Exploitation of a Semantic Platform to Store and Reuse PLM Knowledge

Giulia Bruno, Dario Antonelli, Roman Korf, Joachim Lentes, Nikolas Zimmermann

Exploitation of a semantic platform to store and reuse PLM knowledge

Giulia Bruno[1], Dario Antonelli[1], Roman Korf[2], Joachim Lentes[3], Nikolas Zimmermann[4]

[1] Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[2] USU Software AG, Karlstrasse 53, 76137 Karlsruhe, Germany
[3] Fraunhofer Institute for Industrial Engineering IAO, Nobelstr. 12, Stuttgart, Germany
[4] University of Stuttgart IAT, Nobelstr. 12, Stuttgart, Germany

Abstract. Products generate a large amount of information during their lifecycles. Small and medium enterprises are often not structured enough to enable the efficient management of such amount of information. Several tools of product lifecycle management have been developed in the last years to address this issue, but they are rarely exploited by companies, especially SMEs. The aim of our work is to present a semantic platform to integrate data along the whole product lifecycle to allow semantic search and knowledge reuse. The integration of data is realized with a reference PLM ontology, containing the main concepts and relations to describe a PLM. This ontology has a modular structure, so that it can be easily extended to describe concrete product lifecycles. An example of a real application of the semantic platform in an industrial case is reported.

Keywords: PLM, knowledge management, semantic model, ontology, UML

1 Introduction

The process from the idea for a new product over its development and production to the market is typically fragmented across different functional units, but requires input and activities from experts from a variety of disciplines using different methods and tools. This leads to a high coordination effort to synergize work and information transfer, to sub-optimal decisions, and unused knowledge as well as experiences. The resulting waste in engineering processes results in an unnecessary extension of time-to-market and time-to-production of new products and to a loss of competitiveness of companies. To tackle the resulting challenges for engineering in manufacturing companies, the amePLM (advanced platform for manufacturing engineering and PLM) project is based on an ontology that serves as an interoperable model and integrating element for an open engineering system. Furthermore, the usage of an ontology-based approach advances the information provision in activities during product creation.

An essential advantage of the application of ontologies in product development is knowledge sharing. Bradfield and Gao determined three main problem categories for knowledge sharing in the new product development (NPD) process of a manufacturing company: inappropriate information about the knowledge in the NPD process, multilingualism as well as multidisciplinary, and insufficient information provision to users [1]. By means of an ontology-based approach, knowledge sharing in NPD may be facilitated. Lutters et al. work to apply information management based on an ontological approach on design and engineering processes under special consideration of manufacturing, i.e. process planning and cost estimation [2]. Young et al. showed the benefits of applying ontologies to support knowledge sharing in PLM with a focus on manufacturing processes [3]. By using a product ontology as pivotal element, Panetto et al introduced an approach to support interoperability in Product Data Management (PDM) [4]. Matsokis and Kiritsis developed an ontology of concepts and rules to support PLM, emphasizing the product and its role in closed-loop PLM [5]. Raza et al. tested an approach building up on existing work by usage of ontologies for knowledge management [6]. Furthermore the ongoing work of Fiorentini et al. using ontologies to model the engineering data of nuclear power plants to leverage interoperability with external information systems shows the potential of an ontological approach [7]. So, the principal applicability of ontology-based approaches to PLM as in the platform amePLM (cf. [8]) has been shown, but there still is a potential for improvement in automated information provision in PLM to reduce manual efforts for information management and retrieval.

The rest of the paper is organized as follows. Section 2 describes the semantic platform for PLM knowledge structuring and reusing, while Section 3 reports the structure of the PLM ontology at the basis of the platform. Section 4 shows how the developed semantic platform can be used for semantic search and information reuse. Finally, Section 5 draws conclusions and states future works.

2 Advanced platform for manufacturing engineering and PLM

The architecture of the amePLM platform is shown in Fig. 1. It functions as a middleware by allowing the integration of information as well as it provides interfaces for several engineering modules and applications.

