HAL
archives-ouvertes.fr

Architecture for Open, Knowledge-Driven Manufacturing Execution System

Sergii Iarovyi, Xiangbin Xu, Andrei Lobov, Jose Martinez Lastra, Stanislaw Strzelczak

▶ **To cite this version:**

Sergii Iarovyi, Xiangbin Xu, Andrei Lobov, Jose Martinez Lastra, Stanislaw Strzelczak. Architecture for Open, Knowledge-Driven Manufacturing Execution System. Shigeki Umeda; Masaru Nakano; Hajime Mizuyama; Hironori Hibino; Dimitris Kiritsis; Gregor von Cieminski. IFIP International Conference on Advances in Production Management Systems (APMS), Sep 2015, Tokyo, Japan. IFIP Advances in Information and Communication Technology, AICT-460 (Part II), pp.519-527, 2015, Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth. <10.1007/978-3-319-22759-7_60>. <hal-01431141>

HAL Id: hal-01431141
https://hal.inria.fr/hal-01431141

Submitted on 10 Jan 2017

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.

Distributed under a Creative Commons Attribution| 4.0 International License

Architecture for Open, Knowledge-driven Manufacturing Execution System

Sergii Iarovyi, Xiangbin Xu, Andrei Lobov, Jose L. Martinez Lastra
Tampere University of Technology, Finland
{sergii.iarovyi, xiangbin.xu, andrei.lobov, jose.lastra}@tut.fi

Stanisław Strzelczak
Warsaw University of Technology, Poland
sstrzelc@wip.pw.edu.pl

Abstract – Manufacturing Execution Systems (MES) are a bridge between the enterprise solutions and factory shop floors. MES allows the performance of contemporary factories by closer integration of the needs of enterprise stakeholders with the actual manufacturing hardware and software components. Considering different nature of the factories the application of MES has significant difference across industry. Such constrain requires a sophisticated solutions to contain the generality of application. Most of currently available solutions are proprietary and tightly coupled to the ecosystem of devices and enterprise resource planning systems. In current paper the architecture for the open, knowledge-driven MES is discussed. Authors argue that such MES will provide smart extensibility based on base-of-breed approach and hence will improve the quality of MES application, while reducing the introduction costs and system downtime due to reconfiguration.

Keywords: Manufacturing Execution System, Knowledge-driven Approach, Open architecture, Web Services, Factory Automation, Smart Factory

1 Introduction

Contemporary factories enjoy significant benefits from the application of Manufacturing Execution Systems (MES). MES provides a link between the level of the enterprise planning and the level of the production shop-floors. Such connection introduces faster feedback between layers and new functionalities which without MES are to be done manually. The studies confirm the direct correlation between the employment of MES in enterprises and enterprise performance. For example, in the research made by Manufacturing Enterprise Solutions Association International (MESA) industry group and Industry Directions Inc. the MES-using companies have had the Key Performance Indicators (KPI) generally better than the companies without such systems (Manufacturing Execution Systems - Accenture, 2010).

adfa, p. 1, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The MES solutions generally are provided by the companies already represented either on higher or lower levels of factory information systems (e.g. Siemens, SAP). This leads to orientation of the MES towards particular factory ecosystems and leads to issues of interoperability that restricts the possibilities for the choice. As it would be discussed further, the MES solutions should provide a wide specter of functions.

Different MES solutions naturally have different approaches to implement these functions, which vary in applicability in different factories. The particular case is possible where the company exploiting MES solution is satisfied by some of the functions but does not require others. Such cases lead to possibility to purchase only some parts of MES solutions. Nevertheless the possibility to get different MES functions from different providers are limited, and commonly lead to redundancy in the information system and customized integration which introduces additional costs to the MES solutions.

Furthermore, contemporary MES solutions are designed to satisfy the mass-production paradigm. This makes the existing solutions suitable for the big enterprises operating with large quantities of similar goods to be produced, but limits their application to the smaller ones which largely have more diversified portfolio. Last but not least, the MES solutions currently available on the market has significant cost for software itself, for the service of introduction of such solution to companies and for adjustment of business processes accordingly. Concluding, currently available MES solutions have some limitations for their applicability due to their cost, complexity, collateral changes and limitations casted on the enterprise.

