2015 IEEE European Modelling Symposium

Potentials of web standards for automation control in manufacturing systems

Borja Ramis Ferrer, Sergii Iarovyi, Andrei Lobov, José L. Martinez Lastra
Tampere University of Technology,
Factory Automation Systems and Technologies Laboratory (FAST-Lab.),
Tampere, Finland
{borja.ramisferrer, sergii.iarovyi, andrei.lobov, jose.lastra}@tut.fi

***Abstract*—Web standards developed mainly by W3C and OASIS shape general IT domain and its applications. Due to the scale of web applications, the web standards have matured to deal with the typical situations of finding the right node on the network, reconfiguring the routing for messaging, using common standards for representing graphical information and many others. Industrial manufacturing can benefit from the web standards due to the interoperability and simplified application integration. This article reviews the current use of web standards in the industrial automation domain. In addition, the manuscript describes and discusses the potentials of using web standards at all the levels of automation system: from high level web-based user interfaces to the industrial controllers, which are located in the lowest layer of the well-known automation pyramid. Aligned with such description, the article presents a framework for Open, Knowledge Driven Manufacturing Execution Systems (OKD-MES), which allows using systematically web standards and technologies in factories.**

Keywords— Web standards; industrial automation; OKD-MES; Semantic Web

I. INTRODUCTION

In the domain of industrial manufacturing throughout last decade the level of competition for customers and resources is constantly increasing. One of the most important approaches for reducing operational costs related to manufacturing and customization of goods is linked to possibilities of more interoperable automation systems. The concept of Industrial Internet is one of the possible approaches to achieve new quality of factories.

Concept of Industrial Internet requires more capable devices, which can interact in the web system. In the last several years certain efforts were made to obtain such smart devices. Some of devices are capable to interact with significant levels of autonomy and to provide more interactive and adaptable functionalities to industrial systems [1], [2], [40].

Modern factory shop floor equipment and software provide more heterogeneous and interoperable infrastructure for automation. One of the key problems in such large-scale systems is lack of interoperability and need for customized integration of components. In fact, as the nature of the Industrial Internet system is very similar to the one of consumer Internet, it provides a possibility to employ solutions of later one for factories.

Web standards are critical for enabling factors of Internet success. World Wide Web Consortium (W3C) and OASIS are the leading standardization organizations in the domain and provide a comprehensive and versatile set of standards to enable the web based systems. The application of such web standards in industrial systems may allow not only resolving technical difficulties, but also remove the actual barrier that does not permit the complete integration of two domains: general IT and factory automation.

According to its use, specifications and demonstrated implementations, web standards are open, mature and usually efficient. Some of standards such as the concept of Web Services (WS), and some particular implementations of it, have been already adapted with successfully to factory shop floor control devices [3]–[5]. In fact, based on service-enabled devices, the application of Semantic WS for manufacturing has been already researched and documented in several articles [6]–[8]. Also, the evaluation of applying web standards for system and knowledge representation systems, decision support and visualization is presented in [9].

Right now, the ongoing EU Project eScop[1] (Embedded systems Service-based Control for Open manufacturing and Process automation), employs web standards for multiple purposes in factory wide Manufacturing Execution System (MES). Within the use of web standards, the eScop project is developing a framework for realizing a new concept: Open, Knowledge Driven Manufacturing Execution Systems (OKD-MES). Descriptions about the novel OKD-MES concept are found in [15], [32], [46].

The research for applying web standards in different levels of manufacturing systems is presented in this paper as follows: Section I describes the possibilities of different applications of WS standards for factory shop floor devices. Section III describes a set of standards that are applicable for modeling industrial systems. Afterwards, Section IV presents the possibility of applying web standards for coordinating systems, which employs described in Section II and Section III. Then, Section V provides discusses some possibilities for exploiting some of the web standards for industrial system visualization. In Section VI, the synergy of the standards applied in all levels of the system is described and practical examples are also offered. In fact, the description of such integration of standards permits the presentation of the OKD-MES concept. Finally, Section VII concludes the paper and suggests the further work.

---

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

978-1-5090-0206-1/15 $31.00 © 2015 IEEE
DOI 10.1109/EMS.2015.59

II. DEVICES

The main goal of industrial automation is controlling and monitoring processes, in an automated manner. To achieve this, industrial controllers are used for controlling sensors and actuators that, conjunctly, are the devices that permits the physical execution of process. Usually, industrial controllers are deployed near to the location where the sensors and actuators reside. The deployment of industrial controllers in a facility implies that these units have a small form factor that, in return, constraints its computational resources [10]. In addition, industrial controllers need to execute the control logic and exchange information with other controllers in same level (horizontal communication) or with higher-level components (vertical communication) of the entire industrial automation system.

There exist several communication protocols that enable horizontal integration between controllers as CAN, etherCAT and Modbus, among many other fieldbuses. Nevertheless, web-based technologies are the most convenient and suitable for allowing the vertical communication of controllers and other industrial components like MESs or even Enterprise Resource Planning (ERP) modules [11].

