Cyber–Physical Systems for Open-Knowledge-Driven Manufacturing Execution Systems

This paper describes and illustrates an approach for designing open-knowledge-driven manufacturing execution systems on top of CPS that controls robot workstations and conveyor-based transportation system of a pallet-based production system.

By Sergii Iarovyi, Wael M. Mohammed, Andrei Lobov, Borja Ramis Ferrer, and
Jose L. Martinez Lastra

ABSTRACT | Manufacturing execution systems play an important role of bridging high-level enterprise functions and production or manufacturing operations. The embedded systems are usually in charge of controlling execution of the operations. Modern embedded systems have become capable of simultaneous and deterministic execution of control algorithms and IP-based communication, making it possible to create complex cyber–physical systems (CPSs), where the computational and communication resources of a device can be used directly for various control, supervisory, or monitoring functions. The complexity for defining open-knowledge-driven manufacturing execution system (OKD-MES) is in maintaining awareness of overall system state to avoid disruptive actions as various functions may be requested from a system. The problem is that obtaining such information on system state may necessitate collecting data from a number of devices, as there may not be a single data point for state information. This paper describes and illustrates an approach for designing OKD-MES on top of CPSs that controls robot workstations and conveyor-based transportation system of a pallet-based production system.

KEYWORDS | Cyber–physical systems (CPSs); knowledge-based systems; manufacturing automation; manufacturing systems

I. INTRODUCTION

Contemporary manufacturing enterprises both big and small require new generation information systems to enable efficient operation of factories by reducing the time and costs of building and extending manufacturing system functionality, being aware of the current state and therefore being able to make better decisions. The desired efficiency can be achieved through the deployment of affordable manufacturing execution systems (MESs) that are easy to integrate with heterogeneous cyber–physical systems (CPSs) [1] operating at different levels of an enterprise. In order to improve the integration capabilities, various standards and methodologies have to be used for building such distributed networked systems. In addition, it could be beneficial to have system information represented in a common format at all the levels, so that the same language is used to describe heterogeneous components, which in its turn may reduce the time for system components integration.

The next section provides background in the areas of MESs, CPSs, distributed systems, and knowledge-driven approach, which are required to illustrate the approach and case study presented, respectively, in Sections III and IV. These sections demonstrate how the conjunction of the mentioned areas could improve the development

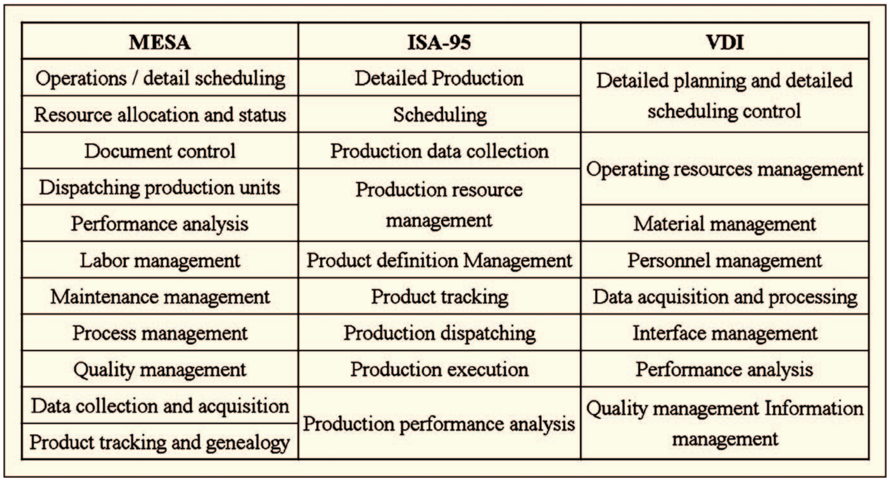
---

Manuscript received June 4, 2015; revised October 30, 2015; accepted November 27, 2015. Date of publication March 10, 2016; date of current version April 19, 2016. This work was supported by the ARTEMIS Joint Undertaking under Grant 332946 and by the Finnish Funding Agency for Technology and Innovation (TEKES), correspondent to the project with the short title eScop, Embedded Systems for Service-Based Control of Open Manufacturing and Process Automation.

The authors are with the FAST laboratory, Tampere University of Technology, Tampere 33101, Finland (e-mail: sergii.iarovyi@tut.fi; wael.mohammed@tut.fi; andrei.lobov@tut.fi; borja.ramisferrer@tut.fi; jose.lastra@tut.fi).

Digital Object Identifier: 10.1109/JPROC.2015.2509498

Table 1 Functions of MES systems



MESA	ISA-95	VDI
Operations / detail scheduling	Detailed Production	Detailed planning and detailed scheduling control
Resource allocation and status	Scheduling	
Document control	Production data collection	Operating resources management
Dispatching production units	Production resource management	
Performance analysis		Material management
Labor management	Product definition Management	Personnel management
Maintenance management	Product tracking	Data acquisition and processing
Process management	Production dispatching	Interface management
Quality management	Production execution	Performance analysis
Data collection and acquisition	Production performance analysis	Quality management Information management
Product tracking and genealogy		

and exploitation of CPS in manufacturing. The main improvements and challenges related to the proposed approach are given in Section V. Section VI presents the conclusions and proposals for future work.

II. BACKGROUND

A. Manufacturing Execution Systems

The information systems applied in manufacturing enterprises are complex systems which should provide various functionalities and be adjusted to the needs of particular enterprise. The more complex the manufacturing process becomes the more challenging is the problem of efficient management of the enterprise. Addressing a problem of complexity, it is usually beneficial to apply a divide and conquer approach defining the independent problem parts and resolving them separately. The efforts to determine which functions the information system components have to provide have been duly applied and resulted in several standards, the most prominent of which are Purdue Enterprise Reference Architecture (PERA)[1] and ANSI/ISA-95.[2]

ANSI/ISA-95 defines a hierarchy of control between the enterprise resource planning (ERP) systems and control systems. In this hierarchy, the MESs assume the role of connecting the office ERPs with the shop-floor equipment implementing manufacturing operational management (MOM) functions in the enterprise. In [2], the following definition of MES is proposed: "A manufacturing execution system . . . is an online integrated computerized system that is the accumulation of methods and tools to accomplish production," where "online" is used as connected. Another important organization Manufacturing Execution Systems Association (MESA) [3] generally assigns MESs the same role. Moreover, as Table 1 shows, ISA-95, MESA, and Verein Deutsche Ingenieure (VDI)[3] propose that almost identical sets of functions be provided on MOM level.

