# An Onto-Based Interoperability Framework for the Connection of PLM and Production Capability Tools

Maxime Lafleur[1], Walter Terkaj[2], Farouk Belkadi[3(✉)], Marcello Urgo[4], Alain Bernard[3], and Marcello Colledani[4]

[1] AUDROS Technology, 41 rue de la cite, 69003 Lyon, France
[2] ITIA-CNR, Institute of Industrial Technologies and Automation, via Bassini 15, 20133 Milan, Italy
[3] Ecole Centrale de Nantes – IRCCyN – UMR CNRS 6597, 1 rue de la Noë, 44300 Nantes, France
farouk.belkadi@irccyn.ec-nantes.fr
[4] Department of Mechanical Engineering, Politecnico di Milano, via La Masa 1, 20156 Milan, Italy

**Abstract.** This paper proposes a model-driven interoperability framework as a technical support of co-evolution strategy of product structure and production systems with a frugal innovation perspective. Based on the modularity concept, the role of this framework is to connect possible product modules managed in the Product Life cycle Management tool to all possible production capabilities managed on the Manufacturing Process Management tool, and able to realize each module. This will help designers to define the optimal product architecture based on technical features of modules regarding the functional requirements as well as the optimal production strategy.

**Keywords:** Interoperability · Co-evolution · Modularity · Production capabilities

## 1 Introduction

To reach economic sustainability in context of hard competitiveness, companies extend their resource optimization strategy across the whole value chain, including the consideration of product, processes and production systems co-evolution as well as change propagation and decisions' interdependencies. According to Tolio et al. [1], the term "co-evolution" represents the "ability to strategically and operationally manage the propagation of engineering changes to gain competitive advantage from the resulting market and regulatory dynamics".

The success of any co-evolution strategy should be based on robust models ensuring the global consistency between all development stages, from product design to production network configuration. Modularity concept is laid out in the literature as a potential solution to handle co-evolution [2] because of its property to decompose a complex system into independent but interconnected parts that can be treated as conceptual, logical, physical or organizational units. This is the scope of the modular approach developed in ProRegio project to support frugal innovation process [3].


R. Harik et al. (Eds.): PLM 2016, IFIP AICT 492, pp. 134–145, 2016.
DOI: 10.1007/978-3-319-54660-5_13

According to this approach, a critical task on using modularity for co-evolution perspective is the definition of the mapping strategy between designed product modules and related possible production capabilities, which inform about the standard manufacturing processes and technologies able to provide the designed module with the desired features.

In practice, the problem of mapping between product modules and related production capabilities can be extended to the more generic problem of system interoperability that aim to support the communication between two or more systems at conceptual, organizational and/or technical level [4]. The first category of tools is dedicated to store and manage product architectures and features, based on a modular organization. PLM systems are the main accepted tool for such a topic. The second category of tools aims to support process and manufacturing data. Manufacturing process management (MPM) and Enterprise Resource Planning (ERP) are currently used to reach this goal. Thus, the mapping of product modules and production capabilities requires the connection of PLM systems and MPM/ERP tools. In addition, even if a PLM tool includes few manufacturing data, the adopted semantic is not already conforming to the semantic used by the MPM. The success of any co-evolution strategy requires then robust semantic interoperability between the concepts used by involved software tools.

This paper investigates how a semantic interoperability framework (based on the Virtual Factory Framework - VFF [5]) can be integrated with a commercial PLM tool (i.e. Audros PLM [6]) to facilitate the exchange of information needed to support a co-evolution strategy between product architecture and production system. Audros solution is French PLM systems which provide a set of flexible tools able to be adapted to any functional domain through an intelligent merge of the business process model, the data model generator and the user interface design.

The next section presents a survey of the literature related to the problem of interoperability between enterprise software; the VFF and PLM tools taken into considerations are also described. Section 3 describes the proposed model-driven approach used for the definition of the connector framework. The last section describes the mapping and query mechanisms used for the implementation of the interoperability between the RDF store (used for the assessment of production capabilities) and the Audros tool.

