Including Human Tasks as Semantic Resources in Manufacturing Ontology Models

Borja Ramis Ferrer, Wael M. Mohammed, Andrei Lobov, Amalia Moreno Galera, Jose L. Martinez Lastra
Tampere University of Technology, Tampere, Finland
{borja.ramisferrer, wael.mohammed, andrei.lobov, amalia.morenogalera, jose.lastra}@tut.fi

Abstract—**Human operators play an important role especially in high added value manufacturing. The use of knowledge representation for decision making at runtime becomes more important. Modern automatic control systems should be capable to seamlessly include and assist operators. This paper describes how to represent information on operator's skills and include it into service-oriented orchestration approach for a production line. This approach allows operator to act not as a passive element, but as active asset for the orchestration engine to consider in production plans.**

Keywords—Industrial automation; Ontologies; human-machine interaction; knowledge-driven systems.

I. INTRODUCTION

Currently, the industrial community is working towards the definition and design of systems that operate in the next industrial revolution or, commonly referred as, the Industry 4.0 [1], [2]. This will enhance the industrial automation field within the implementation and deployment of Cyber-Physical Systems (CPS) [3]–[6]. These systems integrate the cyber and physical domains and must enable the interoperability of local and/or remote resources.

An important issue when systems are automated, mostly in environments wherein humans are involved, is to consider the role of persons that will interact with such systems. Frequently, the considerations cause the development of different solutions for specific situations and needs. For example, engineers develop systems that ensure the safety conditions at locations wherein humans and machines share tasks [7]–[9]. On the other hand, focusing not only in safety but also in the enhancement of human capabilities, software engineers develop friendly and intuitive interfaces that allow operators to monitor or even control efficiently industrial equipment [10]–[12]. Finally, human capabilities are considered in some research works about human and robot interaction [13], [14].

This kind of considerations must be also accounted for new systems that operate following the Industry 4.0 concept. The fact is that, due to the incessant increment of numerous tasks that are automatically performed by machines, industrial workplaces are getting dehumanized. Therefore, there is a clear challenge on keeping human operators in the control loop, being still important assets for an efficient industrial productivity. Moreover, high level education is also affected by the next industrial revolution [15]. In the presented scope, this article proposes the inclusion of descriptions about human tasks and skills in the semantic models that are already utilized as knowledge repositories of CPS e.g., [16], [6], [17], [18]. In practice, semantic models, such as ontologies [19], [20] can be employed by service orchestrators [21] and other support systems [22] for planning and ordering the execution of operations. However, human skills are not always included in knowledge repositories because systems only require to know if the outcome of human operations is produced. The incorporation of such information may be beneficial for multiple issues as e.g., i) enhancing the quality of the production within performance analysis, ii) finding more efficient manners of performing human operations or iii) consider middle-step actions in the performance of one operation that are not directly visible by control engines. This research work has been performed during the execution of the Cloud Collaborative Manufacturing Networks (C2NET) project[1], which aims the enhancement of supply chain assets. Then, consideration of humans (i.e., resources) that are performing tasks in the value chain of products might be useful for the main objective of the project.

Then, the main objective of this paper is to present a formal model that permits the modelling of operations that are performed by human operators. The ultimate goal of this paper goes beyond including role-based information of operators in models that are only used for adapting and customizing HMIs [23]. The presented approach considers also i) which type of actions, ii) how they are performed and iii) the availability of operators. As it can be seen in further sections, this permits orchestrators to consider, assign and control the operations that are performed by workers at shop floors. In order to present a proof of the concept, this research work implemented some parts of the approach in a retrofitted assembly line [24] which is referred as the FASTory line. This system has recently incorporated a manual workstation. Then, operators must perform specific tasks in a workplace which is attached to the production line. The operator tasks and the FASTory line are partially modelled in order to show how the approach could be implemented in a real scenario.

The rest of the paper is structured as follows: Section II presents a literature review and industrial practices on the scope of the Industry 4.0. Section III presents an approach for including information about human operations in semantic models. Then, Section IV presents an implementation of the presented approach, based on a real industrial case. Finally, the results, potential and further work of the approach are concluded in Section V.

---

[1] http://c2net-project.eu/

II. INDUSTRY 4.0 AND KNOWLEDGE-DRIVEN SERVICE ORIENTED SYSTEMS

A. Industry 4.0

Industry 4.0, or fourth industrial revolution, aims to bring computerization and digitalization of various aspects in the factories. Thanks to the use of Cyber-Physical Systems (CPS) capable of communicating and actively participating in the decision making at the factory floor, it should become possible for the production system as a whole to dynamically and rapidly adapt to the changes in demands and/or in the capabilities of a production line. Furthermore, Industry 4.0 sets four design principles that include interoperability, transparency, assistance (for human operators) and decentralization of decision making. In order to follow these principles, one could require proven standards. Web standards[2] developed and broadly tested can be seen as good candidates for executing outlined design principles.

One of the common paths is to propose some multi-layered architecture having dedicated levels for decision making. For example, [25] proposes network virtualization approach to improve overall operations of network environment. The authors claim that network management, e.g. between different department, is one of the prerequisites for building smart systems. On the other end, the mere ability to connect a robot using TCP/IP sockets [26] sometimes allows speaking about Industry 4.0. That fact could show the weight of the Internet protocol as a key enabler for interoperability and transparency of next generation of industrial systems.