For this study, MES function's classification by MESA model [3] was selected. In the model, "resources allocation and status" functions provide access to all resources in MESs and manage the needs for material from suppliers. "Operations/detail scheduling" functions translate orders to shop-floor schedules and may provide the estimated time for product delivery. Decomposition of an order to a task for the shop floor and updates of the order during execution are implemented within "dispatching production unit" functions. "Document control" functions facilitate document generation and flow in the manufacturing system. "Data collection/acquisition" functions accumulate all data in the manufacturing system, especially from shop-floor hardware and provide these data to other functions. Management of shift schedules for workers belongs to the scope of "labor management" functions. "Quality management" functions facilitate the quality assurance in the manufacturing system in general. "Process management" function takes control to ensure the correct process on the shop floor and issues instantaneous alarms to anticipate and mitigate faults early. "Maintenance management" function provides maintenance schedules for shop floor by alarm management or by periodic maintenance. "Product tracking and genealogy" functions record production information for each product in order to apply performance analysis or for faults recall. "Performance analysis" function analyzes the performance of the facility (production rate, energy consumption, and order manipulation) with the support of graphs.

Commercially available MES solutions are mostly centralized systems, tightly coupled horizontally (between MES functions) and vertically (with ERP and shop floor). Such systems are usually developed by business

---

[1] http://www.pera.net/.

[2] https://www.isa.org/isa95/.

[3] https://www.vdi.de/.

intelligence or industrial automation companies (e.g., SAP and Siemens, respectively). This leads to a strong connection between such MES solutions and the ecosystem of other software provided by the company and arguably reduces the openness for integration with third party components. A tightly coupled approach increases the challenge of customization for enterprise needs, and consequently the introduction of an MES solution usually entails a tradeoff between the modifications of software solution and business process.

Currently, MESs are mostly used by large enterprises following a mass production paradigm. For such enterprises the production volumes are significant, while the variation in the products is limited [4]. Nevertheless, the trends in industry are heading toward increasing customization [5] and an increasing role of SME in economy [6].

The adaptation of conventional MES solutions may be possible, but the authors discuss a more efficient concept for MES system. The requirement of modularization for MESs is suggested by Kletti in 2007 [7]. In more detail, the concept of modularized MESs is proposed in [8]. Bratukhin and Sauter in 2011 suggest the concepts for distributed MESs. The eScop[4] solution—open-knowledge-driven MES (OKD-MES)—proposes combining the modularity of the system with the knowledge-driven approach.

The core concepts on which the architecture of OKD-MES is based are CPSs, service orientation, and knowledge-driven approach. Contemporary ERP solutions commonly provide web service interfaces for integration. Given the service-oriented nature of OKD-MES, the challenge of integration with ERP could and should be addressed through the shared knowledge about the services available in MESs and ERP. Integration with the factory shop floor in turn is more challenging, as in addition to the shared knowledge it is mandatory to enable web services on the constrained equipment of the factory shop floor. More details on OKD-MES concepts in relation to CPSs will be provided in Section III, while later in this section the background on the named core concepts will be provided.

B. Cyber–Physical Systems

As the computational power of embedded systems has increased and information and communication technologies (ICT) standards have been refined to support the development and integration of large-scale networked systems, it has become possible to design, implement, and distribute networked embedded systems currently known as CPSs.

Rajkumar *et al.* [9], [10] outlined the profound effects that development of CPSs may have on society. These include improvements in the energy sector, which could start using smart power grids [11]. CPS also tends to affect the automotive industry and transportation systems in general, the first successful experiments having already been carried out [12] with autonomous vehicles. In agriculture, deployment of CPS may increase yields due to better monitoring of the condition of crop fields as well as the utilization of autonomous machines. CPS requires new approaches to create real-time embedded software [13].

The development of CPS requires multidisciplinary knowledge. Besides knowledge of the domain in which the system and its components physically operate, knowledge of software, communication, control theories, methods, and tools is required. Therefore, the management of interdisciplinary knowledge in order to build a CPS demands good engineering methodologies. In order to address issues arising in the implementation of generally loosely coupled CPS [14], a cyber-application framework is proposed. The framework advocates the use of a number of patterns to handle partially ordered knowledge for building pervasive applications.

New standards and new methodologies make it possible to seamlessly integrate information available on different levels into a single CPS. The decisions in such a system can be made autonomously based on the most recent situation observed at the "physical end"—the process, and throughout the "cybernetics" part of the system—the control functions which can be integrated at all the levels. Also, the manufacturing domain starts to benefit from recent approaches such as cloud computing [15] that provides tools, methods, and techniques to enlist additional computational resources on demand to support the enterprise.

OKD-MES can be positioned at level 3 according to ISA-95, between the factory shop floor and ERP components. MES, as an intermediate element in this perspective, should contain all the necessary functionality not only for covering traditional MES functions, but also to seamlessly link factory floor and enterprise business functionality.

C. Distributed Systems

Distributed systems are formed by components residing in networked computers, which coordinate their actions through the exchange of messages. One of the most valuable benefits and the main motivation for using distributed systems is the option to utilize resources that do not belong to the entity. The definitions and principles of distributed systems are described in [16].

In recent decades, aside from the utilization of PLCs for distributed process control, the factory automation domain has also introduced new types of embedded devices that permit the implementation of CPS in modern production lines. In the last few years, one approach that has been intensively researched and tested is the implementation of the service-oriented architecture (SOA) paradigm, which permits encapsulating the functionality of system components and exposing it as web services (WS) [17], [18]. WS may be deployed in industrial devices employing different approaches such as WS-* from OASIS, OPC-UA model, or following RESTful architectural style. For WS-* one of the most important implementations is a device profile for web services (DPWS),

---

[4] http://escop-project.eu/.

which is a modified WS specification stack based on SOAP protocol that can be implemented in resource-constrained embedded devices. In this manner, SOA can be handled by networked devices located at different levels of an enterprise for controlling and monitoring control processes [19], [20]. These processes are physically executed on shop floors. Being based on the previous OPC standards, OPC-UA is widely accepted in industry and also provides SOAP web services. OPC-UA defines a complex data model which introduces the limits of the application, as it may bring unnecessary complexity to the business applications.

DPWS and OPC-UA are both based on SOAP-based web services rivalled significantly in recent years by the newer representation states transfer (REST) web services. The architectural constraints of REST have been developed to utilize the nature of the web, allowing more scalability and requiring simpler infrastructure to support the applications. The community using REST for general applications exceeded that of SOAP[5] at the beginning of the 2010s and as a result the share of open RESTful APIs is currently over 70%.[6] Constrained application protocol (CoAP) in turn is commonly employed in the Internet of Things domain, implementing simple representational state transfer (RESTful) architecture which enables simple mapping to generic HTTP RESTful services. Basic HTTP RESTful service may also find its niche in the industrial applications domain, mainly for monitoring and non-time-critical tasks.