A. *Web standards for devices*

Some of the service-oriented communication standards and approaches that have got special attention by the academia and the industry include: OPC Unified Architecture (OPC UA), Device Profile for Web Services (DPWS) and the Representational States Transfer (REST) architecture [12], which are described below.

OPC UA is the successor of Object Linking and Embedding (OLE) for process control (OPC). This communication stack offers several features like: scalability, multi-threading and security. An outstanding characteristic is that the OPC UA is a Service Oriented Architecture (SOA). It exposes methods and device functionality as services, which are protocol independent. Two protocols are defined for this purpose: binary Transmission Control Protocol (TCP) protocol and Web-service oriented. The binary protocol is highly efficient and reduced significantly the transactions overhead [13]. The Web Service Simple Object Access Protocol (SOAP) protocol facilitates the integration with traditional IT components, tools and technologies. It is understood easily by firewalls and uses Hypertext Transfer Protocol (HTTP), which is the foundation protocol of the World Wide Web. The main drawback of OPC UA is that it is required to be part of the OPC foundation in order to get access to the stack specification.

The DPWS standard [14] defines and tunes a set of SOAP based WS protocols for devices. This stack profile enables WS capabilities on resource-constrained devices. Such capabilities include, secure invocation of WS operations, description and dynamic discovery of WSs and mechanisms to subscribe and receive events from WSs. An important characteristic is that devices that implement DPWS are fully aligned with the WS technology. This facilitates the vertical integration of devices with high-level applications. This synergy between physical devices and cyber systems realizes one of the topics being nowadays widely discussed in the industrial automation: cyber-physical systems (CPS) [1], [2], [38], [39]. Nevertheless, one drawback of this protocol is its verbosity due to the fact that SOAP messages are XML formatted. It is important to mention that DPWS is an open standard and, hence, any manufacturer can adopt it. In the industrial domain, there are already commercial industrial controllers that implement the DPWS stack as, for example, the S1000 from Inico Technolgies[2].

Both OPC-UA and DPWS use Remote Procedure Call (RPC) as an architectural style for the services. In this case the large part of HTTP capabilities is not being used. It may be useful if the system employs services that are employing different protocols, but it is rather rare situation. As well RPC services tend to be more tightly coupled with clients than some other implementations of WS.

Representational States Transfer (REST) is another software architectural style applicable for WS. REST is defining set of constrains which makes WS more compatible with Web infrastructure and technologies. Constrains of REST are enabling more scalable, loosely-coupled and efficient services compared to RPC. The RESTful WSs often employ HTTP/HTTPS as the transport and application protocol [16]. These web mechanisms are very well understood and can be easily ported over the web. Currently the payload in RESTful WSs often is formatted as JavaScript Object Notation (JSON), which is less verbose than XML and still human readable. There is a vast amount of technologies, frameworks and tools from traditional IT that can be exploited in the industrial automation field if industrial controllers are implemented with RESTful capabilities [17].

For the formal description of WS, several description languages may be applied. The most important for description of SOAP services is the Web Service Description Language (WSDL). WSDL provides complete technical description of services. In order to enrich this description with metadata about services, Semantic Annotations for WSDL (SAWSDL) is the one that is usually employed. It allows connecting the service with related concepts in the model. It should be noted that WSDL2.0 might be applied as well for RESTful services. However, different approach is usually applied in this case for reasons, partially described further.

B. *Application of REST*

Application of RESTful WS on the factory shop floor level may require some different approach then implementation of RPC services. It is related mainly with different perception of the system in these approaches. If RPC usually focuses on the actions in the system, REST suggests concentrating on resources. HTTP verbs such as *GET*, *PUT*, *POST*, and *DELETE* are used with the resources to describe the action to be done on the resource. Furthermore, HTTP response codes and other headers may provide additional information about service execution.

---

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

The resources in REST are being defined by their representation. The concept of Hypermedia (linked media) is usually applied for resource representation in order to define resource relations with other resources. This approach allows further exploitation of web architecture, as links on other resources may be dereferenced without any extra tools or technologies. Furthermore, it is suggested to employ Hypermedia as the Engine of Application States (HATEOAS). This approach allows services to instruct clients on further possible operations and, hence, to remove a need for a client to guess the next operation. In means that in case of changes in service implementation it will inform client immediately on execution of the service. An approach to have semantically rich HATEOAS descriptions is being developed within Hypermedia Driven Web APIs (Hydra) [18], [19].

III. MODELS

The models are another important element for contemporary industrial automation systems. According to stated conditions, knowledge of system properties and their interrelations allows adequately acting and reacting in the modeled system. While the previous section concentrated on the question how to provide interaction of automated system with physical world, this section has a focus on representation of available knowledge about the system.

