Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 28 (2015) 137 – 142
www.elsevier.com/locate/procedia

3rd CIRP Global Web Conference

A Virtual Factory Data Model as a support tool for the simulation of manufacturing systems

W. Terkaj[a,]*, M. Urgo[b]

[a] Istituto Tecnologie Industriali e Automazione (ITIA), Consiglio Nazionale delle Ricerche (CNR), Milano, Italy
[b] Mechanical Engineering Department, Politecnico di Milano, Milano, Italy
* Corresponding author. Tel.: +39-02-2369-9607 ; fax: +39-02-2369-9915 .*E-mail address:* walter.terkaj@itia.cnr.it .

Abstract

The design of a manufacturing systems is a complex and critical activity entailing decisions with an impact on a long time horizon and a major commitment of financial resources. Indeed, the modelling, simulation and evaluation of manufacturing systems are relevant activities both in the design and the operational phases of a factory. This paper grounds on the results of the Virtual Factory Framework (VFF) Project and addresses the use of an ontology based model of a production system to support the construction of a performance evaluation model.

Selection and peer-review under responsibility of the International Scientific Committee of the "3rd CIRP Global Web Conference" in the person of the Conference Chair Dr. Alessandra Caggiano.

Keywords: Data model; Ontology; Performance evaluation; Finite state machine; Production systems; Cyber-physical systems;

1. Introduction

The design of a production system involves various and different business processes, as well as different classes of analysis, since it entails decisions with a medium to long time horizon impact and also having a significant influence on the financial commitment of an industrial company [1]. Complex decisions ask for a consistent support through digital tools implementing design methodologies and/or providing an assessment of key performance indicators to be taken into consideration. However, when a large set of digital tools are used, it is also difficult to guarantee an effective interoperability among them, thus reducing the effectiveness of the solution. Moreover, the current standard for design tools asks for a virtual representation of the production system that also needs to be continuously updated during both the design and operational/execution phase, thus guaranteeing an overall coherence of the obtained results. The described requirements call for a common framework to support the interoperability and exploiting the cooperation of different actors with different competences and expertise in the design and management of a factory.

Semantic Web technologies, and in particular the Web Ontology Language (OWL) [2], have been used as an enabler for the interoperability between systems using different data structures and heterogeneous technologies. The use of Semantic Web technologies in the context of factories and manufacturing systems has been addressed in the European projects LinkedDesign [3] and Virtual Factory Framework [4]. 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 [6,7], this paper focuses on providing a complete data model for production systems and their resources by linking the

Selection and peer-review under responsibility of the International Scientific Committee of the "3rd CIRP Global Web Conference" in the person of the Conference Chair Dr. Alessandra Caggiano.
doi:10.1016/j.procir.2015.04.023

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.

2. Structuring production systems' data

A comprehensive 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). From the other side, several scientific contributions and technical standards have been developed in the manufacturing system area, each of them focusing on a set of aspects of the overall problem.

ANSI/ISA-95 [8] is an international standard for developing an automated interface between enterprise and control systems addressing all industries and all sorts of processes (batch processes, continuous and repetitive processes, etc.). ANSI/ISA-95 standard enables the user to freely define customized properties that can be attached to most of the classes representing processes and production resources. However, such flexibility can be a drawback from the interoperability point of view, furthermore, it does not provide a complete support for modeling physical data such as the placement and shape representation of objects.

A different approach is offered by the Process Specification Language (PSL) standard [9], an ontology to formally describe a process and its characteristics. This standard is however scarcely adopted in the industrial domain, probably due the perceived complexity at the enterprise level.

The Industry Foundation Classes (IFC) standard by buildingSMART [10], partially based on STEP standard [11], was mainly conceived for Architectural Engineering Construction (AEC) industry domains (e.g. Building Controls, Structural elements, Structural Analysis, etc.) and 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.

Ontologies represent a possible 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 [12]. Moreover, ontologies provide methods for integrating fragmented data models into a unique model without losing the notation and style of the individual ones [13]. Various ontologies have been developed in the scope of manufacturing domain. Lin et al. [14] 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. Léger et al. [15] presented a Manufacturing's Semantics Ontology (MASON) that is built upon three main concepts: entities, operations and resources. Similarly, Martin and D'Acunto [16] developed an ontology decomposed into product, process and resource areas.