As for the “smartness” or more advanced decision-making capabilities, a number of research works were performed in particular in the fields of prediction and maintenance. For instance, [27] came with the proposal of the methodology to analyse sensors data to detect anomalies in a machine. Neural networks were used to implement the solution. Still how that method can be scaled and integrated for a different system somewhat open. In fact, one could think that it is an attempt to fit an application of Neural networks into currently trendy Industry 4.0. However, sensor data, and especially when it comes in large quantities, falls under concept of Big Data. [28] shows data-driven approach for Quality Management for industrial processes. The presented approach also makes use of Web standards and in particular RESTful web services. In principle that can give an idea how various system components can be integrated and potentially rearranged into new setups if demanded by changing system goals.

On the other hand, the use of virtual reality in the context of Industry 4.0 is proposed in [29]. This can be used to support learning activities and prevent braking actual machines as the operational steps can be first checked in the virtual environments. Still, no elaborated discussion in the mentioned source is given for integration of the approach with the actual field devices or industrial controllers besides of having the cloud networks integrating all the sensors and actuators into the Internet of Things (IoT). The role of the Internet and mobile networking is mentioned in [30] to support integration of various functionalities such as data collection, analysis, processing, online monitoring, etc.

In order to achieve desired design principle and reduce maintenance and reconfiguration costs the same knowledge representation framework on top of proven architectures should be employed.

B. Knowledge-driven service-oriented system architectures

Service-oriented architecture can be depicted as a triangle, which vertices are service provider, service broker and service requestor. The service provider can first register its capabilities at the service broker and the service requestor can use the service broker to search for the necessary web services. Once the required service is found, the service requestor can directly invoke it on the service provider. There can be some variants to the described triangle. For example, the discovery process for Device Profile for Web Services (DPWS)[3] can go via multicasting. However, the multicasting is only applicable to the local networks. As soon as internetworking is required, service brokers are unavoidable.

Discovery of the services can go by basic type alias names. For example, user or application can discover the “conveyor” service just by that English language word. It assumes that system designer should have some prior knowledge about “conveyors” so that system can use those at the run time. In order to allow dynamic discovery of services semantic web services (SWS) should be used. SWS mean that there are knowledge representation languages used that can be automatically processed to check on service capabilities. In particular Web Ontology Language (OWL)[4] can be used for such purposes. The application of knowledge-driven approach for Manufacturing Execution Systems was described in [31]. More precisely, [32] shows how the dynamic composition of web services for orchestration purposes can be implemented. Multiple orchestrators each handling its own service workflow (e.g. fulfilling its product’s needs) can be run in parallel. Still, it is important to investigate how the human operators can be in a best way integrated into the service workflows.

III. INCLUDING HUMAN TASKS IN MANUFACTURING ONTOLOGIES

This section presents an approach to enrich existing ontologies with semantic resources about operations which are performed by humans. This model is inspired on proposed ontologies [33], [34] to be employed by engines that control systems in the industrial domain. In particular, [33] presents an implementation of a knowledge-based industrial system that employs a generic ontology for modelling assembly lines. The presented model is queried and updated by an orchestrator in order to execute operations in such production system. On the other hand, [34] presents an implementation that shows the model coupling of different but connected domains. Fundamentally, it shows how product, process and resource domains can be mapped within i) semantic rules description and ii) the inference of reasoning engines. Furthermore, the model designed by the Manufacturing Enterprise Solutions Association (MESA)[5] not only presents a set of functions that describe several categories of the production operations management; but

---

[2] https://www.w3.org/community/kiss/

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

[4] https://www.w3.org/OWL/

[5] http://www.mesa.org/en/index.asp

also how they are linked with other aspects of manufacturing enterprises [35]. In this scope, some parts of the *Production Management* include activities as *resource allocation and status*, *performance analysis*, *labour management* or *quality management*. Among others, these activities can benefit of approaches that implement knowledge-driven service oriented architectures with an ontology that describes all resources of the system [4], [21], [36].

The concept of the presented research in this article is similar to aforementioned research works in terms of system control and ontology implementation. However, this is extended with the inclusion of semantic human resources. This permits the orchestration engines to consider human operations and include them in their assembly planning. As shown in Fig. 1, this approach tends to employ several coupled ontologies as a knowledge provider to the manufacturing system. Then, an orchestrator engine will act as the brain of the system that deals with i) semantic descriptions from the knowledge layer and ii) events generated at the factory shop floor layer.

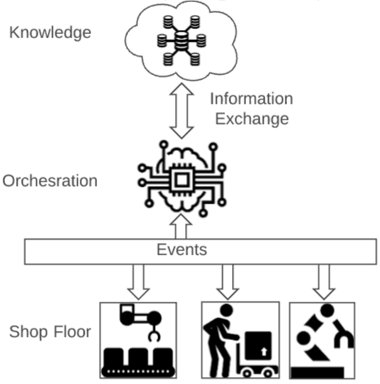


Fig. 1. The concept

A. Main elements for the ontology model

The principal objects and their relationships for the OWL model are depicted in Fig. 2. As it can be seen in the presented UML class diagram, there are three main packages: *Product Ontology*, *Process Ontology* and *Resource Ontology*. As described in [34], these are the different domain ontologies that are needed to model a production system, such as an assembly line. Then, each ontology includes the knowledge description of concepts that belong to such domain.

Firstly, the *Product Ontology* contains the product variants that may be manufactured by the production system. This is done throughout the *Product* class. Each instance of this object represents a different type of product.

