Ontology-based Modeling of Production Systems for Design and Performance Evaluation

Walter Terkaj
Istituto Tecnologie Industriali e Automazione (ITIA)
Consiglio Nazionale delle Ricerche (CNR)
Milano, Italy
walter.terkaj@itia.cnr.it

Marcello Urgo
Dipartimento di Ingegneria Meccanica
Politecnico di Milano
Milano, Italy
marcello.urgo@polimi.it

***Abstract*— This paper presents the extension to an ontology-based data model supporting the design and performance evaluation of production systems. This extension aims to link the modeling of the spatial representation of physical objects with the characterization of their states and behavior. Furthermore, the formalization of the performance history of a production system and its components is addressed to capture both the spatial and state evolution of the objects. Such history can be provided by simulation runs or gathered from a monitoring system. A test case is described and then used to show how different software tools can be integrated to support the integrated design and evaluation of a production system.**

Keywords—Data Model, Ontology, Performance Evaluation, Finite State Machine, Production Systems

I. INTRODUCTION

Various and different business process are involved in the design of a production system, since it requires to take decisions having an impact on a mid-long time horizon and on the financial commitment of an industrial company [1]. The variety and complexity of these decisions asks for a more and more significant support from digital tools implementing design methodologies, as well as structured approaches to evaluate the performance of the designed systems. Usual performance indicators for a production systems include the throughput, the quality of the output, the incurred cost, and also more specific indicators such as the utilization of production resources, the average flow time of products, the average level of the work in progress. However, the larger is the set of adopted digital tools, the higher is the risk of reducing the efficiency of the information exchanges because of the lack of interoperability, thus leading to less effective solutions. The design tools can be more useful if they are based on a virtual representation of the production system that is continuously updated during both the design and operational/execution phase to guarantee an overall coherence of the obtained results. Therefore, a common framework to support the interoperability would benefit the cooperation between various actors with different competences and expertise in the design and management of a factory. ICT companies (e.g. Siemens PLM, PTC and Dassault Systèmes) already offer all-comprehensive Product Lifecycle Management (PLM) suites containing software tools that have been developed or acquired in the recent years. Even if these tools deal with most of the factory planning and design phases, the current solutions still do not fully meet the demands of the industry and are often based on proprietary data structures, thus endangering their interoperability. Moreover, Small and Medium Enterprises (SME) cannot afford the complete expensive PLM software suites and usually employ a subset of digital tools provided by different ICT companies. Semantic Web technologies and in particular the Web Ontology Language (OWL) [2] have been used to cope with such interoperability problems between systems that may be developed according to different data structures and employing heterogeneous technologies. Recent studies in the context of factories and manufacturing systems (e.g. the European projects LinkedDesign [3] and Virtual Factory Framework [4]) investigated if Semantic Web technologies can be indeed a valid solution. In particular, the Virtual Factory Framework (VFF) is an integrated collaborative virtual environment aimed at facilitating the sharing of resources, manufacturing information and knowledge, while supporting the design and management of all the factory entities along the phases of the their lifecycles. The VFF architecture is based on a Virtual Factory Data Model (VFDM), i.e. a coherent, standard, extensible, and common data model for the representation of factory objects related to production systems, resources, processes and products [5]. Starting from the preliminary results reported in previous works [6], this paper focuses on providing a more complete data model for production systems and their resources by linking the static characterization of the production resources with the results of performance evaluation activities in terms of a performance history that can be as detailed as needed, thus enhancing the set of software tools that can be integrated.

Section II gives an overview of the current state of the art on data models for production systems and present the background of this work. Section III delves into the proposed modeling of states and history of a production system. Section IV presents a prototype of the software platform for system design and performance evaluation and, finally, conclusions are drawn in Section V.

