Behaviour and integration of service-oriented automation and production devices at the shop-floor

J.M. Mendes* and F. Restivo

Department of Informatics Engineering,
Faculty of Engineering – University of Porto,
Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal
E-mail: marco.mendes@fe.up.pt
E-mail: fjr@fe.up.pt
*Corresponding author

A.W. Colombo

Schneider Electric Automation GmbH,
Steinheimer Str. 117, D-63500,
Seligenstadt, Germany
E-mail: armando.colombo@de.schneider-electric.com

P. Leitão

Department of Electrical Engineering,
Polytechnic Institute of Bragança,
Quinta Sta. Apolónia, Apartado 134,
5301-854 Bragança, Portugal
E-mail: pleitao@ipb.pt

Abstract: Automation and manufacturing systems are changing in the direction of cooperative ecosystems with heterogeneous entities. An important feature that should be considered is the vertical integration from the business needs down to the shop-floor where the real action takes place. This paper analyses the integration of shop-floor devices into the IT-enterprise but maintaining also a certain degree of independence in terms of behaviour. Service-oriented paradigm is used as the main backbone due its proven merits in the business levels and recently also in automation and production systems. In the provided example, high-level Petri nets (HLPN) demonstrate a set of useful features, namely the partial behaviour description and analysis and some parameters in the integration. The resulting application leads to an easy integration of autonomous devices in the IT-enterprise, taking especially in account the requirements of the shop-floor level.

Keywords: service oriented architecture; IT-integration; Petri nets; industrial automation.

Reference to this paper should be made as follows: Mendes, J.M., Restivo, F., Colombo, A.W. and Leitão, P. (2011) 'Behaviour and integration of service-oriented automation and production devices at the shop-floor', *Int. J. Computer Aided Engineering and Technology*, Vol. 3, Nos. 3/4, pp.281–291.

Biographical notes: J. Marco Mendes is a Student Researcher of the Artificial Intelligence and Computer Science Laboratory (LIACC) at the Faculty of Engineering of the University of Porto. He is also a collaborative Researcher for anticipation projects at Schneider Electric GmbH in Seligenstadt, Germany. He is currently working on his doctoral thesis in Informatics Engineering, entitled 'Service-oriented engineering of automation components based on high-level Petri nets'. His research interests are service-oriented architectures in automation and production systems, Petri nets, distributed systems, software engineering and multi-agent systems.

Francisco J. Restivo is an Associate Professor at the Department of Informatics Engineering of the School of Engineering of the University of Porto since 1988 and the Scientific Director and member of the Board of IDIT – Institute for Development and Innovation in Technology, Santa Maria da Feira, Portugal, since 1999. His research interests include digital signal processing, intelligent production systems, complexity management and e-learning. He holds a DPhil degree from the University of Sussex (1981) and he has been awarded the Third Millennium Medal by the IEEE in 2000.

Armando W. Colombo is a Manager of Anticipation/Advanced Projects in the CAPS Department, Schneider Electric BU Automation, France, Germany and the EU. He received his Doctor degree in Engineering from the University of Erlangen-Nuremberg, Germany in 1998. Currently, he is coordinating the FP6 European Integrated Project on 'Service-oriented cross-layer infrastructures for smart embedded devices (SOCRADES)'. He is also acting as a Deputy Leader of the cluster 'Production Automation and Control (PAC)' of the FP6 NoE on 'Innovative Production Machines and Systems' (I*PROMS)'.

Paulo Leitão received his MSc and PhD degrees in Electrical and Computer Engineering, both from the University of Porto, Portugal, in 1997 and 2004, respectively. From 1993 to 1999 he developed research activities at the CIM Centre of Porto, and from 1999 to 2000 at IDIT – Institute for Development and Innovation in Technology, Santa Maria da Feira, Portugal. He joined the Polytechnic Institute of Bragança, Portugal, in 1995, and currently he is a Coordinator Professor in the Department of Electrical Engineering of that institute. His research interests are in the field of intelligent production systems, agent-based and holonic control, reconfigurable factory automation and bio-inspiration.

1 Introduction

New information technologies (IT) have begun to have a strong presence in the new generation of manufacturing systems. After decades of parallel development, the paths of information systems tools and manufacturing systems are converging to provide the impetus that will allow the integration of the total business enterprise (Fitzgerald, 1992). A special attention is given where the main action relies (the production itself) and its incorporation in the IT-enterprise.