## 2 Literature Survey

Even if the PLM tools are greatly evolving in terms of efficiency and functionalities, there is still a gap between technologies for product design and production system design that jeopardize the realization of a proper digital factory platform. In particular, the generation and management of digital factory data is a key problem. Lee and Noh [7] suggested creating digital factory models on the basis of an information data model taking advantage as much as possible on the data schemas provided by the STEP standard. Zhai et al. [8] identified the need of a proper Virtual Factory Data Management System (VFDMS) within their Integrated Simulation Method aimed at supporting Virtual Factory Engineering. Other approaches have been proposed in the scope of the Core Manufacturing Simulation Data initiative proposed by NIST [9]. Indeed, proper enabling technologies are still being investigated by academics to fulfil the concepts sketched by the early literature works, whereas large ICT players in the market propose all-comprehensive software suites that still lack full integration [10] and/or are not affordable for a large share of industrial companies. The digital tools available in large software suites are typically missing to support an effective interoperability, both between tools belonging to different suites and sometimes also between tools of the same suite.

A solution to the interoperability problem, both at academic and industrial level, is represented by the use of Semantic Web technologies and ontology that offers advantages like (1) key enablers for interoperability, data distribution, extensibility of the data model, querying, and reasoning; (2) the re-use of general purpose technologies for data storage, consistency checking and knowledge inference. Indeed, ontologies can help to meet the goals of (meta-)modelling and interoperability [11].

Several works have proposed ontology-based approaches to facilitate the data exchange between design and other engineering activities in collaborative tasks. For instance, Barbau et al. [12] presented the OntoSTEP initiative dealing with the conversion of EXPRESS schemas to OWL (Web Ontology Language), focusing in particular on the semantic enrichment of product models by adding behavioral and functionality concepts to the more traditional geometry concepts (as in the STEP-family standard).

Ontology-based models are also proposed to support data structuring and interoperability in PLM field [13]. Panetto et al. [14] have proposed the ONTO-PDM framework, as a common core model to provide an interoperability solution between product data (encapsulated in PLM) and manufacturing process management (MPM) applications. An ontology-based approach to support the factory design is represented by the Virtual Factory Framework (VFF) [5]; Colledani et al. [15] showed how the VFF approach can be employed to support the design and performance evaluation of a production system to quickly assess its production capability.

The next section takes advantages from the ontology-based VFF approach to build a software connector mapping between product modules and production capabilities.

## 3 Semantic Mapping of Product Module to Production Capability

During the product design process, a solution is defined based on technical characteristics implementing the product functions but also on the optimal production strategy. Designers should analyze all possible modules able to answer one product function as well the related manufacturing alternatives (M.A.) producing the modules.

### 3.1 The Connector Framework

Taking into consideration the constraints coming from production technologies during the product design process has a potential benefit but, within the PLM system, the M.A. must be connected to a product module and characterized by a list of production resources (a set of machines, a real system, etc.) able to manufacture the module. Therefore, the production capabilities linked to a M.A. must be considered from the earlier times of the design stage, to help the selection of the best product modules. This implies the need to assess the production capabilities for each M.A. of a single module, but also for its integration into the whole production network.

Herein, the data about a production system are formalized according to an ontology-based data model, named Virtual Factory Data Model (VFDM), that was first developed in the scope of VFF project to formalize the concepts of building, product, process and resource while taking into consideration geometric, physical and technological characteristics of the factory that are required to support its planning processes [16]. The VFDM acts as a semantic repository supporting MPM applications to represent the characteristics and classify the types of a manufacturing system in terms of products types, manufacturing processes, and involved resources. The current version of the VFDM [17] is mainly based and extends Industry Foundation Classes (IFC) standard [18], automatically converted from an EXPRESS schema specification into an OWL ontology (named *ifcOWL*) [19].