In order to address the named constrains of MES solutions this article suggests a new architecture for it. New architecture developed in eScop Project[1] is based on the synergy of *embedded devices*, *service-orientation* and *knowledge-driven approach*. Furthermore, the concept of the architecture suggests the open nature of the solution. The MES following the suggested architecture should reduce the cost of introduction and maintenance of the solution to the enterprise, make it more applicable for mass-customization production and enhance the capability to handle changes in the enterprise, hence to make it cost efficient so it can enter the Small and Medium Enterprises (SME) level.

Embedded devices in suggested architecture are providing more dynamicity on the factory shop-floor keeping it loosely coupled to the MES solution. The factory shop-floor using proposed concepts would not require custom or tight integration with the MES reducing the cost of changes in it. Service-orientation serves the purpose to facilitate the loosely-coupled integration of the MES components as well as to integrate the MES in factory ecosystem. Knowledge-driven approach is employed to maintain the knowledge about system correct, organized, and accessible. Such knowledge about the system allows to facilitate the reconfiguration with little or no human interactions, and as result make the system capable to embrace the changes in factory ecosystem such as introduction of new production recipes, equipment or other components.

---

[1] http://www.escop-project.eu/

The following sections will present background followed by description of Open Knowledge-Driven Manufacturing Execution System (OKD-MES) concept. Section 2 defined concepts required for background. In the third section the architecture and concept for new MES will be described. Then the fourth section outlines conclusions and future work.

2 Background

The core technologies and concepts applied for the proposed architecture are discussed in this section.

2.1 Manufacturing Execution Systems

M. McClellan (1997) provides following definition for MES: *"A manufacturing execution system (MES) is an online integrated computerized system that is the accumulation of methods and tools to accomplish production"*. The author argues on importance of **online** and **integrated** nature of the system. In this definition online means connection to upper (business planning and logistics) and lower (factory shop-floor control and monitoring) level systems, providing vertical integration of MES solution in factory automation hierarchy. Integrated nature of MES represents the horizontal integration of the MES solution on the level of Manufacturing Operation Management (MOM). The importance of both types of integration is supported by (J. Kletti 2007).

MES solution should provide a certain set of the functions to the enterprise which are generally located between Enterprise Resource Planning (ERP) systems and factory shop-floor. Dissimilar organizations have tried to provide a classification for such functions. The efforts of Manufacturing Enterprise Solutions Association (MESA), Verein Deutsche Ingenieure (VDI) and ISA-95 are generally coherent. The functions defined by MESA are following: *operations / detail scheduling, resource allocation and status, document control, dispatching production units, performance analysis, labor management, maintenance management, process management, quality management, data collection and acquisition,* and *product tracking and genealogy*.

Besides the need for implementation of the aforementioned list of functions in industry there exists a demand for the manner of MES implementation. Among such requirements in AMR researches from the end of 00's by R. Martin argues that future MES should use the web-services and have more flexible data model. MES users naturally desire to use most efficient tools for MOMs disregard of the solution owner. The future MES should provide modularization of the services to satisfy this need, argues F.K. Johnson in (2010). The modular MES may use the concepts of core and complimentary components. It is important to keep core interoperable and transparent for to maintain capability to develop the complimentary components. In same paper author states the possible role of cloud computing to optimize its costs.

2.2 Service orientation and Web Services

The one of most prominent design paradigm for interoperable components is service-oriented approach (SOA). The core principles service orientation are presented in (Erl, 2007) by T.Erl and are argued to achieve the goals such as increased ROI, increased organizational agility and reduced IT burden which are the benefits per se.

SOA is being commonly implemented in form of Web Services (WS). WS are using the web as a media for service interactions. There exist different standards to implement the WS. Among the most employed ones are RESTful services defined by Fielding in his thesis (Fielding, 2000), WS-* specification by OASIS ("OASIS Devices Profile for Web Services (DPWS)," n.d.). RESTful web services are dominant in consumer internet. WS-* still maintain a big share in the enterprise integration solutions, but for some cases may be replaced with RESTful ones to reduce complexity. Web Services are advancing towards factory shop floor devices employing expanded capabilities of **embedded systems**.

2.3 Embedded devices

Embedded devices in the factory shop floor may support the web service implementations. Among the examples there is a commercially available INICO S1000[2]. Besides the development of industrial computers and credit-card sized computers as Raspberry Pi[3] are providing a solid base for other WS-enabled devices. In eScop Project the Remote Terminal Unit (RTU) is being developed which should support RESTful web services in factory shop floor as well as execute control logic.