Secondly, the *Process Ontology* includes the information that is needed to control and execute the operations that are performed at the factory shop floor. The *Recipe* object includes instances that will be linked to *Task* individuals. Each recipe is to be created and queried by the orchestrator engine that, in this manner, it will know which task is the next one to perform. The recipes include set of steps that are needed for manufacturing a specific product variant. In the proposed design, the process of products is decomposed in *Task* and *Operation* class instances. Furthermore, *Route* and *Production* types are subclasses that define operation for i) routing pallets through the system and ii) executing operations for product manufacturing, respectively. Fundamentally, tasks are general needs for a product to be manufactured e.g., *transfer*, *drill*, *stop* or *load*. Then, *Operation* instances are the actions that are performed in the production system, connected and mapped to specific system resources and services. For example, if a pallet must be transferred from a zone A to a zone B of a transport system, the task could be defined as *transfer* and the operation as *transfer_A_to_B*. Obviously, this will depend on the manner in which the designer implements i) the model and ii) orchestrator service descriptions. In any case, *Operation* instances are the ones to be linked with system resources of the production systems because only specific resources are capable of performing their corresponding operations. In addition, the product variant will be linked to required set of operations within the object property *requiresOperation*. This will be asserted by the system designer, following the requirements of product variants.

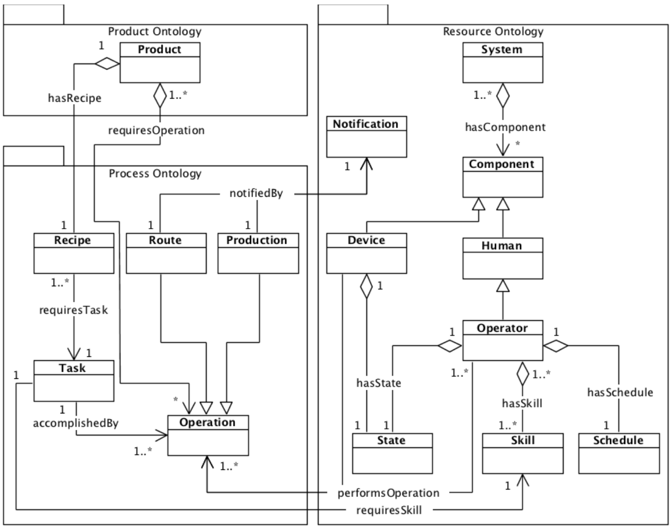


Fig. 2. Hierarchy and relationships of ontology classes

Thirdly, the *Resource Ontology* describes the information of any resource that may be useful for controlling and monitoring processes of the production system. Principally, the *System* class will contain instances for each production system that is described in the ontology. Then, each instance of a system will be linked to the components that perform operations for it. As it can be seen in Fig. 2, the class diagram shows the generalization between the superclass *Component* and the two subclasses *Device* and *Human*. In addition, *Operator* is shown as a subclass of *Human*. Basically, as the main aspect of the presented approach, other research works using ontologies as a centric model for orchestration of services [21], [33] only consider as resources the industrial equipment, which is included in the *Component* class. However, this research work proposes to include human operators (i.e. *Operator* instances) as other resources that may be considered by orchestration engines.

Both devices (i.e. industrial equipment) and human operators will have i) are represented as a class that describes their state within *State* instances and ii) a link with *Operation* instances throughout the *performsOperation* object property. The state instances defined according to the CAMX Equipment State Diagram (included in the IPC-2541). Moreover, each human operator will have a connection within *Skill* instances which are skills that operators are capable to perform and are, in turn, linked to *Task* instances. It could be claimed that devices might be also related to *Skill* instances. However, the presented model does not consider such design fact because inspiring models have already implemented the direct mapping between resources and operations. Anyway, this option could be also implemented for devices as it is proposed for human operators. Finally, the human operator is also mapped one-to-one with *Schedule* instances. This has been added into the model for two main reasons. First, the orchestrator engine will be capable of deciding to who a manual task can be assigned. Second, for further research, this can be later exploited by data mining engines in order to improve the efficiency of processes and conclude best performances at workplace.

In addition, the *Notification* object incorporates notifications that are triggered whenever an operation is performed. This includes both machine and human operations. In fact, notifications are the main semantic resource that the orchestrator engine will use in order to determine that an operation has been performed. Then, the next operation listed in the recipe will be requested. This is further described in next section.

Finally, it should be noted that following the approach presented in [34], the three domain ontologies are all merged into one ontology. This ontology is referred as the PPR (i.e., Product Process Resource) ontology. To achieve such integration, all models are imported into the PPR model and, in the latter, the linking object properties are added. In addition, if desired, Semantic Web Rule Language (SWRL)[6] rules can be added in order to automate certain links within the inference of implicit knowledge [34]. This is anyway an optional practice. On the other hand, another aspect to notice in proposed model is that the service description is not attached. Basically, this section intends to present the relevant classes of the approach for the inclusion of human skills and their relationships with other domain ontology instances. Some possible strategies for incorporating the service description can be found in [21], [37].

B. *Model interaction*

The ontology model provides system knowledge description to the orchestrator. For example, subscription of resource events, resource operations and services, and mapping between operations and notifications. In this subsection, two UML diagrams are presented. The first one depicts the interaction between the orchestrator and the knowledge model during the creation of new recipes when new orders arrive. On the other hand, the second diagram illustrates the interaction between the factory resources, the orchestrator and the knowledge model during the production period.