Several proposals have been made to enhance the flexibility, usability and integration, from holonic and multi-agent systems (see Bussmann and Schild, 2001; Deen, 2003; Leitão et al., 2005), HLPN (see Colombo, 1998), to the more novel service-oriented production systems (see Jammes and Smit, 2005; Lastra and Delamer, 2006). One promising approach that has the potential to surmount the technical, organisational and financial limitations inherent in most current approaches, is to consider the set of production entities as a conglomerate of distributed, autonomous, intelligent and reusable units, which operate as a set of collaborating entities (Bepperling et al., 2006). Each of these entities is typically constituted by hardware – mechatronics, control software and embedded intelligence, and is able to communicate with others. From a functional point of view, each collaborative unit can, at each time, initiate actions and dynamically interact with other ones in order to achieve both local and global objectives, considering that they are within a cross-layer infrastructure like a manufacturing enterprise (see Colombo and Harrison, 2006, and the references therein).

Contributions have been made about the integration of service-oriented components from the factory's shop-floor into the IT-enterprise, in order to provide vertical information and control sharing. One possible solution is that these components provide the required services to the upper levels of the enterprise and thus are controlled by them. This fits with the service-oriented paradigm, but the traditional master-slave hierarchy from the top-down perspective is not favourable when speaking about collaborative and proactive devices. A partial solution is a mix of autonomous devices from the shop-floor level that may provide services, but also may request services provided by the other levels, such as decision making systems (DMS), manufacturing execution systems (MES) (see Qiu and Zhou, 2004), and enterprise resource planning (ERP), (see Gibson et al., 1999).

The aim of this paper is to present a methodology for analysis of the operational behaviour of autonomous service-oriented automation and production devices at the shop-floor and their integration into a flexible IT-enterprise. The goal is to provide a more proactive control to the shop-floor devices, contrasting with the more common service-oriented factory automation systems where devices are only providers of services to the upper levels (such as DMS/MES/ERP).

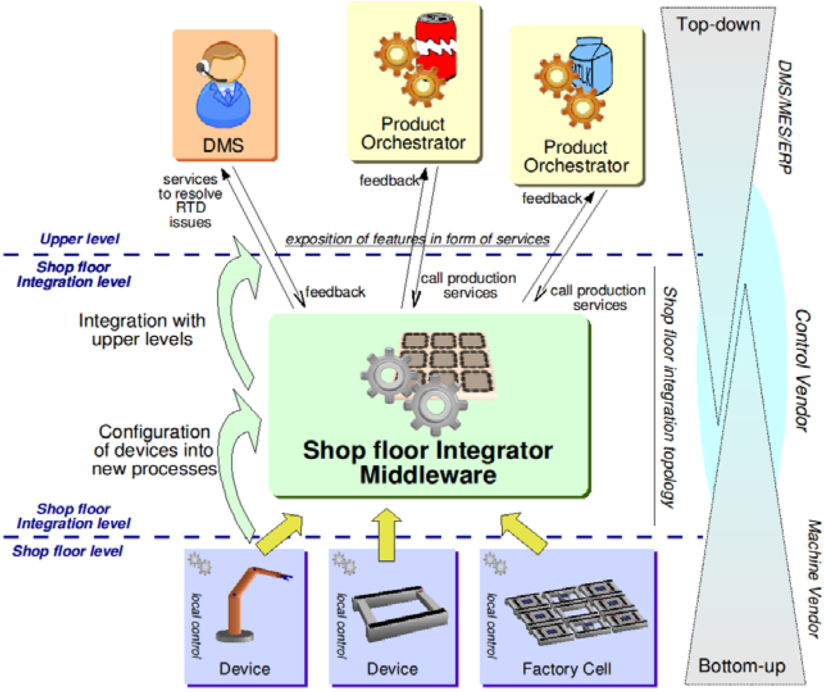
The paper is structured in the following sections: first, the integration middleware is presented in Section 2 to provide an easy and feasible consolidation of DMS/MES/ERP and other upper level systems with the shop-floor. Section 3 goes deeper into the shop-floor, where the automation devices are running and explains the autonomous behaviour of these service-oriented devices. An application example of these concepts using HLPN is given in Section 4 and finally Section 5 terminates the paper with the conclusions.

2 Integration middleware for service-oriented shop floor automation systems

The proposed approach describes a middleware to integrate service-oriented shop floor automation components into a flexible IT-enterprise architecture. In terms of context, it focuses on automation and production systems based on distributed and reconfigurable devices and their integration from the shop floor level into the higher levels (such as the DMS/MES/ERP systems). The resulting middleware infrastructure manages and interfaces the shop floor to the higher level by using a function vectorial and layout based process control of devices. All inter- and intra-level interactions are service-oriented.

