Virtual Factory Data Model

Walter TERKAJ[a], Giulia PEDRIELLI[b], Marco SACCO[a]
[a] Istituto Tecnologie Industriali e Automazione (ITIA), Consiglio Nazionale delle Ricerche (CNR)
[b] Dipartimento di Ingegneria Meccanica, Politecnico di Milano

Abstract. Manufacturing companies have to manage a large amount of data and information. This is due to both the presence of advanced sensors systems at the shop floor level and to the spreading of simulation and decision support software tools. The exchange of simulated and real data can be effective only if interoperability is guaranteed. However, most of the software tools are based on their own data models and proprietary file formats, thus jeopardizing the interoperability. This paper presents the outcomes of a European research project, named Virtual Factory Framework (VFF), that aims at developing an integrated virtual environment to enable the interoperability between software tools supporting the factory processes along all the phases of its lifecycle. The cornerstone of the proposed VFF consists in a common Virtual Factory Data Model (VFDM) that can be considered as the shared meta-language providing a common definition of the data that are shared among the software tools connected to the framework. Semantic web technologies have been adopted to develop the VFDM because of their ability in representing formal semantics, and efficiently modeling and manage distributed data.

Keywords: Virtual Factory, Data Model, Ontology, Manufacturing Systems

1. Introduction

Manufacturing has to cope with a more and more complex and evolving market, therefore factories and their production systems are required to continuously evolve so that the changes in the market demand and in the technologies can be effectively faced and exploited [1]. The digital and virtual factory paradigm [2] [3] can support the design and management of a production environment by addressing various key issues like: (1) reduction of production times and material waste thanks to the analysis of virtual mock-ups, (2) development of a knowledge repository where people can find stored information in different versions, with both advisory role and support to the generation of new knowledge, (3) improvement of workers efficiency and safety through training and learning on virtual production systems, (4) creation of a collaboration network among people concurrently working on the same project in different places [4].

The complexity of the factory design and management problem calls for support tools to effectively address all the phases of the factory lifecycle. Indeed, the Information and Communication Technology (ICT) players in the market (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. Although these tools deal with most of the factory planning and design phases, the current approaches still do not meet the demands of the industry and fail to provide all the required functionalities, mainly because of the lack of interoperability. Moreover, Small and Medium Enterprises (SME) cannot afford the present expensive PLM software suites.

The preliminary analyses highlighted the need of a novel framework for the Virtual Factory (VF) enabling a step forward in the state of the art, by describing the factory as a whole consisting of resources, processes, dependencies and interrelations, data and material flows [5]. The European project "Virtual Factory Framework" (VFF) is currently carrying out the development of a framework for the VF [6]. The Virtual Factory Framework (VFF) can be defined as "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" [7]. The VFF architecture (Figure 1) is based on three main pillars: (1) semantic Virtual Factory Data Model, (2) semantic Virtual Factory Manager, (3) decoupled Virtual Factory modules.

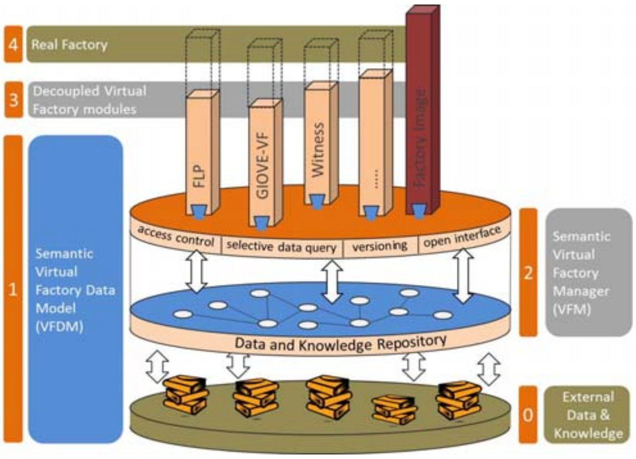
4
Real Factory
3
Decoupled Virtual Factory modules
FLP
GIOVE-VF
Witness
.....
Factory image
access control
selective data query
versioning
open interface
1
Semantic Virtual Factory Data Model (VFDM)
2
Semantic Virtual Factory Manager (VFM)
Data and Knowledge Repository
0
External Data & Knowledge

Figure 1: Virtual Factory Framework architecture

The *Virtual Factory Data Model* (VFDM) establishes a coherent, standard, extensible, and common data model for the representation of factory objects related to production systems, resources, processes and products. The common data model can be considered as the shared meta-language providing a common definition of the data and knowledge stored in the shared repository that is governed by the Virtual Factory Manager (VFM) [8] [9]. The VFM is the core of the VFF and orchestrates the *decoupled Virtual Factory modules* (*VF modules*) by providing a controlled access to the different virtual factory projects. The *VF modules* are the software tools that implement the various methods and services to support the activities related to factory design, performance evaluation, management, production monitoring, etc. The VF modules can be commercial or non-commercial applications located on a remote workstation or on the server where the VFM resides. The integration of VF modules, endowed with different functionalities and level of detail, but insisting on the same factory representation, will offer the possibility to reach a *wide range of users*.

