**Preprints of the 19th World Congress**
**The International Federation of Automatic Control**
**Cape Town, South Africa. August 24-29, 2014**

# Supporting Sensing Enterprise Operations with Polymorphic Service Infrastructures

**Anthony Karageorgos*, Nikolay Mehandjiev*,**
**Elli Rapti****

**Manchester Business School, University of Manchester, UK*
*(e-mail: {anthony.karageorgos, n.mehandjiev}@mbs.ac.uk)*
***Institute of Research and Technology – Thessaly, CERTH, Volos, Greece*
*(e-mail: erapti@ireteth.certh.gr)*

**Abstract:** Sensing enterprises span both physical and virtual boundaries and require support for dynamic re-configurability and decentralised collaboration among large numbers of data retrieval and information processing nodes. Current information technology infrastructures lack sufficient support for decentralised service formation requiring either concentration of all relevant information to a central decision making point or considering static distributed processing node configurations. In this paper we propose a service-oriented infrastructure for supporting sensing enterprise operations which is based on polymorphic services. Such services are capable of adapting and evolving dynamically according to fluctuations of the environmental context. The applicability of the proposed approach is discussed in an exemplar case study scenario concerning a Medium Density Fiberboard production process.

*Keywords:* Sensing Enterprises, Internet of Services and Service Science, Multi-agent Systems.

## 1. INTRODUCTION

The term 'Sensing Enterprise', originally introduced in 2011, refers to enterprises that process multi-dimensional information captured dynamically through physical and virtual objects to obtain added value information that enhance their global context awareness and shape their future decisions (2011). Unlike virtual enterprises, whose main objective is to allow a number of organisations to rapidly develop a common working environment (Martinez et al., 2001), Sensing Enterprises focus on the interaction among objects and systems and are characterised by decentralised intelligence and fusion between virtual and real world. Furthermore, Sensing Enterprises are characterised by context awareness, dynamic configurability and multi-identity oriented virtual entities (Santucci et al., 2012) having self-* properties (Li, 2011).

In this paper we propose an information technology infrastructure based on a multi-layered software architecture that supports sensing enterprise operations through decentralised provision of polymorphic services. Therefore, the main contribution of the proposed approach is the decentralized provision of polymorphic services in the context of the sensing enterprises. Such services capable of adapting and evolving according to changes in their environmental context enable self-organization, self-configuration, self-optimization, self-adaptation and self-evolution of the supported enterprise operations.

The remainder of this paper is organized as follows; Section 2 outlines the motivation behind our work. Section 3 summarizes relevant work concerning infrastructures that support sensing manufacturing enterprises. Our proposed approach in presented in Section 4. In Section 5 we describe the applicability of the proposed infrastructure in an indicative case scenario emphasizing the need of polymorphic services in a shop floor manufacturing environment, while in Section 6 we conclude our work and present future research goals.

## 2. MOTIVATION - BACKGROUND

The Sensing Enterprise is organically linked with the Internet of Things (IoT) metaphor (Santucci et al., 2012), that is the pervasive presence of a variety of things or smart objects, such as RFID tags, sensors, actuators, mobile phones, which are able to interact with each other and cooperate with their neighbours to reach common goals (Atzori et al., 2010). The IoT metaphor aims to bridge the gap between the physical world and its representation in information systems (Haller et al., 2009). Even though the integration of things and smart objects in traditional enterprises may bring tremendous opportunities, at the same time it poses new challenges, primarily concerning decentralisation of control and support of self-* properties (2011).

### *2.1 Decentralisation of control*

Sensing enterprise operational decisions are the collective result of local information processing and actions taking place at several decentralised control points that operate in an autonomous manner. This view of operational control distribution is aligned with the notion of autonomous processing node networks such as smart dust in clouds. Smart dust refers to systems consisting of many tiny micro-electromechanical systems (MEMS), such as sensors, robots, or other devices, that can detect environmental stimuli (light, temperature, vibration, magnetism, pressure) and are distributed over some area forming a wireless network to carry out tasks in a decentralised and collective manner. By exploiting the power of such processing node networks and decentralised intelligence sensing enterprises are able to perform distributed analysis and decision making both in real and virtual worlds (Lázaro et al., 2012).