2.4 Knowledge Representation

(Davis, Shrobe, & Szolovits, 1993) is providing the wide and comprehensive description of Knowledge Representation (KR). Particular implementations of KR are developed in form of linked data, and in case of higher semantic complexity – in the form of ontologies. The realization of linked data and ontologies is based on certain XML based standards such as RDF and OWL. RDF representation is built of subject-predicate-object triples. More complicated formalization leads to wider reasoning capabilities in cost of limitations of representation. Each valid OWL representation is also valid RDF, but not vice versa. SPARQL Protocol and RDF Query Language (SPARQL) is a dedicated technology for manipulating the data in RDF based languages.

The possibility of RDF to OWL compatibility is an important benefit of this language family, as level of formalization may be adjusted without a need to change technology stack significantly. As well complexity of the data representation may be leveraged by creation of domain ontologies as it is proposed in chapter: Ontology-based modeling of manufacturing and logistics systems of this book. Overall

---

[2] http://www.inicotech.com/

[3] https://www.raspberrypi.org/

Knowledge Representation together with Web Services concepts are successfully used for manufacturing domain. For example B. Ramis et al. in (2014) are offering an approach to use such combination for integration of an industrial automation system. In (Puttonen, Lobov, & Lastra, 2013) authors argue about the synergy of KR and WS for representation of a dynamic automation system and even composition of factory automation processes.

3 Architecture

The goal of development of OKD-MES is to provide Open solution, which employs Knowledge Driven approach to implement MES. Considering first term "Open" in this work it has both meaning of being open source solution, open to be studied, modified and distributed. Also it means open from perspective of Open Architecture, e.g. the OKD-MES should be open to addition or replacement of modules developed by community. Second part of definition is "Knowledge Driven". Open Knowledge-Driven MES is being developed to be capable to understand the data within the system and hence to be capable to infer additional information based on reasoning mechanisms within OKD-MES.

- ***Open***. Beyond openness from point of view of rights, OKD-MES architecture should be open for modules extending its functionality. For this part the concept of Core MES functionalities and complimentary modules discussed in (Johnson, 2010) may be employed.
- ***Modular***. Considering relatively high complexity of MES solutions, the open and adaptable solution driven by community should be developed as a set of independent interoperable component to leverage lack of coordination.
- ***Knowledge-driven***. The complex, service-oriented and modular system which should dynamically adapt to new components and services should facilitate the knowledge about the system.

Architecture of OKD-MES core, presented on Fig. 1, includes 5 main components: Physical Layer (PHL), Representation Layer (RPL), Orchestration Layer (ORL), Visualization Agent (VIS), and Interface Layer (INT). Integration of OKD-MES core with shop floor equipment is provided employing functionality of PHL. Complimentary functions of OKD-MES are expected to be implemented as services and/or clients of OKD-MES core functionalities. Access to such core functionalities is provided via Interface Layer in order to provide required level of access to core. Finally services implementing core or complimentary functions of MES are able to employ a Visualization agent to provide user interface for the system.

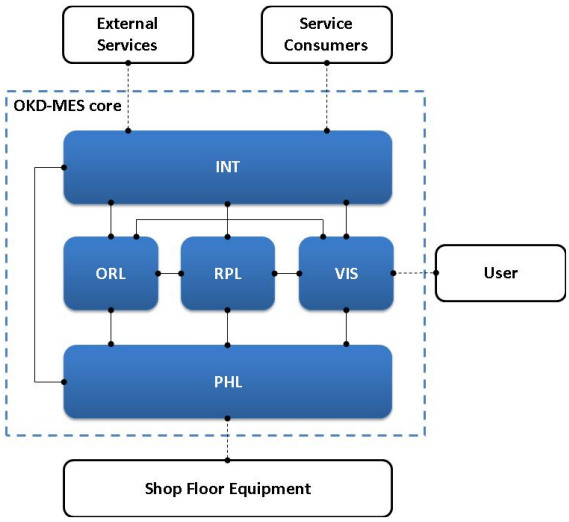

Fig. 1. OKD-MES core architecture

3.1 Interactions between layers

As it was already mentioned, interface layer should be capable to define level of access of external service or user and enable this access to underlying services of OKD-MES core. It is also possible to use adapters in interface layer to connect the MES to providers and consumers which are not implementing RESTful WS or even are not services at all.