Thus, a production system is formalized as an RDF graph according to the VFDM ontology. RDF graphs can be serialized into files or can be stored into more advanced storage solutions (named triplestore or RDF store) providing also remote access and SPARQL endpoint to ease the input/output data exchanges (Fig. 1). The VFDM facilitates the interoperability between different software tools if they are endowed with a specific a software connector taking care of input/output data conversion from the ontology format to proprietary data structures and vice-versa.

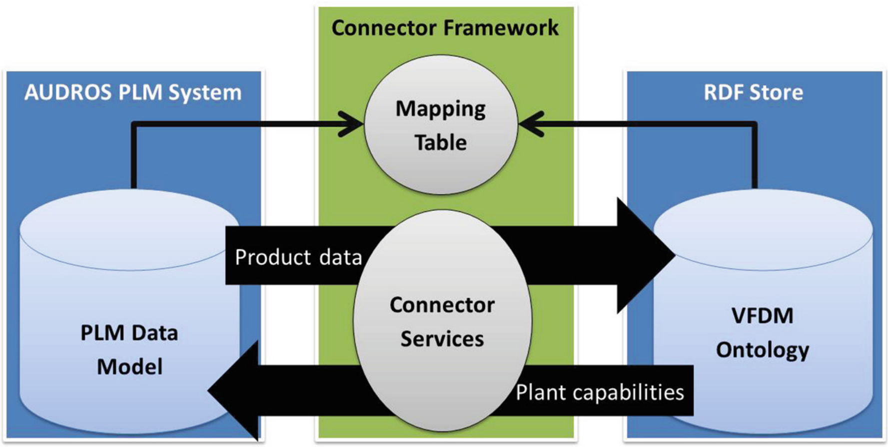


**Fig. 1.** Global interoperability scheme

Since both the scope of product design and production system design are very broad, the mapping of data and the interactions between the two systems of data storage must be limited and flexible. For instance, data will be transferred from the PLM system to the RDF store as soon as a M.A. or a product module is created or modified. Specifically, in a PLM system, the completion of the validation workflow to endorse and formalize the new (or first) version of an M.A is the right moment to disseminate information within the connected Information Systems.

## 3.2 Semantic Interoperability Between Audros and VFDM-Based Models

Semantic interoperability concerns the definition of concepts and relationships supporting the communication between heterogeneous data models. For this, the concept matching strategy and data exchange processes are defined in this section.

### 3.2.1 Exchange Process Definition

Based on the specification presented in previous paragraphs, two processes of communication must be defined, the import of the capabilities into the PLM system and the export of product and M.A. information to an RDF store having VFDM as reference ontological model, which includes classes, types and rules.

The import is triggered by the product specification validation and must take place before the design office receives the go for product definition. The systems will query production capabilities for each M.A. of each module compatible for every function the product will answer. A query will consist of the interrogation of the RDF store for currently capable systems. It will retrieve which production systems are able to execute a process plan, i.e. all its nested process steps.

The query, as is, executes the assessments taking in consideration all the process plans and production systems that are defined in the ontological model. The query will return a table containing, for each line, a production system, a process plan and a Boolean variable representing the capability.

Finally, the export of object types is required to extend the content of the RDF store. The module, as a product, and the M.A. as a process plan is characterized by the M.A. operations (reference, machines, and operation time) and Operations sequence.

### 3.2.2 Concept Matching and Association

The PLM system can feed the RDF store with information related to the product, its processes and their operations details. Consequently, we naturally make the following associations, while referring to classes defined in the ifcOWL version of IFC standard

- Product modules will be mapped to non-abstract subclasses of `IfcTypeProduct`
- The manufacturing alternatives will be mapped to `IfcTaskType` or its subclasses
- Operations will be mapped to the `IfcTask` or its subclasses.