Centralised information processing approaches can deprive real-time enterprise decision making since information is commonly gathered in an a-posteriori fashion (Thoma et al., 2013). Furthermore, centralisation suffers from scalability and fault-tolerance problems which negatively affect information system functionalities and limit their capabilities (Corchado et al., 2012).

On the other hand, decentralised models where functionality is executed in a distributed way by low-cost devices have several advantages including (Marin-Perianu et al., 2007, Trentesaux, 2009, Souza et al., 2008):

- Reducing the load on the back-end system
- Decreasing process execution and transactional costs
- Providing better response in time-critical situations,
- Improving service quality and the overall system flexibility of the system
- Facilitating system evolution due to unexpected changes in the external or internal environment
- Improving overall system functionality by combining distributed service orchestration and task-specific processing.

### *2.2 Self-* Properties*

In (Li, 2011) the term re-invention was introduced in order to describe the self-* properties in the sensing enterprise concept. Due to the use of IoT technologies in future enterprise systems the scale and complexity of ICT systems is exponentially increased. That leads them to become self-adaptable, self-organised, self-optimising, self-configuring, self-protecting, self-healing, self-descriptive and self-discovery networks in which the objects themselves can make autonomous decisions on behalf or for humans (Santucci et al., 2012). Moreover, the lack of need of human control in expanding areas of the system will lead to a system that will be able to take care of itself and will be characterized among others by self-repairing, evolving and emergent behaviour (2011). Self-organization and emergent behaviour will be key issues to support the new generation of reconfigurable control systems (Shena et al., 2007).

Current solutions do not sufficiently support these requirements either by adopting a centralized point of view to data and processing organization or by considering distributed nodes that lack autonomous functionality. We propose an approach based on polymorphic services where large numbers of interconnecting autonomous nodes capable of sensing input data and providing adaptive and evolving services are formed dynamically in a decentralized and collective fashion.

## 3. RELEVANT WORK

The integration of software agent technologies with service-oriented computing provides a promising solution for cooperative distributed systems integration, and particularly for next generation collaborative manufacturing systems (Shena et al., 2007). Wang, Ghenniwa and Shen (Wang et al., 2008) proposed a distributed manufacturing scheduling framework at the shop floor level where dynamic scheduling is achieved by the cooperation of the scheduler agent, the real time control agent and resource agents and distributed scheduling is conducted through Web services. Shen et al. (Shena et al., 2007) proposed an agent-based service-oriented integration architecture to leverage manufacturing scheduling services on a network of virtual enterprises where the scheduling process is orchestrated on the Internet through the negotiation among agent-based Web services. Both approaches lack context-awareness which constitutes one of the most important aspects of sensing enterprises.

In order to add context awareness, some approaches combine agent and service oriented architectures with RFID technology and 'smart objects', while also integrating social relationships. Kosmatos, Tselikas, and Boucouvalas in (Kosmatos et al., 2011) proposes a layered architecture by integrating both RFID and smart object-based infrastructures, while also exploiting the social aspect of the participating objects, shaping the "Social Internet of Things". Castelli et al. in (Castelli et al., 2011) propose a middleware that exploits the graph of a social network (e.g., Facebook), in conjunction with relations deriving from spatial proximity, to drive and rule the actual topology of interactions among devices, users, and services so that it will support decentralized and effective service discovery and orchestration, and will enable tackling critical privacy issues. While our work focuses on sensing enterprises, it also aims to enhance the aforementioned approaches by introducing the use of polymorphic services in order to offer self-adaptation and self-evolution to the system.

In the manufacturing industry, Souza et al. proposed the SOCRADES middleware (Souza et al., 2008), an architecture focused on coupling web service enabled devices with enterprise applications through the use of DPWS. Zhang et al. in (Zhang et al., 2010) propose the SO-Gateway which is used to connect and centrally manage multiple types of smart objects for capturing real-time manufacturing data according to a specific logic flow for further application in enterprise information systems. In the context of sensing enterprises Lázaro et al. introduced the FASyS (Lázaro et al., 2012), a centralized model for personalized risk management. Moreover, Thoma et al. (Thoma et al., 2013) poposed the use of linked services to access sensor devices in order to assure interoperability and connect the shop floor to the top floor. The aforementioned approaches focus on aspects of dynamic service discovery and composition. Our approach on the other hand is intended to addresses the adaptation and evolution aspects in service provision, which can offer self-* properties to the system. In addition, while the majority of them focus in a centralized approach, the decentralized approaches do not fully support the sensing enterprise concept.