The pillars of VFF were conceived to obtain openness, scalability and easiness by plugging in the decoupled VF modules, thus reducing the *investment costs* compared to "all-in-one" software. Moreover, the VFF aims at promoting major *time* and *operating cost* savings.

This paper focuses on the VFDM of VFF and is organized as follows. Section 2 gives an overview of the present state of the art on data models for virtual enterprises in manufacturing domain. Section 3 describes how the VFDM was developed, whereas Section 4 presents a test case. Finally, Section 5 draws the conclusions and suggests future developments.

2. Data Models for virtual enterprises in manufacturing domain

The problem of developing a comprehensive data model for the manufacturing domain has already been addressed in the literature and the main contributions can be traced back to two main approaches: standard data models and reference frameworks for information representation based on ontologies.

A lot of effort has been devoted by the data modeling research community towards the standardization of the information related to the manufacturing domain. For instance, the Standard for the Exchange of Product Model Data (STEP) [10] supports the exchange of information by aiming to create an interlingua for exchanging manufacturing product data. The Process Specification Language (PSL) [11] standard proposes a general ontology for representing manufacturing processes for the exchange of process information and knowledge. In particular, a neutral, standard language for process specification is defined to integrate multiple process-related applications throughout the manufacturing life cycle.

The Industry Foundation classes (IFC) standard by buildingSMART [12], partially based on STEP, represents an open specification for *Building Information Modeling* (BIM) data that is exchanged and shared among the various participants in a building construction or facility management project. The IFC model is structured as a set of schemas that are grouped into four layers: Resource layer (i.e. general purpose or low level concepts/objects), Core Layer (where the most abstract concepts of the model are defined), Interoperability Layer defining concepts or objects common to two or more domains, and the Domains/Application Layer. Even if the standard was mainly conceived for Architectural Engineering Construction (AEC) industry domains (e.g. Building Controls, Structural elements, Structural Analysis, etc.), its data structures can be specialized for other industrial domains, such as the manufacturing domain. The IFC standard is available as an EXPRESS schema specification [13].

Most of the aforementioned standards are focused on particular areas of the factory domain and therefore it is necessary to integrate various contributions to cover all the knowledge domains required by the VFDM, as already highlighted by Colledani et al. [14] [15] and Valente et al. [16] in previous related works.

Ontologies represent a possible way to generate a more flexible data model integrating different knowledge domains. Indeed, ontologies [17] have been developed and investigated in the field of artificial intelligence and natural language processing to facilitate knowledge sharing and reuse [18]. Ontologies are an enabler for knowledge sharing between several applications and, at the same time, they enable a fluent flow of data between different entities [19]. Beyond traditional data models like UML class diagrams or entity relationship diagrams, ontologies provide methods for integrating fragmented data models into a unique model without losing the notation and style of the individual ones [20]. In addition, the first order logic underlying ontologies ensures data consistency on a semantic level and reasoning can be performed. Ontologies are well suited to model logical relationships between different variables in axioms which can then be used to derive assertions based on available data (e.g. data coming from the shop floor).

Various ontologies have been developed to support virtual enterprise (e.g. CIMOSA [21], FDM [22], MOSES [23] and MISSION [24]) and then extended focusing on the operational scope. For instance, the TOVE project [25] and Enterprise Project [26] proposed taxonomies along with an explicit virtual enterprise modeling. However, these projects did not address the manufacturing domain.

Lin et al. [27] 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. The MSE ontology is based on seven key classes that the authors determined based on Manufacturing System (MS) information models [22] [24] [21] [28]: Project, Flow, Process, Enterprise, Extended_Enterprise, Resource and Strategy.

Léger et al. [29] presented a Manufacturing's Semantics Ontology (MASON) that is built upon three main concepts: *entities*, *operations* and *resources*. Similarly, Martin and D'Acunto [30] developed an ontology decomposed into product, process and resource areas. ADACOR (A Collaborative Production Automation and Control Architecture) could be classified as general-purpose manufacturing ontology [31].

More recently, Badra et al. [32] proposed a Semantic Web compliant representation of the objects, concepts and services related to Renault Product Range Specification (PRS) that is used to specify the set of all possible car configurations.

Baqar Raza and Harrison [33] developed an ontology aimed at the formalization of product data throughout its lifecycle, in particular for the automotive sector. The research focused on the integration of product, process and resource information from multiple sources and making the information available via web services.

Moser et al. [19] presented a tool to support real time decision making, by providing an integrated view on relevant engineering knowledge in typical design time and runtime models, which were not originally designed for machine-understandable integration.

Although the domain of the aforementioned ontologies results wider than the one which is typically handled within the previous technical standards, it can be noticed that the classes characterizing the ontologies are not imported from any of the available technical standards.