Every object will be identified by a concatenation of its reference, version and revision. In addition, in order to deduce the feasibility of an operation in a particular system, the instances of `IfcTask` will need to be characterized by the duration of the operation, and the machine or type of machine capable of doing it. Modules or products are exported only to characterize what a manufacturing alternative realize. Finally, the instances of `IfcTaskType` are mostly characterized by their operations. This considerations lead to the mapping shown in the diagram of Fig. 2.

A final mapping issue concerns the relation between concepts. It is relatively easy to identify equivalent concepts between the two systems, but the relationships are managed in a different way. For instance, while in the PLM system there is a direct link between a M.A. and its module, in the ifcOWL ontology the link between an instance of `IfcTypeProduct` and an instance of `IfcTaskType` must go through an instance of the objectified relationship `IfcRelAssignsToProduct`.

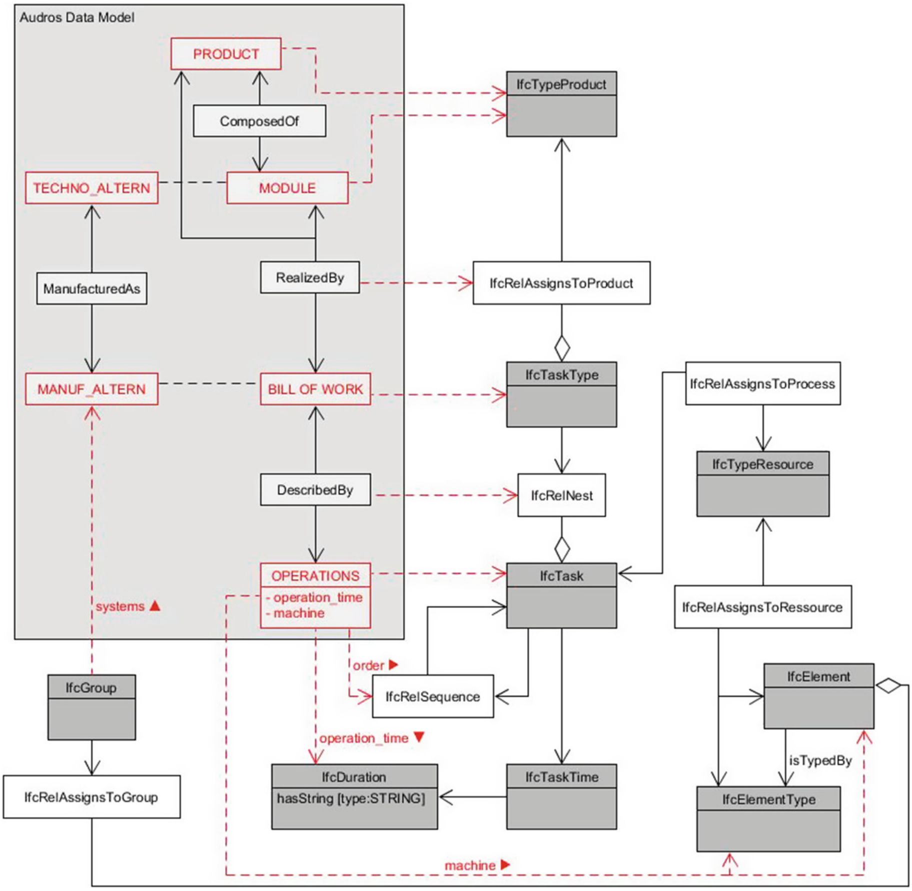


**Fig. 2.** Semantic interoperability between Audros data model and VFDM ontology mainly based on ifcOWL.

Therefore, the following mapping rules are specified in the proposed method:

- To declare a module:
  - An instance of a subclass of `IfcTypeProduct` is declared
- To declare a M.A.:
  - An instance of `IfcTaskType` is declared to represent a process plan.
  - An instance of the objectified relationship class `IfcRelAssignsToProduct` is generated and linked to the instances of `IfcTypeProduct` and `IfcTaskType`.
- To declare operations:
  - Each of operation is declared as an instance of `IfcTask` (or a subclass) to represent the steps in the process plan.
  - An instance of the objectified relationship class `IfcRelNest` is also declared to link the process plan to its operations.