A. *Models for design and runtime support*

For making information systems capable of managing manufacturing processes, it is required to transform real world assumptions into a formal model, which must be understandable by cyber systems. Ontologies may be used for representing the real world because they provide a pragmatic means for modeling domain specific knowledge. An ontology is an explicit and formal specification of a shared conceptualization [20]. In other words, ontologies are employed for representing knowledge, of certain domain, as a set of concepts, their definitions and interrelationships. Web Ontology Language (OWL) is one of the ontology languages used for constructing ontologies. The language is characterized by formal semantics and RDF/XML-based serializations for the Semantic Web. As illustrated in Fig. 1, a hierarchical stack represents the architecture of the Semantic Web [35].

In the presented stack, XML is a surface syntax of structured documents and it does not have any semantic constraints on the document. Then, XML Schema defines the structure constraints of XML documents. RDF [22] is a data model of resources and their relationships expressed by XML syntax that provides simple semantics for the data model. RDF Schema [23] is a vocabulary describing the attributes and types of the RDF resources. Hence, it provides generic semantics for the attributes and types. Finally, OWL adds more vocabulary to describe attributes and types, such as disjoint and cardinality constraints in types and symmetry in attributes.

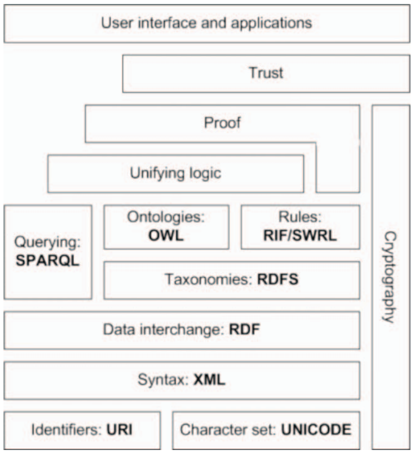

Fig. 1. The Semantic Web Stack [35]

OWL is often presented as one of the richer ontology languages because it has more mechanisms for representing semantics in comparison with XML, RDF and RDFSchema. Nevertheless, Table I shows a comparison of various syntaxes for OWL 2, which is an extension and revision of OWL developed by the W3C Web Ontology Working Group and published in 2004 [21]. As shown in Table I, several syntaxes can be used to store OWL 2 ontologies and to exchange them among tools and applications. It should be noted that the primitive exchange syntax for OWL 2 is the RDF/XML. Thus, RDF/XML is the syntax that must be supported by all OWL 2 tools.

TABLE I. COMPARISON OF SYNTAX FOR OWL 2

Name of Syntax	*Status*	*Purpose*
RDF/XML	Mandatory	Interchange (can be written and read by OWL 2 software)
OWL/XML	Optional	Easier to process with XML tools
Functional Syntax	Optional	Easier to see the formal structure of ontologies
Manchester Syntax	Optional	Easier to read/write DL Ontology
Turtle	Optional	Easier to read/write RDF triples

Users and applications can interact with ontologies and data by querying the ontology model using the SPARQL query language [24], which was standardized in 2008 by the W3C. The standard query evaluation mechanism is based on sub-graph matching and is called simple entailment because it can equally be defined in terms of the simple entailment relation between RDF graphs [25]. Like other database query language, SPARQL uses several keywords to form a pattern to query data. A SPARQL query contains three parts. Firstly, the pattern matching part includes fundamental features of pattern matching of graphs as, for example, optional parts, union of patterns, nesting, filtering (or restricting) values of possible matches. Secondly, the solution modifiers part allows modifying the obtained query values applying classical operators like *DISTINCT*, *ORDER* and *LIMIT*. Thirdly, the output of a SPARQL query can be of different types: Boolean queries (true/false) for *ASK* query type, selections of values of the variables which match the patterns for *SELECT* query type, construction of new triples from these values for *CONSTRUCT* query type. Moreover, SPARQL Update [34] is an extension of SPARQL, which permits the update of the model within different operations as *DELETE* or *INSERT* for deleting or inserting triples. It should be noted that the output of such queries is an execution message for indicating the success of the update operation.

Ontology also supports reasoning, which is deriving facts that are not explicitly asserted in the ontology. For example, if a model describes that A is ancestor of B and B is ancestor of C, then, although the conclusion that A is also ancestor of C is trivial, the model needs an additional engine called reasoner, which is capable of inferring these new fact. Then, a reasoner (or reasoning engine) is a piece of software cable of performing reasoning tasks. In other words, a reasoner is capable of inferring logical consequences from a set of asserted facts in the ontology. Nowadays, there are several reasoners that are available as, for instance, FaCT++, Pellet and HermiT. Nevertheless, Pellet seems to be one of the most common reasoning engines used for reasoning OWL models in current implementations. It should be noted that rule languages, as the Semantic Web Rule Language (SWRL) [41], are used for defining rules in ontological models that are interpretable also by reasoning engines. Within the employment of semantic rules, automatic mapping of ontological individuals can be achieved, as demonstrated in [42], [45].

