Available online at www.sciencedirect.com
SciVerse ScienceDirect
Procedia CIRP 00 (2013) 000–000
www.elsevier.com/locate/procedia

Forty Sixth CIRP Conference on Manufacturing Systems 2013

Integrated Virtual Platform for Manufacturing Systems Design

Marcello Colledani[a], Giulia Pedrielli[a,b], Walter Terkaj[b], Marcello Urgo[a]

Politecnico di Milano, Via G. La Masa 1, Milan 20100, ITALY
Istituto di Tecnologie Industriali e Automazione, Via Bassini 15, Milan 20100, ITALY
* Corresponding author. Tel.: +39-02-2369-9610; fax: +39-02-2369-9616.*E-mail address*: walter.terkaj@itia.cnr.it

Abstract

The design of manufacturing systems is a critical task to be addressed throughout the factory life-cycle phases, including the early design, detailed design, ramp-up, reconfiguration, and monitoring. An efficient and effective system design platform may have a relevant impact on the profitability of industrial companies facing these challenges. Although several commercial applications are available for supporting different activities within the manufacturing system design and operation these stand-alone tools are usually supplied by different software vendors and cannot be easily integrated, thus entailing a massive and time-consuming integration effort. This paper proposes the integration of heterogeneous software tools supporting factory design activities over a common platform. A virtual factory environment, based on a shared data model providing to all the applications a common language to exchange data, is developed. A test case is presented that shows the integration of five methods and the related software tools to support different activities for the design of a manufacturing production line, hence the benefits derived by the application of this integrated approach in industry.

Keywords: manufacturing system design; interoperability; software platform

1. Introduction and Motivation

The design and management of manufacturing systems is a complex task entailing the need of taking decisions with a long time horizon impact and involving a major commitment from the financial point of view. The different decisions to be taken at the design stage may regard the set of products to be manufactured, the production technologies, the production resources and the transportation system to be used in a plant. Besides this, during the factory life cycle, different problems must be tackled, i.e., managing the production, controlling the behavior of the production resources, defining proper maintenance policies and taking decisions related to possible reconfigurations of the plant.

Different applications and methods are commonly applied as a support during these phases. More specifically, this paper will focus on tools for the evaluation of the performance of a manufacturing system, methods for optimization and tools for 3D visualization and design of a manufacturing plant.

Concerning performance evaluation, both *analytical methods* and *simulation methods* have been proposed in the literature [1]. *Analytical methods* are based on a mathematical model of the dynamic behavior of the system. Despite their indisputable efficiency, these methods require strict assumptions. As a result, they can be adopted as evaluation tools especially during the early design phase of a plant. When detailed design has to be performed, usually, simulation methods are more applied providing the representation of the behaviour of a production system through the execution of a computer program [2]. Simulation modelling of manufacturing systems usually takes advantage of commercial software tools (e.g. Arena, Simio, Plant Simulation, Visual Components, etc.).

Concerning optimization, many commercial applications to solve typical manufacturing optimization problems are also available (OptQuest for Arena [3] is an example). These applications are usually based on heuristics or mathematical programming. As for performance evaluation, also in the case of optimization mathematical programming gives more insight into the system behaviour although requiring more assumptions; hence, it is usually adopted during early design. Heuristics are usually more flexible and they are used when the system of interest complexity grows (i.e., during design phase).

Besides these, many tools for 3D visualization have largely replaced the 2D traditional applications providing the user with a friendly environment and fastening the design activity.

Despite tools availability, their heterogeneity is such that, most of the times, they cannot be integrated, thus decreasing their potentiality as support applications [3]. More specifically, one of the main problems, when dealing with multiple and heterogeneous analysis tools, relies in the availability and consistency of data related to the system under study to be used and shared.

This paper contribution goes in the direction of the integration among these software applications. More specifically, a simple test case has been designed to test the integration of four different tools, one for each of the aforementioned categories. In particular, the integration is achieved by means of a novel framework for the Virtual Factory (VF). This framework describes the factory as a whole consisting of resources, processes, dependencies and interrelations, data and material flows. As a result, the paper shows how this platform can be exploited to support the design and management of manufacturing systems throughout their whole life cycle.

