The Challenges Along the Road to the Realisation of a Factory Automation Lifecycle.

F Ubis Lopez[1], T Kirkham[2], B Matthews [3], J Lastra[1], R Harrison[2], V Villaseñor[1], A[2] Chowdrey[2]

[1]Tampere University of Technology
[2]Loughborough University
[3]Science and Technology Facilities Council UK
[1] fernando.ubis@tut.fi
[2]t.d.kirkham@lboro.ac.uk
[3] b.m.matthews@rl.ac.uk

Abstract:

Lifecycle support of a process from conception, implementation and evaluation can be enabled in a distributed cross organisational environment using Service Orientated Architectures (SOA). In the automation domain the development of such lifecycles are both vital for the competitiveness of European manufacturing and challenging due to the environments they exist in. This paper explores the main innovations needed to achieve SOA based application lifecycle management in the automation sector initially looking at a successful lifecycle implementation in the eScience community. This highlights the need for clearly defined standards architecture and ontology building methodologies for emerging SOA communities to achieve lifecycle management around clearly defined and managed metadata.

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

Introduction

The field of industrial automation is currently embarking on the course to adopt a SOA approach at factory element level. Factory elements are grouped together to form automation lines, and a element is usually a part such as a drill or conveyor. The SOA enablement of these devices will bring this area of industry within the wider SOA application community.

The development of the automation domain into wider distributing computing applications is an area of research that depends on common ontology's being defined. Common ontology's form the backbone of understanding between SOA computing applications. This paper
examines how integration of factory floor elements can be achieved suggesting a methodology for ontology definition.

The result of the integration of computing elements surrounding automation in a SOA will enable the development of an automation lifecycle. This lifecycle is the support of production lines from design to implementation. Using emerging Service Orientated Architectures (SOA) this process can be made more efficient and tightly linked. Central to the development of the lifecycle is the challenge of implementing the SOA and the creation of services at element level.

The concept has parallels in other SOA implementations such as in the eScience community. In addition to the Ontologies supporting this SOA approach, the paper will also explore the technologies and architecture to achieve this in the automation model. Within the automation domain this starts from the ideas motivating the use of services instead of agents in the automation domain.

Lifecycle

The concept of a lifecycle in this paper, can be illustrated by the work of the Information Management Group (IMG) at the Science and Facilities Council (STFC) in the UK [1]. STFC is home to some of the largest science facilities in the UK and Europe and a centre for international research. The aim of the IMG has been to support on site and visiting scientists with their planning, execution and evaluation of experiments. To aid this group has developed a integrated eScience lifecycle management system based around the management of Metadata and associated Ontologies.

The lifecycle is the application of the metadata model on all interactions that any eScience user and experiment has in the facility. This enables the distributed resources in the organisation such as storage, processing power and visualisation tools to be highly compatible**.** Thus aiding the development of new applications within the model and the easier integration of new users, data and equipment.

For the automation domain this type of management of data has its obvious benefits. These stem from the increased productivity enabled by reduced cost in the integration of new equipment and users to similar models. However before this can be achieved the infrastructure to support this needs to be achieved. Central to this is the support of distributed computing at factory floor level.

From Agents to Services

Research into the Web Service enablement of factory production line elements is exposing the well established research area of factory automation to a rapidly developing field of service based computing. Projects such as EU SOCRADES [2] are focused at enabling elements at factory floor level to be part of wider Service Orientated Architectures. Technically this is a challenge, as the implementation of services to operate in production line applications and environments is one which is met by extremes at both ends. For example the SOA enabled elements are expected to support real time communication speeds in environments that present limited computational power[3].

These challenges have held back the development of web services to enable SOA applications to embrace factory floor elements, and currently the development of SOA based enterprise solutions are only just beginning to include provision for resources at this low level [4]. To date the use of agents has been achieved in research as a link between higher level enterprise logic to production line routing [5]. However, agent based systems in the automation domain have tended to fit application and vendor specific applications, as agents tend to be task specific.

