

Ontology-Based Modeling of Manufacturing and Logistics Systems for a New MES Architecture

Luca Fumagalli, Simone Pala, Marco Garetti, Elisa Negri

► **To cite this version:**

Luca Fumagalli, Simone Pala, Marco Garetti, Elisa Negri. Ontology-Based Modeling of Manufacturing and Logistics Systems for a New MES Architecture. Bernard Grabot; Bruno Vallespir; Samuel Gomes; Abdelaziz Bouras; Dimitris Kiritsis. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2014, Ajaccio, France. Springer, IFIP Advances in Information and Communication Technology, AICT-438 (Part I), pp.192-200, 2014, Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World. <10.1007/978-3-662-44739-0_24>. <hal-01388243>

HAL Id: hal-01388243

<https://hal.inria.fr/hal-01388243>

Submitted on 26 Oct 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Ontology-Based Modeling of Manufacturing and Logistics Systems for a New MES Architecture

Luca Fumagalli¹, Simone Pala¹, Marco Garetti¹, Elisa Negri¹

¹ Department of Management, Economics and Industrial Engineering, Politecnico di Milano,
Piazza Leonardo da Vinci 32, 20133 Milano.
{luca.l.fumagalli, simone.pala, marco.garetti, elisa.negri} @polimi.it

Abstract. The paper illustrates the role of modeling of shop floor to support an innovative solution for the control architecture of automated manufacturing systems. One of the main characteristics of manufacturing systems domain is, in fact, the variety of configurations that manufacturing systems can assume and this may prevent the possibility to easily adapt and reconfigure the control solution for advanced manufacturing systems. To this end, the paper presents a proposal on how to cope with this issue, coming from a collaborative project, where important European universities and companies are involved. The proposal is based on a structured modeling (i.e. ontology) of manufacturing systems. The paper proposes a practical example of the modeling, envisioning how this can be then exploited within the proposed architecture that defines a new concept of the Manufacturing Execution System of manufacturing equipment.

Keywords: Manufacturing Systems Ontology, Factory and process automation, Service Oriented Architecture, Manufacturing Execution System.

1 Introduction

Efforts of many researchers toward the generalization of methods for the design and management of manufacturing systems have been often limited by the variety of the context of their applications. On the other hand, many authors have addressed this issue in the past; nevertheless it has become available for better handling of modeling problems only in the more recent years thanks to the development of new information technology tools and languages. For instance, the Manufacturing Execution Systems Modeling Language (MES-ML) integrating all necessary views important for Manufacturing Execution Systems (MES) and pointing out their interdependencies has been developed by Witsch and Vogel-Heuser [1].

This paper presents a part of a research developed in the scope of an ongoing European funded project on this topic, named eScop (*Embedded systems Service-based Control for Open manufacturing and Process automation*). The central concept of eScop is to combine the power of embedded systems with an ontology-driven service-based architecture for realizing a fully open automated manufacturing environment. The innovation of the solution proposed by eScop is the fact that it merges the power of ontology knowledge and SOA control approaches; this allows the control to be automatically configured by the ontology to a specific manufacturing

system. Thus, one result is the substitution of the traditional control model based on hierarchical hardware architecture, with a single level population of embedded systems plus a free series of pure software control levels. The deployment of such approach to manufacturing has been named Open Knowledge-Driven Manufacturing Execution System (OKD-MES). The kernel of the eScop platform will be presented in the remainder of this paper, showing the novelty that it introduces.

This document is structured in the following way: section 2 details the benefits for the overall approach proposed by eScop project introducing the new vision provided by the project research activity and the Representation Layer, namely the ontology modeling concept that is included in the proposed architecture. Section 3 explains how a demonstrator has been deployed for modeling, thanks to the eScop approach. Eventually, section 4 provides the conclusions and envisions future challenges in this field of research.