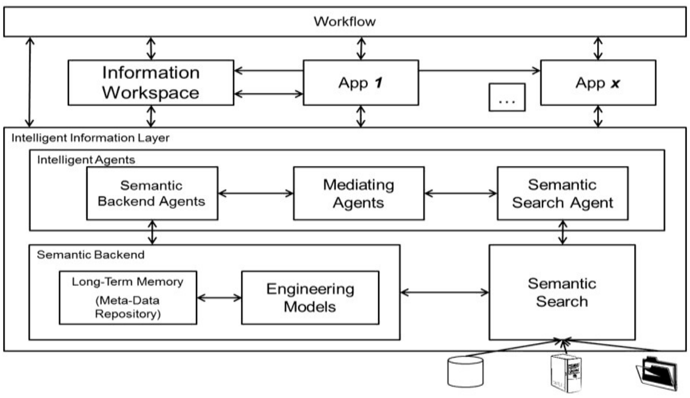

Fig. 1. Architecture of the semantic platform.

The platform is based on the semantic web framework Apache Jena [9]. The technology decision for Apache Jena is based on an original review of semantic web technology [10]. It is open source, provides a wide range of APIs to manipulate and reason with RDF and OWL ontologies and allows querying and manipulating ontologies using SPARQL 1.1 [11].

Based on the engineering models (i.e., the PLM ontology, cf. section 3), the knowledge is structured and concrete relationships between knowledge artifacts within the PLC are established. The semantic search provides access to arbitrary information sources of the PLC like databases, document management systems, CAx systems and documents on file systems. The knowledge is indexed based on the PLM ontology which directly integrates indexed knowledge artifacts into the knowledge base. An intelligent agent layer allows applications and other engineering modules to access knowledge using a single access point. The workflow orchestrates the services and knowledge access along the PLC. An example to illustrate the interaction of graphical user interface (GUI), i.e. the 3D-workspace, the working memory, the agent-based platform and the semantic backend is provided in Fig. 2.

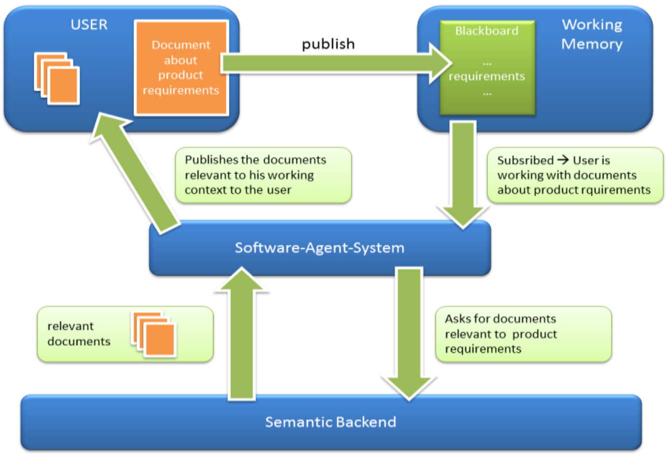

Fig. 2. Context-based user support.

In the simplified example, the user is working in a document concerning requirements of a specific product (e.g. with id “4711”). The related user input and the name of the document as well as the used software module are published to the blackboard of the working memory. By reasoning, the memory can determine that the user’s current work is concerned with product requirements of product 4711. This finding enables the consequence to support the user with related information. So, a related request is sent to the semantic backend that in turn delivers information about product requirements and about product 4711 (if available). The information or documents provided by the backend to the respective software agent can then be supplied to the user, preferably in a non-obtrusive manner, e.g. in a background window of his virtual working desktop.

3 PLM ontology

The PLM ontology at the basis of the semantic platform is shown in Figure 2, according to the UML class diagram formalism.