II. DATA MODELS FOR PRODUCTION SYSTEMS

An effective data model for the holistic representation of a production system needs to consider several aspects ranging from tangible (e.g. machine tools, part types to be produced, etc.) to intangible (e.g. process plans, production logics, etc.), from geometric (e.g. placement of production resources in the factory layout) to organizational (e.g. roles of the involved actors), from static (e.g. nominal power consumption of a machine tool) to dynamic (e.g. evolution of the states of a resource). As a consequence, even if several scientific contributions and technical standards have faced one or more aspects of the problem, it can be hardly said the research on this topic is completed.

Among the available technical standard, ANSI/ISA-95 [7] is an international standard for the development of interfaces between enterprise and control systems. This standard has been developed for applications in all industries and enables the user to define customized properties that can be attached to most of the classes representing processes and production resources. However, ANSI/ISA-95 does not provide a complete support for modeling physical data such as the placement and shape representation of objects.

A different approach in the modeling of manufacturing process is offered by the Process Specification Language (PSL) standard [8]. PSL is an ontology providing a way to formally describe a process and its characteristics. Grounding on an ontology, the PSL standard provides a robust and reliable framework to formalize the knowledge related to a process and guarantee an adequate level of interoperability.

The Industry Foundation Classes (IFC) standard by buildingSMART [9], partially based on the STEP standard [10], was mainly conceived for Architectural Engineering Construction (AEC) industry domains and therefore provides most of the definitions needed to represent tangible elements of a manufacturing systems. Furthermore, generic definitions of intangible characteristics (e.g. processes, work plans, etc.) are provided, so that its data structures can be specialized for other industrial domains, such as the manufacturing domain.

Ontologies provide a way to generate flexible data model integrating different knowledge domains, enabling knowledge sharing between several applications and a fluent flow of data between different entities [11]. Moreover, ontologies provide methods for integrating fragmented data models into a unique model without losing the notation and style of the individual ones [12]. Various ontologies have been developed in the scope of manufacturing domain. For instance, Lin et al. [13] designed a Manufacturing System Engineering (MSE) ontology to provide a common understanding of manufacturing-related terms and to enhance the semantic and reuse of knowledge resources within global extended manufacturing teams. Lemaignan et al. [14] presented a Manufacturing's Semantics Ontology (MASON) that is an upper ontology based on three main concepts: *entities*, *operations* and *resources*. However, the currently available ontologies for manufacturing are usually too generic for real application (e.g. they are upper ontologies) or are focused on specific domains and lack an integrated approach, thus jeopardizing a broader applicability.

A. Virtual Factory Data Model

The advantages of the an ontology-based data modeling were exploited for the development of the Virtual Factory Data Model (VFDM) [5] in the scope of the Virtual Factory Framework, while trying to keep a practical approach. The VFDM aims at formalizing and integrating the concepts of product, process, production resource and building while providing enough details to feed digital tools supporting the factory life-cycle phases. Moreover, the VFDM was designed to exploit already existing technical standards and extends their definitions to represent the characteristics of a manufacturing system in terms of the products to be manufactured, the manufacturing process they must undergo and the resources entitled to operate the different manufacturing operations. The current version of VFDM is mainly based on the IFC [9] and STEP-NC [15] standards that were translated into a set of ontology modules.

The Entities in the IFC standard were mapped to OWL Classes in the VFDM. Most of the classes derived from IFC are specializations of two fundamental classes named *IfcTypeObject* and *IfcObject*, both being subclasses of *IfcObjectDefinition*. The former class is the generalization of any thing or process seen as a type, the latter seen as an occurrence. *IfcObject* has the following subclasses: *IfcProduct*, *IfcProcess*, *IfcResource*, *IfcControl*, *IfcActor*, *IfcGroup*. *IfcProduct* represents the occurrence of a generic object that can be related to a geometric or spatial context (e.g. manufactured products, machine tools, transport systems, etc.). *IfcProcess* defines a process that can be used to transform an input into output (e.g. assembly operation, machining operation, etc.). *IfcResource* represents the information related to a resource needed to execute a process. *IfcControl* is the generalization of the concepts that control or constrain the use of products, processes, or resources. *IfcActor* defines the actors or human agents that are involved in a project. *IfcGroup* is the generalization of any group. The subclasses of *IfcTypeObject* (i.e. *IfcTypeProduct, IfcTypeProcess, IfcTypeResource*) can be paired with the corresponding subclasses of *IfcObject*.