2.1. Virtual Factory Data Model

The advantages of 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. The VFDM aims at formalizing and integrating the concepts of product, process, production resource and building as handled by the 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 [10] and STEP-NC [17] standards that were translated into a set of ontologies [18].

The Entities in the IFC standard are 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. a manufactured products, machine tools, transport systems, etc.). IfcProcess defines a process that can be used to transform an input into output (e.g. an assembly operation, machining operation, etc.). IfcResource represents the information related to 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 physical and logical 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 (as individuals of IfcTypeProduct), the process plans (as individuals of IfcTypeProcess), the required type of production resources (as individuals of 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 (as individuals of IfcProduct), the actually executed operations (as individuals of IfcProcess), and the usage of production resources (as individuals of 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.

3. Behavior and history of a production system

The problem of evaluating/monitoring the performance of a production system and formalize the relevant output requires a data model able to structure the representation of the configuration of the production systems and the representation of the production logics. These aspects can be addressed through the VFMD. However, also other aspects must be taken into consideration.

On is the behavior of production resources/systems and their link with the production logics. Another one is the physical and logical evolution of the production resources/systems while executing the production processes and in relation to the detail of the performance evaluation or monitoring. Moreover also the connection between the state of production resources/systems and their behaviour in relation to the processes to execute. These last requirements are particularly critical for the performance evaluation of a system and, hence, a further development of the VFDM has been carried out and is presented in the following sub-sections.

3.1. Modeling the behavior

The behavior of a system can be defined in terms of events or actions that a system can perform, their order and other aspects such as timing, probabilities and transitions [19]. If the behavior can be defined through a discrete model, then the process is also called a discrete event system. Several approaches have been proposed to model a discrete event system: State Charts, Petri Nets, Finite State Automata, Queuing Networks, etc.

Herein, it was decided to follow an approach based on Finite State Machine (FSM) [20], 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) [21] 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* represents 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 using the objectified relationship abstract class *FsmRelDecomposes* with 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, a state can be directly decomposed either by a hierarchical nesting or by orthogonal regions.

This modeling allows state machines with several levels of decomposition as detailed as needed, thus enabling also the development of a multi-scale approach. Fig.1 shows an excerpt of the ontology, where boxes represent classes and arcs represent property restrictions linking classes according to the Manchester OWL Syntax [22].

Further OWL classes can be defined to represent the concepts of *events* and *transactions*, as discussed in [23, 24, 25]. 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).

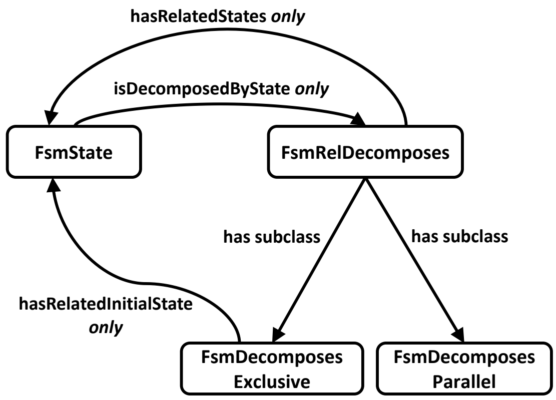

Fig. 1. Modeling of object states

3.2. Modeling the history

The output of a performance evaluation or monitoring tool 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).

The standard IFC proposes the class *IfcPerformanceHistory* to describe an object history that can be associated with a time interval. The individuals of the class *IfcPerformanceHistory* and its subclasses can be linked to their reference objects 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 an object that is physically or logically contained by the object referenced by the decomposed history). 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 reference 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. Therefore, it was needed to design an extension by specializing and further characterizing the class *IfcPerformanceHistory* with the aim of linking the concepts of history and behavior of the production resources. The class *IfcPerformanceHistory* has been specialized by subclasses such as *IfcProductHistory, IfcProcessHistory, VffMachineryElementHistory* to describe the history of individuals of class *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, if the product behavior is described by a set of possible states, then the class *FsmStateFrequency* can be exploited to define which percentage of the time interval was spent in a specific state, how many times the product entered that state, and other statistics as well. Finally, if the product is a physical object, each component of its history can be associated with a placement in the space, so that trajectories can be represented with the required level of detail. This information about the placement of a product is additional and does not replace the information about the designed placement.