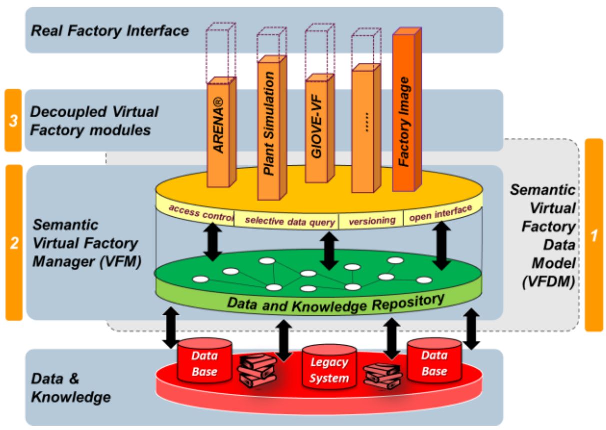

Fig. 1. Virtual Factory Framework architecture

2. Virtual Factory Framework

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, from a single product to networks of companies, along all the phases of the their lifecycles" [4-5]. The VFF architecture (Fig. 1) consists of the following components: Data & Knowledge, Semantic Virtual Factory Data Model (VFDM), Semantic Virtual Factory Manager (VFM), Decoupled Virtual Factory modules, Real factory interface.

The common VFDM (level 1, Fig. 1), based on the semantic web technology [6], assures a comprehensive vision of the information and the possibility to share it with different actors along the factory life-cycle.

The VFDM [7] can be considered as the shared language providing a common definition of the data and knowledge stored in the shared repository, i.e., it can be considered as a meta-language. It is based on 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 VFDM is mainly based on the Industry Foundation classes (IFC) standard release IFC2x4 RC2 [8-11]. Specifically, the Semantic Web approach was selected as the technology to implement the VFDM. Indeed this technology offers the possibility to (1) represent formal semantics, (2) merge ontologies dealing with different domains, (3) efficiently model and manage distributed data and (3) ease the interoperability between different applications. As a result of the adopted technology, the *entities* in the IFC standard were mapped to Web Ontology Language (OWL) *classes* in the VFDM. Hence, the data repository is created as an ontology (set of ontologies) instead of a relational Data Base. Since an official translation of the IFC standard into OWL has not been provided yet, the classes included in the VFDM were translated to the OWL following the suggestions of [13-14-15].

The Semantic Virtual Factory Manager (VFM), was developed to manage the communication of the actors and tools involved [16] (level 2, Fig. 1).

The decoupled Virtual Factory modules implement specific functionalities to support factory configuration activities through the whole lifecycle (level 3, Fig. 1). The modules are commercial or non-commercial software tools and applications used to support the activities related to factory design, performance evaluation, management, production monitoring, etc.

To integrate the modules, each software has to be provided with a *VFconnector*, i.e., a software layer providing: (1) the translation of the data from the VFDM language (which the data repository is based upon) into the application language and vice versa, (2) the upload/download of data from/to the data repository.

The real factory integration, i.e., the synchronization between the virtual representations and the real factory (level 4, Fig. 1) represents the last level, closing the loop with the External Data & Knowledge (level 0 in Fig. 1).

3. Test Case

This section briefly presents a test case aimed at demonstrating how the proposed design platform can be employed to support several design activities.

The test case consists of a data repository made of four ontologies, thus exploiting the data distribution empowered by the Semantic Web Approach. These ontologies are hierarchically organized: one ontology (*factory project*) *ProductionLine* imports three ontologies (*factory libraries*) (*BuildingLib*, *MachineLib*, *ProcessLib*). Moreover, each ontology (either libraries or project) is characterized by basic settings needed to correctly interpret its contents (e.g. the definition of the unit of measurements).

BuildingLib ontology defines a production site and a building. The building is associated with a 3D shape representation. *MachineLib* ontology defines four types of machinery element. Each machinery element is associated with a 3D shape representation and characterized by specific attributes. Specifically, each machinery element type can be characterized by failure modes that may characterize the machine type.