This approach employs an event driven methodology for invoking operations and exchanging information. Thus, all components will react whenever an event is present. As depicted in Fig. 3, the initialization process requires a factory engineer to populate the PPR model according to the production system entities. This includes all classes in the PPR model except the *Recipe* and the *State* classes. These two objects are populated during the system runtime. After that, the orchestrator requests all available notifications related to operations in order to be notified whenever an operation is ended.

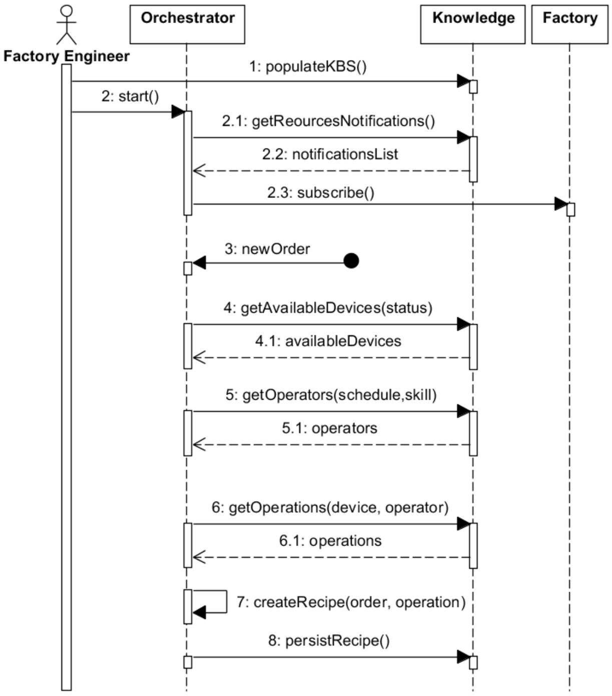


Fig. 3. Creating recipe sequence diagram

After the manufacturing system is initialised, the orchestrator starts listening to the *"new order"* event. When this event is triggered, orchestrator creates the recipe in the product ontology model. As seen in Fig. 2, the orchestrator requests information, such as i) the availability of devices with respect to their state or ii) operator with respect to their skills, schedule and skills. This results of two lists: i) the available devices and ii) proper operators, respectively. Then, the orchestrator requests the list of operations that available devices and operators are capable to perform. Finally, the orchestrator creates the recipes that not only matches the requirements in the order; but also matches the operations that the operations provides. As discussed in the model description, the recipe is a list of tasks that are mapped to operations.

After the order is persisted in the manufacturing system, causing the creation of recipes, the production process is ready to start. Simply, the production process can be seen as a list of tasks that compose a recipe. Each task leads to certain operations in the factory shop floor. Therefore, the orchestrator is required to subscribe to all events in the factory shop floor in order to receive required notifications. With the information collected from the notifications and the information in the tasks, the orchestrator is able to execute the operation in the factory resources. The sequence diagram shown in Fig. 4 represents the interaction during the production process.

---

[6] https://www.w3.org/Submission/SWRL/

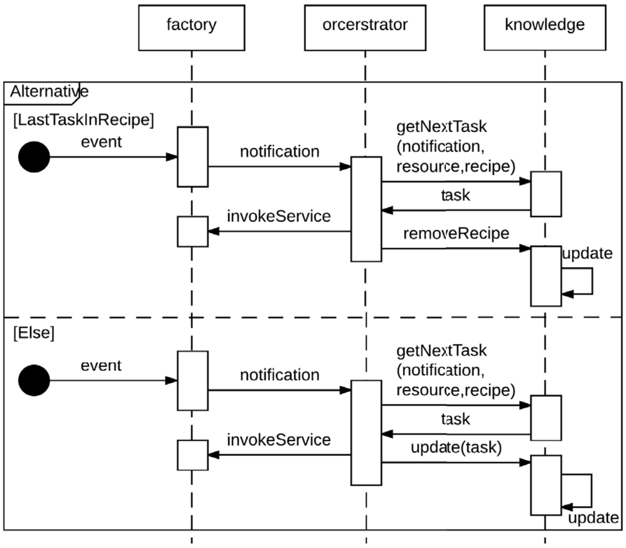


Fig. 4. Production interactions sequence diagram

As it can be seen in previous diagram, the orchestrator receives notifications from the factory shop floor after the finalization of each invoked operation. Thanks to the event driven methodology, the orchestrator is able to synchronize the invocation of the operations according to the created recipe. In this context, the orchestrator first checks if the next operation in the recipe is the last one. If so, the orchestrator deletes the recipe, which means the product has been manufactured. Otherwise, the orchestrator updates the recipe by removing the invoked operation.

C. *Approach usability*

Previous subsections show i) an approach for modelling a production system considering human resources and ii) how to implement the interaction between the designed model and an orchestration engine for controlling production systems. Mainly, this approach may be employable by organizations that:

a) Own an assembly line that requires operations which are performed by both machine and human operators,
b) Implement web service technologies for executing industrial equipment operations and
c) Are aligned or open to implement knowledge-driven service oriented system architectures.

There are many different kind of manual operations that operators still perform at shop floors. Nevertheless, not all the systems are the same in terms of physical layout and performance. Mainly, there are two common ways of reporting when a manual operation is finished: i) by sending a notification throughout a device, such as a button, or ii) by sensing specific objects that are picked and placed by humans at specific locations [38]. Besides detecting such kind of manual operation end, this approach permits to exploit descriptions about the skills of humans and, hence, the creation of information to feed other systems that permit e.g., data mining or data analysis. In fact, as discussed, even the enrichment of the model within SWRL rules may allow semantic reasoners to infer explicit knowledge. Moreover, this is feasible to be created, monitored and managed remotely because ontologies can be implemented in RDF-based languages which are W3C standard that can be queried e.g., within SPARQL over HTTP[7].