3. VFDM - Virtual Factory Data Model

The Virtual Factory Data Model (VFDM) aims at formalizing and integrating the concepts of building, product, process and production resource handled by the digital tools supporting the factory life-cycle phases. These tools typically deal with input/output data in the following domains:

- *Building*, i.e. the physical structure of the factory (e.g. walls, columns, etc.).
- *Product*, i.e. the product seen as the production output of the factory.
- *Process*, i.e. the processes that are executed by a system to directly or indirectly transform a product.
- *Production Resource*, i.e. the resources that are used by a system with the final goal of transforming the product (or a work in progress). These resources can be human operators, machines, conveyors, AGV, etc.

- *Production System*, i.e. the transformation system (e.g. manufacturing system, assembly system) that affects a product by means of physical resources and/or human resources within a process.
- *Factory*, i.e. the factory seen as whole during its lifecycle.

As highlighted in Sect.2, there is not a unique data model that is already able to cover all the listed domains. Instead of creating a brand new data model, the development of the VFDM aimed at exploiting as much as possible the already existing technical standards for manufacturing, thus trying to favor the interoperability between software tools. Therefore, the work focused on the integration (managing redundancies and inconsistencies) of various knowledge domains represented by different technical standards. This approach requires both the translation of the existing standards into a common language and the development of the required extensions to represents the concepts in the scope of the VFF.

At first, it was evaluated the opportunity of implementing the VFDM as a set of XSD files [34], thus defining the structure of the XML files that would be stored and managed by the Virtual Factory Manager (VFM). This solution allowed carrying on a successful study of the feasibility of the approach [8]. However, the XSD technology alone is not suitable for knowledge representation and an explicit characterization of data with their relations on a semantic level is missing. Moreover, intra-document references are supported but inter-document references (i.e. cross-references) are poorly modeled, thus endangering referential consistency. Therefore distributed data can be hardly managed and the integration of different knowledge domains can be cumbersome. These limitations led to evaluate and finally adopt the Semantic Web technologies [35] for developing the VFDM as an ontology [9]. Indeed, the Semantic Web technologies offer the possibility to the whole VFF to represent a formal semantics, efficiently model and manage distributed data, ease the interoperability of different applications, and exploit generic tools that can infer from and reason about an ontology, thus providing a generic support that is not customized on the specific domain. The key features of the ontology-based VFDM can be summarized as follows:

- the OWL 2.0 vocabulary [35] of the Web Ontology Language [17] was adopted to define all the classes, properties and restrictions that can be used to create the individuals to be stored in the data repository of the VFM.
- after the analysis of the state of the art, the Industry Foundation Classes [36] and STEP-NC [37] were selected to be integrated in the VFDM for modeling the factory building and the most abstract classes of objects, and for modeling the manufacturing processes, respectively.
- the selected standards were partially translated (based on the scope of the VFF) into ontologies. Indeed, an official implementation of IFC and STEP-NC as ontologies does not exist yet. In particular, the IFC standard is available as EXPRESS schemas [13]. The translation of the schemas into ontologies took as a reference the suggestions provided by Beetz et al. [38], Krima et al. [39], and Schevers and Drogemuller [40].
- the technical standards have been extended thus leading to the creation of novel data structures.

3.1. VFDM Architecture

The various knowledge domains of the VFDM are organized into *macro areas* consisting of one or more ontologies, thus creating a hierarchical structure of ontologies while decomposing the problem and downsizing its complexity. The VFDM defines only the so-called meta data (i.e. the classes, properties and restrictions), whereas the actual instances (i.e. the OWL individuals) will be stored in the data repositories. Table 1 lists the ontologies contained in each VFDM macro area, thus representing the VFDM complete architecture that is shown in Figure 2 as well. In summary, the VFDM consists of 593 classes, 508 object properties and one data property. The following sub-sections present an overview of the VFDM macro areas, highlighting their main features.

The following rules were adopted while creating the VFDM:

- The ontologies containing only novel definitions of classes have a name with prefix "Vff" (e.g. *VffComons*), whereas the ontologies that are created by importing/transforming third party ontologies or technical standards have a name with prefix equal to the acronym of the source (e.g. "Ifc" for IFC standard, "StepNc" for STEP-NC standard).
- Novel classes are named with prefix "Vff" (e.g. *VffDataType*), whereas classes that are imported from a technical standard are named with a prefix after the standard itself (e.g. *IfcProject* is imported from IFC standard).

Table 1: Macro Areas in the VFDM

VFDM – Macro Area	**Ontologies**
Commons	VffCommons;IfcActorResource, IfcCostResource, IfcDateTimeResource, IfcUtilityResource, IfcRepresentationResource, IfcExternalReferenceResource, IfcTopologyResource, IfcGeometricConstraintResource, IfcGeometricModelResource, IfcGeometryResource, IfcMaterialResource, IfcMeasureResource, IfcPropertyResource, IfcQuantityResource; IfcKernel, IfcProductExtension, IfcProcessExtension, IfcControlExtension; VffStochasticResource, VffFailureResource
Building	IfcSharedBldgElements
Product	StepNcAP10, VffProduct
Process	VffProcess
Production Resource	VffProductionResource
Production System	VffSystem
Factory	VffFactory

Given the large number of ontologies that were imported a special attention has been dedicated to avoid the generation of redundancies and inconsistencies between classes derived from different standards. If redundancy could not be avoided, then equivalent classes have been defined or hierarchical relationships have been established between classes by defining multiple inheritances.