The exposed services represent diverse and sufficient functionalities of the shop floor to be managed and integrated into the higher levels, providing a high order of control and information feedback. Some of the exposed features include, but not limited to: topology information, maintenance, several control operations, conflict resolution, and process analysis and monitoring. The approach is based on a bottom-up perspective (from the devices) to interface the top-down approach from the upper levels. There is a loose-coupled inheritance, so the higher levels are not limited to one specific business activity, but to any depending on the provided features by the middleware. The methodology applies and extends the concept of service-orientation by the adoption and integration from the device level up to the enterprise levels. Since services are available in both sides, the integration middleware is also capable of using the services of the upper-levels when required by the middleware itself or forwarded by some device from the shop-floor.

Figure 1 Shop floor integration middleware in the IT-enterprise architecture (see online version for colours)



One main outcome of the approach is that the middleware works as a system integrator for service-oriented shop-floor with other service-oriented components of IT-enterprise architecture (see Figure 1):

- Integrate service-oriented production devices (control vendor and machine vendor).
- Integrate shop-floor with upper-levels, such as production management systems (control vendor and DMS/DES/ERP).

Each production device (integrating mechatronics, communication and control aspects) is a process controller that offers services that have been defined and developed by the device manufacturer. The sequence and link of services offered by each device follows properties and restrictions that are generated by the hardware (mechatronics), communication and possible control aspects of the device. The set of services offered by a device can be mapped into a set of functions (vectors). Composing the function vectors allows knowing all possible combinations (aggregation of services), thus the vector space of the device contains all possible (allowed) individual and aggregated services.

Configuring a layout implies composing devices and of course, formally spoken, composing their vectorial spaces, generating new restrictions and a new set of aggregated services. Again, the analysis of the vectorial space gives all necessary acceptable specification about the complete set of services offered by a given configuration. Vectorial spaces may also represent executable processes bounded to the services that are accessible and manageable by any level that has access to them. Thus, the integrated behaviour is driven by processes that access available services. Processes can be distinguished from the automation processes (at the shop floor), integration processes, production processes to the business and enterprise processes.

After fixing a given layout, a very well defined set of services has been defined. It is possible to speak here about process definition of the shop floor topology. Putting a fixed layout running implies that the sequence of called services will have to respect mechatronics, communication and control restrictions like shared-resources, capacity of processing, concurrency, etc., that are explicitly included in the topology of the aggregated services.

The integration topology for a given automation/production system is now able to be interfaced with external service-oriented components, such as product orchestrator and decision-making systems (DMS). For example, a defined workflow associated to a product will be able to call services (individual or aggregated), offered by the integrator topology of the shop floor only if the right interface exists and the controller associated to a product recognise which services are to be found in the shop floor integration topology.

3 Autonomous behaviour of service-oriented devices at the shop-floor

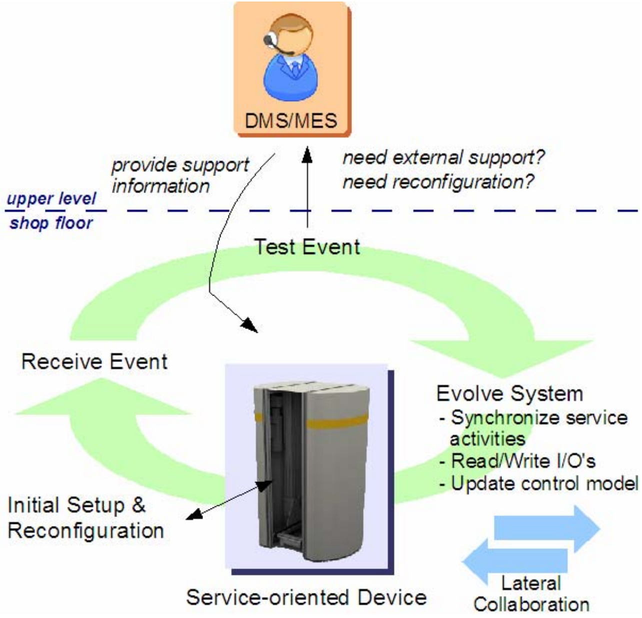
From the previously explained integration, nothing was pointed about the devices itself, i.e., their behaviour and how they contribute to the overall system. The addressed automation and production units at the shop-floor are sometimes recognised differently, for instance ‘modular intelligent automation unit’, ‘physical agent’, ‘holon’, ‘collaborative automation units’, ‘smart mechatronic component’, etc. (see Leitão, 2004; Colombo, 2005; Bepperling et al., 2006).