The embedded devices enabling web services can serve as gateways permitting cross-layer integration between physical equipment and cybersystems, managed and coordinated within the distribution of information and referred to as remote terminal units (RTUs). As a commercial example, S1000 (by Inico Technologies)[7] is a WS-enabled controller that has been used in several research works for controlling the operations of modern production lines. Each device exposes the equipment functionality as service operations which can be invoked. Moreover, information messages are distributed among all the networked devices that participate on the process control, which allows the coordination of each device operation. Some studies using these devices for distributing data and controlling processes are described in [21] and [22].

D. Knowledge-Driven Approach

The preceding sections described the application of CPSs in the industrial domain and presented tools and technologies enabling it in modern manufacturing enterprises. Today, one of the biggest problems that engineers confront is managing the large volumes of information, which flow between the different layers of enterprises. In fact, the issue is not only to handle it but also to extract valuable data that permit the analysis, tracking, evaluation, and understanding of ongoing processes in production lines. This need is crucial in a time when each decision on manufacturing processes directly affects the economies of organizations.

Knowledge representation (KR) and reasoning are a part of the artificial intelligence field that is concerned with describing world information in certain formalisms that can be interpreted and used by computer systems for accomplishing complex tasks. Main concepts and several formalisms (e.g., frames and ontologies) are described in [23]. In the industrial automation domain, the main benefit of KR is the creation of a system model which incorporates all the required information that is generated and consumed by manufacturing systems in both human and machine readable form. Recent works in the field describe how the utilization of KR and its combination with SOA implementation facilitate the management of manufacturing system information [24], [25]. In fact, the scenarios presented in these works envision requirements for implementation of the CPS integration.

Recently, the knowledge-driven approach and main concepts for manufacturing systems have been presented in [26] as the solution for achieving the eScop project objectives. Garetti *et al.* present a multilayer architecture for organizing the entire manufacturing system that manages the orchestration of service operations execution by consulting a central system model, which is updated on runtime with the actual status of the system.

This knowledge-driven approach proposes the use of ontologies as a technology that in recent years has been utilized in the industrial automation domain. Among other benefits, ontologies offer a hierarchical and well-organized framework for the description of the system model. Additionally, ontologies enable reasoning—the process of the generation of new facts in the model based on available facts and predefined sets or axioms and rules. The reasoning automates runtime update and consistency check for the model. Applying semantic rules in ontologies it is also possible to implement data mapping and classification.

There are many languages for implementing ontologies [27] but the most used ones are the resource definition framework (RDF)-based ones [28]. Following the semantic web stack presented in [29], RDF is built on the top of the XML syntax, which does not include any semantic constraint, into the structured documents. Then, RDF Schema is used for defining the taxonomy of document resources, which means semantic constraints (attributes and types) that allow the interrelation of RDF resources in XML-based documents. Ontology models implemented within RDF are RDF graphs that are a set of RDF statements, or RDF triples. Such RDF triples consist of a concrete relation: subject–predicate–object. Ontology web language (OWL) [30] is an extension of

---

[5] http://www.google.com/trends/explore#q=%2Fm%2F077dn%2C%20%2Fm%2F03nsxd&cmpt=q&tz=Etc%2FGMT-2.

[6] http://www.programmableweb.com/search.

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

RDF language that permits richer knowledge descriptions because it adds more vocabulary for describing types (e.g., disjoint and cardinality constraints) and attributes (e.g., symmetry). It should be noted that OWL is classified into distinct sublanguages that offer different description capabilities: OWL-Lite, OWL-DL, and OWL-Full.

Since the decision by manufacturing system components depends on the status of the system, model information can be consulted within queries. Queries can be formulated within different languages but for querying RDF-based models, SPARQL protocol and RDF query language (SPARQL) [31] are the most used. On the other hand, RDF-based graphs can be updated within SPARQL update (SPARUL) [32], which is an extension of SPARQL. Briefly put, SPARUL is a language used for adding, removing, or modifying RDF triples. In fact, RDF graphs can be totally populated with instances through SPARUL queries. Recent studies showing some of the functionality of SPARQL and SPARUL languages in the industrial automation domain are described in [33]–[36].

III. APPROACH

This section presents a description of the approach for the creation of an open-knowledge-driven manufacturing execution system. As mentioned in the background, the MES operates on the level between ERP and shop floor and should connect them. From the bottom-up perspective ERPs are centralized cybernetic solutions (although technically ERP may be implemented in several components), while the control equipment on the shop floor is distributed and resource constrained. This means that the OKD-MES needs to interconnect two systems of different natures.

The basic set of requirements for an MES solution has been summarized in [7] by Kletti and includes ERP production system requirements mapping, modular and expandable nature, adaptability to process and functional needs, and standardized interfaces on all levels.

RTUs are commonly used as a gateway between physical shop-floor devices and systems on higher levels of the automation hierarchy. Considering the increasing computation and communication capabilities of such RTUs, the devices may provide their functions in the form of services, data, and descriptions from their physical components. Such capabilities reduce the required depth of coordination during the development of the system components. Although the recent developments in the field of embedded systems enable more functionality and data on the factory shop-floor level, such functions and data have limited scope and are machine centric (e.g., manufacturing operations, or device-specific data).

The MES solution should handle the shop-floor CPS and provide the functionalities of higher levels of abstraction and complexity. Moreover, a contemporary MES solution should be able to use data and descriptions hosted by the shop-floor device. This information provided by this device can be processed in order to obtain higher level knowledge of the manufacturing system. Such knowledge may be used to make the MES solution more flexible and loosely connected to the underlying shop floor.

Besides providing flexibility on the factory shop-floor level, a contemporary MES must itself be flexible. Such consideration transforms to a design requirement for modularity of MES architecture. The modular architecture should allow the selection of the optimal solution for MES functionality. Such an approach provides great benefit to the domain and is constantly requested by consumers. The concept for OKD-MES is being developed taking the aforementioned points into consideration. In the following sections the architecture of OKD-MES will be described.

The methodology to implement modular OKD-MES is based on SOA. Service orientation should allow the required level of loosely coupled integration to keep the parts interchangeable while maintaining the capability for the interaction of the system. A particular approach for the system architecture REST is selected in this approach for several reasons. First, REST RESTful-WS exploits the web-based nature of the system and uses ubiquitous Internet standards, thus providing an accessible toolkit for interactions with other systems and users. Finally, the application of RESTful services provides easy integration with the tools to place the orders in the system, whether through a third-party ERP system or directly from the system user.

The components of the OKD-MES may be grouped into two sets—those that provide core functionalities to the system and those introducing the additional MES functions exploiting the underlying core. The core functionalities should:

- define and handle configuration of the MES with relation to internal and external components;
- facilitate interactions with shop-floor equipment;
- facilitate interactions with MES users;
- facilitate interactions between the internal components of MES and external systems.

Following the needs for named core functionalities four main layers of OKD-MES core were defined: physical layer (PHL), representation layer (RPL), orchestration layer (ORL), and visualization layer (VIS).