IV. USE CASE

This section presents an implementation that has been performed for proving the concept. Below, the presented industrial scenario allows this research work to perform first tests about how to consume and update the human operator model information. Besides the production system, some parts of the model, its usability and the interface of the system are also demonstrated.

A. *The FASTory line*

As an educational organization, Tampere University of Technology provides several opportunities to students and researches to interact with real industrial equipment. One of such test-beds is located at Factory Automation Systems and Technologies Laboratory (FAST-Lab) premises. Among several real systems, the FAST-Lab owns a discrete assembly line, which is known as the FASTory line.

The FASTory line demonstrates the assemble of mobile phones by drawing three parts (frame, screen and keyboard) on papers located on special machined pallets. As shown in the upper part of Fig. 5, the FASTory line consists of 12 workstations.

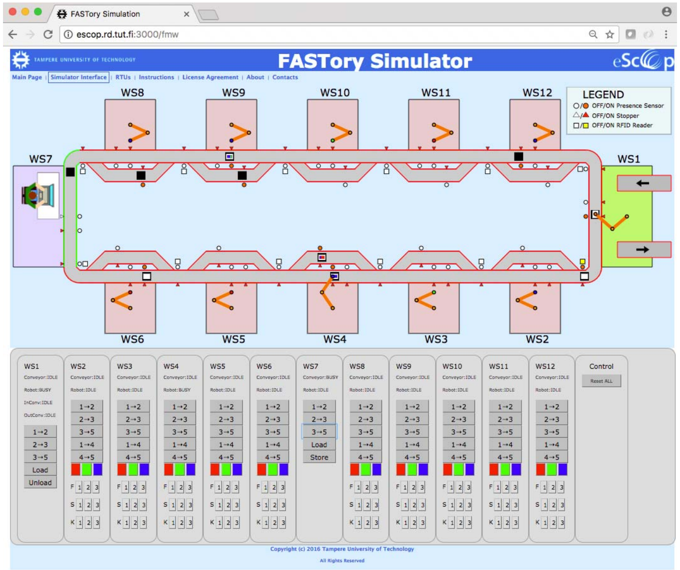


Fig. 5. The FASTory simulator in action

In recent configuration, a manual workstation has been installed for loading and unloading pallets to the central transport system. This workstation is labelled as WS7 and is also presented in Fig. 6. On the other side, the WS1 is dedicated for loading and unloading papers to and from the pallets. The remaining 10 workstations are similar cells that are capable of performing drawing processes. Particularly, WS2–WS6 and WS8–WS12 include a SCARA robot that draws using different pen colours which are available in pen feeders. In addition, these stations contain a double path conveyor. Meanwhile one route is

---

[7] https://www.w3.org/TR/sparql11-http-rdf-update/

used for the drawing purposes; the other one is used for bypassing the cell since the topology of the line is designed as closed loop. Basically, this feature eliminates the blockage if any cell is busy and a pallet can be routed to a next available workstation.

Besides the physical real line, the FAST Lab developers created a web-based tool for monitoring and controlling the real line via REST interface. This application is known as the FASTory simulator[8], previously presented in [39]. In fact, Fig. 5 shows only the line visualization of the simulator. The simulator mimics the same functionalities of the real line in a virtual environment. It runs on the university servers for allowing remote access to both international partners and students. Then, the FASTory simulator is a web-accessible tool for users to test operations of the real assembly line. The major advantage of this system is that the simulator is a safe environment for both humans and machines wherein the performance of the overall physical system can be virtually tested and validated.

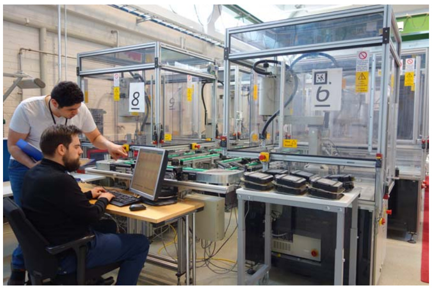


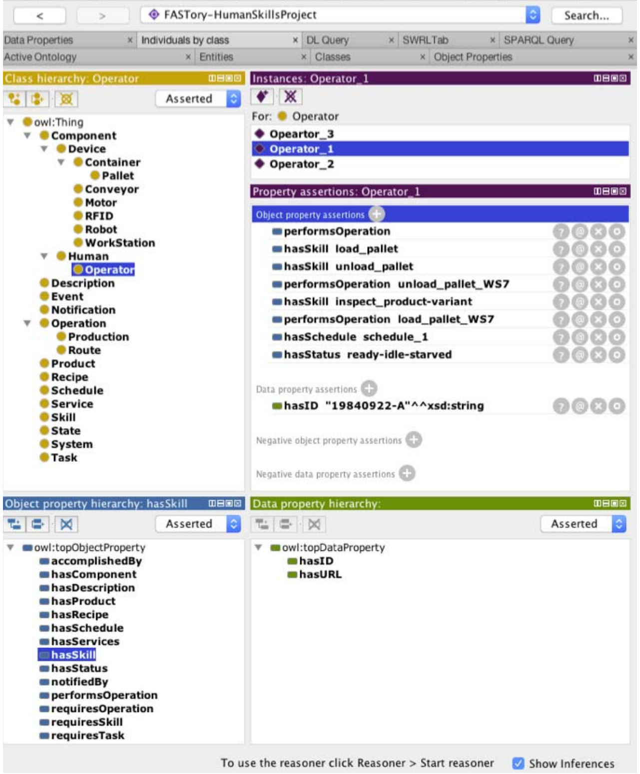
Fig. 6. The manual workstation of the FASTory assembly line