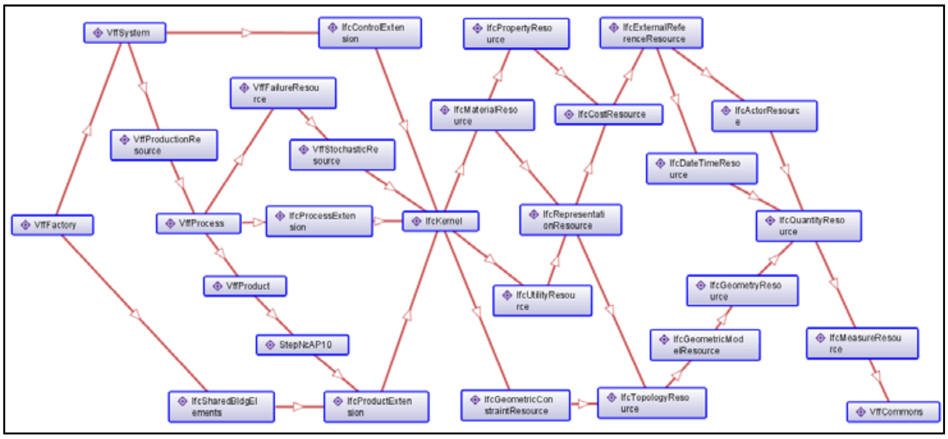

Figure 2. VFDM Architecture

3.2. Commons

The *Commons* area contains the basic definitions exploited by the other ontologies in the VFDM. The ontology in the *Commons* area are organized into sub-areas according to their source and content:

- VFF Commons (Sect.3.2.1) and VFF Resource (Sect.3.2.2) containing novel ontology developed in the scope of VFF.
- IFC Resource (Sect.3.3.3) and IFC Core (Sect.3.3.4) containing ontologies derived from the IFC standard.

3.2.1. VFF Commons

The *VFF Commons area* consists of the *VffCommons* ontology and models the common classes and properties that are imported and specialized by the ontologies belonging to the other VFDM areas. The *VffCommons* ontology defines three basic classes: *VffDataType, VffList* and *VffEnumeration*. *VffDataType* class is specialized by subclasses *VffBoolean, VffInteger, VffLogical, VffNumber, VffReal* and *VffString*.

3.2.2. VFF Resource

The *VFF Resource* area consists of the novel ontologies *VffStochasticResource* and *VffFailureResource* defining fundamental concepts not included in the IFC standard.

VffStochasticResource ontology models the stochasticity characterizing the objects represented within the data model (e.g. machine processing times, failures distributions, etc.). Two superclasses are defined within this ontology: *VffProbabilityDistribution* and *VffStochasticQuantity*. *VffProbabilityDistribution* represents a general probability distribution and is specialized by *VffDiscreteProbabilityDistribution* and *VffContinuousProbabilityDistribution* to model the main discrete (e.g. Bernoulli, Poisson, Geometric) and continuous probability distributions (e.g. Beta, Exponential, Gamma, Log-normal, Normal, Triangular, Uniform, Weibull).

VffStochasticQuantity is a generic class that represents complex or simple quantities that are sampled from a probability distribution and can be associated with an actual value and a unit of measurement. The subclass *VffStochasticQuantityTime* allows defining a stochastic time quantity (e.g. a stochastic time to failure of a machine tool). A generic time quantity can be defined as an individual of the superclass *VffTimeValueSelect* that is specialized by classes *IfcQuantityTime* and *VffStochasticQuantityTime* for deterministic and stochastic time, respectively.

VffFailureResource ontology defines two main classes: *VffFailureMode* and *VffFailure*. *VffFailureMode* represents a generic failure mode, i.e. the manner in which a component, system or process could potentially fail to meet or deliver the intended function(s). *VffFailure* models the occurrence of a specific failure event.

3.2.3. IFC Resource

The IFC Resource area defines supporting data structures and consists of 14 ontologies that are created by importing the IFC schemas belonging to the Resource Layer of the release IFC2x4 RC2 [36]. The ontologies in this area import the ontology from the *VFF Commons area* and are imported by all the ontologies contained in the other areas, included the ontology derived from the standard Step-NC within the *Product area*.

3.2.4. IFC Core

The *IFC Core area* defines the basic structure, fundamental relationships and common concepts of IFC that are then specialized by the other VFDM areas. The IFC Core area consists of four ontologies (i.e. *IfcKernel*, *IfcProductExtension*, *IfcProcessExtension*, *IfcControlExtension*) that are created by transforming with a one-to-one mapping the IFC schemas belonging to the Core Layer [36].