According to previous explanations, it is obvious that ontologies enable the construction of models for design and runtime support of systems. This is feasible because ontologies allow the description of models that are understandable in machine-to-machine, human-to-machine and even human-to-human communications. In addition, the computational inference that is achieved thanks to reasoning engines permits an automatic solution for deriving new facts, which are later included in the model. Finally, it should be noted that ontologies are reusable, extendible and flexible. This means that population of data in ontological models is possible after the model definition at runtime. This feature makes them even more powerful in the industrial domain because the system model changes continuously due to the large amount of events that succeed in manufacturing systems. In fact, the benefits of ontologies for manufacturing and logistics operations management are presented in [47].

B. *Use of ontologies at runtime*

Modeling within ontology is one aspect of knowledge driven information systems, which involves use of ontologies at development phase. The other aspect is its usage at runtime. From the second aspect, the ontological models act as semantic data as opposed to the mere syntactic data managed in databases. An Ontology Management System (OMS) is needed to manage the semantic data such as Sesame [36] and Jena [37] frameworks.

The OMS serves as a repository where ontology models can be imported or exported. On the other hand, an OMS may need to manage the ontology models, for instance, updating the concepts and relationships. Implementing a SPARQL query engine in OMS is one of the easy ways to manage ontology models upon the demand of external use. An inference engine of the OMS can also be employed so that for retrieval information, this engine could return the entailed information derived by a reasoner from the asserted facts and axioms. In addition, OMS can be a universal access to data sources. Heterogeneous data sources can be accessed by mapping schema based on certain rules so various specific data sources can be wrapped into one source. Users or applications can simply use OMS to obtain information from different sources independent of their locations and schemas.

IV. SYSTEM COMPOSITION

In automated service-oriented manufacturing system composition it is required to coordinate service executions. One of the problems for composing systems is the description of processes in certain format that can later be executed by machines. This problem is generally addressed by using workflow or orchestration engines. Such engines are capable of executing services according to their technical descriptions. This approach works well with static service inventories. But once the changes are happening in the service inventory in result of for example equipment replacement, the process must be reconfigured. This brings a second problem: the problem of process composition. In systems with rich knowledge models, it should be possible to address the second problem by employing more abstract process descriptions, which are later mapped to particular service implementations. This section will firstly review some of applicable web standards for the system composition and then describe the composition process available within proposed set of technologies.

A. *System composition standards*

One of the standards which may be applied to describe the executable process is Web Services Business Process Execution Language (WS-BPEL) [26]. WS-BPEL defines the sequences and patterns of service executions. There exist several engines capable to process WS-BPEL description and to execute related process. The applicability of WS-BPEL for industrial automation was presented by J. Puttonen in 2008 [27]. Some other efforts were directed in another language for process description: Business Process Model and Notation (BPMN) [28]. Originally being developed for modeling processes, the BPMN 2.0 addresses the problem of execution of such compositions.

Another approach for composition is application of OWL to WS is OWL-S [29]. OWL-S aims to provide means for service discovery, invocation and composition. It describe services from technological aspects (grounding) through the description of communication patterns to the logical pre-conditions of service execution and impact of execution on the system [30], [31].

B. *Composition process in automation systems*

The service composition is a knowledge heavy process. Information is required about service inventory, equipment status, processes and requirements, schedules and others as well as relations between domains. In most of the cases it is also time consuming process.

Querying ontology is slower than direct data access, noting that the complexity of queries will affect directly to the speed of the process, being slower with more complex queries. Therefore, the load on ontology should be minimized by simplifying queries are reducing their amount when possible.

Firstly it is possible to employ the ontologies for configuration of the process executions. In such case knowledge base provides not a particular data but rather source of this data as direct data access is faster that querying. Once the system composition tool is being connected to required data source it may directly interact with it as long as configuration of the system does not change.

Beyond the access for data sources information from ontologies the engine executing an automated process might require some data not available in any component of the system separately. But considering reasoning capabilities of ontology the knowledge base may be used as source for such data. Following mentioned approach it is possible for providing the orchestration tools with required information, which can be efficiently accessed in the system.

V. VISUALIZATION

The interaction with a human is another important function of industrial systems in which web standards might be applied. Currently multiple hand-held or even wearable devices are becoming one of the most practical interface for human machine interactions. The application of web browser based user interfaces provides the benefits of being cross-platform, powerful and user-friendly framework.

Most basic standard for web-based visualization is the Hyper Text Markup Language (HTML). Besides its capabilities to represent the industrial system it provides some out-of-the-box compatibility with RESTful architecture. In fact, through its integration with JavaScript (JS), HTML may be employed to provide user with access to all WS functionality. Additionally, to enrich data driven capabilities of representation Cascading Style Sheets (CSS) and Scalable Vector Graphics (SVG) may be applied.

A. *Visualization with support of Ontologies*

Visualization applications are developed for monitoring the manufacturing system. These applications form the front end of the OKD-MES concept, by providing a graphical visualization of the system in web-enabled devices as computer, smart phones or tablets. The Manufacturing System Ontology (MSO) for OKD-MES aims to define general description for the components of manufacturing system [32]. These descriptions should also include information to generate visualization of the system based on some rules. Such rules allows mapping MSO data that is required for configuring, for example, the visualization symbols.