FASTory line workstation operations are represented within the INICO S1000[9] controllers, which are web service-based Remote Terminal Units (RTUs). Each station includes two RTUs, one for controlling the robot and another for controlling the conveyor. These RTUs host RESTful services that can be invoked for performing operations. In addition, the RTUs issue pre-defined notifications for the subscribers in order to inform about events and change of status. Since the drawing workstations are identical, they perform the same operations in the related equipment. More precisely, the conveyor RTUs are able to perform the operation *TransZone12*, which moves pallets from *zone1* to *zone2*. Similarly, *TransZone23*, *TransZone35*, *TransZone14* and *TransZone45*. As it can be noticed in Fig. 5, each zone is represented in the FASTory Simulator interface within a small circle. On the other hand, the Robot RTUs host the *Draw* operation for drawing the aforementioned mobile parts. Also, it Robot RTUs include *ChangePen* operation for changing the pen, which will permit drawing mobile parts with different colours. More detailed information can be found in the simulator instructions[10].

On previous configuration, the WS7 used to be an automated storing workstation with a Cartesian robot [40]. However, it has been removed from the line due to the lack of space in the laboratory facilities. Such transformation forced researchers to exchange the workstation automatic operation with a manual one. This included installation of a computer with screen for allowing the operator to interact with the system. Besides, a web server is dedicated to this workstation allowing the operator to receive http RESTful requests from the orchestrator and other devices of the FASTory line.

B. Including human resources in the FASTory model

In order to implement the described approach in Section III, an ontology that describes the industrial scenario must be developed. As it can be seen in Fig. 7, an ontology has been implemented within Protégé[11] by following the class diagram depicted in Fig. 2.



performsOperation
hasSkill load_pallet
hasSkill unload_pallet
performsOperation unload_pallet_WS7
hasSkill inspect_product-variant
performsOperation load_pallet_WS7
hasSchedule schedule_1
hasStatus ready-idle-starved

hasID "19840922-A"^^xsd:string

Fig. 7. The implemented FASTory model within Protégé

The Protégé interface depicts at left hand side the class hierarchy of the model, which, in comparison to the structure shown in Fig. 2, shows that the *Device* class is extended. Basically, the FASTory line includes three several component types, such as *Container*, *Motor*, *RFID*, *Conveyor*, *Robot* and *WorkStation*. In any case, as these objects are subclasses of the *Device* type, they inherit the object property *performsOperation* and, hence, can be linked to *Task* instances, which are included in the *Process Domain*. In addition, the components that perform operations are linked service concepts i.e., *Description*, *Event* and *Service*. The service description is based on the one shown in [4].

On the other hand, Fig. 7 shows that all; product, process and resource, domain classes are in the same model. This occurs because the three domain ontologies, which are represented as packages in the UML class diagram of Fig. 2, are imported in the shown model. In any case, each class keeps its own

---

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

[9] http://www.inicotech.com/s1000_overview.html

[10] http://escop.rd.tut.fi:3000/instructions

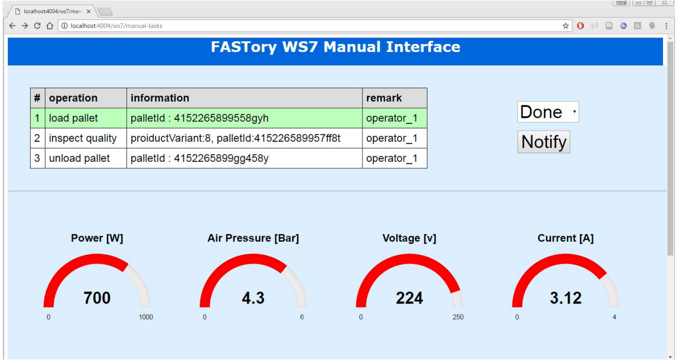
[11] http://protege.stanford.edu/

Internationalized Resource Identifier so this must be considered when querying the model. In addition, Fig. 7 shows the object and data property hierarchies. As it can be seen, the object properties shown in Section III are added. Moreover, datatype properties are values to be inserted in certain instances of the model in order to describe needed variables for the orchestration engine. As an example, the ontology editor interface includes *hasURL* that a datatype property string value with the required URL for REST operation execution [4].

Finally, the interface of the ontology editor shows a list of instances which are the ones of the real FASTory line. In fact, they are interrelated following the object properties shown in Fig. 2. As an example, Fig. 7 depicts how a human operator is linked to his/her schedule, certain skills and their corresponding operations. In addition, as the ontology is updated on system runtime, the operator is described to be in a specific state. This state will be changed by the orchestrator according to the performance of the operator.

C. A tentative interface for the FASTory WS7 operation

As mentioned before, the FASTory line was considered as fully automated assembly line. However, a change in WS7 forced the inclusion of a manual workstation. During the implementation of the eScop project[12], WS7 was used to host an interface for allowing users to monitor the workstation and to interact accordingly. At that time, the user was able just to monitor the status of the line using dedicated terminals. Following Fig. 8 depicts a tentative interface for the described approach and its implementation for the manual control and monitoring in WS7 of the FASTory line.