The use of task specific agents implies that the agent contains specific logic that is either passed onto the agent from another source or implicit within the agent. Also it suggests that the number of agents is a finite resource. A broad use of agents that can be seen to have its dangers, as it can often lead to the creation of complex agents that often carry out multitasks and respond to various prompts. A problem in particular in terms of scalability for a system that relies on agents often as the main link between the lower level services and higher level application.

In a factory environment where there is little change and dynamism this use of agents can be seen as not an issue. But the idea in using SOA is to enable an embrace of the new flexible and dynamic approaches. Collaborations based on service state need to be enabled by agents and assessments made on production routing based on real time flexible data.

The use of services naturally build upon the applications that agents began to expose to business level software. Service development using web service toolkits present more of an open canvas for the application designer. This is due to Web Service standards being universally agreed upon and their ability to support with a large degree of flexibility a wide range of service functionality [6].

Therefore, by using services the application designer has greater flexibility and control over the elements at factory floor level. This greater functionality is the main attraction behind research into this area of SOA enablement. As the goal of enabling the flexible and reconfigurable manufacturing line is cited as vital if western manufacturers are to remain competitive in the face of mass production at low cost from their far eastern competitors [7,8]. A motivation that is pushing the development of services at factory floor level to become part of wider distributed applications.

Service Implementation

The challenge of linking services at element level to wider SOA applications is specifically linked to the lack of support on elements of Web Services. This development of services at device level was pioneered by the SERENA project [9]. This research project used the Device Profile for Web Services (DPWS) toolkit to enable the representation of factory floor devices on relatively low power computational devices. The implementation and development of DPWS [10] essentially uses G-SOAP [11] as a base technology building more advanced standards for Web Service communication on top of it.

The SOCRADES project is the focus of this paper and aims to build upon the developments in SERENA and other related projects in the development of DPWS and SOA at factory floor level. The implementation of the services at element level uses Arm 9 [12] hardware and supports DPWS over a Real Time Operating System (RTOS).

Service Infrastructure

In order to improve the agility of a manufacturing system the service infrastructure of the SOA needs to be defined. This infrastructure along with the ontology building to be discussed later are the main cornerstones of designing distributed service base applications and therefore the automation lifecycle. The agility of manufacturing will be enhanced by the greater pervasiveness enabled by element level Web Services in Manufacturing planning and management software. This level of service will exist at traditionally the ERP level, and will depend on a middle service architecture to link the services on the factory floor with higher level enterprise service requirements.

The key aim of the middleware refinement is the definition of the types of service functionality needed between the lower and higher levels. In the Grid community architectures in the form of the Open Grid Standards Architecture are central to the application of SOA based Grid resources [13].

Metadata Management

The need for a clear architecture to enable the development of elements in a wider SOA has been discussed along with the service implementation challenges in the field. However in order for the architecture and services to be effectively integrated in the application domain and linked to other domains a clear roadmap for the development of metadata management needs to be made.

Multiple vendors have created different definitions in the domain, but are the natural starting point when defining the Ontologies needed to link logic across the wider applications. Although in the SOA world already competing approaches are emerging, or example the OPC-UA[14] consortium are developing a rival web service grouping from DPWS to support their classification standards.

Therefore the development of SOA has to overcome this real hurdle in the way of SOA enabled automation. A picture that is made more complex when the various ways of building Ontologies are taken into account. For example, Lassila and Mc Guinnes categorize them based on the richness of their internal structure [15], Mizoguchi et al categorize them based on the subject of conceptualization [16]. Van Heijst et al present a two-dimensional categorization based on their internal structure and the subject of conceptualization [17]. Guarino categorizes them based on their level of dependence on a particular task or point of view [18].

Ontology Building

In this document the classification proposed by Gomez-Perez is followed [19]. Their classification is based on the subject of conceptualization; particularly, two types of Ontologies are considered. *General or common Ontologies*: specify knowledge that can be reused across domains, and *Domain Ontologies*: allow specifying tasks, activities and processes; usually as a specialization of a top-level ontology, but still at a generic level to be reused across a domain.