The previously described generic classes can be exploited to model a wide range of manufacturing systems while taking into consideration both hardware and software aspects. The subclasses of *IfcTypeObject* can be used to specify the designed characteristics of a manufacturing system, e.g. the part types to be produced (*IfcTypeProduct*), the process plans (*IfcTypeProcess*), the required type of production resources (*IfcTypeResource*). On the other hand, the subclasses of *IfcObject* can be used to represent the execution phase of a manufacturing system by defining the workpieces in process (*IfcProduct*), the actually executed operations (*IfcProcess*), and the usage of production resources (*IfcResource*).

Furthermore, the VFDM consists of novel ontology modules that were developed to define concepts non included in the reference technical standards or to specialize some classes, thus leading to the inclusion of concepts such as stochastic distributions (e.g. class *VffProbabilityDistribution* and its subclasses), production processes (e.g. class *VffManufacturingProcess* as subclass of *IfcProcess*), production resources (e.g. class *VffMachineryElement* as subclass of *IfcProduct*) and production systems (e.g. class *VffManufacturingSystem* as subclass of *IfcGroup*).

A key feature of the IFC standard consists in the availability of predefined objectified relationships that enable to characterize the data by specifying assignments, associations, connections, and decompositions according to the required level of details. For example, these relationships allow to explicitly define the following links:

- assignment of a process step to the production resource that can execute it;
- nesting of a process into its sub-processes and decomposition of a machine tool into its components;
- precedence constraints between the processes;
- input and output entities of a process.

III. Modeling the Behavior and History of a Production System

The problem of evaluating/monitoring the performance of a production system and formalize its relevant output requires a data model that is able to characterize:

- the static configuration of the production systems;
- the production management policies;
- the designed behavior of production resources/systems and the link with the production management policies;
- the evolution of the production resources/systems according to the level of details provided by the specific evaluation/monitoring tool;
- the link between the evolution of the production resources/systems and their static characteristics.

The first two requirements can be met, at least partially, by the VFDM briefly presented in Section II.A. However, the last three requirements are particularly critical and asked for a further development that is presented in the following sub-sections.

A. Modeling the Behavior

The behavior of a system can be defined as the total of events or actions that a system can perform, the order in which they can be executed and maybe other aspects of this execution such as timing, event and transition probabilities, or continuous aspects [16]. As an example, a machine can be idle waiting for a part to process and then be working as soon as a raw part is provided. In addition, it can be failed if a failure occurs and, hence, it needs a maintenance action to be able to operate again. Moreover, some machines can be operative only if some conditions are guaranteed, e.g. a given temperature is reached. If the actions are discrete, then the process is also called a discrete event system. Several approaches to model the behavior of a discrete event system have been proposed in the literature, such as State Charts, Petri Nets, Finite State Automata, Queuing Networks, etc.

Herein, it was decided to follow an approach based on Finite State Machine (FSM) [17], thus explicitly modeling the possible states of an object and the events that may cause the transition from a state to another. Moreover, the extensions included in UML state machine (statechart) [18] were considered. The most relevant enhancements that were exploited consist in the decomposition of states, including the definition of both hierarchical nesting (OR-decomposition) and orthogonal regions (AND-decomposition). This feature enables to mitigate the problem of dealing with a large number of states and transitions when addressing real cases.

The novel class *FsmState* was defined in the *FiniteStateMachine* ontology module to represent the state of a generic object in the scope of a Hierarchical Machine State modeling. A state can be decomposed into one or more substates by means of the objectified relationship abstract class *FsmRelDecomposes* having as subclasses *FsmDecomposesExclusive* and *FsmDecomposesParallel*. The former subclass can be used to formalize a hierarchical nesting, whereas the latter to formalize orthogonal regions. In case of hierarchical nesting it is possible to specify which is the initial state to be entered. Moreover, it must be stressed that a state can be directly decomposed either by a hierarchical nesting or by orthogonal regions. This modeling allows to define state machines with several level of decomposition to be as detailed as needed, thus enabling also the development of a multi-scale approach. Fig.1 shows an excerpt of the ontology, where the boxes represent classes and the dashed arcs represent property restrictions linking classes according to the Manchester OWL Syntax [19].