#	operation	information	remark
1	load pallet	palletId : 4152265899558gyh	operator_1
2	inspect quality	proiductVariant:8, palletId:415226589957ff8t	operator_1
3	unload pallet	palletId : 4152265899gg458y	operator_1

Fig. 8. Web-based interface for the operator in WS7

As appears in the figure, the interface includes a table of operations that the operator needs to perform. Fig. 8 depicts a list of operations for loading and unloading specific pallets, and inspecting the quality of a product variant. The information column in the interface specifies those pallets or products. Following the event driven concept, the orchestrator must be notified when operations are performed by human operators. This is achieved by allowing the operator to send a confirmation through the shown *Notify* button. Then, the orchestrator will update the list of operations linked to a corresponding recipe as described in the model interaction section. On the other hand, this interface is implemented with web-based standards. Thus, if desired by the factory owner, such tool can be also accessed and manipulated from any device that can open a web browser.

V. CONCLUSION AND FURTHER WORK

This article presents an approach for describing the information required by an orchestrator engine in order to control operations of production lines. Furthermore, this research work provides the possibility of including human operator skills and linked operations. This permits that the orchestrator can consider humans as another resource for processing product variants in production system. Thus, this approach allows human operators to be an active asset that may be considered in the production plan.

The approach and use case implementation presents an alternative for designing larger models that can include human operator information for enhancing the productivity in tasks that ae undoable (or expensive to do) by machines. To achieve this, the current model should be extended and connected to analysis engines that are able to not only assess how the tasks are performed by humans, but also how they can be improved. Moreover, the semantic descriptions and software-based approach should be complemented with certain sensing devices for such quality of performance assessment.

On the other hand, there are previous research works in the literature that consider human information e.g., for creating schedules [41], [42], adapting interfaces [23] or modelling the human-machine interaction [13], [14]. In the future, the authors plan to integrate some of the strategies of aforementioned works for implementing a valid solution that can be beneficial to control operations that are carried out for both humans and machines in a knowledge-driven service-oriented environment. Then, the model will be extended with information about work performance. In fact, a complex event processing engine will be integrated for the analysis of such new information. On the other hand, the proposed interface will be tested and evaluated in order to modify it according to human operator feedback.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement n° 636909, correspondent to the project shortly entitled C2NET, Cloud Collaborative Manufacturing Networks.

REFERENCES

[1] J. Lee, B. Bagheri, and H.-A. Kao, “A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems,” *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.

[2] A. J. C. Trappey, C. V. Trappey, U. Hareesh Govindarajan, A. C. Chuang, and J. J. Sun, “A review of essential standards and patent landscapes for the Internet of Things: A key enabler for Industry 4.0,” *Adv. Eng. Inform.*

[3] B. Vogel-Heuser, C. Diedrich, D. Pantförder, and P. Göhner, “Coupling heterogeneous production systems by a multi-agent based cyber-physical production system,” in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 713–719.

[4] S. Iarovyi, X. Xu, A. Lobov, J. L. M. Lastra, and S. Strzelczak, “Architecture for Open, Knowledge-Driven Manufacturing Execution System,” in *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*, S. Umeda, M. Nakano,

---

[12] http://www.tut.fi/escop/

H. Mizuyama, H. Hibino, D. Kiritsis, and G. von Cieminski, Eds. Springer International Publishing, 2015, pp. 519–527.

[5] 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.

[6] B. Ramis Ferrer and J. L. Martinez Lastra, “Private local automation clouds built by CPS: Potential and challenges for distributed reasoning,” *Adv. Eng. Inform.*, vol. 32, pp. 113–125, Apr. 2017.

[7] C. Vogel, M. Fritzsche, and N. Elkmann, “Safe human-robot cooperation with high-payload robots in industrial applications,” in *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2016, pp. 529–530.

[8] H. Haggag, M. Hossny, S. Haggag, S. Nahavandi, and D. Creighton, “Safety applications using Kinect technology,” in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 2164–2169.

[9] Y. Shen, G. Reinhart, and M. M. Tseng, “A design approach for incorporating task coordination for human-robot-coexistence within assembly systems,” in *2015 Annual IEEE Systems Conference (SysCon) Proceedings*, 2015, pp. 426–431.

[10] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, “Human-machine-interaction in the industry 4.0 era,” in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 289–294.

[11] R. Harrison, D. Vera, and B. Ahmad, “Engineering Methods and Tools for Cyber #x2013;Physical Automation Systems,” *Proc. IEEE*, vol. 104, no. 5, pp. 973–985, May 2016.

[12] J. Jokinen, M. B. Ambat, and J. L. M. Lastra, “Condition monitoring for distributed systems with reconfigurable user interfaces and data permissions,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 5705–5710.

[13] A. Björkelund *et al.*, “On the integration of skilled robot motions for productivity in manufacturing,” in *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)*, 2011, pp. 1–9.

[14] A. Björkelund, J. Malec, K. Nilsson, and P. Nugues, “Knowledge and Skill Representations for Robotized Production,” *IFAC Proc. Vol.*, vol. 44, no. 1, pp. 8999–9004, Jan. 2011.

[15] M. Baygin, H. Yetis, M. Karakose, and E. Akin, “An effect analysis of industry 4.0 to higher education,” in *2016 15th International Conference on Information Technology Based Higher Education and Training (ITHET)*, 2016, pp. 1–4.