The visualization symbols may be simple or complex. The simple symbols adopting visualization standards like SVG can be directly represented in the knowledge base with different levels of details of the graphics. A basic SVG symbol defines the shape, position of shape (x, y), size (height, width) and style (fill, stroke and color) [33]. By modeling these features, the data required to create SVG symbols can be stored in ontology.

For supporting the visualization, the ontology must provide mapping between the real component of the system and its visualization symbol. For example, if a conveyor is located in the system and is represented by an instance of *Conveyor* class in the MSO, then, it can be mapped to its corresponding symbol instance in *Symbol* class using the "*hasSymbol*" object property. This kind of mapping offers flexibility to store the visualization related information about the components in MSO and provide the information to create visualization displays. Also, it enables a dynamic graphical visualization that reflects the changes in the system at runtime. Whenever a new device is added to the system, the ontology will be updated (e.g. via SPARQL Update for a RDF-based model) with the description of the device, which also includes visualization data. Hence, the added device appears in the visualization display making the visualization dynamic.

Another important concept to be represented in the ontology is the display composition. The display usually includes visual elements like screens, graphs, tables, maps, buttons and objects (single symbol). An element can be composed of other elements, for e.g. screen might include graphs, objects etc., and they are present at a certain position on the element. The same element can be on different position on different displays. All these concepts related to display composition should be represented in MSO to facilitate creation of visualization display.

Some visualization information may not be available during modeling and might depend on the user to some extent to provide the missing data to configure the display screen. The MSO will not contain the metadata information like position of an element, if it is not possible to pre-define such data initially. User Interface (UI) could be a solution in this case to obtain the missing information from the user. MSO is flexible enough for storing the metadata obtained from the user and together with other visualization information it is feasible the configuration of the display screen.

Therefore, the visualization of the system is supported by the ontology because it makes possible a dynamic graphical visualization, supporting extendibility and evolution of UIs. The visualization information is also represented in the knowledge base as like any other system information providing universal information representation format and simplifies system management.

VI. INTEGRATING DIFFERENT LAYERS OF AUTOMATION SYSTEMS WITH WEB STANDARDS: THE OKD-MES POTENTIALS

This section will discuss the benefit exploitation of the approaches and standards proposed before in integrated industrial automation system. The general discussion of possible benefits and risks of web standards exploitation in automation will be followed by practical outcomes of application of such standards in manufacturing systems. Firstly the most important features of the OKD-MES architecture, which is being developed in the EU project eScop, will be outlined. In fact, some of the features related to the representation of manufacturing equipment and services for OKD-MES have been recently presented in [43]. Further, some smaller use cases will be described.

A. *An overview of the OKD-MES architecture*

EU Project eScop pursues a goal of creation of a framework for OKD-MES. In OKD-MES, web standards are employed in all the levels of the system. The main goal of the project is to develop a system for providing a set of core components that facilitate the basic manufacturing functions. The core components must be capable to accommodate new services, equipment or processes being introduced to the system. Due to its distributed nature, interoperability and community driven development is a part of the core features that OKD-MES provides by employing web standards. Following Fig. 2 presents the OKD-MES architecture.

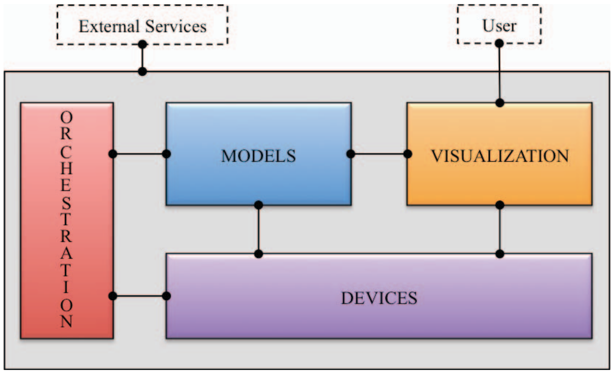

Fig. 2. The OKD-MES architecture

The integration of different layers of industrial automation systems can be implemented as shown in previous OKD-MES architecture. In fact, each one of the OKD-MES modules has been described separately in anterior article sections, which also present which web standards are employed. It should be noted that the orchestration module is referred in Section IV as the system composition, which is required for orchestrating service executions. Then, following the order in which each layer appears in this article, the OKD-MES is formed by four core modules: *Devices*, *Models*, *Orchestration* and *Visualization*.

Firstly, the *Devices* module represents the equipment that is mostly located in factory shop floors. Devices are interconnected with any other module of the architecture. The connection with the *Models* and *Visualization* modules is needed for representing and accessing at runtime to the status of the equipment. On the other hand, the connection with the *Orchestration* module is required for the execution of services, which are located in the industrial controllers.