Ontologies reusability resides on the language in which the ontology is described. Particularly, the practical ontology development has been carried out using the Description Logic (DL) formalism implemented with the Web Ontology Language (OWL).

The fundamental premise in DL is "to represent the knowledge of an application domain (the "world") by first defining the relevant concepts of the domain (its terminology), and then using these concepts to specify properties of objects and individuals occurring in the domain (the world description)" [20]..

Ontology Implementation

The best known current approach to Ontology modelling is done by using OWL-DL. However, there is a large knowledge of interest in the domain and new Ontologies are rapidly being developed. Ubis built the ABAS Domain Ontology [21]. This practical ontology implementation is related to the Agent-Based Assembly System concept developed in TUT by Martinez Lastra [22]. In this case, the process taxonomy is the framework of the ontology, where it is represented through a hierarchical classification of the assembly activities. Because the ontology is built under the OWL-DL the different classes and subclasses are defined.

The terminology used in the ABAS Domain Ontology is derived from the classification developed by Vos [23], which is based in the standard DIN 8593 [24], later extended with contributions of Groover [25] and Boothroyd et al. [26]. The use of the appropriate terminology is crucial during ontology modelling. Appropriate terminology will facilitate ontology reusability and lately ontology reconciliation if different Ontologies with the similar domain are merged.

The different upper-classes are defined: *Manufacturing Processes*, *Assembly Tasks*, *Assembly Process* and *Assembly Operations*. These upper-classes contain correspondent subclasses which define the different manufacturing processes, operations and tasks.

The classes are linked with each other by the different properties which describe the relations between the classes and subclasses, and the relations between the different operations are fixed. As a result the domain ontology is created and it is ready for its later use.

Under the scope of SOCRADES different demonstrators have been developed by the different partners. The Factory Automation Systems and Technologies Laboratory (FAST Lab) as active member of the SOCRADES consortium has developed its own equipment ontology, this is the based on the demonstrator equipment. The modularity of the systems has allowed building modular Ontologies where the equipment parts and features have been implemented. The modular ontology building has extended the reusability of the different modules.

The product ontology describes the products to be manufactured. The product ontology describes the parts which comprises the product and the relationships between the different parts. The assembly features and the geometrical constraints together with the product assembly operations are modelled in the ontology. As a result the product ontology introduces classes of the different product parts, linked with the respective properties which represent the different features.

Future Development

The development of the automation lifecycle is made more complex by the wide ranging metadata models and Ontologies being developed in the field. The OWL-DL approach in this paper can be seen as the best way forward at the current moment in time, and it may be the case that the development is split along application specific lines. Although effort is being made to merger Ontologies within this domain.
Currently the application of the Ontology has been made on lines implemented at the Fast Lab, as the SOCRADES projects develops the number of demonstrators will expand. At Loughborough University effort is being made in the area of Engineering tools to support the design of lines. These tools link to the SOA based elements on the line to aid planning and monitoring of factory environments. Merging this design and monitoring phase with the common Ontologies is a current area of work.

Conclusion

The automation domain is unique in that it has developed back to front in comparison with many other SOA enabled computing domains. The enablement of element level services to create a true SOA of automation involves the development of specialised embedded web services and has restricted the adoption of SOA in the sector. Whilst conversely the establishment of equipment classification from various vendors in the domain can be seen to have created a potential metadata mine field. This is now being compounded by the potential development of various separate Ontologies to navigate the field.
The future of the domain however does rely on useful SOA at this level to increase Enterprise competitiveness. Therefore, the development of an automation lifecycle in the domain is vital. The challenges in implementing the services are gradually being overcome. The creation of a Architecture for all interested parties to conform to should be a first step to aid the development of the goal, and projects such as the SOCRADES project have a important role in highlighting this challenge. But vendors have to recognise and industry demand that the collaboration is needed.