The autonomous behaviour methodology given in this section focuses on the setup and running phases in the life cycle of collaborative service-oriented production systems. These systems are composed of distributed, reconfigurable smart production automation devices that expose their functionality as services or aggregation of them. It permits, after the initial setup of the automatic operation of the device and the exposition of their services, the collaborative interaction/cooperation in order to follow flexible and customised workflows associated to the products to be manufactured.

Under the umbrella of the service-oriented architecture paradigm, the services exposed by the production devices at the shop floor have (see Figure 2):

- Lateral/horizontal (device-device) relationship. Calling services or responding with services as e.g., move-pallet-in, pick-gripper, transport-pallet-up, etc.
- Vertical (device-upper-level) relationship. Calling services offered by upper-level components for dynamic rescheduling of production/automation tasks, responding with services as monitoring, diagnosis/information indexes, etc.

Figure 2 Concept of the independent behaviour of autonomous components (see online version for colours)



The devices have a degree of autonomy in sense of auto-sustainable control and exposition of necessary services to permit lateral/horizontal collaboration with other devices (aggregating services), requesting and providing services for decision-making information from DMS/MES and integration. All interactions and resource sharing are via service-orientation. There is a loused-coupling inheritance in form of bottom-up perspective (from the devices/shop floor level), enhancing the autonomy and consequent reconfiguration capabilities.

Having the shop-floor mapped into a service-oriented system, the behaviour of each smart production automation device is part of a middleware shell formally specified by e.g., HLPN models and supported by routines to handle undocumented events and decide over present conflicts in the behaviour of the device. The operational behaviour of these devices follows the token-game of the HLPN (i.e., the behaviour control of the HLPN), it is then self-controlled and guided by internal/external events that link the shell with other components of the system. These events may also correspond to service calls.

Several procedures are defined for the behaviour life-cycle of these components (see Figure 2):

- Initial setup of the device, including configuration, exposition of services, establishment of connections to other devices/components and putting it on waiting initial state.
- Events are received via service operations, internal device interface to I/O and generated directly by the control (e.g., from decisions/conflicts).
- The received event must be tested. If it corresponds to some description of the control model's actual state, then the system is evolved by updating the control model, synchronising service activities, accessing to I/O and report diagnosis/monitoring indexes. In case of being an exception, an undocumented event or an internal conflict, some decision is required. If the device has the necessary information to resolve it, special procedures are taken to interfere in the normal system's control and new events are generated. In the case of not having sufficient control over the event, it may ask for services provided by external components of the system (e.g., DMS) to provide more useful information for a concrete decision over the problem.
- After processing the event and evolving according to the token-game of the model-based middleware shell, the system reaches the next state and it is again able to receive other events.

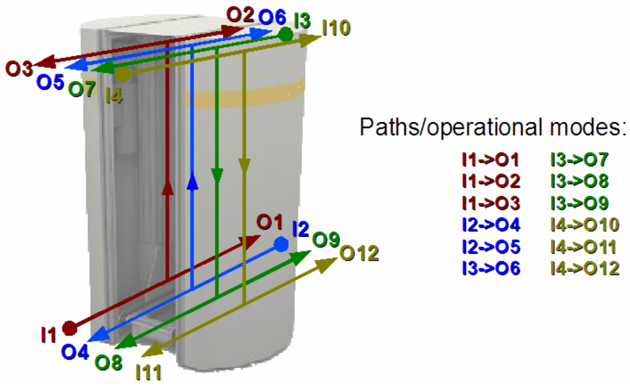
The application of this method results on autonomous devices that are self-controlled and have less dependency to other components, especially from the upper levels such as the DMS. In short terms, the features of these devices are:

- service-orientation, event-based evolution according to the rules of the token-game of a HLPN
- autonomous control and consequent behaviour
- event-based life-cycle following the rules of e.g., the token-game of a HLPN
- handling of documented events and exceptions to the normal control.

4 Application example using high-level Petri nets

The methodology is applied to a mechatronics device corresponding to a lifter with two levels and four different ports where pallets can be inputted and outputted. These ports should be used to connect to other devices, such as conveyors, but can be triggered manually by placing a pallet in the lifter (since a sensor detects it). Figure 3 shows a representation of the lifter with all 12 possible predicted operation modes (that corresponds to the paths that a pallet may take) and that are represented in a service-oriented specification as 12 aggregated services.