ProcessLib ontology defines a part type and which process plan must be executed to obtain it. The process plan is decomposed into five manufacturing operations that are characterized by processing time, sequence and needed types of production resource. The processing time was assumed equal to 22 [s] for each manufacturing operation.

ProductionLine ontology contains the factory project that imports and enriches the data provided by the libraries. In particular, the production site and the building are imported from *BuildingLib* and their placement is defined. Five machine tools are defined after the machine types available in *MachineLib* (M_1, …, M_5 in Fig. 2). Each machine tool is characterized by a shape representation and by a placement. It is specified which machine tool can provide the production resources needed to execute each manufacturing operation. Furthermore, a failure mode was defined for each machine. Specifically, in addition to the description of the failure, the distribution of the Time To Fail (TTF, Fig. 2) and Time to Repair (TTR, Fig. 2) were both defined (all the TTF and TTR are exponential and the value in parenthesis represents the mean of the distribution reported in *hours* [h].

Moreover, the machine tools are grouped into a manufacturing system and are connected with each other thanks to decoupling buffer elements (B_1, …, B_4 in Fig. 2) according to a flow line pattern. Each buffer element is characterized by a finite capacity (c_1, ..., c_4 in Fig. 2 represent the maximum capacity for each buffer).

4. Tools integrated in the platform

The test case presented in the section 3 was generated, used and enriched by a set of five software tools that were able to communicate by importing/exporting data stored into a VFDM-compliant Data Repository thanks to their customized VF Connector. The development of the VF Connector, for the software tools herein presented, was supported by a programming library named *VfConnectorLib* 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 *VfConnectorLib* is based on the Redland C libraries [17] that provide support for the Resource Description Framework (RDF), based on which data in the VFDM are structured.

4.1. VF-GUI

VF-GUI is a software tool developed in C++ providing a graphical user interface for managing the ontologies that contain factory libraries or projects. The main window of the tool represents a control panel that allows creating and exploring ontologies, by defining also dependencies to specify a hierarchy of ontologies. VF-GUI provides also a graphical interface with other software tools and it allows exploring and generating OWL individuals that populate an ontology. The individuals can be searched according to the class they belong to and relationships between individuals can be defined according to the restrictions defined in the VFDM.

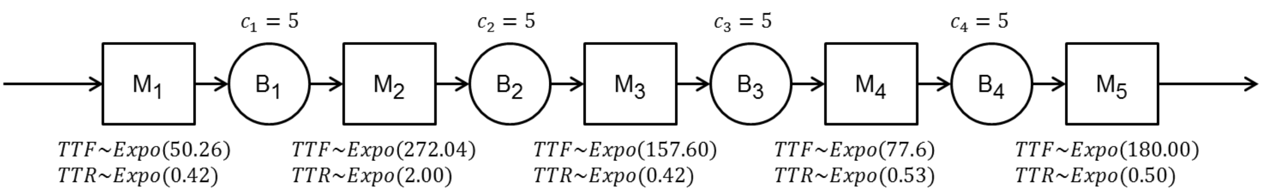

Fig. 2. Test case, production line configuration

Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2011) 000–000

www.elsevier.com/locate/circp

4.2. GIOVE-VF

GIOVE Virtual Factory (GIOVE-VF) is a virtual reality collaborative environment aimed at supporting the factory layout design [18]. In particular, GIOVE-VF 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. GIOVE-VF enables the collaboration between managers, experts and also workers in an intuitive way. Furthermore, GIOVE-VF provides a shared virtual environment where users can collaborate on the same activity and can exchange data with other tools thanks to standard input/output file formats (.xml, .rdf, .owl). GIOVE-VF was developed onto the C++ library GIOVE (Graphics and Interaction for OpenGL-based Virtual Environments), that is a set of libraries and tools designed for the creation of collaborative virtual environments and the realization of real-time 3D interactive scenes including working digital representations of objects, products, systems that are being designed and evaluated.