The *IfcKernel* ontology defines the most abstract classes that are based on three generic concepts like *object*, *property* and *relationship*. The abstract class *IfcRoot* is specialized by its subclasses *IfcObjectDefinition*, *IfcRelationship* and *IfcPropertyDefinition*. *IfcObjectDefinition* class is the generalization of any semantically treated thing or process, either being a type (i.e. *IfcTypeObject*) or an occurrence (i.e. *IfcObject*) (Figure 3). *IfcRelationship* class is the generalization of all relationships between things that are treated as objectified relationships and is specialized by subclasses *IfcRelAssigns*, *IfcRelAssociates*, *IfcRelConnects*, *IfcRelDeclares*, *IfcRelDecomposes*, *IfcRelDefines*. *IfcPropertyDefinition* is the generalization of all characteristics that may be assigned to objects.

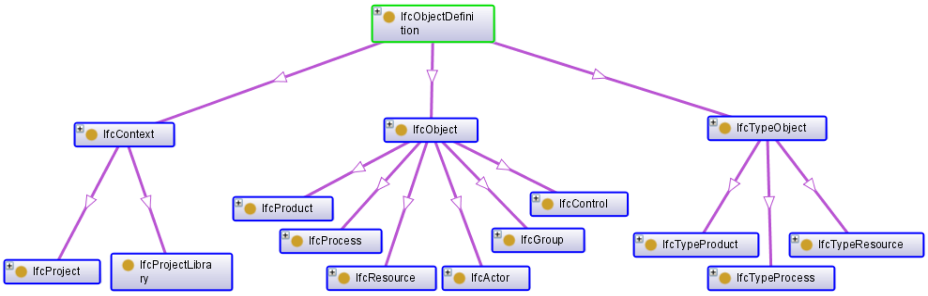

Figure 3. *IfcObjectDefinition* class and its subclasses

IfcKernel ontology is imported by *IfcProductExtension*, *IfcProcessExtension*, and *IfcControlExtension* ontology. *IfcProductExtension* ontology further specifies the concepts of a (physical) product, i.e. a component likely to have a shape and a placement; for instance, *IfcElement* is a subclass if *IfcProduct* and represents a generic component that can be contained in a spatial structure, whereas *IfcElementType* is a subclass of *IfcTypeProduct* and represents a generic type of component.

3.3. Building

The *Building area* consists of the *IfcSharedBldgElements* ontology that imports the *IfcProductExtension* ontology. *IfcSharedBldgElements* ontology models the shared building elements (e.g. wall, beam, column, slab, roof, stair, ramp, window, door and covering) that are the main components of the raw building. Two main classes are defined: *IfcBuildingElement* and *IfcBuildingElementType*. The former is a subclass of *IfcElement*, the latter is a subclass of *IfcElementType* (Sect.3.2.4).

3.4. Product

The *Product area* consists of the *StepNcAP10* and *VffProduct* ontologies and models the data related to the product that is meant as the production goal of the factory.

StepNcAP10 ontology partially translates the STEP-NC ISO 14649-10 standard [10] into an ontology. This ontology is placed in the *Product area* of the VFDM even if it models some aspects related to production processes that are exploited by the *Process area* as well. The following classes are defined in the *StepNcAP10* ontology: *StepNcWorkpiece*, *StepNcManufacturingFeature*, *StepNcOperation*, *StepNcWorkingstep*. According to ISO 14649-10, *StepNcWorkpiece* contains the entire description of the workpiece, including material, surface condition and geometric data. *StepNcManufacturingFeature* is the superclass of all manufacturing features characterizing a workpiece. *StepNcOperation* is the superclass defining the generic data required by all operations (e.g. machining, rapid movement, touch probing). *StepNcWorkingstep* represents the essential building block of an ISO 14649 NC programme and consists in the machining process for a specified area of the workpiece.

VffProduct is a novel ontology defining the basic concepts of a factory product by importing and merging the definitions from the ontologies *IfcProductExtension* and *StepNcAP10*. Two main classes are defined: *VffProduct* and *VffProductType*. *VffProduct* is a subclass of *IfcProduct* and represents the occurrence of a generic product that is realized by a factory. An individual of *VffProduct* can be typed only by an individual of *VffProductType* class that represents a generic product type. *VffProductType* is a subclass of *IfcTypeProduct* (Sect.3.2.4) and is specialized by class *VffWorkpieceType* that is a subclass of *StepNcWorkpiece* as well, thus defining a relationship between the two imported technical standards and the novel class.

3.5. Process

The *Process area* consists of the *VffProcess* ontology and models the data and information related to the transformation processes that are executed in a factory. *VffProcess* ontology imports the *IfcProcessExtension* and *VffProduct* ontologies and defines three main superclasses: *VffProcess*, *VffProcessType*, *VffProcessProperties*.

VffProcess is a subclass of *IfcProcess* (Sect.3.2.4) and represents the occurrence of a generic transformation process. *VffProcess* is specialized by subclasses *VffManufacturingProcess*, *VffAssemblyProcess*, *VffMaintenaceProcess*, and *VffHandlingProcess*. These classes represent the main production processes that can occur in a factory (i.e. manufacturing, assembly, maintenance and handling, respectively).