## 4. OUR APPROACH

In this paper, we propose a decentralized architecture for polymorphic service provision in sensing enterprises. The proposed approach offers:

- Self-adaptation: polymorphic services can adapt according to environmental context parameters.
- Self-evolution: polymorphic services are able to evolve following evolution of their execution context.
- Decentralised control: the lack of central management controlling the composition and execution of polymorphic services allows each service to be executed independently offering increased scalability and reliability, and improving thus overall performance.

### *4.1 Polymorphic Services*

The ubiquity of computing power has created additional issues in service provision. Contexts are no longer concrete and static for service adaptation methods to operate by selecting the best candidate among pre-recorded service-to-context mappings, as is normally the case in service adaptation methods (Brogi and Popescu, 2007). Context can also change and hence service adaptation needs to consider evolving contexts leading to what is commonly termed service evolution (Carl K. Chang, 2009).

Service evolution refers to altering service functionality in order to match contexts and requirements that were not specified in exact form (Ming et al., 2008, Carl K. Chang, 2009, Papazoglou, 2008). When services have both adaptation and evolution capabilities then they can deliver their functionality in different forms, and therefore we term them polymorphic services (Karageorgos et al., 2013). Polymorphic behaviour in services can be exhibited either on the composed services or before and during service composition. We envision that polymorphic services are going to be the next trend in service provision in extremely dynamic environments such as IoT settings and mobile clouds (Fernando et al., 2013).

Service adaptation is often achieved by applying certain execution strategies that modify service functionality. In this view adaptive service applications continuously monitor and modify the applicable strategies based on changes in the service environment (Cheung et al., 2008). In addition, software evolution refers to gradual and constant change of service functionality in particular situations over a period of time (Mittermeir, 2001). Therefore, service evolution can be seen as a gradual response to context-aware feedback received from the service environment.

Adaptation and evolution can be carried out by a single service component, for example when services resulting from a composition process adapt and evolve their execution parameter values according to context fluctuations. However, in the general case polymorphic behaviour is the result of a collective process involving a number of atomic services that interact and dynamically modify the composed services in an autonomous manner. In the latter case there is no central point of authority and all atomic services act rationally according to a decentralised behavioural model.

### *4.2 Service Provision Points*

Polymorphic software services execute within computational environments which we term Service Provision Points (SPPs). Each SPP comprises at least one module of four possible types: service modules, agent modules, interface modules and support modules (Fig. 1).

Support modules enable binding of real world objects, such as RFID readers, sensors and smart screens, to middleware and expose their capabilities as polymorphic services, which are stored in appropriate service modules. This “servitisation” of real world object functionality allows utilisation of their capabilities through established methods for service management, discovery and composition. For example, an SPP can be an RFID reader that is able to retrieve information from RFID tags and provide them as an information service. Furthermore, support modules enable SPPs to communicate with each other forming SPP networks. For example, the temperature and humidity sensors in a living room can generally be interconnected and coordinate associated effector devices, such as the air-condition, based on decentralised optimisation models.

Services exposed by SPPs are represented internally by software agents, which we term service agents, according to a technique known as service “agentification” (Murguzur et al., 2013). Service agents are stored in agent modules and each agent functions on behalf of a service with respect to its interaction with other services and the environment. Since multiple services can be hosted in an SPP, depending on the capabilities of the represented real-world object, multipleagents may reside in each SPP. While services contribute to interoperability in the system, the use of agent technology enables building of high-level models with flexible interaction patterns (Shena et al., 2007).