4.3. Flowline

Flowline is a software application developed by the Department of Mechanical Engineering at PoliMI to evaluate the performance of manufacturing lines based on approximate analytical methods. In the stand-alone version of the software, the input data are provided manually by the user through a user-friendly interface (see Fig. 3).

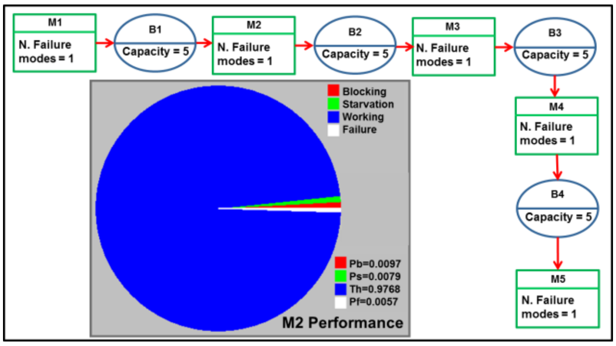

Fig. 3. System Performance Evaluation with Flowline

In particular, these data concern the system layout, including the resources composing the system and the connections between machines and buffers; machines' reliability parameters, including the number of failure modes for each machine and the specific values of the *MTTF* and *MTTR* for each failure mode; buffer sizes, machines' cycle time.

Afterwards, the logical connections between resources are validated by the user while the approximate analytical method assesses the performance applying the system decomposition provided in [1]. The calculation typically requires few seconds. The output provided by the software for a specific system configuration includes:

- Synthetic graphical of the main performance measures of the system (throughput, WIP).
- Detailed charts reporting the fraction of time each machine stays in a given state, for each possible blocking and starvation cause. This information is important for system improvement purposes.

The connector developed for the integration of Flowline automatically feeds the application with the required data exporting the information from the VFDM-based repository. Moreover, it allows to export the estimated performance measures to the ontology for further analysis (for example economic analysis of the system) or to generate new possible reconfiguration actions to be further evaluated by Flowline.

4.4. CHRONOS

The optimization problem to solve in the Test Case presented in section 3 is the Buffer Allocation Problem [19]. In particular, we want to minimize the buffer cost related to the five machine line in the test case, honoring a predefined input target performance (the production rate).

CHRONOS (Continuous Hedging Relaxation ON Optimization and Simulation) [23] finds an approximate solution to the Buffer allocation problem in multistage tandem lines with finite capacity buffers and a single class products. This tool was developed in C++ adopting CPLEX 12.1 libraries developed by ILOG-IBM.

Grounding on the description of the manufacturing system in the VFDM compliant repository, the mapping of CHRONOS data structures against VFDM classes allows the automatic generation of the needed system information to initialize the mathematical model for optimization. However, parameters specifically related to the setting of the adopted optimization algorithms are extracted by the connector from a dedicated repository which is not based on the VFDM meta-language. Furthermore, the connector can save the generated solution within the common repository to make the information available to all the connected tools.

4.5. Simulation

Avoid Among the great number of available general-purpose commercial off-the-shelf (COTS) simulation packages, we choose Arena, one of the most used simulation environment both in the academic and industrial world for applications in the manufacturing field [2]. The information stored into the VFDM are exploited in order to define an Arena simulation model by mapping the Arena data structures against the VFDM classes. This mapping has been implemented by a software component named Arena VF Connector, developed in C++ language, able to import/export ontologies serialized in RDF/XML format.

The Arena VF Connector provides functionalities to parse the ontologies and, grounding on the description of the manufacturing system in the VFDM, allows the generation of an Arena simulation model to evaluate the performance of the system under study (Fig. 4).

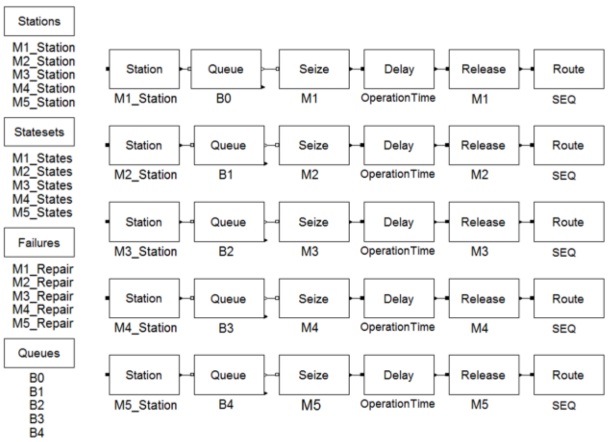

Fig. 4. The simulation model in Arena.