PHL in OKD-MES is embodied in service-enabled RTUs which expose the descriptions of controlled devices as well as available services and data. PHL implements discovery protocols based on multicasting of Hello/Bye messages and on listening for discovery probing/heartbeat requests. Discovery functionality allows synchronization representation of the factory shop floor in RPL with real-world in-system runtime.

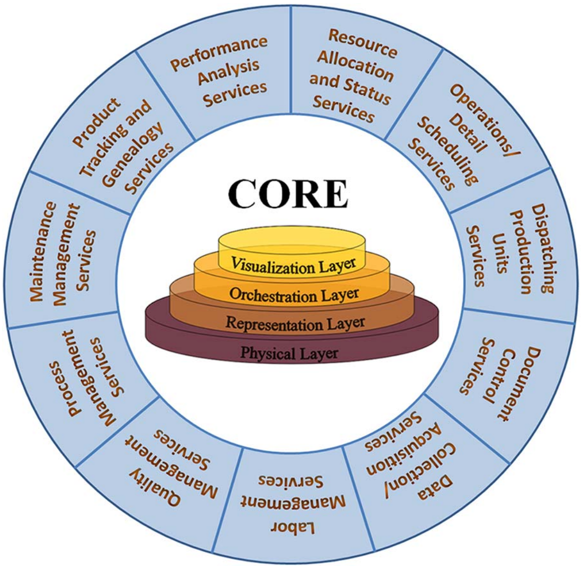


Fig. 1. ***Structure for OKD-MES.***

RPL serves to maintain the KR of the MES and related components. The knowledge about MES is represented in the manufacturing system ontology (MSO) [24]. Exploiting OWL as a language to describe the manufacturing system enables reasoning and querying for the required information when required in the system. RPL contains the system configuration, such as knowledge about component functionality and relations. Employing discovery protocols the system representation is being synchronized with the real world.

ORL enables the execution of sequences of operations provided by the system components according to defined processes. Considering that for the execution of most simple RESTful operations only URL is required, ORL is capable of requesting the next operations from other system components and of dynamically executing them in the system while maintaining the closed loop in the system.

Finally, the visualization layer exposes the user interfaces to interact with system components. This layer is developed to dynamically adjust the user interfaces based on the system configuration available in the ontology. Such an approach reduces the burden on the system reconfiguration related to the development of the user interfaces. Fig. 1 presents the structure for OKD-MES including core layers and complimentary MES functions defined by MESA.

The system status is primarily defined by the status of factory shop-floor equipment. Efficiency of dispatching is based on how precise the representation of the system status is at the moment of dispatching decisions and the possibility to predict the status of the system when the dispatched operation is executed. In case of an integrated OKD-MES solution the decision about dispatch may be made close to the moment of execution of the required operation, thereby reducing the possibility of an erroneous prediction of system status. Moreover, as the status of the equipment is provided by the equipment itself, possible distortions in the precision of system status representation can be improved. The improvement is possible because each piece of equipment starts to assess its status locally instead of reporting often-limited set of data to some central diagnostic application, which may become difficult to change or extend to support the extension of a production line. In mission-critical applications, where also the situation of wrong status reporting by a machine should be avoided or mitigated, the role of "external" observers for particular equipment can be taken by its neighbor machines. It becomes more computationally expensive in comparison to the use of a single supervisor, but can be paid back by simplified dynamic reconfigurability available at system runtime.

IV. CASE STUDY

To demonstrate the capabilities of the OKD-MES the sample MES function implementation will be described. This section presents selected MES function, its design and implementation, and the application case.

The case study for a proposed solution is based on dispatching production unit (DPU) MES function. The DPU function should analyze the production order and dispatch it on the manufacturing equipment in the production line. This demonstrates vertical and horizontal integration for visual and understandable process. As well in OKD-MES DPU has to interact with all layers and may interact with other MES function implementations. Concluding, the DPU function implementation provides a comprehensive example of MES functionality, and consequently was chosen to be used as for current case study.

A. FASTory Line

As a testbed for the implementation of the DPU the FASTory production line (see Fig. 2) was selected. The line is used to simulate the process of mobile phone assembly. The real operations of mobile phone assembly are imitated by drawing the components on the pallet. Considering the nature of the robotic operations required for the real assembly process, scribed simulation is a relatively close approximation. The imitation process includes drawing the three main parts of mobile phone (frame, screen, and keyboard) in different colors and shapes, these variations provide 729 different products.

The FASTory line contains ten identical workstations which can draw the phone parts on paper, one buffer station for loading/storing empty pallets and one station for loading new paper onto the pallet and unloading the ready papers. In this sense, the paper represents the

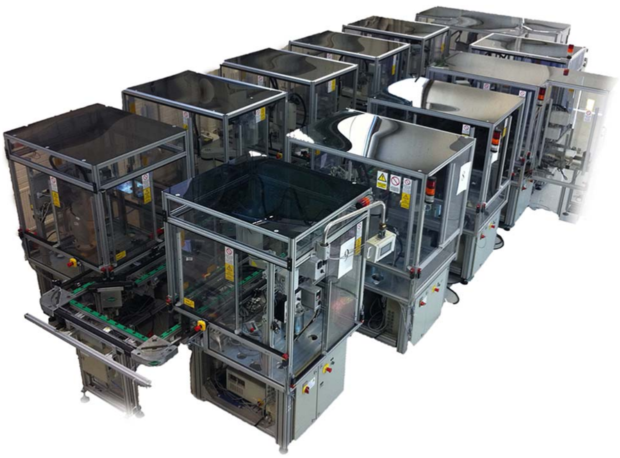


Fig. 2. ***FASTory line.***

product while the colored shapes represent the components of the mobile phone.

Fig. 3 presents the physical structure of FASTory workstations (to be referred to as W#). The pallet buffer station is labeled W7, paper loading station is labeled W1, and finally, the ten processing workstations are labeled W2–W6 and W8–W12. Each processing workstation contains two conveyors paths: a main conveyor to deliver a pallet to the robot and a bypass conveyor moving the pallet to the next station once the workstation is busy. The production line arranged in the closed loop. Such typology provides a continuous path for pallets, thereby increasing the productivity/space ratio [21].

All conveyors are divided into different zones (to be referred to as Z#) as marked in Fig. 3. The inlets and outlets of the workstations are located at Z1 and Z5, respectively, in all stations but W7, where the inlet is in Z2. The possible positions of the pallets in main and bypath routes are marked as Z2, Z3, and Z4. The processing point of each station is in Z3.

With this structure, FASTory is considered to be a flexible assembly line as a pallet can reach any workstation from any position. For each zone, there is a sensor to detect the pallet and a stopper to precisely stop the pallet. Z1 of each station also contains an RFID reader to read the pallet ID.