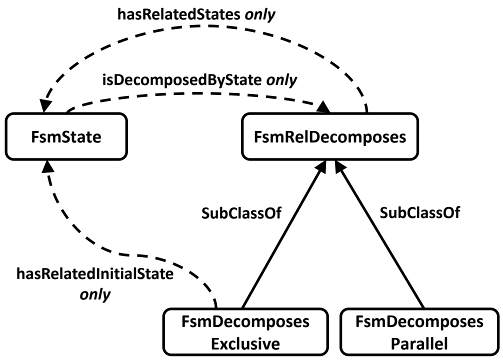

Fig. 1. Modeling of object state.

Further OWL classes can be defined to represent the concepts of *events* and *transactions*, as discussed in [20] and [21]. Moreover, the class *FsmState* can be specialized to better characterize the states of different objects, such as a product (e.g. class *IfcProductState*) or a machine tool (e.g. class *VffMachineryElementState*). In case of a machine tool, further subclasses of *VffMachineryElementState* can be defined to characterize different behaviors (e.g. idle, working, and failed states). Fig. 2 shows how the class *IfcProduct* defining a generic product is linked with class *IfcProductState* for characterizing its state machine. In Fig. 2 the novel classes are highlighted by a grey background color, whereas the dashed arcs represent again property restrictions. Taking into account the link between a product and its shape representations (see class *IfcShapeRepresentation* in Fig. 2) already defined in the IFC standard and in the proposed VFDM, a direct link between a product state and its possible shape representations has been added to reach a thorough characterization of the behavior of a physical object, thus supporting also software tools for dynamic visualization (see Sect.IV.D).

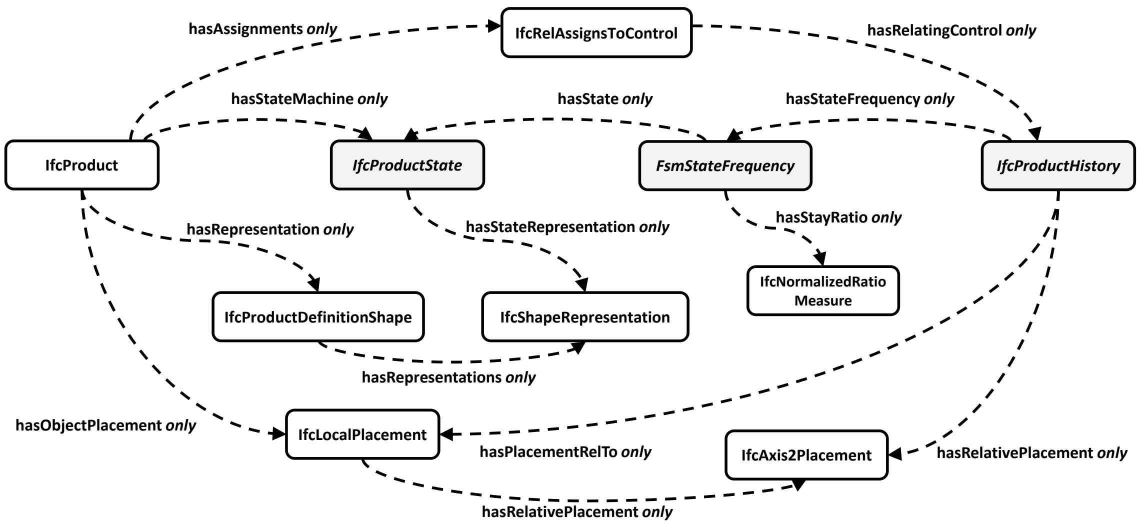

Fig. 2. Relations between classes in the extended VFDM.