VffProcessType is a subclass of *IfcTypeProcess* (Sect.3.2.4) and models generic transformation processes that can be used to define various process occurrences. VffProcessType is specialized by subclasses *VffManufacturingProcessType*, *VffAssemblyProcessType*, *VffMaintenanceProcessType* and *VffHandlingProcessType*. *VffManufacturingProcessType* is further specialized by *VffMachiningProcessType* that is a subclass of *StepNcMachiningWorkingstep* as well.

VffProcessProperties represents a set of predefined properties for a generic transformation process. The individuals of this class can be associated with individuals of both *VffProcess* and *VffProcessType*.

3.6. Production Resource

The *Production Resource area* consists of the *VffProductionResource* ontology and models the data related to the production resources that are used by a system to transform a product (or a work in progress).

VffProductionResource ontology extends the IFC standard by modeling both the physical and logical aspect of a production resource. The former is modeled by classes *VffMachineryElement* and *VffMachineryElementType*, the latter by classes *VffProductionResource* and *VffProductionResourceType*.

VffMachineryElement class is a subclass of *IfcElelment* (Sect.3.2.4) and models a generic machinery element that can be typed by individuals of class *VffMachineryElementType.* Being a physical element, an individual of *VffMachineryElement* may be associated to a shape representation and a geometric placement thanks to the restrictions inherited from its superclasses.

VffProductionResource class is a subclass of *IfcResource* (Sect.3.2.4) and the generalization of a resource occurrence used in a factory (and its production systems), mainly labor, material, equipment, product resources, and subcontracted resources and aggregations, such as a crew resource.

VffProductionResource represents the *use of something* and does not necessarily correspond to a single item (e.g. person, machine), but represents a pool of items having limited availability (e.g. raw materials, labor). Physical objects may be assigned to a production resource thanks to the objectified relationship class *IfcRelAssignsToResource* (Sect.3.2.4). For instance, an individual of *VffMachineryElement* class may be assigned to an individual of *VffProductionEquipmentResource* class.

3.7. Production System

The *Production System area* consists of the *VffSystem* ontology and deals with the description of transformation systems. *VffSystem* ontology defines *VffTransformationSystem* class as a sublclass of *IfcGroup* (Figure 3) representing a generic transformation system that is a logical aggregation of parts for a common purpose or function or to provide a service. *VffTransformationSystem* is specialized by subclasses *VffManufacturingSystem*, *VffAssemblySystem*, *VffTransportationSystem*, and *VffQualityControlSystem.*

3.8. Factory

The *Factory area* consists of the *VffFactory* ontology and models the information related to the factory during its whole lifecycle at an high level of description. *VffFactory* ontology defines two main classes: *VffFactory* and *VffFactoryLibrary*.

VffFactory is a subclass of *IfcProject* (Figure 3) and represents the root of every factory project. Some of the definitions made in a project have meaning only in its context (e.g. placement of an object), whereas other definitions can be imported from a library. More than one project can make use of the same libraries.

VffFactoryLibrary is a subclass of *IfcProjectLibrary* (Figure 3) and represents the root of every factory library. Both the individuals of *VffFactory* and *VffFactoryLibrary* are associated with the units of measurement to be used in their context. Furthermore, individuals of *VffFactory* can be associated with one or more representation contexts.

4. Test Case

This section briefly presents a test case that has been developed to evaluate how the VFDM can be employed to create factory projects and share them among different digital tools. The test case consists of six ontologies, thus exploiting the data distribution empowered by the Semantic Web approach: five factory libraries (*VffLibrary01*, *VffLibrary02*, *VffLibrary03*, *VffLibrary04*, *VffLibrary05*) and one main factory project (i.e. *VffProject01*). All these ontologies import the VFDM ontologies. The whole test case consists of 582 individuals and 1927 corresponding triples.

VffLibrary01 ontology defines a production site (as individual of class *IfcSite*) and a building (as individual of *IfcBuilding*). The building is associated with two possible shape representations in VRML and 3DS format. *VffLibrary02* ontology defines six types of machinery element (as individuals of class *VffMachineryElementType*). Each machinery element is associated with two possible shape representations in VRML and 3DS format. *VffLibrary03* ontology imports *VffLibrary02* ontology and defines one occurrence of machinery element (as individual of class *VffMachineryElement*).

VffLibrary04 ontology defines a part type (as individual of class *VffWorkpieceType*). *VffLibrary05* ontology imports *VffLibrary04* ontology and defines the process types (as individuals of class *VffManufacturingProcessType*) that must be executed to obtain the demanded part type. The process types are characterized by processing time, operations sequence and needed types of production resource (as individuals of *VffProductionResourceType*).

