more effectively according to the choreography of composite parts. The inclusion of the machine into enterprise level software enables the services on the machine to become part of a wider SOA. Thus enabling the services to become part of distributed enterprise applications, thus adding the machine lifecycle to new application contexts, such as production planning and supply chain managment.

4. SERVICES AT THE ENTERPRISE LEVEL

4.1 Implementation

The Web Services on the FTBs are implemented using C. Since the Web Services are exposed over the network and accessible using SOAP messages, higher level enterprise services can communicate to the device level services. In the SOCRADES implementation SAP MII directly communicates with the elements by constructing SOAP messages to make requests and parse them to read the responses from the FTBs. The architecture of this approach can be seen in Figure 3.

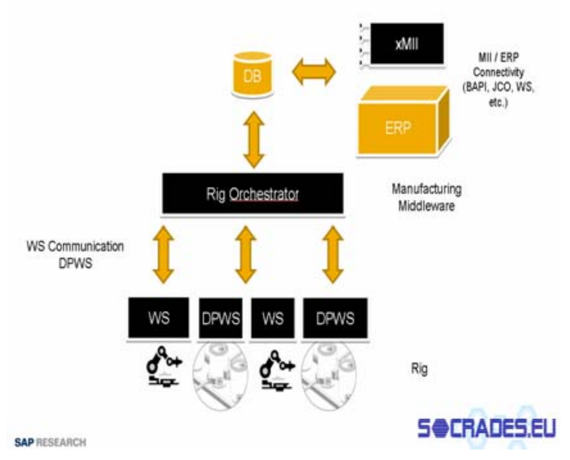

Figure 3: The vertical connection to the SAP ERP systems.

In the implementation, the SAP MII is fully aware of the changes happening on the shop floor. Any faults or competition of tasks are updated as status messages to the SAP MII. Additionally, the ERP components like Supply Chain Management (SCM) Production Planner can use the status of the shop floor to plan further new orders coming from different locations of its sales organization. The services in the shop floor are also presented in the SAP MII with business data so that the shop floor manager is fully aware of which production order is currently being executed and which ones are in the pipeline.

4.2 Key Benefits

Providing Web Service on the shop floor is effectively bringing the business logic closer to the shop floor. It enables the use of sensor data on the shop floor to make decisions on the next stages of interlocked business processes with or without human intervention. For example, if a temperature sensor is available nearby a robotic arm, the overheating of the robotic arm can be easily sensed and reported as a Web Service event to the SAP MII which could invoke the robotic arm to stop further action. It can also update the ERP system about a possible production delay at this particular place of the shop floor.

Therefore, services at device level begin to form important parts of wider enterprise level applications. The development of manufacturing intelligence applications like MII will merge and enhance existing production and supply chain management applications available to businesses. This places the device in a vital role not only to enable the completion of the physical production line, but also to correctly and expediently represent current state to reflect into wider business level applications.

4.3 Issues

When the number of Web Services is increasing on the shop floor, it may be difficult to have an overview on them. It may also become difficult to subscribe to a particular set of events. Imagining there are ten temperature sensors on a welding line, when the SAP MII is updated with the temperature events, they have to be associated with context information.

In addition to the messaging ability of the services to higher level applications, the messages and devices also need to have meaning. Hence Semantics play a major role as the device level Web Services are developed and integrated into higher applications. A more global view on semantics is emerging and collaboration will and is needed as the number of Web Services increase on the shop floor.

In order to support business models in the wider distributed enterprise application levels of Quality of Service (QoS) need to be expressed across all layers of the design. This QoS also needs to be packaged as Service Level Agreements (SLA) that can be associated with specific manufacturing line layout execution behaviour and wider application functionality. The defined metrics associated with these needs to be modelled into the machine lifecycle process for the machine in particular within the Evaluation Time phase. In order to manage this and other features of the wider architecture supporting services have to be defined and regulated to reflect the process flow of an operation.

5. AUTOMATION MIDDLEWARE

5.1Architecture Development

The wider business process approach from the machine to the enterprise application workflow, has to be defined in order for devices to become part of next generation SOA applications. In addition to a generic Web Service architecture, this infrastructure will include ontologies to support specific applications. Greater application development using Web Services can be achieved on an application by application basis. However experience in other distributed computing communities expresses the need for the development of a middleware architecture to form the basic level of service functionality as a support for development progression [20]. This functionality needs to be defined in an architecture document with specific references to the business models supported in the subject area. Similar efforts have been made in other communities adopting Web Services such as the Grid community [21].

However, the start of this process comes from the definition of the business models that need to be supported and types of applications the machine services will become part of. Research on the SOCRADES project has begun this process. The initial work on the machine lifecycle can be expanded and linked to a wider automation lifecycle concept.

5.2 Engineering Services

This expansion of the automation Web Service architecture is currently being defined around the concept of Engineering Services enabling the design, monitoring and evaluation of process execution on the engineering level. Also with the development of services such as QoS monitoring and provision of to the enterprise or higher level. Figure 4 represents how the Engineering Services should be implemented to cut across device, machine / line and enterprise level services.