B. *Modeling the History*

The output of a performance evaluation or monitoring system can be characterized with significantly different levels of detail and type of information, ranging from geometric (e.g. the placement of an object) to behavioral data (e.g. state transitions). When using a performance evaluation approach like the simulation, we build a model representing the behavior of the system and, hence, we simulate how the system would work in a given condition and register the results. These results can be in the form of a log of sequential events/states of the system along the simulation time, i.e. an history of what happened during the simulation. Moreover, aggregate indicators (e.g. the mean and variance of a performance indicator) can elaborated in place of a more detailed history. Not all the performance evaluation approaches provide a complete history, but also aggregate estimations can be considered as a simple type of history.

The standard IFC proposes the class *IfcPerformanceHistory* to describe an object history that can be associated with a time interval. The class *IfcPerformanceHistory* is linked to its reference object by means of the objectified relationship class *IfcRelAssignsToControl*. A piece of history can be decomposed either time-wise (i.e. defining sub-intervals) or hierarchically (i.e. defining the history of object components). Such flexibility allows to formalize detailed and specific history data coming from a real monitoring system, but also aggregate simulated data coming from performance evaluation methods that may or not refer to specific time intervals. However, it must be noted that the part of the IFC standard related to the object history has not been thoroughly developed yet.

The PSL ontology, in principle, can be convenient in capturing the performance history, since it can formalize ternary relations that are used in representing complex spatio-temporal scenarios. However, it must be noted that making alignments between the Common Logic-based PSL and ontologies that are based on the description logic SROIQ [22] is not a straightforward task (e.g. OWL is limited to unary and binary relations). Therefore, in order to capture in OWL the complex history scenarios that can be also integrated in the scope of VFDM, it was decided to extend the IFC and its OWL implementation by specializing the class *IfcPerformanceHistory* with the aim of linking the concepts of history and behavior of the production resources, while compensating the shortcomings of OWL related to its expressivity. The class *IfcPerformanceHistory* was specialized by subclasses such as *IfcProductHistory*, *IfcProcessHistory*, *VffMachineryElementHistory* that are linked to classes *IfcProduct*, *IfcProcess* and *VffMachineryElement*, respectively. Fig. 2 shows the case of *IfcProductHistory* and *IfcProduct*, highlighting how the concepts of product, product state and product history are all linked and demonstrating how the proposed data model provides a unified representation for the problem of production system modeling. A product history and its sub-components can be characterized by the start and end time of the corresponding time interval. Moreover, since the product behavior is described by a set of possible states, the class *FsmStateFrequency* was introduced to define which percentage of the time interval was spent in a specific state together with other statistics. Finally, the history of a product can be associated with a placement in the space, so that trajectories can be represented with the required level of detail.

IV. System Design and Performance Evaluation Platform

To demonstrate the viability and the benefits of the proposed ontology-based data model for manufacturing, this section presents a prototype of *system design and performance evaluation platform* that results from the integration of different software tools thanks to the interoperability offered by the extended VFDM. The platform is built on a common data repository providing access to different but integrated digital tools that support the following phases: 1) early production system design; 2) layout design; 3) performance evaluation; 4) visualization of production system performance. These phases can be sequential or more frequently iterated in loops. It must be stressed that it was possible to coherently link phases 3 and 4 with the previous ones thanks to the enhancement presented in Sect.III. Each phase will be presented in the following sub-sections while referring to a common test case. This test case consists in the design of a production line made of five working stations and four inter-operational buffers. The flow of data between the various software tools integrated in the platform and the shared data repository is shown in Fig. 3. The data repository for ontology data can be implemented adopting different technologies ranging from file-based solutions to relational databases and to native triple stores [23]. The adoption of an efficient data repository is a key requirement for the scalability of this ontology-based platform. For each of the presented software tools, a customized *connector* was developed to import/export ontology defined according to the VFDM. The development of the *connector* was supported by a programming library named *VfConnectorLib*, based on the Redland C libraries [24], that allows creating and modifying the data within the repository thanks to a C++ library that maps the definitions of the VFDM to C++ classes and methods.