– The order of operation is extracted from PLM process structure.
– An instance of the objectified relationship class `IfcRelSequence` is created to link each pair of direct predecessor/successor operations.

# 4 Implementation Strategy Based on SPARQL

If the employed RDF store provides a SPARQL endpoint, then it is possible to query and update the knowledge base remotely using the SPARQL protocol. SPARQL [20, 21] is a query language developed and validated by the W3C (World Wide Web Consortium) allowing reading and writing semantic queries on RDF stores.

Using the SPARQL protocol, this section presents the implementation process of different mapping rules defined in the Sect. 3.2.2. The first step concerns the definition of mapping variables that implement concepts' mapping. These variables are used for the declaration of prefixes and objects. The last stage concerns the characterization of operations necessary for the assessment of production capabilities.

RDF syntax consists of **triples** composed of a **subject**, a **predicate** and an **object**. A **subject** possesses a property represented by the **predicate** and for which the value is the **object**. A set of triples is called a **graph**. In SPARQL we use such triples to describe the information we are looking for or the information we want to insert. The following query examples refer to the latest version of VFDM that is based on the ifcOWL ontology. The OWL language is based on and extends RDF.

## 4.1 Product and Related Processes Data Export to "Feed" the RDF Store

To implement the semantic mapping with the SPARQL language, mapping variables are defined based on PLM and RDF store objects ID (Table 1). Except for `moduleID`,

**Table 1.** Definition of mapping variable

| Variable name | Description |
|---|---|
| `graphName` | URI of the graph where to insert data |
| `moduleID` | Concatenation of the PLM object reference, version and revision to identify the instance of `ArtifactType` (a subclass of `IfcTypeProduct`). |
| `manufAlterID` | Concatenation of the PLM object reference, version and revision to identify the instance of `IfcTaskType`. |
| `opeID` | Concatenation of the PLM object reference, version and revision to identify the instance of `IfcTask`. |
| `relProdProcID` | ID of the relation object between a module/product and its M.A. |
| `relProcOpeID` | ID of the relation object between a M.A and its operations. |
| `relSeqOpeID` | ID of the relation object describing the sequence between 2 operations. |
| `durationID` | ID of the duration object characterizing an operation. |
| `tasktimeID` | ID of the time definition characterizing an operation. |
| `relOpeResID` | ID of the relation object between an operation and is production resource object. |
| `prodResID` | ID of a production resource object. |
| `relResMachineID` | ID of the relation object between a production resource and a machine. |
| `machineID` | ID of the machine as an instance of `MachineTool` (a subclass of `IfcElement`) |

manufAlterID and opeID, all the IDs characterize concepts exclusive to the RDF store. For reuse possibility those IDs will be constructed based on the objects they link and the type of property/relation they characterize. Thus it will be possible to retrieve them for further modification. (i.e. declaring new M.A for existing module).

### 4.1.1 Prefixes Definition

The prefixes below are required in the following statement; <graphName> represents a new RDF graph where data are exported from the PLM system.

```
PREFIX list: <https://w3id.org/list#>
PREFIX express: <https://w3id.org/express#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX ifc: <http://ifcowl.openbimstandards.org/IFC4_ADD1#>
PREFIX factory: <http://www.h2020-proregio.eu/Factory#>
PREFIX PROnto: <graphName>
```

### 4.1.2 Declaration of an Object

In order to declare an object in the ontology, we must declare it as a NamedIndividual and type it to the class it belongs, e.g. ArtifactType (i.e. a subclass of IfcTypeProduct) in the case of a product module.

```
PROnto:<moduleID> a owl:NamedIndividual , factory:ArtifactType .
```