2 New MES Architecture for Manufacturing and Logistics Systems

In factory automation the evolutionary path of real-time control architecture is directed to overcome the rigidity of current traditional solutions. In fact, the current architecture of MES real-time control is based on fragmented and scattered information from the field, joined with a rigid hardware structure, hence being suitable and efficient only in stable contexts. In the remainder of the document this situation is called “AS-IS scenario” in contrast with the “TO-BE scenario” envisioned by the research. In the AS-IS scenario, in order to manage a production system, the supervisor generally requires an external input (i.e. information about product features and order portfolio), while business logic, scheduling algorithms and configuration data (i.e. information on process, transportation and storage equipment) are stored in the system database. Part of the required information is directly programmed in the Devices Control Units (DCUs) managing physical equipment. Depending on the hardware architectural solution, DCUs can be made of Manufacturing Control Units (MCUs), Computer Numeric Controls (CNCs) or Programmable Logic Controllers (PLCs). In the TO-BE scenario, in order to overcome the rigidity and limits of the traditional control architectures, the development of communication technologies is exploited to make it possible and affordable to integrate heterogeneous devices into a large network. To this end, specific attention is paid to web services protocols [2]. Web Services are “self-contained, self-describing, modular applications that can be published, located and invoked across the web” [3]. The scientific community has started to prove the validity of the use of web services and SOA for manufacturing automation and monitoring through industrial test-beds (e.g. [4,5]).

Due to the advent of SOA and Web service technologies, the task of monitoring and controlling locally distributed heterogeneous devices has become an effortless task. However, knowledge is required about the physical and logical structure of the application domain, if the collected data must be transformed into useful information. In the AS-IS scenario, this knowledge is encapsulated into the machine control systems, according to the specific situation. Hence, due to its static implementation,

this configuration does not allow easy re-configurability and also interoperability of components by different manufacturers [6]. To overcome this limit, one possible attempt is to abstract this knowledge and to store it, thus allowing quick and effortless reconfigurations [7]. To this end, ontologies are claimed as important to capture the conceptual structure of a domain [8,9].

Therefore, ontology seems to be a good solution for fulfilling the need of a dynamic knowledge base of the production system to be controlled. Nevertheless, if ontology is used just to describe the static aspect of a manufacturing system, most of its power is wasted and not used. The innovation about using ontology in factory automation is, in fact, to consider it within the tools in charge for the control of the manufacturing system. In this case, ontology highlights the unique feature to be queried and updated by web services with languages and tools belonging to the world of the semantic web [10]. Thus, ontology is definitely better than other databases to represent and manage web services, given that it can interact with them directly.

This research aims at showing the innovation in the ontology use for the control of the shop floor, with the ultimate aim of setting up the TO-BE scenario. Indeed, potentiality of knowledge-based semantic web services in factory automation has been already postulated by Lastra and Delamer [11]; the shift from the traditional control system architecture to the schema based on ontology and web services enables the transition towards a knowledge-based production control system (as previously postulated by Lobov et al. [12]), in which the key is the semantic coordination of standardized production components. To this end, Long [13] provided an overview on how to improve MESes, dealing with a generic ontology tier (level), even if not specifically detailed with manufacturing knowledge and Garetti et al. [14] further analyzed the integration of ontology and web services.

The platform proposed by eScop project is fully based on this research background. The platform architecture is represented in Figure 1 by a view of its kernel module, controlling the physical system and connecting it to the upper applications. The kernel is made by 3 main layers (Physical Layer, Representation Layer and Service Orchestration Layer), plus an Interface Layer to the applications part. Physical Layer corresponds to the hardware components of the production system, where each one of them is insulated by an embedded system providing intelligent device control to the physical equipment and acting as a SOA-based Remote Terminal Unit (RTU). Web services do not substitute RTUs (PLCs, CNCs and so on), but they are a layer above the machine-level controllers whose functionalities are seen as encapsulated, thus achieving a great degree of flexibility in various dimensions.

The main element for Representation Layer consists of the instance of the production system under control, taken from a reference ontology of production systems (i.e. the so called eScop Manufacturing System Ontology - MSO). This production system modeling allows to formally instantiate any specific production system, thus being able to drive the set of SOA based orchestration tools which act as the Supervisory Control System. eScop MSO is based on the Politecnico di Milano - Production System Ontology, the P-PSO, which represented manufacturing systems as outcome of a large research activity carried out at Politecnico di Milano. The research started back in the 90's, while recently the P-PSO has taken shape aiming at

a complete representation of a manufacturing reality. Details are available in Garetti and Fumagalli [15].