In addition to basic format translation functionalities, the Arena VF Connector makes use of the COM interface provided by Arena to drive the execution of the simulation model, collect the statistics related to the addressed performance measure and store the results in the VFDM [24].

5. System Design Platform

In light of section 4, herein, we present the interaction and exchange of data between the presented software tools, thus describing the workings of the system design platform (Fig. 5).

The OWL individuals manager functionality provided by the VF-GUI (section 4.1), was used to populate the libraries *BuildingLib*, *MachineLib*, *ProcessLib* (see section 3).

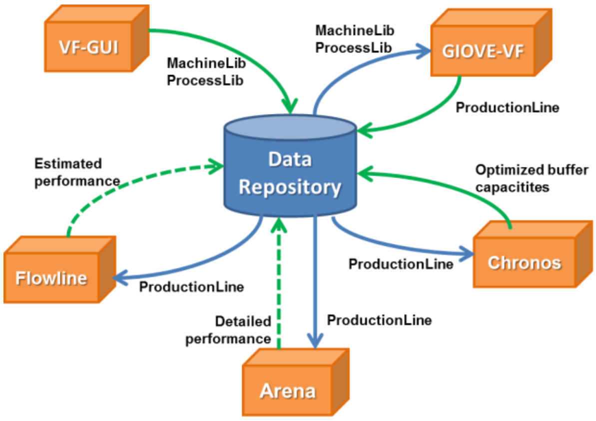

Fig. 5. System Design Platform, data exchanged

GIOVE-VF was used to generate the ontology *ProductionLine* containing the actual factory project that imports the factory libraries generated with the VF-GUI. The GIOVE-VF functionalities allowed generating the instances of machine tools after the types defined in the *MachineLib*, placing the machine tools in the 3D environment, creating connections between production resources. Fig. 6 shows the visualization of Test Case in GIOVE-VF.

Fig. 6. Test Case Production Line in GIOVE-VF

Thanks to the related connector, CHRONOS can download the ontology *ProductionLine* generated by GIOVE-VF. The input information, needed to create the optimization model, is extracted and the optimizer is launched by the related connector. Once the module has generated the solution, i.e. the configuration in terms of buffer capacities for each buffer in the line, it is saved in the project *ProductionLine*, again adopting the connector. The information related to the optimized buffer capacities is then available to the simulation tool. Either Flowline or Arena can be used at this point to assess the performance of the obtained configuration, based on the assumptions that can be done on the system behavior and on the available level of detail to describe the system dynamics. In particular, the Arena (or Flowline) connector can load the *ProductionLine* project and automatically generates the simulation model of the case (see Fig. 4).

The interactions among modules just described can iterate as long as the platform user has not solved the design problem of interest. For example, after receiving the results from the simulation module, the designer might wish to use the VF-GUI to add a machine in order to increase the whole efficiency and so forth until the accomplishment of the design objectives.

6. Conclusions

This paper has presented a framework for an Integrated Virtual Platform to design and analyze manufacturing systems. The integrated platform grounds on a common and shared data repository modeling the relationships between physical and logical entities of a manufacturing system. These data can be shared by the tools integrated within the platform. This integration is realized through connectors whose development was eased by the presence of C++ libraries to manage data formalized as ontologies. The shared data have been used to demonstrate the interoperability between software tools supporting the design, management and performance evaluation of a manufacturing system.

The proposed data model needs further developments in order to deal with complex production logics usually characterizing manufacturing systems.