A. Early production system design

The early design of a production system requires to define a wide range of data related to part types, the process plans, the selectable machine types, the already available machines, the factory building, etc. If an ontology-based approach is adopted, then it is necessary to formalize all this data in terms of ontologies and this activity demands user-friendly graphical interfaces and/or tools to synchronize the contents of databases or legacy systems. Herein, the early design phase is supported by the prototype software tool OntoGUI (see *step 1* in Fig. 3) that was developed by ITIA-CNR in C++ language to enable the management and instantiation of ontologies. This software tool consists of a control panel that allows creating and loading ontologies and a set of other applications for the exploration and generation of OWL individuals. The individuals can be searched according to the class and relationships between individuals can be viewed, added and deleted according to the restrictions defined in the adopted data model. Moreover, an integrity check based on a Closed World Assumption (CWA) approach can be run on the individuals.

B. Layout design

The layout design phase takes as input the relevant information about the available equipment together with the production requirements to generate as output the geometric layout of the production system, defining also the number of production resources that are needed. GIOVE Virtual Factory (GIOVE-VF) is a virtual reality collaborative environment aimed at supporting the factory layout design [25]. GIOVE-VF, if used in design-mode (see step 2 in Fig. 3), offers the user the possibility to design factories by selecting machines, operators and other resources from available catalogues and place them in the 3D scene of the virtual factory.

C. Performance Evaluation

The applicability of the extended VFDM to model a production system and evaluate its performance, has been tested focusing the attention on the case of Discrete Event Simulation (DES). The capability of generating simulation models in a (semi-)automatic way has been considered one of the great challenges in the simulation of manufacturing systems [6] because it would greatly speed up the process and also reduce the time needed to verify a model. Among the great number of available general-purpose commercial off-the-shelf (COTS) simulation packages, Arena by Rockwell Automation is one of the most used both in the academic and industrial field. An Arena simulation model can be automatically generated thanks to a *connector* that maps the Arena data structures against the VFDM classes (see step 3 in Fig. 3). This *connector* was developed in C++ language using the COM interface of Arena [6]. The connector drives the execution of the simulation model, collects the statistics related to the addressed performance measure and generate a simulation log.