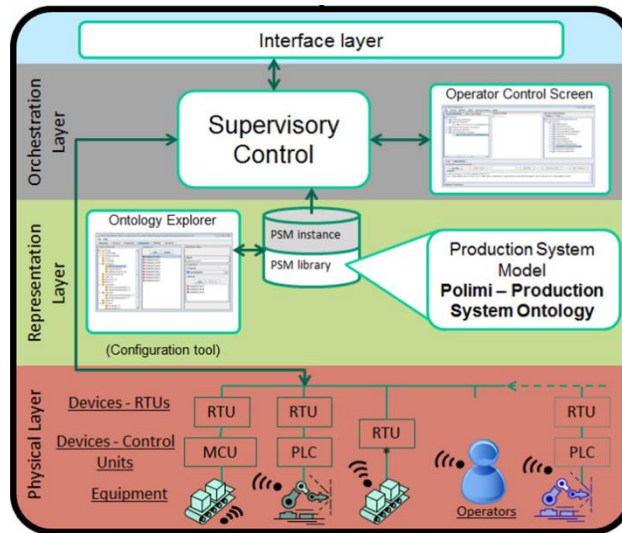


Figure 1. Detail of the eScop Kernel

The eScop MSO can be considered as a meta-model of the manufacturing systems domain, since it specifies the entities (building blocks) it is made of, their attributes and their relations, thus defining a standardized data format for its description. The manufacturing system modeling in eScop MSO addresses three different aspects separately, i.e., the physical aspect, the process aspect and the control aspect.

eScop MSO ontology can directly interact with web services through SPARQL queries that allow to retrieve or update information stored within the ontology, activities needed for the successful run-time control of the system. The language in which eScop MSO is written and which allows querying and the related reasoning activities is OWL, as it is clearly explained by Pan [16].

Orchestration layer acts as the Supervisory Control System of the manufacturing equipments and it is made of two main components: i) the set of Orchestration Tools and Services which interact, via the Representation Layer models, with the shop floor (i.e. Level 1 devices) and ii) the Production Scheduler which feeds the Orchestration Tools and Services with work orders coming from the factory Order Entry System. The ontology is exposed as a service itself through the Ontology Service on the Orchestration platform.

The resulting architecture will be flexible, reliable and scalable and tailored for control (and even monitoring) in industrial environments.

3 Application domain: picking system case

When the control system for a given shop floor system has to be established, the eScop MSO instance of the system is created by the operator through the Ontology

Manager starting from the ontology model. The instance knowledge base is connected to Web Services Orchestration that, then, can send commands to the production layer through production web services controlling the physical system. This way, flexible command capability is achieved through the configuration knowledge content of the ontology.

Within the eScop project a test case is a picking system, composed by four main subsystems, as it is shown in Figure 2, namely: a carousel ring (indicated by “A” letter), a input/output warehouse station (B), a picking buffer station (C) and a gravity conveyor (D).

The purpose of the ring carousel is to create a buffer of the most used pallets to speed up picking operation and avoid time-consuming movement in/from the warehouse. For this reason, a high level control system is needed that manages the in/out coming pallets.

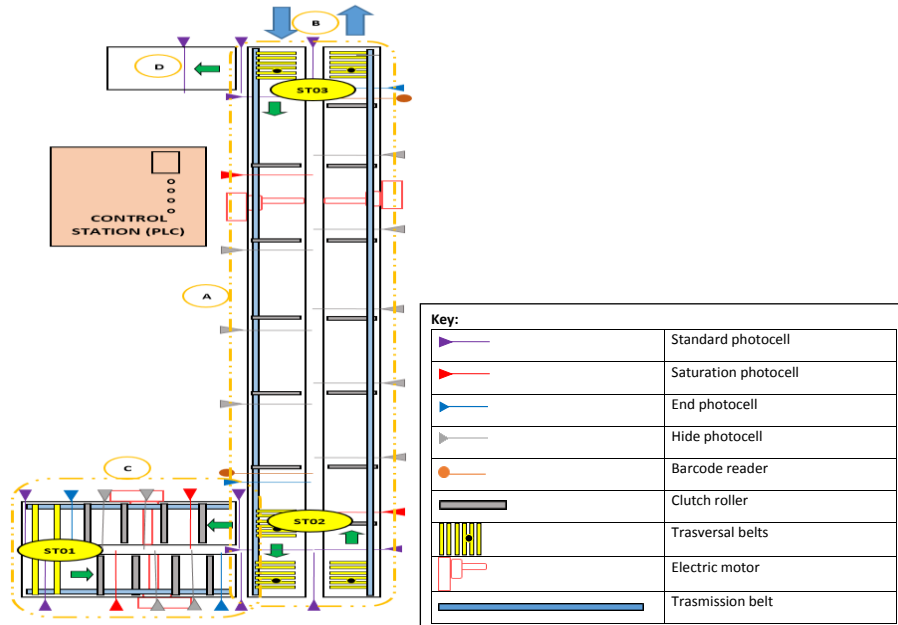


Figure 2. Layout of the logistic system

The actual control system, the so called “AS-IS scenario”, is based on a rigid and hierarchical structure on a three level control schema (according to the IEC 62264 [17]): level 0, 1 and 2. The level 0 is mainly composed by electro-mechanical devices and controllers that can autonomously take decisions (e.g. stop a pallet in a compartment on ring carousel because the next compartment is occupied by another pallet), level 0 can interact only with level 1. Level 1 is composed by the PLC that acts as the control station, its role is to coordinate the actions for correct sorting of pallets; level 1 interacts with level 0 by collecting information and coordinating the action of level 0 actuators, the “decision points” ST01, ST02 and ST03 has been identified as the main decision points of level 1. Level 1 also interacts with the supervisory level, level 2, which provides the orders of the goods to be picked by operators and coordinates the action with other parts of the system to allow the correct