**Declaration of the Link Between a Module and a M.A.** The relation is represented by an instance of IfcRelAssignsToProduct characterized by links with both module and M.A.

```
PROnto:<manufAlterID> a owl:NamedIndividual , ifc:IfcTaskType .
PROnto:<relProdProcID> a owl:NamedIndividual,ifc:IfcRelAssignsToProduct;
          ifc:relatingProduct_IfcRelAssignsToProduct PROnto:<moduleID> ;
          ifc:relatedObjects_IfcRelAssigns PROnto:<manufAlterID> .
```

**Declaration of the Link Between a M.A. and its Operations.** This relation is enabled through the creation of an instance of IfcRelNests that groups all the operations belonging to the M.A.

```
PROnto:<manufAlterID_item> a owl:NamedIndividual , ifc:IfcObjectDefinition_List.
PROnto:<opeID> a owl:NamedIndividual , ifc:IfcTask .
PROnto:<relProcOpeID> a owl:NamedIndividual , ifc:IfcRelNests ;
          ifc:relatingObject_IfcRelNests PROnto:<manufAlterID> ;
          ifc:relatedObjects_IfcRelNests PROnto:<manufAlterID_item> .
PROnto:<manufAlterID_item> list:hasContents PROnto:<opeID> .
```

Nevertheless, to transcribe the sequence, an instance of another objectified relationship class is needed: `IfcRelSequence`. This instance will link an operation to its predecessor, independently to any other association.

```
PROnto:<relSeqOpeID> rdf:type owl:NamedIndividual , ifc:IfcRelSequence ;
          ifc:relatingProcess_IfcRelSequence PROnto:<opeID_1> ;
          ifc:relatedProcess_IfcRelSequence PROnto:<opeID_2> .
```

### 4.1.3 Characterization of Operations

**Associate Time to Operation.** To characterize an operation by its duration, we declare:

- Duration instance (`IfcDuration`) carrying a string value representing the duration.
- Task time instance (`IfcTaskTime`) linked to the operation instance (`IfcTask`).

```
PROnto:<durationID> a owl:NamedIndividual , ifc:IfcDuration ;
          express:hasString "PT10S"^^xsd:string .
PROnto:<tasktimeID> a owl:NamedIndividual , ifc:IfcTaskTime ;
          ifc:scheduleDuration_IfcTaskTime PROnto:<durationID> .
PROnto:<opeID> ifc:taskTime_IfcTask PROnto:<tasktimeID> .
```

**Associate Machine/Machine Type with Operation.** To link an operation to the machine tool that can execute it, we declare:

- A production resource instance (`IfcTypeResource`).
- A machine tool instance (`MachineTool` as a subclass of `IfcElement`) or an instance of machine tool type (`MachineToolType` as a subclass of `IfcElementType`).
- Two instances of objectified relationship classes linking the operation with the production resource and then the machine.

```
PROnto:<prodResID> a owl:NamedIndividual , ifc:IfcTypeResource .
PROnto:<machineID> a owl:NamedIndividual , factory:MachineTool .
PROnto:<relOpeResID> a owl:NamedIndividual , ifc:IfcRelAssignsToProcess;
      ifc:relatingProcess_IfcRelAssignsToProcess PROnto:<opeID> ;
      ifc:relatedObjects_IfcRelAssigns PROnto:<prodResID> .
PROnto:<relResMachineID> a owl:NamedIndividual,
                        ifc:IfcRelAssignsToResource ;
      ifc:relatingResource_IfcRelAssignsToResource PROnto:<prodResID> ;
      ifc:relatedObjects_IfcRelAssigns PROnto:<machineID> .
```

### 4.1.4 Production Capabilities Assessment

The plant capability assessment will consist of the following SPARQL query that is designed under the following assumptions:

- only one level of nesting in the process plan, i.e. the process steps of the process plan are not further nested;
- a system is a group of elements, e.g. machine tools;
- elements are not decomposed;
- a system is not decomposed into sub-systems.