4. System design and performance evaluation platform

This section presents a prototype of a *system design and performance evaluation platform* that results from the integration of different software tools, thanks to the interoperability offered by the ontology-based data model. These tools support the following phases: 1) early production system design; 2) layout design; 3) performance evaluation; 4) visualization of the performance of the production system. Each phase will be briefly presented in the following sub-sections while referring to a common test case. This test case consists in the problem of designing a production line made of five working station 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 [26].

For each of the presented software tools, a customized *connector* was developed to import/export ontology defined according to the VFDM. The development of *connector* was supported by a programming library named *VfConnectorLib,* based on the Redland C libraries [27], 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.

The prototype starts from an ontology containing the set of information describing a production system in terms of part types, process plans, selectable machine types, already available machines, factory building, etc..

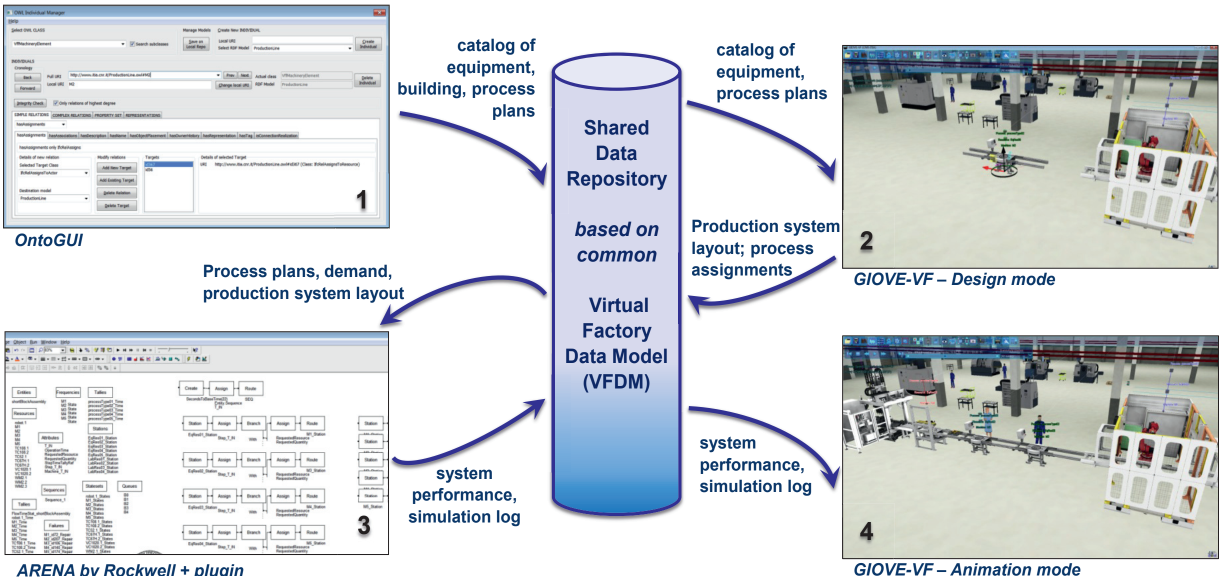

Fig. 2. System design platform highlighting the involved software tools and the exchanged data

The definition of these data can be supported by the software tool OntoGUI (see step 1 in Fig. 2) that was developed in C++ language to enable the management of ontologies. The software tool consists of a control panel that allows creating and exploring ontologies and a set of other applications that support the exploration and generation of OWL individuals. The individuals can be searched according to the class and relationships between individuals can be defined according to the restrictions defined in the VFDM.

4.1. 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 [28]. 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.

4.2. Performance evaluation

The applicability of the VFDM to model a manufacturing system and its behavior, aiming at evaluating its performance, has been tested focusing the attention on the case of Discrete Event Simulation (DES). The capability of generating simulation models in an automatic (or semi-automatic) way has been considered one of the great challenges in the simulation of manufacturing systems [6]. The automatic generation of a simulation model answers to the need of speeding up the building of a simulation model and should 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 world for applications in the manufacturing 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. In addition to basic format translation functionalities, the connector drives the execution of the simulation model, collects the statistics related to the addressed performance measure and generate a simulation log.