Finally SPPs comprise appropriate interfaces for providing

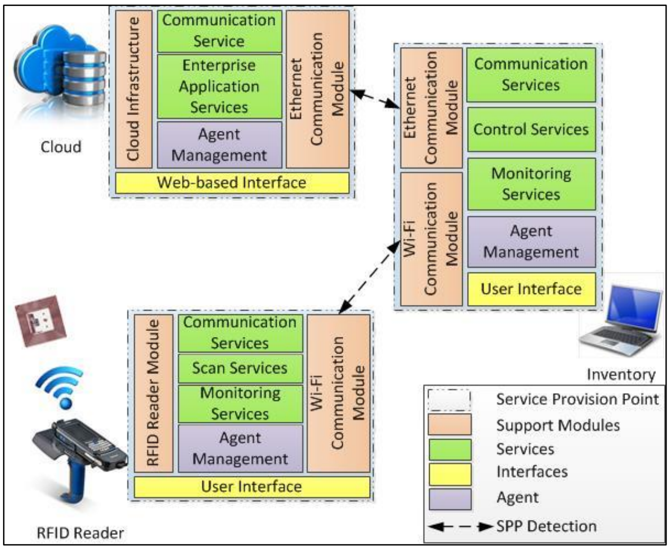


Fig. 1. SPPs software architecture in a sensing enterprise

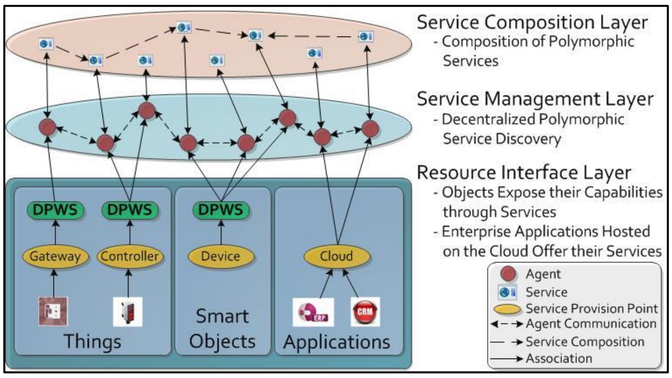


Fig. 2. The proposed system layered architecture

service functionality and sensing external input. Service provision is driven by and carried out by the respective service agents.

### 4.3 *Software Architecture*

Along the lines of generic IoT architectures, such the one described in (Kosmatos et al., 2011), we propose a three-layered software architecture. which comprises layers responsible for representing real-world object functionality as services, managing and configuring available services, and composing services in a decentralised manner through service agent interactions (Fig. 2).

The Resource Interface layer translates service specifications to device specific commands and vice versa, binding this way real-world objects to middleware and exposing their capabilities as services to upper layers. This functionality is realized by semantically describing object capabilities enabling thus automated service discovery, composition and monitoring.

The Service Management layer provides decentralized service discovery, for example by using epidemic spreading or gossiping algorithms, as well as service status monitoring and service configuration.

Finally in the Service Composition layer services are composed by joining and combining services discovered at the Service Management layer, for example using message passing algorithms. Service composition is carried out dynamically and the correct sequence of execution of the polymorphic services is determined according to the pre-conditions and post-conditions of the individual services.

### 4.4 *Proposed Technologies*

We propose to use agents as service orchestrators and manipulators as is widely adopted in service research and has been proven a promising computing paradigm for efficient service discovery and integration in enterprise information systems (Wang et al., 2008, Shena et al., 2007, Zhang et al., 2010, Li et al., 16-18 December 2013). Agent technology provides a natural way to realize enterprise integration effectively and has been widely implemented in manufacturing applications for its autonomy, flexibility, reconfigurability, and scalability. On the other hand, service-oriented architectures enable the creation of components that can be assembled and deployed in a distributed environment and have provided a new and excellent solution to the data integration among heterogeneous and distributed systems (Zhang et al., 2010).

Service agents create mobile ad-hoc networks in order to be able to search for available services in their closest peers (nodes). The created networks are also connected to the internet which enables the nodes to search for candidate services on the cloud. We propose using Device Profile for Web Services (DPWS) protocol, which defines a minimal set of implementation constraints to enable secure Web Service messaging, discovery, description, and eventing on resource-constrained devices. Its objectives are similar to those of Universal Plug and Play (UPnP) but, in addition, DPWS is fully aligned with Web Services technology and includes numerous extension points allowing for seamless integration of device-provided services in enterprise-wide application scenarios (Hellbruck et al., 16-18 December 2013). Devices that are not web service enabled (RFID, barcode, sensors) are connected to wrapper/Gateway and/or service Mediator devices that encapsulate the device functionality and offer a service to the outside world. For example, an RFID reader is used as a Gateway depicting the capabilities of an RFID through services.

For service specification we propose using the Linked Unified Service Description Language (USDL) (Leidig and Pedrinaci, 2012). Linked USDL instead of only being a typical service description language, goes beyond the technical interface and consists of different modules which cover functional, operational and business aspects. In addition, it is modelled in Resource Description Framework (RDF), which allows the usage of already existing domain specific vocabularies (Thoma et al., 2012).