Secondly, the *Models* module is formed by (1) the system descriptions and (2) the interfaces that permit the access, update and retrieval of any model information. Similarly to connections of *Device* module, the *Models* module is interconnected with any other one of the architecture. The connection with the *Devices* module is needed for the representation of them. In addition, *Models* module is connected with the *Visualization* module because it must be capable of answering requests in form of data that it will be needed for web UI management. On the other hand, as the hosted models also contain service descriptions of devices, *Modules* module also must be interconnected with the *Orchestration* module.

Thirdly, the *Orchestration* module is connected within *Models* and *Devices* modules. As explained before, the reason of this connection is the need of system composition to have access to all service descriptions. These descriptions include the information about operations that will be invoked for executing processes in the system.

Finally, the *Visualization* module is connected with *Models* and *Devices* modules. As explained previously, the reason of this connection is the need of having access to system status information. This information permits the *Visualization* module to create and display web UIs.

As depicted in previous Fig. 2, the system contemplates also outside interaction within (a) external services and (b) users. External services are supposed to be interconnected with the whole architecture components so they can be consumed by any module, which requests any service which is not provided inside the system. On the other hand, users are only interconnected with the *Visualization* module because they interact within the system through the displayed web UI.

B. *Exploitation of web standards: FASTory as the use case*

This section will discuss the benefit exploitation of the approaches and standards proposed before in integrated industrial automation system. The general discussion of possible benefits and risks of web standards exploitation in automation will be followed by practical outcomes of application of such standards in a specific manufacturing system, named the FAStory (shown in Fig. 3).

Fig. 3. The FASTory line in Tampere University of Technology

The FASTory, is an automated mobile phone assembly line, which has been used as a testbed in many research works of EU projects. This line is composed by a set of independent workstations. Each workstation is provided with safety equipment, a segment of the central transport system and a Selective Compliance Assembly Robot Arm (SCARA) robot. The FASTory line was retrofitted [44] within the addition of S1000 for controlling each component, which allowed the implementation of WSs.

The transportation of the work pieces in the line is performed by a closed loop modular conveyor system. Among the robotic cells, there is a pallet buffer station, which loads and unloads the pallets to the conveyor line. Another robotic cell provides material deployment on the pallet. The remaining robotic cells are providing the production operations. Aiming the reduction of the line operation cost, the real assembly process is simulated by drawing of the mobile components. Nevertheless, the complexity of the drawing process is similar to the assembly operations.

During the development of OKD-MES, the FASTory is employed as a solution demonstrator. The FASTory line is being controlled by WSs enabled devices, combining the real-time PLC like control with the exposure of the RESTful WSs. Among the exposed services, the control devices provide subscription functionality for enabling event driven behavior of the solution. The configuration and status of the system are represented in the ontology. Then, the access to the ontology is provided in form of the RESTful services by the ontology management component. The information about available services from the devices is automatically discovered by the ontology management component and exposed to the consumers. Among the most important consumers of the information about the system status and configuration is used by the orchestration component. The orchestration component executes the manufacturing process in the line according on the process description based on BPMN. Finally, the user interactions for the system in general are being provided by the visualization component. Besides the four basic modules presented in Fig. 2, the additional MES functions are provided by external services. Particularly, the important functions provided by the external services are dispatching and scheduling. The scheduling service provides the high level decision to which order to assign incoming pallet. The dispatching function handles lower level decision which if any operations is to be performed in the station when the pallets reaches a workstation.

By the application of the web standards on all levels, the developed system enables seamless integration of all of its components and other web based solutions, including ERP. Dynamically updating representation of the system status, based in availability and configuration of the shop-floor devices, enables easy re-configurability of the system hardware as well as the introduction of new products in manufacturing systems. The orchestration with decision support from MES functions is capable to handle repetitive preconfigured modules in various configurations. Additionally, the managers, operators, customers and other manufacturing stakeholders have direct controlled browser access to the data generated in the manufacturing process. Furthermore, the dynamically generated screens permits to visualize such basic system data. Concluding, the FASTory demonstrator provides significant level of adaptability to configuration changes on all levels of the system (representation, control, monitoring) in case of exploitation of predefined modules or their variations, within implementing web standards.

VII. CONCLUSION

This paper firstly presents an overview of different web standards that are currently being employed in ongoing research works in the industrial automation domain. Then, the article shows the potentials of web standards for automation control in manufacturing systems within results achieved in the ongoing EU project eScop.

The promising benefits of the suggested solution are discussed within a presented framework: the OKD-MES. This particular application for the industrial domain, demonstrates how the integration of the different automation levels is possible thought the employment of web standards and the synergy between cyber and physical systems, realizing the implementation of CPS in manufacturing systems.

Finally, the authors argue, that the use of web standards, especially the ones, which are directly supported in the web browsers, gives greater chances for applications developed following these standards to be compatible with the future technologies. Thus, it may reduce the efforts needed to update or integrate the application to its evolving environment in the future.

ACKNOWLEDGMENT

The work reported in this paper was partially supported by the ARTEMIS Joint Undertaking under grant agreement n° 332946 (eScop, Embedded systems for service-based control of open manufacturing and process automation).