References

[1] Kirkham T, Matthews B et al Using User Driven Dynamic Grid Services to Integrate New Users and Old Systems in e-Science Facilities. eChallenges 2007.
[2]SOCRADES homepage : http://www.socrades.eu/
[3] François Jammes, Harm Smit, "Service-Orient Paradigms in Industrial Automation", Schneider Electric Corporate Research, France.
[4] Karnouskos, S et al. "Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure." 12th IEEE Conference on Emerging Technologies and Factory Automation
[5] S. Bussmann, N. R. Jennings and M. Wooldridge. "Multiagent Systems for Manufacturing Control," Springer-Verlag, 2004.
[6] M. P. Papazoglou and D. Georgakopoulos, Service oriented computing (special issue), *Communications of the ACM* **46**(10) (2003).
[7] R. Harrison, A. W. Colombo, A. A. West and S. M. Lee: "Reconfigurable Modular Automation Systems for Automotive Power-Train Manufacture", *International Journal of Flexible Manufacturing Systems, Springer-Verlag, London. 2007.*
[8] Helo, P. (2004). "Managing agility and productivity in the electronics industry." Industrial Management & Data Systems **104**(7): 567-577.
[9] SIRENA project homepage www.**sirena**-itea.org *last accessed 1st Marchy 2008.*
[10] The DPWS Specification http://schemas.xmlsoap.org/ws/2006/02/devprof/ last accessed 20-09-07
[11] Robert A. van Engelen and Kyle Gallivan, "The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks" Proceedings of the 2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002), pages 128-135, May 21-24, 2002, Berlin, Germany.
[12] ARM homepage www.**arm**.com/
[13] The Open Grid Service Architecture Specification: http://www.globus.org/ogsa/ last accessed 20-09-07
[14] OPC-UA homepage: http://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC
[15] Lassila O., McGuinness D., The Role of Frame-Based Representation on the Semantic Web, Technical Report KSL-01-02, Knowledge Systems Laboratory. Stanford University, Stanford, California, 2001.
[16] Mizoguchi R., Vanwelkenhuysien J., Ikeda M., Task Ontology for Reuse of Problem Soling Knowledge, In: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, University of Twente, Ensched, The Netherlands, Amsterdam, 1995.
[17] van Heijst G., Schreiber ATh., Wielinga BJ., Using Explicit Ontologies in KBS Development, International Journal of Human-Computer Studies, 1997.
[18] Guarino N., Formal Ontology in Information Systems, In: Conference on Formal Ontology in Information Systems (FOIS'98), Trento, Italy, 1998.
[19] Gómez-Pérez, A., Fernández-López, M., Corcho O., Ontological Engineering, Ed. Springer, London, 2003.
[20] F. Baader, and W. Nutt, "Basic Description Logics", In Description Logic Handbook, F. Baader, D. Calvanese, D. McGuinness, D. Nardi, D., P. Patel-Schneider (Eds.), Cambridge University Press, 2002, pp. 349–372.
[21] Ubis F. Tools and Methodologies for Knowledge Representation: Generation of an ABAS Domain Ontology. Master of Science Thesis, Tampere University of Technology, Finland, 2006.
[22]Martinez Lastra, J.L.. Reference Mechatronic Architecture for Actor-based Assembly Systems. Doctoral Thesis. Tampere University of Technology, Finland, 2004.
[23] Vos, J.A.W.M., *Module and System Design in Flexibly Automated Assembly*, Ph.D. Thesis, Delft University of Technology, The Netherlands, 2001.
[24] Deutsches Institute für Normung e.V., Manufacturing Processes Joining, *DIN 8593 Parts 0-8*, 2003.
[25] Groover M.P., Fundamentals of Modern Manufacturing, Second Edition, John Wiley & Sons, 2002.
[26] Boothroyd G., Dewhurst P., Knight W., Product Design for Manufacture and Assembly, Second Edition, Marcel Dekker, 2001.