The FASTory line is equipped with S1000, WS-enabled controllers, managing the shop floor hardware. In addition to the generic controller functionality S1000 is capable of exposing the procedures and data from the line equipment in the form of RESTful services. Among such service the event subscription mechanism is developed. Such mechanism enables event-driven behavior in the system. The component based on its internal logic may send the predefined events to the dynamic list of subscribers. For example, if orchestrator should trigger some process in response to the appearance of the pallet in Z1 of certain station, it may subscribe to corresponding event—*Z1_Changed*—provided by the controller in the line. For subscription, the client should provide the event sink to which the notification should be sent when status of Z1 changes.

Following the service-oriented constrains on development of control logic it is possible to encapsulate the underlying complexity and expose relevant level of abstraction to shop-floor service consumers. Additionally such approach enables exploitation of the web simulators. The simulator for the FASTory line is described below.

B. FASTory Simulator

FASTory simulator was developed as a platform-independent solution. The web application is considered the strongest candidate among other solutions [35]. The FASTory simulator is a web server which hosts RESTful services and web pages that can be accessed via internet browser. The services are virtualizing the shop-floor functionalities, while the set of web pages provides basic information about simulator as well user interface including the visual representation of the status of simulated system.

Simulator significantly accelerates the development process and reduces the potential risks and cost of

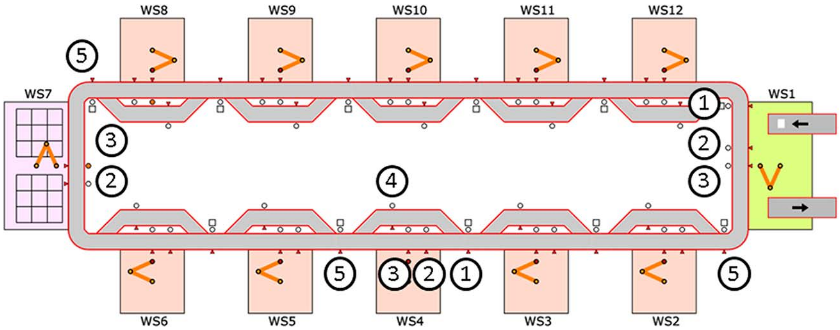


Fig. 3. ***FASTory line typology.***

running a real system. Being online, web-based solution, the simulator provides realistic development environment open for general public. It simulates the real line in terms of interface and functionality.

In the scope of development of OKD-MES the simulator is used for development of ORL, RPL, and MES functions. The solutions developed and tested in the simulated environment are then deployed to real system. If the constrains of simulator were properly addressed in the development, migration from the virtual to real line is a seamless process. In the OKD-MES exploiting the capabilities of RPL and PHL the migration process is reduced to basic connection to the proper networks.

FASTory simulator is accessible online.[8]

C. DPU Implementation

The implementation of the DPU function in OKD-MES requires interaction with PHL, RPL, ORL, and other MES functions. DPU function should provide a service to deploy assign the operation to shop-floor equipment based on the order and status of the system. To facilitate the information about the status DPU should interact with RPL, requesting the information about the available and required functionalities. In order to enforce the decision execution the ORL can be used to manage the complexity of direct interactions with the PHL. Finally some other services such as implementation of operation/detail scheduling (ODS) or resource allocation and status (RAS) functions may be used in dispatcher for decision.

An example of the DPU configuration for FASTory line is provided below. The assumption is made that a certain number of pallets are constantly available to be introduced to the system. In such a case the appearance of the pallet on the inlet of any workstation is the event (*Z1_Changed*) which should trigger the ORL to request the displacement function for particular order in particular station. ORL analyzes the notification to retrieve pallet ID and workstation ID. These are the parameters for which ORL requests DPU to provide the list of production tasks to be dispatched.

Fig. 4 shows the sequence diagram for the scenario. In this representation, ORL focuses only on the request and execution of the task list. RPL provides information once it is required and dispatcher is the part which makes decisions for the production sequence.

The particular decision in the scenario depends on the circumstances in which the DPU was called. If a pallet enters a zone of a workstation, DPU should analyze if the current or any of the following zones of the workstation may provide the services required for the current product. If there are no services to be provided for a product, the pallet is moved from the workstation in a predefined optimal path. If the operations provided by the workstation are required for a product and achievable by the pallet, the pallet has to be moved to the required zone and possible operations are to be executed. In the case of complex and interdependent operations in the product recipe, an additional request for dispatching may be issued after the execution of the manufacturing operations are complete, as new operations may become enabled for the product or in the equipment.

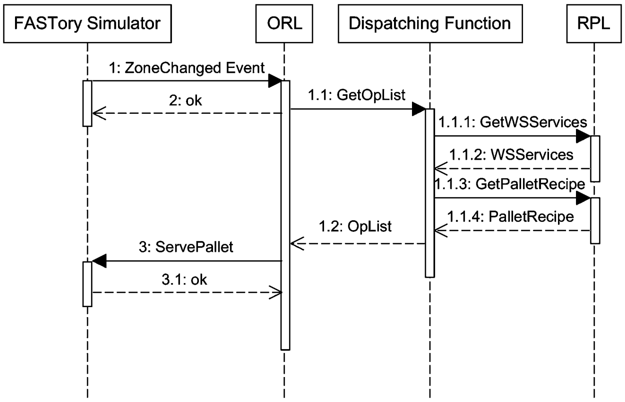


Fig. 4. *Sequence diagram for the dispatching scenario*

Such a decision-making process requires certain models for the representation of the required knowledge in RPL. The part of eScop MSO used in the FASTory DPU use case is depicted in Fig. 5. Conveyor may have event emitted when the pallet appears in one of its zones. Description of event is defined as a triggering event for the DPU, so ORL can subscribe to it. Such an event includes information about the location and ID of the pallet. The pallet belongs to the container concept in MSO and is related to a product from the production order. Product in turn has a production routing or routing, which is a sequence of operations which have to be performed in order to manufacture the product. The operations may match the description of services from the processor in a workstation. In addition to the semantic description, the service may have a technical description required for the invocation of the operation. In the FASTory scenario, the complete technical details are embedded in the URL of the service. Employing such a representation DPU by the set of interactions with the PHL and RPL may retrieve a set of executable URLs which invokes real operations in the manufacturing system based only on information about the location and ID of the pallet.

V. DISCUSSION

Manufacturing systems would generally appear to be doing the same or similar things as before. Just as the appearance of an autonomous vehicle on the road [12] will not generally change its main function, which is to

---

[8] http://escop.rd.tut.fi:3000/.