REFERENCES

[1] E. COMPUTING, ‘Cyber-physical systems’, 2009.

[2] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, ‘Cyber-physical systems: the next computing revolution’, in *Proceedings of the 47th Design Automation Conference*, 2010, pp. 731–736.

[3] J. M. Mendes, A. Rodrigues, P. Leitao, A. W. Colombo, and F. Restivo, ‘Distributed Control Patterns using Device Profile for Web Services’, in *Enterprise Distributed Object Computing Conference Workshops, 2008 12th*, 2008, pp. 353–360.

[4] A. Cannata, M. Gerosa, and M. Taisch, ‘SOCRADES: A framework for developing intelligent systems in manufacturing’, in *IEEE International Conference on Industrial Engineering and Engineering Management, 2008. IEEM 2008*, 2008, pp. 1904–1908.

[5] H. Bohn, A. Bobek, and F. Golatowski, ‘SIRENA - Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains’, in *International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006*, 2006, pp. 43–43.

[6] A. L. Juha Puttonen, ‘A Semantic Web Services-based approach for production systems control’, *Adv. Eng. Inform.*, no. 3, pp. 285–299, 2010.

[7] A. Lobov, F. U. Lopez, V. V. Herrera, J. Puttonen, and J. L. M. Lastra, ‘Semantic Web Services framework for manufacturing industries’, in *IEEE International Conference on Robotics and Biomimetics, 2008. ROBIO 2008*, 2009, pp. 2104–2108.

[8] J. Puttonen, A. Lobov, and J. L. Martinez Lastra, ‘Semantics-Based

Composition of Factory Automation Processes Encapsulated by Web Services', *IEEE Trans. Ind. Inform.*, vol. 9, no. 4, pp. 2349–2359, Nov. 2013.

[9] B. Ramis, L. Gonzalez, S. Iarovyi, A. Lobov, J. L. Martinez Lastra, V. Vyatkin, and W. Dai, 'Knowledge-based web service integration for industrial automation', in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 733–739.

[10] N. A. N. Lee, L. E. G. Moctezuma, and J. L. M. Lastra, 'Visualization of Information in a Service-Oriented Production Control System', in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, 2013, pp. 4422–4428.

[11] S. Iarovyi, J. Garcia, and J. L. M. Lastra, 'An approach for OSGi and DPWS interoperability: Bridging enterprise application with shop-floor', in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 200–205.

[12] S. Sucic, B. Bony, and L. Guise, 'Standards-compliant event-driven SOA for semantic-enabled smart grid automation: Evaluating IEC 61850 and DPWS integration', in *2012 IEEE International Conference on Industrial Technology (ICIT)*, 2012, pp. 403–408.

[13] J. Imtiaz and J. Jasperneite, 'Scalability of OPC-UA down to the chip level enables ";Internet of Things ";, in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*, 2013, pp. 500–505.

[14] 'OASIS Devices Profile for Web Services (DPWS)'. [Online]. Available: http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01 [Accessed: 16-Jul-2015]

[15] G. Marco, L. Fumagalli, A. Lobov, and J. L. Martinez Lastra, "Open Automation of Manufacturing Systems through Integration of Ontology and Web Services," 2013, pp. 198–203.

[16] S. Karnouskos and V. Somlev, 'Performance assessment of integration in the cloud of things via web services', in *2013 IEEE International Conference on Industrial Technology (ICIT)*, 2013, pp. 1988–1993.

[17] V. Gilart-Iglesias, F. Maciá-Pérez, D. Marcos-Jorquera, and F. J. Mora-Gimeno, 'Industrial Machines as a Service: Modelling industrial machinery processes', in *2007 5th IEEE International Conference on Industrial Informatics*, 2007, vol. 2, pp. 737–742.

[18] M. Lanthaler and C. Gütl, 'Hydra: A Vocabulary for Hypermedia-Driven Web APIs.', in *LDOW*, 2013.

[19] 'Hydra Core Vocabulary'. [Online]. Available: http://www.w3.org/ns/hydra/spec/latest/core/. [Accessed: 16-Jul-2015].

[20] T. R. Gruber, 'A translation approach to portable ontology specifications', *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.

[21] 'OWL Web Ontology Language Semantics and Abstract Syntax'. [Online]. Available: http://www.w3.org/TR/owl-semantics/. [Accessed: 16-Jul-2015]

[22] 'RDF 1.1 Concepts and Abstract Syntax'. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. [Accessed: 16-Jul-2015].

[23] 'RDF Schema 1.1'. [Online]. Available: http://www.w3.org/TR/rdf-schema/. [Accessed: 03-Jul-2015].

[24] 'SPARQL Query Language for RDF'. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/. [Accessed: 05-Jul-2015].

[25] I. Kollia, B. Glimm, and I. Horrocks, 'SPARQL query answering over OWL ontologies', in *The Semantic Web: Research and Applications*, Springer, 2011, pp. 382–396.