VffProject01 ontology contains the factory project that imports and enriches the data provided by the libraries. The factory project (as individual of *VffFactory*) defines the units of measurement, the representation context and world coordinate system where the production site and the building imported from *VffLibrary01* are placed. One manufacturing system (as individual of *VffManufacturingSystem*) is designed and consists of six occurrences of machinery element (one imported from *VffLibrary03* and five new occurrences). Each machinery element is characterized by a shape representation and by a placement. Finally, some machinery elements are assigned to the production resource types needed by the process types.

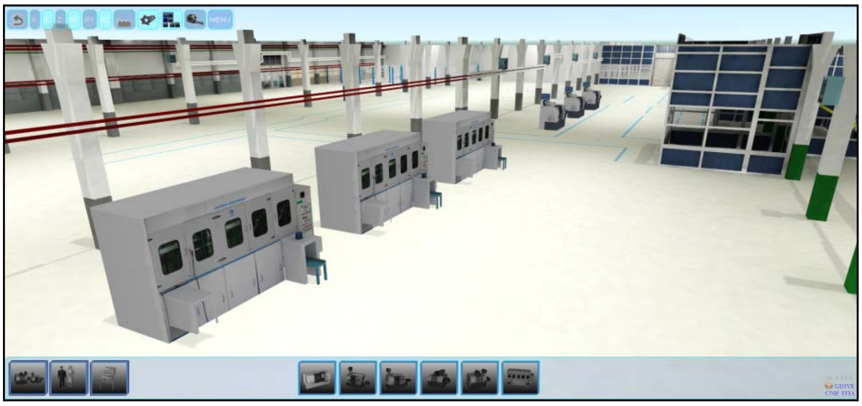

Figure 4. Test Case opened and modified by GIOVE Virtual Factory

Figure 4 shows a representation of the factory project made by a specific VF module named GIOVE Virtual Factory (GIOVE-VF) [41]. GIOVE-VF has been developed in C++ and can import/export ontologies serialized in RDF/XML format thanks to a C++ library named *VF Connector C++ Library* providing functionalities to parse, create and modify the ontologies by exploiting an internal map between OWL classes/restrictions and C++ classes/methods. The instances of C++ classes are used as handlers of the ontology individuals to support and ease the binding between the factory project individuals and the internal data structure of GIOVE-VF. The *VF Connector C++ Library* is based on the Redland C libraries [42] and makes use of an *in-memory* RDF storage.

5. Conclusions

Section 4 has shown an example of non-commercial software tool that is already able to import/export data compliant with the VFDM. However, the potentiality and applicability of the VFDM need to be further investigated. New test cases will delve into the relations between processes (e.g. *VffProcess*), production resources (e.g. *VffProductionResource*) and physical elements (e.g. *VffMachineryElement*). These new test cases will be particularly useful to check if the VFDM already provides all the data structures needed to evaluate the performance of manufacturing systems (e.g. via Discrete Event Simulation) [43]. The VFDM will be further extended by delving into the manufacturing domain, so that more business processes can be supported. VF Connector Libraries will be developed for various programming languages (e.g. C++, Java, C#). Finally, as soon as the size of Factory projects grows, the performance of *persistent RDF storage* will be evaluated and compared to *in-memory RDF storage*.

6. Acknowledgements

The research reported in this paper has been funded by the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement No: NMP2 2010-228595, Virtual Factory Framework (VFF).

7. References

[1] T. Tolio et al., "SPECIES -- Co-evolution of Products, Processes and Production Systems," CIRP Annals - Manufacturing Technology, vol. 59, no. 2, pp. 672-693, 2010.

[2] S. Yang, B. Ahn, K. Seo, "Development of a prototype customer-oriented virtual factory system," The International Journal of Advance Manufacturing Technology, vol. 28, no. 9-10, pp. 1031--1037, 2006.

[3] J. Ding, Y. Wang, K. Chen, "An Interactive Layout and Simulation system of Virtual Factory," Applied Mechanics and Materials, vol. 20, no. 23, pp. 421-426, 2010.

[4] N. Menck, X. Yang, C. Weidig, P. Winkes, C. Lauer, H. Hagen, B. Hamann, J.C. Aurich, "Collaborative factory planning in virtual reality," in Proceedings of 45th CIRP Conference on Manufacturing Systems, Athens, Greece, 2012, pp. 359-366.

[5] P. Pedrazzoli, M. Sacco, A. Jönsson, C Boër, "Virtual Factory Framework: Key Enabler For Future Manufacturing," in Digital Enterprise Technology.: Springer US, 2007, pp. 83-90.

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

[7] M. Sacco, P. Pedrazzoli, W. Terkaj, "VFF: Virtual Factory Framework," in Proceedings of ICE - 16th International Conference on Concurrent Enterprising, Lugano, Svizzera.

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

[9] G. Ghielmini, P. Pedrazzoli, D. Rovere, W. Terkaj, G. Dal Maso, F. Milella, M. Sacco, C.R. Boer, "Virtual Factory Manager of Semantic Data," in Proceedings of DET2011 7th International Conference on Digital Enterprise Technology, Athens, Greece, 2011.