Most, if not all, of these assumptions can be relaxed in the future, by upgrading the content of the query.

```
SELECT DISTINCT ?pplan ?sys (if(?npstepPP=count(distinct ?pstep),"true","false" )
as ?assessment)
WHERE {
    VALUES ?pplan {PRonto:<manufAlterID>}.a
    VALUES ?sys {PRonto:<systemID>}.b
    ?sys rdf:type/rdfs:subClassOf* factory:TransformationSystem .
    ?sys (ifc:isGroupedBy_IfcGroup|^ifc:relatingGroup_IfcRelAssignsToGroup)/
(ifc:relatedObjects_IfcRelAssigns|^ifc:hasAssignments_IfcObjectDefinition) ?elem .
    ?res (^ifc:relatingResource_IfcRelAssignsToResource)/
(ifc:relatedObjects_IfcRelAssigns|^ifc:hasAssignments_IfcObjectDefinition) ?elem .
    ?pstep (^ifc:relatingProcess_IfcRelAssignsToProcess)/
(ifc:relatedObjects_IfcRelAssigns|^ifc:hasAssignments_IfcObjectDefinition) ?res .
    ?pplan (ifc:isNestedBy_IfcObjectDefinition|
^ifc:relatingObject_IfcRelNests)/(ifc:relatedObjects_IfcRelNests)/
list:hasNext*/list:hasContents ?pstep .
    {
        SELECT ?pplan (count(distinct ?pstep) as ?npstepPP)
            WHERE {
            ?pplan rdf:type/rdfs:subClassOf* ifc:IfcTaskType .
            ?pplan (ifc:isNestedBy_IfcObjectDefinition|
^ifc:relatingObject_IfcRelNests)/(ifc:relatedObjects_IfcRelNests)/list:hasNext*/li
st:hasContents ?pstep .
        }
        GROUP BY ?pplan
    }
} GROUP BY ?sys ?npstepPP ?pplan
```

[a] This constraint allows restricting the capability assessment to a subset of process plans
[b] This constraint allows restricting the capability assessment to a subset of systems

## 5 Conclusion

The presented approach allows an asynchronous and flexible data exchange between a PLM system and an RDF store and gives an answer to the problem of interoperability between corporate information systems. The strength of this approach comes from the use of standards as the ifcOWL ontology and SPARQL language. In future work, we foresee studying the possibility to develop domain ontology to better represent manufacturing-specific concepts while taking advantage of ifcOWL and SPARQL standards.

Being developed in the project "ProRegio", the approach will be prototypically implemented together with industrial partners in several use cases. Thus, its feasibility and potentiality to support co-evolution strategy can be test on real use cases. Further works will concerns the extension of the mapping table with the ontology models to support more complex interoperability situations including other tools than MPM.

**Acknowledgments.** The presented results were conducted within the project "ProRegio" entitled "customer-driven design of product-services and production networks to adapt to regional market requirements". This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 636966. The authors would like thank the industrial partners involved in this research.