### 4.5 *Agent Internal Architecture*

The internal architecture of a service agent is depicted in Fig. 3. The service discovery module receives the environmental stimuli through the sensors and defines the agent's goal using the plan library. The plan library is a collection of pre-conditions and post-conditions of services that can be used in order to define each agent's goal in every situation. For example, it could be a fuzzy rule base of an agent that takes as input a temperature and according to the temperature value defines its goal, namely what kind of service output will satisfy its needs. Thereafter, the agent launches a

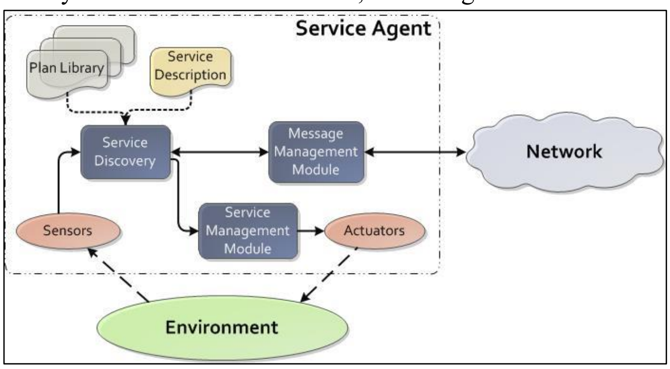


Fig. 3. Polymorphic service agent internal architecture

decentralized algorithm such as one based on epidemic discovery or message-passing based on input/output matching in order to establish if the goal can be satisfied by the service provided by the agent, using the information provided in the service description. If the goal cannot be satisfied by the service provided by the agent the Message management module launches a new service discovery request to the neighbouring nodes by using a variation of the flooding protocol. Last but not least, the service management module is responsible for the composition and translation of the chosen services to a set of device specific commands and vice versa.

## 5. POLYMORPHIC SERVICE PROVISION IN THE MDF MANUFACTURING – A CASE STUDY SCENARIO

In a Medium Density Fiberboard (MDF) production plant the raw material normally consists of wood chips that are typically delivered by truck or rail from offsite locations, such as sawmills and furniture manufacturing facilities. If wood chips are prepared onsite the logs are debarked, cut to more manageable lengths and then they are stored in a log yard to be collected for further processing and sent to the chippers. In order to monitor the storing conditions of the wood chips in the log yard, wireless piezoelectric sensors have been placed to record the weight of the wood chips. If the sensors detect abnormal weight reduction without human intervention, it means that fungi and insects have infected the wood chips (mass reduction due to degradability). To control the infestation, the production plant owns a mobile fungicide sprayer that takes action to avoid further degradation.

In the above settings we distinguish two SPPs: a) the piezoelectric sensor nodes offering the mass loss calculation service and b) the mobile fungicide sprayer that offers and the fungicide spray service respectively. SPPs use a wireless communication protocol, thus they are only able to communicate with SPPs that are within a limited range. Since the fungicide sprayer SPP is not static, the sensor SPPs are only able to communicate directly with it only when it is located within a specific range. Therefore, when a sensor SPP searches for a service offered by the fungicide sprayer SPP, then it would have to communicate and coordinate with the other SPPs in order to locate it.

In exceptional cases, for example due to an equipment failure which leads to the suspension of the production for two whole days, the wood chips will have to remain in the log

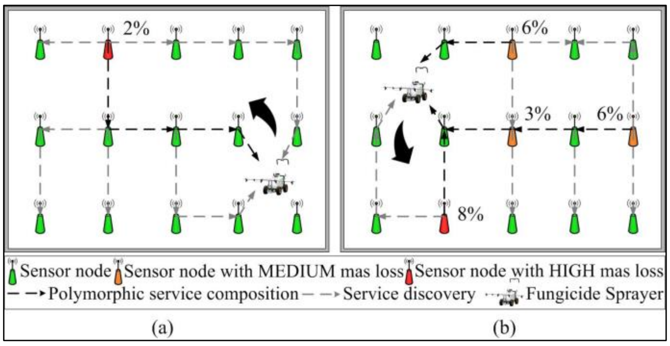


Fig. 4. Fungi and insect infestation control in wood chips stored in a log yard, (a) before the service evolution and (b) after the service evolution

yard for longer than it was expected. According to the context input, which in this case is the measured mass reduction and is noted on each node (Fig. 4(a)), the fungicide spray service can offer different policies, such as different spraying techniques providing adaptation. As more and more nodes start detecting mass loss (Fig. 4(b)), due to their large number the fungicide sprayer will have to evolve its parameters for example in order to define where it should spray each time. Therefore, polymorphic service provision in this highly heterogeneous and dynamically changing environment of this sensing enterprise can offer self-* properties to the system.