[26] 'OASIS Web Services Business Process Execution Language (WSBPEL) TC | OASIS'. [Online]. Available: https://www.oasis-open.org/committees/ tc_home.php?wg_abbrev=wsbpel. [Accessed: 16-Mar-2015].

[27] J. Puttonen, A. Lobov, and J. L. M. Lastra, 'An application of BPEL for service orchestration in an industrial environment', in *IEEE International Conference on Emerging Technologies and Factory Automation, 2008. ETFA 2008*, 2008, pp. 530–537.

[28] 'BPMN 2.0'. [Online]. Available: http://www.omg.org/spec/BPMN/2.0/. [Accessed: 10-May-2015].

[29] 'OWL-S: Semantic Markup for Web Services'. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/#6. [Accessed: 25-May-2015].

[30] D. Redavid, L. Iannone, T. Payne, and G. Semeraro, 'OWL-S Atomic Services Composition with SWRL Rules', in *Foundations of Intelligent Systems*, A. An, S. Matwin, Z. W. Raś, and D. Ślęzak, Eds. Springer Berlin Heidelberg, 2008, pp. 605–611.

[31] H. Li, L. Zhang, and R. Jiang, 'Study of manufacturing cloud service matching algorithm based on OWL-S', in *Control and Decision Conference (2014 CCDC), The 26th Chinese*, 2014, pp. 4155–4160.

[32] 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. Springer Berlin Heidelberg, 2014, pp. 192–200.

[33] 'Scalable Vector Graphics (SVG) 1.1 (Second Edition)'. [Online]. Available: http://www.w3.org/TR/SVG11/. [Accessed: 16-Mar-2015].

[34] "SPARQL 1.1 Update." [Online]. Available: http://www.w3.org/TR/sparql11-update/. [Accessed: 11-July-2015].

[35] "Semantic Web Architecture - Introduction to ontologies and semantic web - tutorial." [Online]. Available: http://obitko.com/tutorials/ontologies-semantic-web/semantic-web-architecture.html. [Accessed: 26-Jul-2015].

[36] "Sesame." [Online]. Available: http://rdf4j.org/. [Accessed: 26-Jul-2015].

[37] "Apache Jena - Home." [Online]. Available: https://jena.apache.org/. [Accessed: 26-Jul-2015].

[38] A. W. Colombo, S. Karnouskos, and T. Bangemann, "Towards the Next Generation of Industrial Cyber-Physical Systems," in *Industrial Cloud-Based Cyber-Physical Systems*, A. W. Colombo, T. Bangemann, S. Karnouskos, J. Delsing, P. Stluka, R. Harrison, F. Jammes, and J. L. Lastra, Eds. Cham: Springer International Publishing, 2014, pp. 1–22.

[39] B. B. Jay Lee, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *SME Manuf. Lett.*, 2014.

[40] S. Iarovyi, José L. Martinez Lastra, Rodolfo Haber, and Raúl del Toro, "From artificial cognitive systems and open architectures to cognitive manufacturing systems", in *2015 13th IEEE International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1225–1232.

[41] "SWRL: A Semantic Web Rule Language Combining OWL and RuleML" [Online]. Available: http://www.w3.org/Submission/SWRL/. [Accessed: 28-July-2015].

[42] B. Ramis Ferrer, B. Ahmad, A. Lobov, D. Vera, José L. Martinez Lastra, R. Harrison, "A knowledge-based solution for automatic mapping in component based automation systems", in *2015 13th IEEE International Conference on Industrial Informatics (INDIN)*, 2015, pp. 262–268.

[43] S. Iarovyi, B. Ramis Ferrer, X. Xiangbin, A. Sampath, A. Lobov, José L. Martinez Lastra, "Representation of manufacturing equipment and services for OKD-MES: from service descriptions to ontology", in *2015 13th IEEE International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1069–1074.

[44] L. Gonzalez, J. Jokinen, C. Postelnicu, J. M. Lastra, "Retrofitting a factory automation system to address market needs and societal changes," Industrial Informatics (INDIN), 2012 10th IEEE International Conference, pp.413- 418, 2012.

[45] B. Ramis Ferrer, B. Ahmad, A. Lobov, D. Vera, José L. Martinez Lastra, R. Harrison, "An approach for knowledge-driven product, process and resource mappings for assembly automation", in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 1104-1109.

[46] S. Iarovyi and X. Xiangbin, Developing Open Knowledge-Driven Manufacturing Execution System (2015). In: Strzelczak S., Balda P., Garetti M., Lobov A. (Eds.), *Open Knowledge Driven Manufacturing and Logistics - the eScop Approach*, Warsaw University of Technology Publishing House, pp. 23-50, 2015. ISBN 978-83-7814-440-3.

[47] Strzelczak S., Ontology-Aided Manufacturing and Logistics (2015). In: Strzelczak S., Balda P., Garetti M., Lobov A. (Eds.), *Open Knowledge Driven Manufacturing and Logistics - the eScop Approach*, Warsaw University of Technology Publishing House, pp. 23-50, 2015. ISBN 978-83-7814-440-3.