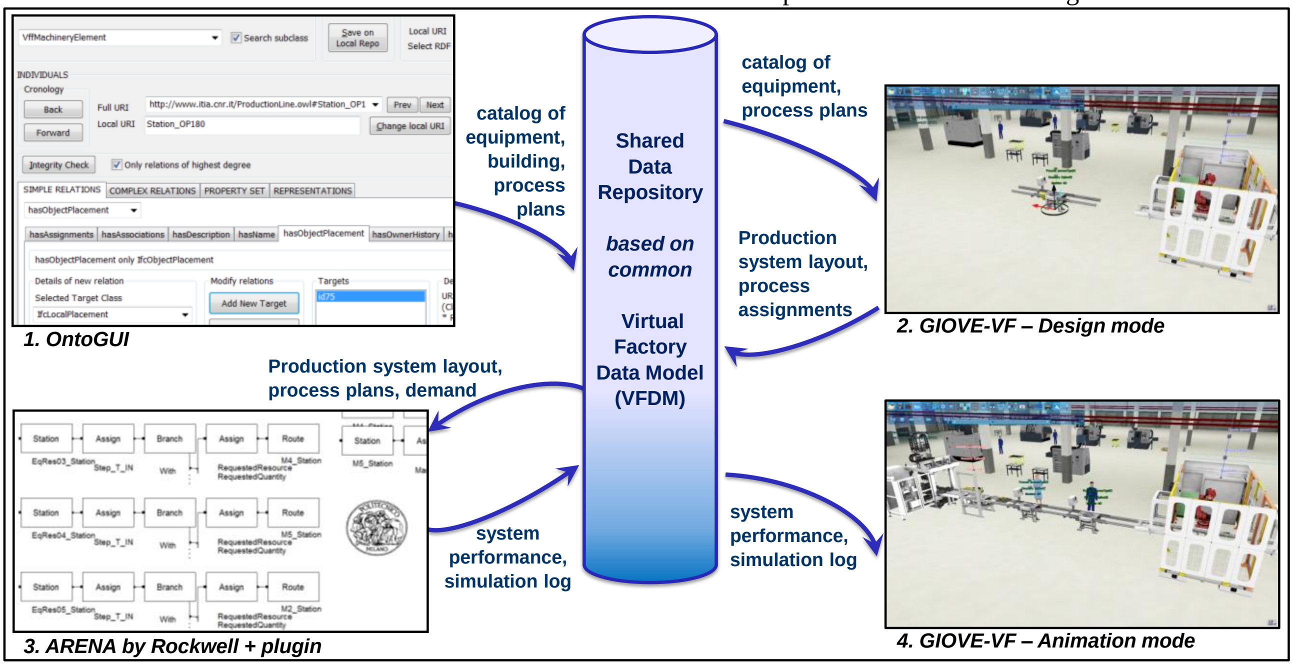

Fig. 3. System design platform highlighting the involved software tools and the exchanged data

D. *Visualization of production system performance*

The analysis of the performance of a production system is a relevant task that, if not carried out properly, may jeopardize the benefits coming from performance evaluation and monitoring activities. Synthetic reports, graphical user interfaces and 3D visualization tools allows to navigate the performance at different levels of detail. The software tool GIOVE-VF, if used in animation-mode (see step 4 in Fig. 3), gives access to synthetic performance indicators and is also able to load and elaborate a simulation log to transform it into an animation. Such animation can be exploited both to detect unexpected/unwanted dynamic behaviors and to assess the coherence of the upstream performance evaluation method, thus leading to system and/or model reconfigurations.

V. Conclusions

This work aims at providing re-usable application-oriented ontologies that exploit the state-of-the-art technical standards, while avoiding the commitment to specific integrated software suites provided by commercial ICT companies. The preliminary results are already exploited at industrial level by implementing an ontology-based platform for a large Italian company playing in the roll milling market. Experiments have been carried out on production systems characterized by one hundred physical objects and on generating the performance history from simulation logs with over 20000 events.

Further research will focus on the modeling of hierarchical state machine and its events and transitions. Moreover, the links between the VFDM and upper ontologies available in the literature will be investigated. In particular, the alignment between the process-related ontology modules in VFDM and the PSL ontology will be addressed. *Ontohub ontology repository* is a platform that can support such alignment and consistency checking of linked ontologies, by exploiting the parsing and inference back end of the Heterogeneous Tool Set [26]. Finally, the potentiality of reasoning will be investigated to assess the integrity and consistence of a production system model, as for example addressed by Abele et al. [27].

Acknowledgment

This research has been partially funded by MIUR under the Italian flagship project “La Fabbrica del Futuro”, Subproject 2, research project “Product and Process Co-Evolution Management via Modular Pallet configuration”, and by the EU 7th FP under the grant agreements No: 314156, “Engineering Apps for advanced Manufacturing Engineering” and No: 262044, “VISION Advanced Infrastructure for Research”, and by Regione Lombardia under the project “Fabbrica Intelligente per la Deproduzione Avanzata e Sostenibile”.

References

[1] T. Tolio, M. Sacco, W. Terkaj, M. Urgo, “Virtual Factory: an Integrated Framework for Manufacturing Systems Design and Analysis,” Procedia CIRP 2013; 7:25-30.

[2] W3C. OWL 2 Web Ontology Language - Document Overview. [Online]. http://www.w3.org/TR/owl2-overview/

[3] A. Milicic, A. Perdikakis, S. El Kadiri, D. Kiritsis, S. Terzi, P. Fiordi, S. Sadocco, "Specialization of a Fundamental Ontology for Manufacturing Product Lifecycle Applications: A Case Study for Lifecycle Cost Assessment," OTM Workshops 2012: 69-72, 2012.