Figure 3 Lifter with possible paths/modes (see online version for colours)



The behavioural control of the lifter is formally represented by a HLPN model that shows the global operation in the different modes. The atomic service *top-left-transfer-in* (see Figure 4) implies production automation operations associated to reading/writing to corresponding I/O and synchronising the service activity (e.g., a conveyor requests the service). The activation of this transition is done when:

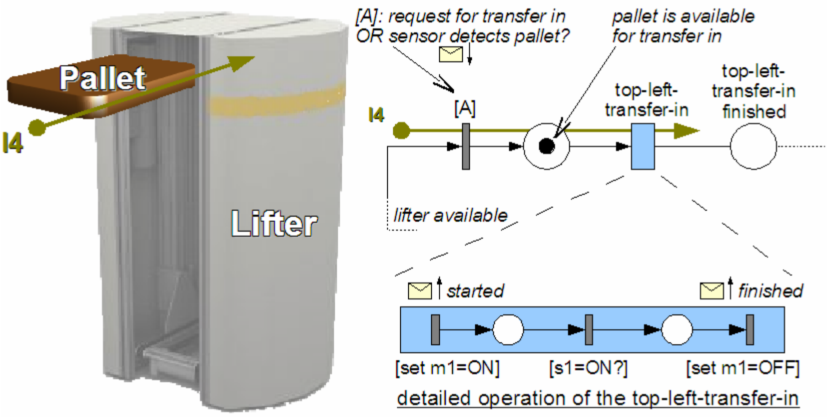
1 the model presents a logic state true (lifter is ready to offer the service, i.e., the lifter is available)

2 when a conveyor asks for the service or a sensor did detect a pallet in that port.

Other situations that are not documented can also be handled and needs special procedures, as referenced previously.

Transfer out operations (such as the *top-right-transfer-out* transition) should be done synchronously with connected transfer devices (e.g., conveyor) to be able to provide a smooth transitional movement of the pallet from one device to another one. This requires that the lifter requests a transfer in service of the connected conveyor.

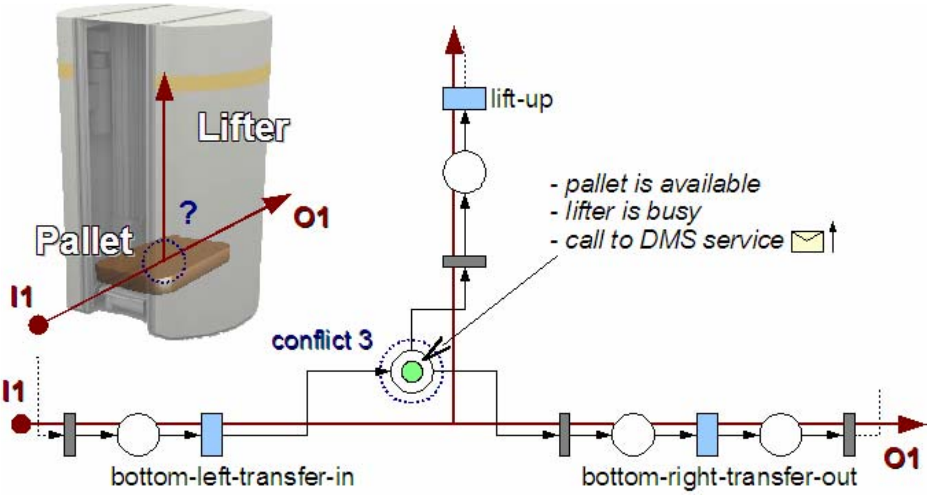
Figure 4 Operation of the top-left-transfer-in service (see online version for colours)



After the initial setup and configuration of the device with the control model and additional routines, the device is available offering/exposing services and waiting for events and call services.

For example, a connected conveyor is requesting the *bottom-left-transfer-in* service (I1). In this case, and since it is a documented event in the control model, the device proceeds to evolve the system by running the HLPN model and taking the related actions. After the *bottom-left-transfer-in* is successfully concluded, the system has to confront an exceptional event that is thrown by a conflict in the model (see Figure 5).