## 6. DISCUSSION

The introduction of polymorphic services, which present adaptable and evolving behaviour according to the environmental stimuli, for the seamless integration of networked things and software applications in the sensing enterprise, offer self-* properties to the system. Therefore, the system is able to respond to the variability of the environmental context beyond those that were envisaged at design time. In addition, the introduction of a decentralized infrastructure for polymorphic service provision in the context of sensing enterprises can offer scalability, flexibility and real-time decision making, which is of extreme importance in enterprises and enterprise networks. Furthermore, to enhance interoperability we proposed the implementation of polymorphic services as linked services as proposed in (Thoma et al., 2013).

## 7. CONCLUSIONS

We proposed a decentralized architecture using polymorphic services for a dynamically changing environment of a sensing manufacturing enterprise. Polymorphic services offer dynamic adaptation and evolution according to the changing context, which offers dynamic configuration and self-* properties to support sensing enterprise operations.

Our future research plans include the experimental evaluation of the proposed approach in terms of scalability, adaptability and performance. Moreover, we plan to extend the proposed model by including social dimensions, such as business relationships between enterprises, in a sensing enterprise network. In addition we aim to address interoperability issues regarding the integration of the various components of the sensing enterprise.

**Acknowledgements:** This research is supported by the EU Marie-Curie Intra-European Fellowship for Career Development grant No. PIEF-GA-2011-301125.

## REFERENCES

Future Internet Enterprise Systems (FInES) 2011. Position Paper on Orientations for FP8. *A European Innovation Partnership for Catalysing the Competitiveness of European Enterprises*

Atzori, L., Iera, A. & Morabito, G. 2010. The internet of things: A survey. *Computer Networks,* 54**,** 2787-2805.

Brogi, A. & Popescu, R. 2007. Service Adaptation Through Trace Inspection. *International Journal of Business Process Integration and Management,* 2**,** 9.

Chang, C. K., H.-Y. J., H. Ming, & K. Oyama 2009. Situ: A Situation-Theoretic Approach to Context-Aware Service Evolution. 15.

Castelli, G., Mamei, M., Rosi, A. & Zambonelli, F. 2011. Pervasive Middleware Goes Social: the SAPERE Approach. *Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops.* Ann Arbor, MI.

Cheung, R., Yao, G., Cao, J. & Chan, A. 2008. A Fuzzy Service Adaptation Engine for Context-Aware Mobile Computing Middleware. *International Journal of Pervasive Computing and Communications,* 4**,** 20.

Corchado, J. M., Tapia, D. I. & Bajo, J. 2012. A Multi-Agent Architecture for Distributed Services and Applications. *International Journal of Innovative Computing, Information and Control,* 8**,** 2453-2476.

Fernando, N., Loke, S. W. & Rahayu, W. 2013. Mobile cloud computing: A survey. *Future Generation Computer Systems,* 29**,** 84-106.

Haller, S., Karnouskos, S. & Schroth, C. 2009. The Internet of Things in an Enterprise Context. *In:* Domingue, J., Fensel, D. & Traverso, P. (eds.) *Future Internet – FIS 2008.* Springer Berlin Heidelberg.

Hellbruck, H., Teubler, T. & Fischer, S. Name-Centric Service Architecture for Cyber-Physical Systems (Short Paper). IEEE 6th International Conference on Service-Oriented Computing and Applications, 16-18 December 2013 Kauai, Hawaii. 77-82.

Karageorgos, A., Mehandjiev, N. & Rapti, E. 2013. A Model for Intelligent Adaptation and Evolution of Polymorphic Services. *6th International Conference on Service Oriented Computing and Applications.* Kauai, Hawaii: IEEE.

Kosmatos, E. A., Tselikas, N. D. & Boucouvalas, A. C. 2011. Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture. *Advances in Internet of Things,* 1**,** 5-12.