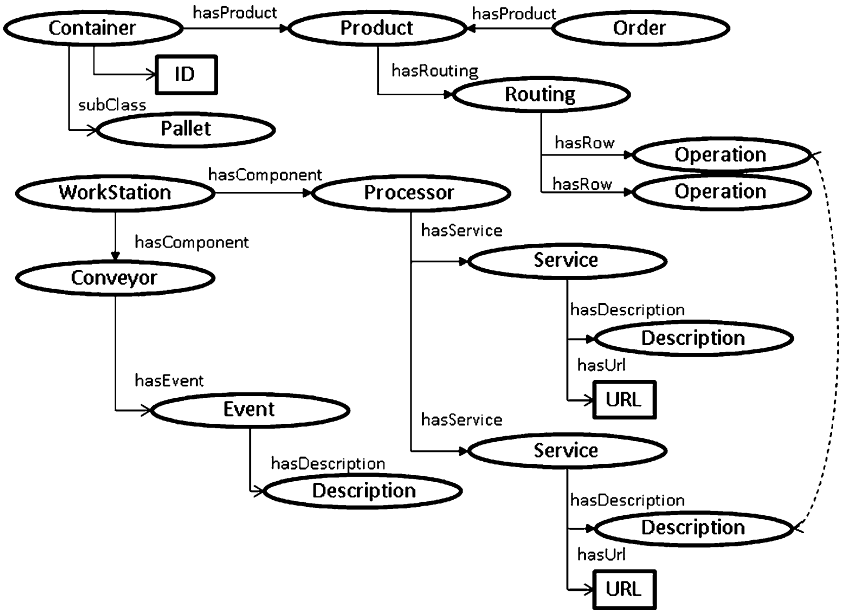


Fig. 5. ***Excerpt from eScop MSO required for DPU service.***

transport people and/or goods from some arbitrary point A to another point B, manufacturing systems equipped with OKD-MES will continue to deliver their main functionality—manufacturing of corresponding products. The adoption of new technologies, methodologies, and standards changes rather the nonfunctional requirements or qualities of a manufacturing system. It is possible to list the four most appealing improvements.

The first one is "time and cost reduction" for the development of MES. The use of common web standards and technologies developed and matured in the field of general ICT reduces development time and costs due to widely available and advanced tools in addition to available expertise and APIs for application development. For instance, it takes just several days for an engineer to develop a fully functioning online simulator of the production system presented in the previous section. It enables fast prototyping evaluating different ideas. Furthermore, the only tool needed to operate the simulator and the actual production line is a basic web browser.

The web browser can be used to visualize information on the line and to interact with it by invoking services on the controllers. For instance, among other tools Advanced REST Client by Google can be mentioned,[9] which is accessible via a web browser. The tool can be directly used as an engineering tool, for example, to test the operations of line controllers. On the other hand, sales and marketing personnel get new opportunities to present solutions to their potential customers as the majority of people are familiar with handling a web browser. In comparison to commercially available products, such as, for example, totally integrated automation[10] by Siemens, presented approach is open for different service-enabled controllers. In fact, the approach is meant for service-oriented systems. Although major PLC vendors did not pick up the approach yet, the general IT sector moves and evolves around the web standards. The authors predict that similarly to the Ethernet, which was adapted to various industrial Ethernet protocols in industry, the adaptation of software engineering paradigms will follow. The authors also understand that major vendors may still want to bind their customers to their solutions by providing virtuous kinds of "integrated" solutions. As good as such solutions can be they will cause a vendor lockup problem. OKD-MES approach can be followed by those, who may also want to avoid being locked up to the particular vendor.

Second, system "extendibility" is increased. New functions can be relatively easily added using web service standards, thus extending the overall system functionality. In general, since web standards and IP-based networks are used on the factory floor, the CPS can be integrated into a global network if necessary. In the example of dispatch function outlined in the previous sections, the set of services required to address particular product needs may be extended or changed over time, but without requiring reprogramming of MES functions. This is possible due to late binding in the loosely coupled

---

[9] Advanced REST client, https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo.

[10] http://www.industry.siemens.com/topics/global/en/tia/pages/default.aspx.

OKD-MES based on the domain knowledge. Consequently to introduce new manufacturing system entities in most cases it should suffice to use the same concept vocabulary or to map the domain vocabularies of the application. In the case of more complex modifications, a change of OWL (knowledge) models may be required to facilitate the representation of additional concepts. Yet even in this case, an engineer can already use online tools[11] (requiring just a web browser). The complete eScop MSO ontology can be accessed using the aforementioned online tools.

Third, systems can be characterized by "adaptability" with respect to new conditions, which may be due to changes in production processes or in production equipment. The change is noted in the knowledge model and the model can be directly updated via a SPARUL query sent to RPL by a device integrated into the physical environment. The word "query" should not be confused here. The "update query" is sent to make a change in the knowledge model rather than to retrieve some information. Thus, an engineer should develop and test valid queries when deploying devices. The same online tools mentioned for "extendibility" can be used for developing and testing the queries.

Finally, system "availability" is improved due to better awareness of system resources and their status. Introduction or removal of the hardware which follows certain constraints can be reflected in the system model within seconds. Given the event-driven and service-oriented nature of the proposed system, the status of system resources can also be updated. For more elaborated resource representation, additional MES services (such as resource status and allocation) may be developed. The awareness of configuration and status of the components contributes to improved decision making and failure handling.

In addition to qualitative improvements, there are certain risks or challenges associated with the use of CPS for OKD-MES. The three most relevant of these can be described as follows.

"Security" is a general demand for any networked system. Since it is now easy to access CPS on the factory floor, as even a basic tool such as a web browser can be used not only to obtain information on the status of a device or the process it controls, but also to invoke an operation on the device, therefore security measures should be carefully implemented. Again, general IT policies and standards developed for global networks can be applied here. As soon as public networks are used for data transmission, the data can be encrypted decreasing the chances of security breaches. The factory floor should be isolated behind network firewalls, with analytical tools and procedures in place for detecting possible attacks.

"Observability" can be seen as a more important challenge specifically for manufacturing. Observability is an ability to know the system state, as and when it is needed. As decision making is pushed to the lower levels to increase system responsiveness and adaptability, it becomes more challenging to observe the overall system state. In manufacturing, many products may be handled in parallel, competing for the same resources of the manufacturing system, and the system must provide mechanisms for handling conflict resolution. Orchestrators that do service composition for making a product should be aware of other production workflows and their statuses.

Another issue for implementing CPS is fast changing standards. For example, there are different versions of simple object access protocol (SOAP)[12] that can be used for invoking web services. There are different versions of device profile for web services (DPWS)[13] that can be used to build service-enabled CPS. There are a number of versions for business process execution language (BPEL)[14] that can be used for service composition. There are different versions of HTML, and so on. A CPS following web standards needs to use several standards and protocols at the same time. Aiming a functional integration of such standards and protocols, the implementation of OKD-MES is developed within open standards, which are mature enough to perform required features. Although web standards evolve, they are often compatible extensions of their predecessors, providing more complex, richer or, simply, new features (e.g., RDF and OWL or SPARQL and SPARUL). On the other hand, APIs developed for one version of the standard may not have forward compatibility, thus solutions developed earlier may not be directly integrated with newer applications. In order to mitigate this risk, a web browser can be used as a benchmarking tool. That is, if the technology and/or protocol is supported by a web browser, then it is more likely to be around for a longer time. Due to the application scale of Internet technologies, those which are supported by the majority of the web browsers would tend to have the most mature and highly developed APIs. Then, applications developed within web standards, which are directly supported in different web browsers, will be compatible with future technologies.