[10] WZL RWTH Aachen. ISO 14649 – STEP-NC architecture overview. [Online]. http://www.step-nc.org/frames/iso14649_frames.htm

[11] International Organization for Standardization, ISO 18629-1:2004 Industrial automation systems and integration -- Process specification language -- Part 1: Overview and basic principles, 2004.

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

[13] 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.

[14] M. Colledani, W. Terkaj, T. Tolio, M. Tomasella, "Development of a Conceptual Reference Framework to manage manufacturing knowledge related to Products, Processes and Production Systems," in Methods and Tools for Effective Knowledge Life-Cycle-Management.: Springer, 2008, pp. 259-284.

[15] M. Colledani, W. Terkaj, T. Tolio, "Product-Process-System Information Formalization," in Design of Flexible Production Systems.: Springer, 2009, pp. 63-86.

[16] A. Valente, E. Carpanzano, A. Nassehi, S.T. Newman, "A STEP compliant knowledge based schema to support shop-floor adaptive automation in dynamic manufacturing environments," Cirp Annals-Manufacturing Technology, vol. 59, no. 1, pp. 441-444, 2010.

[17] W3C. OWL Web Ontology Language - Use Cases and Requirements. [Online]. http://www.w3.org/TR/webont-req/#onto-def

[18] B. Kulvatunyou, H. Cho, and Y.J. Son, "A semantic web service framework to support intelligent distributed manufacturing," International Journal Knowledge-Based Intellingent Engineering Systems, pp. 107-127, 2005.

[19] 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.

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

[21] K. Kosanke, F. Vernadat, M. Zelm, "CIMOSA: enterprise engineering and integration," Computers in Industry, vol. 40, pp. 83–97, 1999.

[22] J.A. Harding, A.R. Omar, K. Popplewell, "Application of QFD within a concurrent engineering environment," International Journal of Agile Management Systems, vol. 1, pp. 88–98, 1999.

[23] T.I.A. Ellis, A. Molina, R.I.M. Young, R. Bell, "The development of an information sharing platform for concurrent engineering," in Proceedings of the International Manufacturing Systems Engineering Workshop, Grenoble, France, 1994.

[24] J.A. Harding, K. Popplewell, D. Cook, "A manufacturing system engineering moderator: an aid for multi-discipline project teams," International Journal of Production Research, vol. 41, pp. 1973–1986, 2003.

[25] M.S. Fox, M. Gruninger, "On ontologies and enterprise modelling," in International Conference on Enterprise Integration Modelling Technology, Torino, Italy, 1997.

[26] M. Uschold, M. King, S. Moralee, Y. Zorgios, "The enterprise ontology," Knowledge Engineering Review, vol. 13, pp. 31–89, 1998.

[27] H.K. Lin, J.A. Harding, M. Shahbaz, "Manufacturing system engineering ontology for semantic interoperability across extended project teams," International Journal of Production Research, vol. 42, no. 24, pp. 5099-5118.

[28] J. Zhao, W.M. Cheung, R.I. Young, R. Bell, "An object oriented manufacturing data model for a global enterprise," in Proceedings of the 15th International Conference on Computer-Aided Production Engineering (CAPE ’99), Durham, UK, 1999, pp. 582–588.

[29] A. Léger et al., "The semantic web from an industry perspective," in Proceedings of the 2nd Reasoning Web Summer School, 2006, pp. 232–268.

[30] P. Martin, A. D’Acunto, "Design of a production system: an application of integration product-process," Int. J. Computer Integrated Manufacturing, vol. 16, no. 7, pp. 509–516.

[31] P. Leitão, F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control.," Computers in Industry, pp. 121-130, 2006.

[32] F. Badra, F.-P. Servant, A. Passant, "A Semantic Web Representation of a Product Range Specication based on Constraint Satisfaction Problem in the Automotive Industry," in Proceedings of the 1st International Workshop on Ontology and Semantic Web for Manufacturing, Heraklion, Crete, Greece, 2011, pp. 37-50.

[33] M. Baqar Raza, R. Harrison, "Ontological Knowledge Based System for Product, Process and Resource Relationships in Automotive Industry," in Proceedings of the 1st International Workshop on Ontology and Semantic Web for Manufacturing, Heraklion, Crete, Greece, 2011, pp. 23-36.

[34] W3C. (2004) RDF/XML Syntax Specification (Revised). [Online]. http://www.w3.org/TR/REC-rdf-syntax/

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

[36] buildingSMART. Industry Foundation Classes - IFC2x Edition 4 Release Candidate 2. [Online]. http://buildingsmart-tech.org/ifc/IFC2x4/rc2/html/index.htm

[37] 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.

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

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

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

[41] G.P. Viganò, L. Greci, S. Mottura, 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.: Springer, 2011, pp. 201-216.

[42] D. Beckett. Redland RDF Libraries. [Online]. http://librdf.org/

[43] W. Terkaj, M. Urgo, "Virtual Factory Data Model to support Performance Evaluation of Production Systems," in Proceedings of 2nd OSEMA (Ontology and Semantic Web for Manufacturing) Workshop, 2012.