4.3. 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/monitoring activities. Synthetic reports, graphical user interfaces and 3D visualization tools can provide a key support to navigate into performance of the system at different levels of detail and according to the expertise of the user. The software tool GIOVE-VF, if used in animation-mode (see step 4 in Fig. 3), provides access to synthetic performance results of a production system (e.g. utilization of the machines) and is also able to load and elaborate a simulation log to transform it into an animation. After the analysis of the performance, if a system reconfiguration is needed, then GIOVE-VF can be switched to design-mode or the use of other tools integrated in the platform can be iterated.

5. Conclusions

The research presented in this paper is currently further developed in European and national research projects. Moreover, the preliminary results are already exploited at industrial level by implementing the ontology-based solution for a large Italian company acting in the market of roll milling. Further research will be focused on the modeling of hierarchical state machine as an ontology, paying particular attention to the representation of the events and transitions and the specialization of the *FsmState* class according to the modeling needs of the specific objects. Moreover, the potentiality of reasoning and inference engines will be investigated to assess the integrity and consistence of the model of a production system.

Acknowledgements

The research reported in this paper has been partially funded by the MIUR under the Italian flagship project "La Fabbrica del Futuro", Subproject 2, research project Product and Process Co-Evolution Management via Modular Pallet configuration (PRO2EVO), and by the European Union 7th FP (FP7/2007–2013) under the grant agreement No: 314156, Engineering Apps for advanced Manufacturing Engineering (Apps4aME) and grant agreement No: 262044, VISION Advanced Infrastructure for Research (VISIONAIR), and by Regione Lombardia under the project "Fabbrica Intelligente per la Deproduzione Avanzata e Sostenibile" (FIDEAS).

References

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

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

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

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

[5] Terkaj, W., Pedrielli, G., Sacco, M., 2012. "Virtual Factory Data Model," Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing. Graz. Austria, p. 29.

[6] Terkaj, W., Urgo, M., 2012. "Virtual Factory Data Model to support Performance Evaluation of Production Systems," Proceedings of the Workshop on Ontology and Semantic Web for Manufacturing. Graz. Austria, p. 44.

[7] Terkaj, W., Urgo, M., Kádár, B., Popovics, G., Kemény, L., 2012. "Modeling and simulation of production systems supported by a Virtual Factory Framework," Proceedings of the 14th International Conference on Modern Information Technology in the Innovation Processes of Industrial Enterprises. Budapest. Hungary p. 368.

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

[9] National Institute of Standards and Technology, 2008. "Process Specification Language (PSL)".

[10] buildingSMART, "IFC Overview". [Online]. http://buildingsmart-tech.org/specifications/ifc-overview

[11] International Organization for Standardization, "ISO 10303 - Industrial automation systems and integration -- Product data representation and exchange".

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

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

[14] Lin, H.K., Harding, J.A., Shahbaz, M., 2004. Manufacturing system engineering ontology for semantic interoperability across extended project teams, International Journal of Production Research 42(24), p. 5099.

[15] Léger, A., Heinecke, J., Nixon, L.J.B., Shvaiko, P., Charlet, J., Hobson, P., Goasdoué, F., 2006. "The semantic web from an industry perspective," Proceedings of 2nd Reasoning Web Summer School, 2006, p. 232.

[16] Martin, P., D'Acunto, A., 2003. Design of a production system: an application of integration product-process, Int. J. Computer Integrated Manufacturing 16(7), p. 509.

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

[18] Beetz, J., van Leeuwen, J., de Vries, B., 2009. IfcOWL: A case of transforming EXPRESS schemas into ontologies, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 23, p. 89.

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

[20] Kohavi, Z., 1978. Switching and Finite automata theory, McGraw-Hill.

[21] Harel, D., 1987. Statecharts: A visual formalism for complex systems, Sci. Comput. Program 8, p. 231.

[22] W3C, "OWL 2 Web Ontology Language Manchester Syntax,".

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

[24] Park, M., 2005. Ontology-based customizable 3D modeling for simulation, Ph.D Dissertation, The University of Florida.

[25] Belgueliel, Y., Bourahla, M., Brik, M., 2014. Towards an Ontology for UML State Machines, Lecture Notes on Software Engineering 2(1), p. 116.

[26] Modoni, G., Sacco, M., Terkaj, W., 2014. "A survey of RDF store solutions," 20th International Conference on Engineering, Technology and Innovation, Bergamo, Italy.

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

[28] Viganò, G.P., Greci, L., Mottura, S., Sacco, M., 2011. "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*, Springer.