The presence of MES solutions in SMEs is limited due to the complexity, inflexibility, and high implementation and customization costs of an existing solution. OKD-MES is being developed to address this niche. In order to address the MES system migration for enterprises already having an MES solution, the migration approach is required. Such an approach might be based on the advantage of the open and knowledge-driven nature

---

[11] eScop online tools, http://www.escop-project.eu/tools.

[12] http://www.w3.org/TR/soap/.

[13] http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01.

[14] https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel.

of the system. Openness makes it possible to develop the solution to bridge the gap once and share it with the community, while knowledge-driven nature supports the reusability of a developed block providing a shared but flexible model of required knowledge.

The challenge of "performance" is often an obstacle to the widespread application of knowledge-driven solutions. In light of the dependence of the performance of the interactions with the knowledge base on the complexity of queries and the size of the knowledge base, the surest approach to verify the required performance may be achieved only through extensive testing and benchmarking. The prototype implementation of OKD-MES solution was capable to process tens of mixed queries per second in persistent mode and hundreds in nonpersistent mode. Regarding the application of the prototype in several medium-sized pilot cases (e.g., FASTory line) it was estimated that it is realistic to maintain the representation of system configuration in the knowledge base and use it for decision making in MES functions. Additional research and optimization may enable better performance and as a result should be capable of managing larger systems.

Having discussed four qualities and challenges of CPS for manufacturing, it is also important to stress that the manufacturing of devices that can be used for controlling industrial equipment is currently in the hands of only a few rather small companies. As APIs, standards, and tools reach maturity, the use of CPSs in conjunction with knowledge-driven approaches in domain of manufacturing will become common. Meanwhile, the adaptation of the approach to the legacy systems can be carried out in two basic ways.

The first approach is servitization of high-level supervisory and monitoring applications providing service interfaces for other high-level enterprise applications. The supervisors or monitors can continue to use traditional fieldbuses to communicate with the equipment wrapping and translating information between the equipment and enterprise functions.

The second approach is servitization of controller devices by installation of gateway devices at the factory floor. The role of the device is the same as in the first case—to translate and wrap the functionality of a controller as a set of web services, but it can be a dedicated solution for particular industrial controller. The cost of the gateway hardware can be relatively low, in terms of tens of euros[15] and it is already capable to run applications using latest service APIs.

VI. CONCLUSION AND FUTURE WORK

The use of CPSs for OKD-MES was illustrated using FASTory production line. The main challenges as well as the improvements of the approach proposed were discussed. Web standards and mature Internet-based technologies made it possible to develop and integrate an application using less time due to affordable and widely used basic tools such as a web browser. Future work on developing integrated methodology that would combine methods, tools, and techniques used in heterogeneous disciplines may be required to improve the adoption of the approach in the manufacturing domain. The development of the consumer market for handheld devices making, for instance, a smartphone a common and widely used tool, the functions of which can be extended with the installation of new applications using the same web standards to interact with the manufacturing systems may contribute to a paradigm shift toward the use of open-knowledge-driven manufacturing execution systems.

Because of the advantages mentioned in Section V, the OKD-MES concept may become more desirable in the manufacturing domain. CPS is one of key systems with significant potential in implementing the OKD-MES concept. Since traditional manufacturing systems can in principle deliver the basic functionality expected from such systems, the OKD-MES concept faces challenges in the adoption of the approach by industrial community. A solution to this challenge would require new system architecture which homogenizes the manufacturing ecosystem applying the proposed approach. ■

REFERENCES

[1] J. Sztipanovits *et al.*, "Toward a science of cyber–physical system integration," *Proc. IEEE*, vol. 100, no. 1, pp. 29–44, Jan. 2012.

[2] M. McClellan, *Applying Manufacturing Execution Systems*. Boca Raton, FL, USA: CRC Press, 1997.

[3] MESA International. [Online]. Available: http://www.mesa.org/en/index.asp.

[4] A. Bratukhin and T. Sauter, "Functional analysis of manufacturing execution system distribution," *IEEE Trans. Ind. Inf.*, vol. 7, no. 4, pp. 740–749, Nov. 2011.

[5] V. Modrak, S. Bednar, and D. Marton, "Generating product variations in terms of mass customization," in *Proc. IEEE 13th Int. Symp. Appl. Mach. Intell. Inf.*, 2015, pp. 187–192.

[6] Manufacturing Execution Systems—Accenture, 2010.

[7] J. Kletti, *Manufacturing Execution Systems—MES*. New York, NY, USA: Springer-Verlag, 2007.

[8] F. K. Johnson, "Future of manufacturing execution systems: The brave new modular world of manufacturing intelligence," *Rev. Manag.*, vol. 1, no. 1, p. 4–14, 2011.

[9] R. Rajkumar, "A cyber–physical future," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1309–1312, May 2012.

[10] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proc. 47th Design Autom. Conf.*, 2010, pp. 731–736.

[11] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," *Proc. IEEE*, vol. 99, no. 6, pp. 998–1027, Jun. 2011.

[12] E. Guizzo, "Autonomous vehicle driving from Italy to China," *IEEE Spectrum*, Sep. 21 2010. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/robotics-software/autonomous-vehicle-driving-from-italy-to-china.

---

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

[13] J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia, and J. Zou, "Distributed real-time software for cyber–physical systems," *Proc. IEEE*, vol. 100, no. 1, pp. 45–59, Jan. 2012.

[14] J.-S. Choi *et al.*, "Application patterns for cyber-physical systems," in *Proc. IEEE 1st Int. Conf. Cyber-Phys. Syst. Netw. Appl.*, 2013, pp. 52–59.

[15] A. Colombo *et al.*, *Industrial Cloud-Based Cyber-Physical Systems: The IMC-AESOP Approach*. New York, NY, USA: Springer-Verlag, 2014.

[16] G. F. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems: Concepts and Design*. New York, NY, USA: Pearson, 2005.

[17] SOA and Web Services. [Online]. Available: http://www.oracle.com/technetwork/articles/javase/soa-142870.html.

[18] A. Lobov, J. Puttonen, V. V. Herrera, R. Andiappan, and J. L. Martinez Lastra, "Service oriented architecture in developing of loosely-coupled manufacturing systems," in *Proc. 6th IEEE Int. Conf. Ind. Inf.*, 2008, pp. 791–796.