Figure 5 Example situation in which a conflict is the actual state of the model (see online version for colours)



If it does not have the necessary information to decide, it must call specialised components to help in the procedure. As in Figure 2, DMS are used for this purpose. The lifter sends a request (call service) for support to the DMS, including supporting information [the ID of the pallet and possible outputs (service to be called): *lift-up* and *bottom-right-transfer-out*]. Based on the workflow of the pallet, a decision is returned in form of an event to the lifter and now it should be able to resolve the conflict and evolve the system. For example, the lifter receives the suggestion to do a *lift-up* from the DMS, so it may proceed to do a *lift-up*. In any case, the final decision is up to the lifter that considers the received suggestion, but may operate differently in case of internal situations (e.g., occupation of the lifter and/or connected conveyors).

The application example shows some of the features and benefits of the used methodologies such as independent control definition for devices, requirements in collaboration, service-oriented lateral and vertical integration and decision support (specially when the device does not have the necessary parameters to decide and should ask others for information). The control of the device was defined using HLPN due its valuable properties, but it can be anything else including the additional support for intelligent routines to enhance the normal control situations.

5 Conclusions

The goal of this paper is to provide an approach to configure flexible shop floors as an integration method and to specify the elements to develop the middleware for a transparent integration of shop floor components (e.g., devices) into a service-oriented IT-enterprise. In account is taken the formalisation of the operational behaviour of autonomous and collaborative automation devices at the shop floor.

Several advantages can be distinguished from the methodology. On one hand, the inherited features of service-oriented architectures merged with traditional configuration approaches for shop floor, and on the other hand, a new and innovative integration approach is formally grounded on vectorial analysis theory. Each device has an autonomous service-oriented control behaviour that is local to the device but that is linked to the other device based on the layout configuration of the shop floor. The middleware shell makes these possible behavioural links based on offering/calling services. The decision mechanisms associated to the behaviour of aggregated services are local to the neighbouring devices but can also be influenced by information associated to the whole system, including shop floor (lateral collaboration) and upper level components (vertical collaboration) of the IT-enterprise system.

Acknowledgements

The authors would like to thank the European Commission and the partners of the EU IST FP6 project SOCRADES, the EU FP6 (I*PROMS), and the EC ICT FP7 project 'Cooperating objects network of excellence' (CONET) for their support.

References

Bepperling, A., Mendes, J., Colombo A., Schoop, R. and Aspragathos A. (2006) 'A framework for development and implementation of web service-based intelligent autonomous mechatronics components', *Proceedings of the IEEE International Conference on Industrial Informatics*, pp.341–347.

Bussmann, S. and Schild, K. (2001) 'An agent-based approach to the control of flexible production systems', *Proc. of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*, Vol. 2, pp.481–488.

Colombo, A. and Harrison, R. (2006) 'Modular and collaborative automation: achieving manufacturing flexibility and reconfigurability', *Int. Journal of Manufacturing Technology and Management*.

Colombo, A.W. (1998) 'Development and implementation of hierarchical control structures of flexible production systems using high-level Petri nets', PhD thesis, Fertigungstechnik, Erlangen, Nuernberg.

Colombo, A.W. (2005) 'Industrial agents: towards collaborative production-automation, -management and -organization', *IEEE Industrial Electronics Society Newsletter*, Vol. 52, pp.17–18.

Deen, S. (2003) *Agent-Based Manufacturing: Advances in the Holonic Approach*, Springer Verlag.

Fitzgerald, A. (1992) 'Enterprise resource planning (ERP)-breakthrough or buzzword?', *Proc. Third International Conference on Competitive Performance through Advanced Technology*, pp.291–297.

Gibson, N., Holland, C. and Light, B. (1999) 'Enterprise resource planning: a business approach to systems development', *Proc. of the 32nd Annual Hawaii International Conference on HICSS-32 System Sciences, Track7*, p.9.

Jammes, F. and Smit, H. (2005) 'Service-oriented architectures for devices – the SIRENA view', *Proceedings of the 3rd IEEE International Conference on Industrial Informatics*, pp.140–147.

Lastra, J. and Delamer, I. (2006) 'Semantic web services in factory automation: fundamental insights and research roadmap', *IEEE Transactions on Industrial Informatics*, Vol. 2, No. 1, pp.1–11.

Leitão, P. (2004) 'An agile and adaptive holonic architecture for manufacturing control', PhD thesis, Faculty of Engineering, University of Porto, Portugal.

Leitão, P., Colombo, A. and Restivo, F. (2005) 'ADACOR: a collaborative production automation and control architecture', *IEEE Intelligent Systems*, Vol. 20, No. 1, pp.58–66.

Qiu, R. and Zhou, M. (2004) 'Mighty MESs: state-of-the-art and future manufacturing execution systems', *IEEE Robotics & Automation Magazine*, Vol. 11, No. 1, pp.19–25.