Physical layer should be capable to provide the information required to create representation of shop floor devices in the Knowledge Base of RPL. This information must include shop floor device description from manufacturing point of view as well as from service perspective. Directly or indirectly the information about equipment status is to be exposed by PHL to RPL.

Orchestration Layer should be capable to execute service on PHL in order to orchestrate service compositions. On other hand it should be capable to read some data from shop floor equipment if it is required for process definition. In some cases it may be required to subscribe for notifications from PHL in order to execute the service composition efficiently.

The definition of the executable processes in ORL may and should be abstracted from real service implementations and be concentrated on functionalities for the reasons of robustness and flexibility of the approach. Hence the mapping of these abstract services to their real implementations should take place. This on the mapping information related to service invocation should be provided. All required information connecting the syntactic and semantic description of the services is persisted in RPL. This mapping functionality is one of the main interactions between ORL and RPL. As

well some information about process execution and status of the system is an important part of interaction of these two layers.

Interactions between Visualization Agent and RPL are required to generate the visualization screens on demand. Information about configuration of the screen including data points for subscription in other layers should be provided by RPL.

Interactions between Visualization Agent and PHL are to be mainly of subscription format. E.g. some information explicitly available in the device may be required. All interactions directed from visualization as input should be developed in a form of executions of certain services in the system.

3.2 Physical Layer

Architecture of physical layer is based on the concept that device functionality should be encapsulated in a service. The data from device should be accessible employing WS interface, the operations on devices may be invoked as services. For obvious reasons PHL should provide capabilities to discover and describe the service in order to enable loose coupled integration in overall system. Moreover some basic services as subscription should be implemented on Physical layer to enable event oriented behavior of the system. A more efficient and flexible solution is to provide an access to a runtime core. In the control programs running in PHL device runtime core events may be defined. The data consumers may subscribe to such events and be connected to more abstract data than one available from a simple I/O module. As well possibility to invoke the programs in PHL device directly using some data from service call is also more flexible approach.

The architecture of PHL developed for OKD-MES includes three main modules: Simple I/O module, Runtime Core and WS Toolkit. WS Toolkit enables web services functionality and may be connected both to Runtime Core and I/O Module. Besides already named service oriented functionality WS Toolkit may provide an interface for configuration of PHL. I/O Module is being employed as a bridge to real inputs and outputs of a device. Runtime Core is an architectural block on which the control logic is being executed. In case if PHL device is being employed as a bridge to some legacy device the Runtime Core may virtually be omitted.

3.3 Representation Layer Architecture

Architecture for Representation Layer is developed based on the services it will provide for other layers. Most of the services of RPL will have interaction with MSO such as CRUD (create, read, update and delete) operations on the Ontology model. Depending on the service consumer, the services can be separated into four modules in order to facilitate the development and maintenance, namely, Device Registration Module, Visualization Provider Module, Service Handler Module and Manager Module. These modules all need some internal functions in order to interact with MSO which is stored in an Ontology database. The components in the architecture are explained as following:

Device Registration Module: A service module to register or unregister eScop devices in the MSO through device catalogue. It enables access to device existence and their capabilities in the network. Mainly works with Physical domain in MSO. Visualization Provider Module: A service module to retrieve, update the visualization information, for example GUI components and their properties, in MSO according to the query from Visualization Agent. Mainly interacts with Visualization, Technological and Physical domains of MSO.

Service Handler Module: A service module to communicate with Service Composer, Orchestrator Services and Production Manager. The module should expose services to provide information requested from the Service composer, receive update requests from Orchestration service and receive update requests from Production Manager. The module must handle description of the services, configuration and status of the system, SPARQL over HTTP could be used for flexibility of the approach, but it is worth to mention it is necessary to introduce authorization and verification mechanism for SPARQL over HTTP in order to keep data integrity of the MSO. Manager Module: It provides configuration interface, model browser/editor and CRUD operation on MSO for IT staff or authorized users. SPARQL Query Factory: Basic common function to generate SPARQL queries. Ontology Connector: Basic Common function to execute SPARQL queries and return the query result.

3.4 Orchestration Layer Architecture

Architecture for Orchestration Layer is developed considering to provide control over system. Service Orchestrator has to enable execution of service compositions and providing information about the execution for other parts of the system. Orchestration of service compositions means that Service Orchestrator should include orchestration engine as well as other components required for interactions with other systems.