[19] I. M. Delamer and J. L. Martinez Lastra, "Self-orchestration and choreography: Towards architecture-agnostic manufacturing systems," in *Proc. 20th Int. Conf. Adv. Inf. Netw. Appl.*, 2006, vol. 2, p. 5.

[20] A. N. Lee and J. L. Martinez Lastra, "Data aggregation at field device level for industrial ambient monitoring using web services," in *Proc. 9th IEEE Int. Conf. Ind. Inf.*, 2011, pp. 491–496.

[21] L. E. G. Moctezuma, J. Jokinen, C. Postelnicu, and J. L. Martinez Lastra, "Retrofitting a factory automation system to address market needs and societal changes," in *Proc. 10th IEEE Int. Conf. Ind. Inf.*, 2012, pp. 413–418.

[22] S. Iarovyi, J. Garcia, and J. L. Martinez Lastra, "An approach for OSGi and DPWS interoperability: Bridging enterprise application with shop-floor," in *Proc. 11th IEEE Int. Conf. Ind. Inf.*, 2013, pp. 200–205.

[23] R. J. Brachman, *Knowledge Representation And Reasoning*. San Mateo, CA, USA: Morgan Kaufmann, 2004.

[24] L. Fumagalli, S. Pala, M. Garetti, and E. Negri, "Ontology-based modeling of manufacturing and logistics systems for a new MES architecture," in *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*, B. Grabot, B. Vallespir, S. Gomes, A. Bouras, and D. Kiritsis, Eds. Berlin, Germany: Springer-Verlag, 2014, pp. 192–200.

[25] B. Ramis *et al.*, "Knowledge-based web service integration for industrial automation," in *Proc. 12th IEEE Int. Conf. Ind. Inf.*, 2014, pp. 733–739.

[26] M. Garetti, L. Fumagalli, A. Lobov, and J. L. Martinez Lastra, "Open automation of manufacturing systems through integration of ontology and web services," in *7th IFAC Conference on Manufacturing Modelling, Management, and Control*, 2013, vol. 7, pp. 198–203.

[27] D. Kalibatiene and O. Vasilecas, "Survey on Ontology Languages," in *Perspectives in Business Informatics Research*, J. Grabis and M. Kirikova, Eds. Berlin, Germany: Springer-Verlag, 2011, pp. 124–141.

[28] W3C, "RDF 1.1 concepts and abstract syntax." [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[29] "Semantic web architecture—Introduction to ontologies and semantic web: Tutorial." [Online]. Available: http://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html.

[30] W3C, "OWL 2 web ontology language document overview (second edition)." [Online]. Available: http://www.w3.org/TR/owl2-overview/.

[31] I. Kollia, B. Glimm, and I. Horrocks, "SPARQL query answering over OWL ontologies," in *The Semantic Web: Research and Applications*. New York, NY, USA: Springer-Verlag, 2011, pp. 382–396.

[32] W3C, "SPARQL 1.1 update." [Online]. Available: http://www.w3.org/TR/2013/REC-sparql11-update-20130321/#sec-intro.

[33] J. Puttonen, A. Lobov, and J. L. Martinez Lastra, "Semantics-based composition of factory automation processes encapsulated by web services," *IEEE Trans. Ind. Inf.*, vol. 9, no. 4, pp. 2349–2359, Nov. 2013.

[34] J. Puttonen, A. Lobov, and J. L. Martinez Lastra, "Maintaining a dynamic view of semantic web services representing factory automation systems," in *Proc. IEEE 20th Int. Conf. Web Services*, 2013, pp. 419–426.

[35] B. R. Ferrer, S. Iarovyi, L. Gonzalez, A. Lobov, and J. L. Martinez Lastra, "Management of distributed knowledge encapsulated in embedded devices," *Int. J. Prod. Res.*, vol. 0, no. 0, pp. 1–18, Dec. 2015.

[36] T. Preobrazhenskaya, M. Bakaev, and T. Avdeenko, "The role of data structures design in modern web applications development," in *Proc. 8th Int. Forum Strategic Technol.*, 2013, vol. 2, pp. 276–279.

ABOUT THE AUTHORS

Sergii Iarovyi received the M.Sc. degree (with distinction) in electromechanics from the National Technical University "KhPI," Kharkiv, Ukraine, in 2011 and the M.Sc. degree in factory automation from Tampere University of Technology, Tampere, Finland, in 2014.

Since 2012, he has worked at the FAST Laboratory, Tampere University of Technology, as a Project Researcher. His research interests are in the application of semantic web services, cyber-physical systems, and enterprise integration for factory automation.

Wael M. Mohammed received the B.Sc. degree in mechatronics engineering from Jordan University, Amman, Jordan, in 2010.

From 2010 to 2011, he was a Research Assistant at Tampere University of Technology, Tampere, Finland, and from 2011 to 2013, he worked as Head of the Technical Department at Etihad Alafandi LLC in Eastern Province, Kingdom of Saudi Arabia. Since 2015, he has been a Research Assistant with the FAST Laboratory at Tampere University of Technology.

Andrei Lobov received the B.S. degree in computer and system engineering from Tallinn University of Technology, Tallinn, Estonia, in 2001, the M.S. degree in automation engineering from Tampere University of Technology, Tampere, Finland, in 2004, and the Dr. Tech. degree on the subject of formal methods in factory automation from Tampere Universtiy of Technology in December 2008.

He is a University Lecturer at Tampere University of Technology. His research interests include development of architectures, methodologies, and technologies for industrial manufacturing systems. He is a technical coordinator in the eScop project.

Borja Ramis Ferrer received the B.Sc. degree in electrical engineering from the Universidad de las Islas Baleares, Islas Baleares, Spain, in 2011 and the M.Sc. degree (with distinction) in factory automation from Tampere University of Technology, Tampere, Finland, in 2013, where he is currently working toward the Ph.D. degree.

He is a Fellow of the President's Doctoral School at Tampere University of Technology. His research interests include the deployment of knowledge-based and cyber-physical systems in factory automation.

Jose L. Martinez Lastra received the Ingeniero Tecnico Industrial degree in electrical engineering from the Universidad de Cantabria, Santander, Spain, and the M.Sc. degree (with Distinction) and the Dr.Sc. in Technology degree (with Commendation) in automation engineering from the Tampere University of Technology, Tampere, Finland.

He joined Tampere University of Technology, Tampere, Finland, in 1997, and became University Full Professor in 2006. His research interest is in applying information and communication technologies to the fields of factory automation and industrial systems. He leads the FAST Laboratory, Tampere University of Technology, with the ultimate goal of seamlessly integrating the knowledge of humans and machines. He has co/authored over 250 scientific papers and holds a number of patents in the field of industrial informatics and automation.

Dr. Martinez Lastra serves as an Associate Editor of the IEEE Transactions on Industrial Informatics, and he is a Technical Editor of the IEEE/ASME Transactions on Mechatronics.