[4] K. Botond, W. Terkaj, M. Sacco, "Semantic Virtual Factory supporting interoperable modelling and evaluation of production systems," CIRP Annals - Manufacturing Technology 2013; 62(1):443-446.

[5] W. Terkaj, G. Pedrielli, M. Sacco, “Virtual Factory Data Model,” Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing, Graz, Austria, 24-27 July, 2012. pp 29-43.

[6] W. Terkaj, M. Urgo, “Virtual Factory Data Model to support Performance Evaluation of Production Systems,” Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing, Graz, Austria, 24-27 July, 2012, pp 44-58.

[7] International Society of Automation, “ISA-95: the international standard for the integration of enterprise and control systems”.

[8] National Institute of Standards and Technology, “Process Specification Language (PSL)”, 2008.

[9] buildingSMART, “IFC Overview”. [Online]. http://buildingsmart-tech.org/specifications/ifc-overview

[10] “ISO 10303 - Industrial automation systems and integration -- Product data representation and exchange”.

[11] T. Moser, W.D. Sunindyo, M. Merdan, S. Biffl, "Supporting Runtime Decision Making in the Production Automation Domain Using Design Time Engineering Knowledge," in Ontology and Semantic Web for Manufacturing 2011, Heraklion, Crete, Greece, 2011, pp. 9-22.

[12] M. Hepp, P. De Leenheer, A. De Moor, Y. Sure, “Ontology Management: Semantic Web, Semantic Web Services, and Business Applications”, Springer-Verlag, 2007.

[13] H.K. Lin, J.A. Harding, M. Shahbaz, "Manufacturing system engineering ontology for semantic interoperability across extended project teams". Int. Journal of Production Research, 42(24):5099-5118.

[14] S. Lemaignan, A. Siadat, J. Dantan, A. Semenenko, "MASON: A Proposal For An Ontology Of Manufacturing Domain," Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006.

[15] “ISO 14649 - Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers, International Organization for Standardization”, 2004.

[16] J.C.M. Baeten, D.A. van Beek, J.E. Rooda, “Process Algebra,” in P.A. Fishwick (ed) Handbook of Dynamic System Modeling, Chapman and Hall, 2007.

[17] Z. Kohavi, “Switching and Finite automata theory,” 2nd ed. McGraw-Hill, 1978.

[18] D. Harel, "Statecharts: A visual formalism for complex systems," Sci. Comput. Program., 8:231-274, 1987.

[19] W3C, “OWL 2 Web Ontology Language Manchester Syntax,”.

[20] P.A. Fishwick, J.A. Miller, "Ontologies for Modeling and Simulation: issues and approaches," Proc. of 2004 Winter Simulation Conference.

[21] Y. Belgueliel, M. Bourahla, M. Brik, "Towards an Ontology for UML State Machines," Lecture Notes on Software Engineering, 2(1), 2014.

[22] I. Horrocks, O. Kutz, U. Sattler, "The Even More Irresistible SROIQ," Proceedings of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning, pp. 57-67. AAAI Press, 2006.

[23] G. Modoni, M. Sacco, W. Terkaj, “A survey of RDF store solutions,” Proceedings of the 20th International Conference on Engineering, Technology and Innovation, Bergamo, Italy, 23-25 June, 2014.

[24] D. Beckett, "Redland RDF Libraries," [Online].http://librdf.org/.

[25] G.P. Viganò, L. Greci, S. Mottura and M. Sacco, "GIOVE Virtual Factory: A New viewer for a more immersive role of the user during factory design," in Digital factory for human-oriented Production systems, 2011.

[26] T. Mossakowski, O. Kutz, M. Codescu, C. Lange, "The Distributed Ontology, Modeling and Specification Language," Proc. of the 7th Int. Workshop on Modular Ontologies, Corunna, Spain, September 15, 2013.

[27] L. Abele, C. Legat, S. Grimm, A.W. Muller, "Ontology-based validation of plant models," Proceedings of 11th IEEE International Conference on Industrial Informatics (INDIN), Bochum, Germany, 2013.