The resulting platform is extensible: the sematic web technology enables the easy and fast modification of the VFDM and the creation of new classes in order to increase the expressiveness of the model. Moreover, the number of applications can be naturally increased without affecting the framework. As long as the connected modules are provided with a connector they can be included in the platform and exchange data with other connected tools as well as download (upload) information from the common repository.

References

[1] T. Tolio and A. Matta, "A method for performance evaluation of automated flow lines," in *Annals of the CIRP*; 1998. pp. 373-376

[2] A. M. Law, Simulation modeling and analysis, Mac Graw Hill, 2007.

[3] H. Panetto, "Towards a classification framework for interoperability of enterprise applications," *International Journal of Computer Integrated* Manufacturing; 2007. Vol. 20 (8). 727-740, 2007.

[4] "VFF, Holistic, extensible, scalable and standard Virtual Factory Framework (FP7-NMP-2008-3.4-1, 228595)," [Online]. Available: http://www.vff-project.eu/.

[5] M. Sacco, P. Pedrazzoli and W. Terkaj, "VFF: Virtual Factory Framework," in *International Conference on Concurrent Enterprises*, Lugano, 2010.

[6] X. Yang, E. Deines, C. Lauer and J. C. Aurich, "A human centered virtual factory," in *IEEE International Conference MSIE*, 2011. 1138 – 1142.

[7] W. Terkaj, G. Pedrielli and M. Sacco, "Virtual Factory Data Model," in *2nd OSEMA (Ontology and Semantic Web for Manufacturing)*, Graz, 2012.

[8] buildingSMART. [Online]. Available: http://buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm.

[9] buildingSMART. [Online]. Available: http://buildingsmart-tech.org/specifications/ifc-overview.

[10] International Organization for Standardization, *ISO 10303-11:2004 Industrial automation systems and integration -- Product data representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual,* 2004.

[11] W3C, "OWL 2 Web Ontology Language," 2009. [Online]. Available: http://www.w3.org/TR/owl2-overview/.

[12] M. Sacco, G. Dal Maso, F. Milella, P. Pedrazzoli, D. Rovere and W. Terkaj, "Virtual Factory Manager," in *Lecture Notes in Computer Science*, R. Shumaker, Ed., Springer; 2011. pp. 397-406.

[13] J. Beetz, J. van Leeuwen and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*; 2009. Vol. 23, 89-101.

[14] S. Krima, R. Barbau, X. Fiorentini, R. Sudarsan and R. Sriram, "OntoSTEP: OWL-DL Ontology for STEP," NIST, NIST Interagency/Internal Report (NISTIR) - 7561, 2009.

[15] H. Schevers and R. Drogemuller, "Converting the Industry Foundation Classes to the Web Ontology Language (SKG 2005)," in *First International Conference on Semantics, Knowledge, and Grid*, 2005.

[16] G. Ghielmini, P. Pedrazzoli, D. Rovere, M. Sacco, W. Terkaj, G. Dalmaso, F. Milella and C. Boër, "Virtual Factory Manager of Semantic Data," in *Proceedings of DET2011 7th International Conference on Digital Enterprise Technology*, Athens, Greece, 2011.

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

[18] 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.

[19] L. Demir, S. Tunali and D. T. Eliiyi, "The state of the art on buffer allocation problem: a comprehensive survey," *Journal of Intelligent* Manufacturing; 2012; pp. 1-22.

[20] E. Koenigsberg, "Production lines and internal storage a review," *Management Science*; 1965. Vol. 5. 410–433.

[21] J. A. Buzzacott and J. G. Shanthikumar, Stochastic models of manufacturing systems, New Jersey: Prentice-Hall, 1993.

[22] C. T. Papadopoulos, M. E. J. O'Kelly, M. J. Vidalis and D. Spinellis, Analysis and design of discrete part production lines, New York: Springer Science & Business Media, 2009.

[23] A. Alfieri, A. Matta and G. Pedrielli, "Mathematical programming formulations for approximate simulation and optimization of closed loop systems," in *Stochastic Models of manufacturing and service operations*, Kusadasi, Turkey; 2011.

[24] R. Automation, Arena User's Guide Version 12.00.00, Rockwell Automation, 2007.