Lázaro, Ó., Moyano, A., Uriarte, M., González, A., Meneu, T., Fernández-Llatas, J. C., Traver, V., Molina, B., Palau, C., López, Ó., Sánchez, E., Ros, S., Moreno, A., González, M., Palazón, J. A., Sepulcre, M., Gozálvez, J., Collantes, L. & Prieto, G. 2012. Integrated and Personalised Risk Management in the Sensing Enterprise. *In:* BANAITIENE, N. (ed.) *Risk Management - Current Issues and Challenges.*

Leidig, T. & Pedrinaci, C. 2012. Linked USDL. http://www.linked-usdl.org.

Li, F., Vogler, M., Claeßens, M. & Dustdar, S. 16-18 December 2013. Towards Automated IoT Application Deployment by a Cloud-based Approach. *IEEE 6th International Conference on Service-Oriented Computing and Applications.* Kauai, Hawaii.

LI, M.-S. The Sensing Enterprise. The Internet of Things Conference, 2011 Poznan, Poland.

Marin-Perianu, M., Meratnia, N., Havinga, P., De Souza, L. M. S., Muller, J., Spiess, P., Haller, S., Riedel, T., Decker, C. & Stromberg, G. 2007. Decentralized Enterprise Systems: a Multiplatform Wireless Sensor Network Approach. *Wireless Communications, IEEE,* 14**,** 57-66.

Martinez, M. T., Fouletier, P., Park, K. H. & Favrel, J. 2001. Virtual enterprise–organisation, evolution and control. *International journal of production economics,* 74**,** 225-238.

Ming, H., Oyama, K. & Chang, C. 2008. Human-Intention Driven Self Adaptive Software Evolvability in Distributed Service Environments. *Proc. 12th IEEE Int'l Workshop Future Trends of Distributed Computing Systems (FTDCS '08)***,** 7.

Mittermeir, R. T. 2001. Software Evolution: Let's Sharpen the Terminology Before Sharpening (out-of-scope) Tools. *Proceedings of the 4th International Workshop on Principles of Software Evolution***,** 8.

Murguzur, A., Truong, H.-L. & Dustdar, S. 2013. Multi-perspective Process Variability: A Case for Smart Green Buildings (Short Paper). *IEEE 6th International Conference on Service-Oriented Computing and Applications***,** 25-29.

Papazoglou, M. 2008. The Challenges of Service Evolution. *Advanced Information Systems Engineering***,** 15.

Santucci, G., Martinez, C. & Vlad-Câlcic, D. 2012. The Sensing Enterprise.

Shena, W., Haoa, Q., Wanga, S., Li, Y. & Ghenniwac, H. 2007. An agent-based service-oriented integration architecture for collaborative intelligent manufacturing. *Robotics and Computer-Integrated Manufacturing,* 23**,** 315-325.

Souza, L., Spiess, P., Guinard, D., Köhler, M., Karnouskos, S. & Savio, D. 2008. SOCRADES: A Web Service Based Shop Floor Integration Infrastructure. *In:* Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F. & Sarma, S. (eds.) *The Internet of Things.* Springer Berlin Heidelberg.

Thoma, M., Antonescu, A.-F., Mintsi, T. & Braun, T. 2013. Linked Services for Enabling Interoperability in the Sensing Enterprise. *In:* Sinderen, M., Oude Luttighuis, P., Folmer, E. & Bosems, S. (eds.) *Enterprise Interoperability.* Springer Berlin Heidelberg.

Thoma, M., Sperner, K. & Braun, T. Service descriptions and linked data for integrating wsns into enterprise it. Software Engineering for Sensor Network Applications (SESENA), 2012 Third International Workshop on, 2012. IEEE, 43-48.

Trentesaux, D. 2009. Distributed Control of Production Systems. *Engineering Applications of Artificial Intelligence,* 22**,** 971-978.

Wang, C., Ghenniwa, H. & Shen, W. 2008. Real Time Distributed Shop Floor Scheduling Using an Agent-based Service-oriented Architecture. *International Journal of Production Research,* 46**,** 2433-2452.

Zhang, Y., Huang, G., Qu, T. & Ho, O. 2010. Agent-Based Smart Objects Management System for Real-Time Wireless Manufacturing. *In:* Huang, G., Mak, K. L. & Maropoulos, P. (eds.) *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology.* Springer Berlin Heidelberg.