working (e.g. it requests to warehouse control system to deliver specified pallets to the ring carousel).

The behavior of the system in the “TO-BE scenario” will be the same as in the “AS-IS one”; this is an aspect that has to be taken into account during the model generation of the logistic line.

The complete model, based on eScop MSO ontology and drawn in UML class diagram notation, is shown in Figure 3. Each object that composes the logistic system, described above and shown in Figure 2, belongs to one class and can be linked by association, by inheritance or by aggregation to another class. Hereafter the main classes that compose the model are presented, explaining how they are logically linked together and providing comments to understand the rationale behind the presented model.

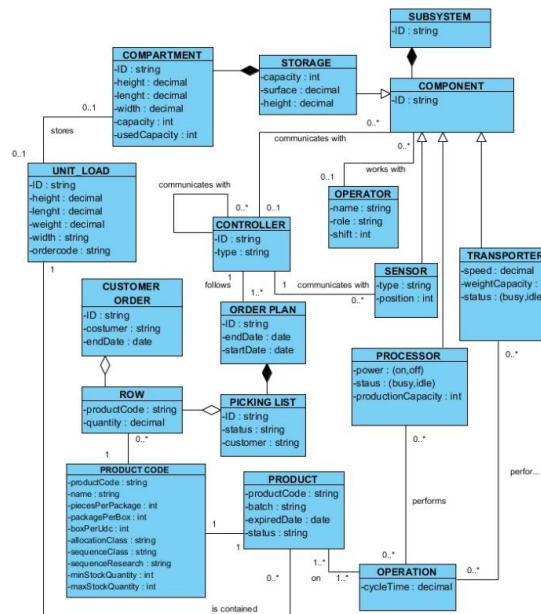


Figure 3. Modeling of the logistic system through eScop MSO

Physical Aspect: The *subsystem* class allows creating customized building blocks that can be used everywhere in the object model. In this paper the class subsystem contains one item, that is the whole logistic line.

The *component* class: represents generic physical items of a production system. It can be further decomposed into subclasses according to the nature of the object: storage, transporter, processor, operator and sensor.

- The *storage* class: comprises all the entities that perform a storage function, i.e. keeping material for later use into the production process.
- The *transporter* class: groups the entities that physically move a product in the plant.
- The *processor* class: is composed by the entities used to perform a production process function, i.e. transforming the material.

- The *sensor* class: groups the items used to get physical information from the field, such as: position, status, temperature, etc.

An operator belonging to *operator* class performs an action on/with a component.

Process aspect: An *operation* can be performed by a *transporter* or a *processor* on a *product*, the latter is identified through a *ProductCode*. Every product item is moved in the system thanks to a *unit load* that contains it and that is the pallet in this industrial case. Each *unit load* instance is stored in a *compartment* instance, that represents the single spaces for *storage*. To this end it is easy to see that the ring carousel (indicated by letter “A” in Figure 2) and the picking buffer station (letter “C”) are two different storage items, each one composed by the aggregation of several compartment instances (namely the space comprised between two clutch rollers). The transporter moves the product instances, contained in a unit load, from one compartment to another one.

Control aspect: A *controller* item interacts with one or more sensors and/or other controllers to launch an *order plan* built as a composition of *picking list* instances. The picking list is an aggregation of *row* of items, each row representing a specified quantity of the same goods to be sent to the customer, so an aggregation of *row* items build a *customer order*. The controller item has to be intended not only as a PLC, but as every item capable to take decisions and to influence the status of the system and acting to fulfill the order plan instances.

Once the control system interacts with information structured as presented above, it is possible to control the entire shop floor flexibly of the described logistics system for picking. The control architecture, in fact, is flexible to any variation of the system (e.g. the change in the number of compartments, namely a change in the number of instances of compartment).