3.5 Visualization Layer Architecture

Visualization Layer has to provide a visualization agent capable to generate visualization screens based on a screen descriptions persisted in RPL. Moreover it is considered that employment of a browser as a visualization client for reasons of ubiquity in modern world.

Considering a need to provide visualization for dynamically composed screens, it is required to have a repository of symbols to which visualization descriptions may be linked. Dynamic Composition Module is to be using this symbol repository in order to generate visualization descriptions which can be transformed to browser readable format by visualization agent. Additionally Dynamic Composition Module of VIS has to create required subscriptions to data in the other parts of OKD-MES in order to represent in on visualization screen. Finally Visualization Agent is a part of architecture which has to transform the visualization screen description to a web page of some form, which can be displayed in browser being connected to a real data in the system.

4 Conclusion and Future Work

The general considerations for the architecture to be used in implementation of new Open Knowledge-Driven Manufacturing Execution System were outlined. The concepts can also be validated using online tools and simulator[4]. In order to increase the chances for the adoption of the proposed OKD-MES approach, tools and models have to be developed in a joint, collaborative manner between the experts. Online, web-based tools requiring no installation of dedicated software except a web browser make it possible for wide group of people with different skills to work together on the same project. This article provided a concise review of the main elements in OKD-MES architecture and the description of expected benefits. In order to achieve the benefits, many of World Wide Web Consortium[5] (W3C) standards are employed to allow communications at all the levels of the enterprise and problem domain modeling and dynamic use of the models at the system run-time.

Future work on the topic will be dedicated to describe in more details the layers of OKD-MES. Furthermore the improvements to architecture and its modules will be revealed in the course of integrating and extending OKD-MES core functions with the modules dedicated for certain MES functions in eScop architecture. The feedback from the first industrial pilots adopting the approach may also require further improvements in different layers of OKD-MES.

5 Acknowledgement

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement n° 332946 and from the Finnish Funding Agency for Technology and Innovation (TEKES), correspondent to the project shortly entitled eScop, Embedded systems for service-based control of open manufacturing and process automation.

References

1. Davis, R., Shrobe, H., & Szolovits, P. (1993). What Is a Knowledge Representation? *AI Magazine*, *14*(1), 17.
2. Erl, T. (2007). *SOA: Principles of Service Design* (1st edition). Upper Saddle River, NJ: Prentice Hall.
3. Fielding, R. T. (2000). *Architectural styles and the design of network-based software architectures*. University of California, Irvine.
4. Johnson, F. K. (2010). Future of Manufacturing Execution Systems: The Brave New Modular World of Manufacturing Intelligence. *Review of Management*, 4.
5. Kalibatiene, D., & Vasilecas, O. (2011). Survey on Ontology Languages. In J. Grabis & M. Kirikova (Eds.), *Perspectives in Business Informatics Research* (pp. 124–141). Spring-

[4] http://www.escop-project.eu/tools/
[5] http://www.w3.org/

er Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-24511-4_10
6. Kletti, J. (2007). *Manufacturing Execution Systems (MES)*. Berlin; London: Springer. Retrieved from http://www.books24x7.com/marc.asp?bookid=30973
7. *Manufacturing Execution Systems - Accenture*. (2010). Retrieved from http://www.accenture.com/us-en/Pages/insight-achieving-high-performance-manufacturing-execution-systems-full.aspx
8. McClellan, M. (1997). *Applying Manufacturing Execution Systems*. CRC Press.
9. OASIS Devices Profile for Web Services (DPWS). (n.d.). Retrieved March 16, 2015, from http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01
10. Puttonen, J., Lobov, A., & Martinez Lastra, J. L. (2013). Semantics-Based Composition of Factory Automation Processes Encapsulated by Web Services. *IEEE Transactions on Industrial Informatics*, *9*(4), 2349–2359. http://doi.org/10.1109/TII.2012.2220554
11. Ramis, B., Gonzalez, L., Iarovyi, S., Lobov, A., Martinez Lastra, J. L., Vyatkin, V., & Dai, W. (2014). Knowledge-based web service integration for industrial automation. In *2014 12th IEEE International Conference on Industrial Informatics (INDIN)* (pp. 733–739). http://doi.org/10.1109/INDIN.2014.6945604