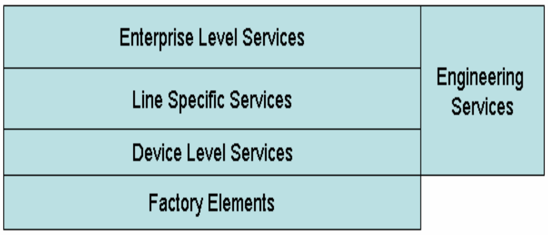

Figure 4: Engineering Services Placement.

The Engineering Services can be seen as specific to the automation domain and will reflect the domain specific ontologies. In terms of wider enterprise architecture the services could fit into existing business application models with the inclusion of specific services at the Enterprise layer.

5.3 Implementation

Currently as part of the project the main demonstrator of the Engineering services is a design tool. This tool enables a Virtual Reality Modelling Language (VRML) view of the machine and individual devices. These devices are linked to the semantic web definitions for their components, state and wider machine characteristics. The VRML is used to design the machine and simulate its working. This simulation can be based on previous machine execution data.

The visualisation services integrate with higher level services in the initial implementation to aid remote error recovery. Here errors generated on the machine can be notified by the engineering services to appropriate higher level enterprise level services. These services can then co-ordinate an error response to repair the lower level elements. In the scenario we are working on this involves a remote engineer using the visualisation tool to simulate the machine execution using the real data from the run that created the error.

6. FUTURE WORK

6.1 Scalability

Scalability test and management techniques need to be developed for the project. For example, as the number of Web Services increase on the shop floor, they needed to be understood more meaningfully in a global view. For example on a shop floor, there might be hundreds of automation devices to be started or stopped at a particular time. Exposing them all as Web Service would need specific management. Hence the Web Services need to be added with more context and semantic syntaxes that management tools can easily distinguish different events and places of the shop floor.

7. CONCLUSION

The development of Web Services at factory device level exposes shop floor devices to higher level applications creating a need for support of these services and applications. This support relies on the use and definition of common ontologies and a defined architecture in terms of Engineering services linking the domains. In order to exploit this new functionality business models need to be developed and supported. By following these lines and building on the research in this paper engineering services can support new business models from the enterprise to factory floor.

8. REFERENCES

1. Schlimmer, J. “A Technical Introduction to DPWS” http://msdn2.microsoft.com/en-us/library/ms996400.aspx

2. Article on ConnectedLife. Homepage http://www.msnbc.msn.com/id/16300970 /

3. Windows Rally technology homepage http://www.microsoft.com/whdc/rally/default.mspx

4. SIRENA Project homepage http://www.sirena-itea.org

5. Francois Jammes, Antoine Mensch, Harm Smit, "Service-Oriented Device Communications Using the DPWS," *21st International Conference on Advanced Information Networking and Applications Workshops (AINA'07)*

6. Chan, K. and Spedding, T. (2003). "An integrated multidimensional process improvement methodology for manufacturing systems." *Computers & Industrial Engineering 44(4): 673-693.*

7. Ong, M.H., West, A.A., Monfared, RP. and Harrison, R., "Application of enterprise modelling technique for specifying a component-based system", *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2199, 1st September 2005, 649-664*

8. Schäfer, C: "On the Modularity of Manufacturing Systems", *IEEE Industrial Electronics Magazine, Vol. 1(3), Fall 2007, 20-27*

9. Helo, P. (2004). "Managing agility and productivity in the electronics industry." *Industrial Management & Data Systems 104(7): 567-577.*

10. SODA (Service Oriented Device and Delivery Architecture) Project homepage http://www.soda-itea.org

11. SOCRADES Web Introduction: http://www.socrades.eu

12. The DPWS Specification http://schemas.xmlsoap.org/ws/2006/02/devprof/

13. UPnP Forum http://www.upnp.org/

14. Web Services Dynamic Discovery (WS-Discovery) http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf

15. Web Services Description Language (WSDL) 1.1 (W3C Note): http://www.w3.org/TR/wsdl

16. SOAP 1.2 (W3C standard) – Part 1: Messaging Framework http://www.w3.org/TR/2003/REC-soap12-part1-20030624/ – Part 2: Adjuncts http://www.w3.org/TR/2003/REC-soap12-part2-20030624/

17. Web Services Addressing 1.0 (WS-Addressing) – Core http://www.w3.org/TR/2006/REC-ws-addr-core-20060509/ – SOAP Binding http://www.w3.org/TR/2006/REC-ws-addr-soap-20060509/

18. Web Services Transfer (WS-Transfer, W3C Member Submission): http://www.w3.org/Submission/WS-Transfer/

19. Web Services Choreography Description Language 1.0 (WS-CDL, W3C Candidate Recommendation) http://www.w3.org/TR/ws-cdl-10

20. T. Dimitrakos, D. M. Randal, F. Yuan, M. Gaeta, G. Laria, P. Ritrovato, B. Serhan, S. Wesner, and K. Wulf, “An Emerging Architecture Enabling Grid Based Application Service Provision,” in *Seventh International enterprise Distributed Object Computing Conference (EDOC’03), Brisbane, Queensland, Australia, September 2003, 240–251.*

21. The Open Grid Service Architecture Specification: http://www.globus.org/ogsa/

22 T. Andrews, F. Curbera, H. Dholakia, Y. Goland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. “Business Process Execution Language for Web Services Version 1.1.” *Technical report, 2003*

All quoted Web references were last accessed on 20 March 2008