[16] J. Zhang, B. Ahmad, D. Vera, and R. Harrison, “Ontology based semantic-predictive model for reconfigurable automation systems,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 1094–1099.

[17] W. Terkaj and M. Urgo, “Ontology-based modeling of production systems for design and performance evaluation,” in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 748–753.

[18] A. Kashevnik, N. Teslya, E. Yablochnikov, V. Arckhipov, and K. Kipriianov, “Development of a prototype Cyber Physical Production System with help of Smart-M3,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 4890–4895.

[19] T. R. Gruber, “A translation approach to portable ontology specifications,” *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.

[20] J. L. M. Lastra, I. M. Delamer, and F. Ubis, *Domain Ontologies for Reasoning Machines in Factory Automation*. ISA, 2010.

[21] 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.

[22] Y. Evchina and J. L. M. Lastra, “Semantic-driven CEP for delivery of information streams in data-intensive monitoring systems,” in *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 2015, pp. 1251–1256.

[23] M. J. Islam, B. R. Ferrer, X. Xu, A. Nieto, and J. L. M. Lastra, “Implementation of an industrial visualization model for collaborative networks,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, 2016, pp. 720–725.

[24] L. E. G. Moctezuma, J. Jokinen, C. Postelnicu, and J. L. M. Lastra, “Retrofitting a factory automation system to address market needs and societal changes,” in *2012 10th IEEE International Conference on Industrial Informatics (INDIN)*, 2012, pp. 413–418.

[25] Y. W. Ma, Y. C. Chen, and J. L. Chen, “SDN-enabled network virtualization for industry 4.0 based on IoTs and cloud computing,” in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, 2017, pp. 199–202.

[26] J. Bohuslava, J. Martin, and H. Igor, “TCP/IP protocol utilisation in process of dynamic control of robotic cell according industry 4.0 concept,” in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 2017, pp. 000217–000222.

[27] T. Y. Lin, Y. M. Chen, D. L. Yang, and Y. C. Chen, “New Method for Industry 4.0 Machine Status Prediction - A Case Study with the Machine of a Spring Factory,” in *2016 International Computer Symposium (ICS)*, 2016, pp. 322–326.

[28] L. Stojanovic, M. Dinic, N. Stojanovic, and A. Stojadinovic, “Big-data-driven anomaly detection in industry (4.0): An approach and a case study,” in *2016 IEEE International Conference on Big Data (Big Data)*, 2016, pp. 1647–1652.

[29] J. Kovar, K. Mouralova, F. Ksica, J. Kroupa, O. Andrs, and Z. Hadas, “Virtual reality in context of Industry 4.0 proposed projects at Brno University of Technology,” in *2016 17th International Conference on Mechatronics - Mechatronika (ME)*, 2016, pp. 1–7.

[30] G. J. Cheng, L. T. Liu, X. J. Qiang, and Y. Liu, “Industry 4.0 Development and Application of Intelligent Manufacturing,” in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, 2016, pp. 407–410.

[31] S. Strzelczak, P. Balda, M. Garetti, and A. Lobov, *Open Knowledge-Driven Manufacturing and Logistics - The eScop Approach*. Warsaw: Warsaw University of Technology Publishing House, 2015.

[32] Andrei Lobov, “On Service Composition - Dynamic Formation and Orchestration of Service Workflows,” in *Open Knowledge-Driven Manufacturing & Logistics, The Escop Approach*, S. Strzelczak, P. Balda, M. Garetti, and A. Lobov, Eds. Warsaw University of Technology Publishing House, Warsaw 2015, pp. 311–318.

[33] B. Ramis *et al.*, “Knowledge-based web service integration for industrial automation,” in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, pp. 733–739.

[34] B. Ramis Ferrer, B. Ahmad, D. Vera, A. Lobov, R. Harrison, and J. L. Martínez Lastra, “Product, process and resource model coupling for knowledge-driven assembly automation,” *- Autom.*, vol. 64, no. 3, Jan. 2016.

[35] MESA International, “MESA White Paper #39: MESA Model Evolution,” 2011.

[36] M. Garetti, “P-PSO Ontology for Manufacturing Systems,” 2012, pp. 449–456.

[37] I.-W. Kim and K.-H. Lee, “A Model-Driven Approach for Describing Semantic Web Services: From UML to OWL-S,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 39, no. 6, pp. 637–646, Nov. 2009.

[38] A. Baechler *et al.*, “A Comparative Study of an Assistance System for Manual Order Picking – Called Pick-by-Projection – with the Guiding Systems Pick-by-Paper, Pick-by-Light and Pick-by-Display,” in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 523–531.

[39] W. M. Mohammed, A. Lobov, B. R. Ferrer, S. Iarovyi, and J. L. M. Lastra, “A web-based simulator for a discrete manufacturing system,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6583–6589.

[40] B. Zhang, C. Postelnicu, J. Jokinen, and J. L. M. Lastra, “An open energy consumption-relevant factory automation dataset in the cloud,” in *IEEE 10th International Conference on Industrial Informatics*, 2012, pp. 792–797.

[41] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau, “HTN planning for Web Service composition using SHOP2,” *Web Semant. Sci. Serv. Agents World Wide Web*, vol. 1, no. 4, pp. 377–396, Oct. 2004.

[42] C. S. Lee, M. H. Wang, M. H. Wu, C. Y. Hsu, Y. C. Lin, and S. J. Yen, “A type-2 fuzzy personal ontology for meeting scheduling system,” in *International Conference on Fuzzy Systems*, 2010, pp. 1–8.