4 Conclusions

The present paper has introduced the general concept of the eScop research project, focusing on the Representation Layer of the eScop platform kernel. The theoretical use of modeling of a system for control logic has been addressed by a specific industrial case.

The case refers to an industrial domain, where automation is particularly important. In fact, automated logistic systems are systems that allow efficiency in many different companies. Moreover, they represent a typical case where the problem of flexibility for new configuration is crucial.

In order to cope with the problem, it has been demonstrated how the proposed approach is useful and can provide interesting results.

This paper aimed at introducing the possibility of a new vision of MES, providing a tangible example related to the modeling of the physical system. This fosters the dissemination for a larger research activity that is ongoing within eScop project and will provide new scientific results in the near future.

Acknowledgements. The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n° 332946 and from the

Italian Ministry of Education, Universities and Research (MIUR), correspondent to the project shortly entitled *eScop, Embedded systems for Service-based control of Open Manufacturing and Process Automation*. The authors would like also to personally thank all the European partners of the project and, in particular, INCAS S.p.a that provided the material and support for the development of the presented case.

References

1. Witsch, M., Vogel-Heuser, B.: Towards a formal specification framework for manufacturing execution systems. *IEEE Trans. Ind. Informat.*, 8(2), 311-320. (2012).
2. Web Services Architecture - World Wide Web Consortium, <http://www.w3.org/TR/ws-arch>
3. Tidwell, D. Web services-the web's next revolution. IBM developer Works. <http://www.ibm.com/developerWorks/>. (2000).
4. Macchi, M., Fumagalli, L., Garetti, M., Tavola, G., CheccoZZo, R., Rusina, F., Vidales Ramos, A., Jokinen, J., Popescu, C., Lastra, J. L. M., Karhumaki, O., Vainio, M. A.: Use case analysis method for the implementation of service-oriented solutions for monitoring and diagnostics. *Proceedings of the 24th COMADEM International*, 2011. - ISBN: 0-9451307-2-3.
5. eSONIA, <http://www.esonia.eu>.
6. Wong, A. K. Y., Ray, P., Parameswaran, N., Strassner, J.: Ontology mapping for the interoperability problem in network management. *10. IEEE J. Sel. Areas Commun*, Vol.23, Oct 2005, pp. 2058-2068.
7. Cai, M., Zhang, W.Y., Zhang, K., ManuHub: A Semantic Web System for ontology-Based Service Management in Distributed Manufacturing Environments, *IEEE Trans. Syst., Man, Cybern., Syst, Part A: Systems and Humans*, 41 (3), 574-582. (2011).
8. Brachman, R.J., Levesque, H.J., *Knowledge representation and reasoning*. Elsevier. (2003).
9. Chandrasekaran, R., Josephson, J.R., Benjamins, V.R., What are ontologies, and why do we need them? [ed.] *IEEE Intell. Syst.* (2009).
10. Hunter, J. Enhancing the semantic interoperability of multimedia through a core ontology. *IEEE Trans. Circuits Syst. Video Technol.*, 13, 49 - 58. (2003).
11. Lastra, J. L. M., Delamer, M., Semantic web services in factory automation: fundamental insights and research roadmap. *IEEE Trans. Ind. Informat.*, February 2006. - 1 : Vol. 2. - p. 1-11. (2006).
12. Lobov, A., Ubis Lopez, F., Villasenor Herrera, V., Puttonen, J., Lastra, J. L. M., *Semantic Web Services Framework for Manufacturing Industries*. *IEEE International Conference on Robotics and Biomimetics*. - Bangkok, Thailand - p. 2104-2108.
13. Long, W., *Research on Development Method of MES based on Component and Driven by Ontology*, JSW, Vol. 5, No. 11, November 2010.
14. Garetti, M., Fumagalli, L., Lobov, A., Lastra, J.L.M., *Open automation of manufacturing systems through integration of ontology and web services*. In *proceedings of 7th IFAC Conference on Manufacturing Modelling, Management, and Control International Federation of Automatic Control*. Saint Petersburg, Russia. June 19-21, 2013.
15. Garetti, M., Fumagalli, L., *P-PSO ontology for manufacturing systems*, 14th IFAC Symposium on Information Control Problems in Manufacturing. Bucharest, Romania - May 23-25, 2012.
16. Pan, J. Z. A flexible ontology reasoning architecture for the Semantic Web, *IEEE transactions on knowledge and data engineering*, vol. 19 no. 2 February 2007.
17. ISO/IEC 19501:2005, *Information technology—Open Distributed Processing—Unified Modeling Language (UML)*. IEC 62264:2013, *Enterprise-control system integration*.