## References

1. Tolio, T., Ceglarek, D., ElMaraghy, H.A., Fischer, A., Hu, S.J., Laperrière, L., Newman, S.T., Váncza, J.: SPECIES - co-evolution of products, processes and production systems. CIRP Ann. Manuf. Technol. **59**(2), 672–693 (2010)
2. Sako, M.: Modularity and outsourcing: the nature of co-evolution of product architecture and organisation architecture in the global automotive industry. In: The Business of Systems Integration, pp. 229–253. Oxford University Press, New York (2005)
3. Belkadi, F., Buergin, J., Kumar Gupta, R., Zhang, Y., Bernard, A., Lanza, G., Colledani, M., Urgo, M.: Co-definition of product structure and production network for frugal innovation perspectives: towards a modular-based approach. In: 26th CIRP Design Conference, Stockholm Sweden, 15–17 June 2016
4. Penciuc, D., Durupt, A., Belkadi, F., Eynard, B., Rowson, H.: Towards a PLM interoperability for a collaborative design support system. In: 8th International Conference on Digital Enterprise Technology, DET, Stuttgart, Germany, 25–28 March 2014
5. Kádár, B., Terkaj, W., Sacco, M.: Semantic virtual factory supporting interoperable modelling and evaluation of production systems. CIRP Ann. Manuf. Technol. **62**(1), 443–446 (2013)
6. Audros Company. http://www.audros.fr/en/
7. Lee, K.I., Noh, S.D.: Virtual manufacturing system—a test-bed of engineering activities. CIRP Ann. Manuf. Technol. **46**(1), 347–350 (1997)
8. Zhai, W., Fan, X., Yan, J., Zhu, P.: An integrated simulation method to support virtual factory engineering. Int. J. CAD/CAM **2**(1), 39–44 (2002)
9. Bloomfield, R., Mazhari, E., Hawkins, J., Son, Y.J.: Interoperability of manufacturing applications using the Core Manufacturing Simulation Data (CMSD) standard information model. Comput. Ind. Eng. **62**(4), 1065–1079 (2012)
10. Chen, D., Kjellberg, T., Euler, A.: Software tools for the digital factory – an evaluation and discussion. In: Huang, G.Q., Mak, K.L., Maropoulos, P.G. (eds.) 6th CIRP-Sponsored International Conference on Digital Enterprise Technology. AINSC, vol. 66, pp. 803–812. Springer, Heidelberg (2010)
11. Agyapong-Kodua, K., Lohse, N., Darlington, R., Ratchev, S.: Review of semantic modelling technologies in support of virtual factory design. Int. J. Prod. Res. **51**(14), 4388–4404 (2013)

12. Barbau, R., Krima, S., Rachuri, S., Narayanan, A., Fiorentini, X., Foufou, S., Sriram, R.D.: OntoSTEP: enriching product model data using ontologies. Comput.-Aided Des. **44**(6), 575–590 (2012)
13. Bruno, G., Villa, A.: The exploitation of an ontology-based model of PLM from a SME point of view. Manuf. Model. Manag. Control **7**(1), 1447–1452 (2013)
14. Panetto, H., Dassisti, M., Tursi, A.: ONTO-PLM: product-driven ONTOlogy for product data management interoperability within manufacturing process environment. Adv. Eng. Inform. **26**(2), 334–348 (2012)
15. Colledani, M., Pedrielli, G., Terkaj, W., Urgo, M.: Integrated virtual platform for manufacturing systems design. Procedia CIRP **7**, 425–430 (2013)
16. Terkaj, W., Pedrielli, G., Sacco, M.: Virtual factory data model. In: Workshop on Ontology and Semantic Web for Manufacturing, OSEMA, CEUR Workshop Proceedings, vol. 886, pp. 29–43 (2012)
17. Pellegrinelli, S., Terkaj, W., Urgo, M.A.: Concept for a pallet configuration approach using zero-point clamping systems. Procedia CIRP **41**, 123–128 (2016)
18. Liebich, T., Adachi, Y., Forester, J., Hyvarinen, J., Richter, S., Chipman, T., Weise, M., Wix, J.: Industry Foundation Classes IFC4 Official Release (2013). http://www.buildingsmart-tech.org/ifc/IFC4/final/html/
19. Pauwels, P., Terkaj, W.: EXPRESS to OWL for construction industry: towards a recommendable and usable ifcOWL ontology. Autom. Constr. **63**, 100–133 (2016)
20. W3C. SPARQL Query Language for RDF. n.d. https://www.w3.org/TR/rdf-sparql-query/
21. Kollia, I., Glimm, B., Horrocks, I.: SPARQL query answering over OWL ontologies. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., Leenheer, P., Pan, J. (eds.) ESWC 2011. LNCS, vol. 6643, pp. 382–396. Springer, Heidelberg (2011). doi:10.1007/978-3-642-21034-1_26