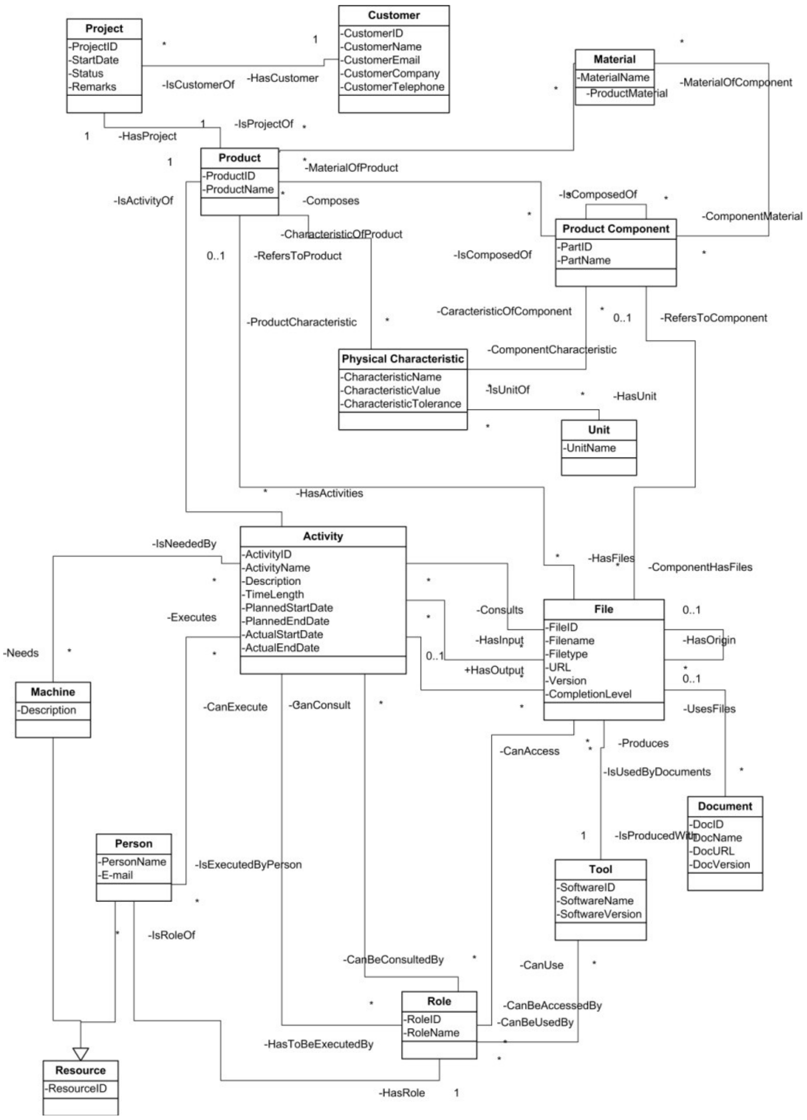

Fig. 3. UML class diagram of the PLM ontology.

The three core concepts are the product (i.e., whatever it is produced by a company which serves a need or satisfies a want), the activity (i.e., an action executed during the product lifecycle for a specific product that can be univocally be identified), and the file (i.e., an electronic information stored as a single object in the file system). A product is associated with all the activities of its lifecycle and with all the files containing information about it. There are several kinds of associations between an activity and a file, due to the different kind of usage of the file done by the activity. An activity can (i) consult a file, if it is simply read, (ii) have a file as input, if it is used and modified by the activity, and (iii) produce a new file as output.

The product class is linked to the project class because a product is developed during a project. The project class is associated with the customer class to store the involvements of customers in projects. A product is also associated to the product component class, because each product can be made of several components. Both the product and the product component are associated to the material class and to the physical characteristic class, to store their material and their characteristics of interest.

To keep trace of the people involved in the activities, the person class, which is a specialization of the resource class, is represented, which includes personal data and contacts. The other specialization of the resource class is the machine class, which represents the machines used in each activity. To each person is also assigned a role. For each activity, it is known the roles that have to execute it and the roles that are allowed to consult it. For each file the software tool exploited to produce it and the roles that can access it are known. Files or groups of files can be used to create documents. Additional information on the PLM ontology can be found in [12,13].

4 Knowledge exploring by using the semantic platform

The following section describes the access of knowledge within the semantic platform along the product lifecycle (PLC). In this example the PLC starts with a customer request for quotation where a customer specifies the requirements on the product enquiry. The responsible needs to get an overview on the details of capabilities of the company based on past projects with similar requirements to assess the feasibility.

A RFQ will result in creating a new customer project for the product 4711. The SPARLQ update statement for the RFQ populating the PLM ontology is the following:

```
PREFIX amePLM: <http://www.amePLM.org/PLM#>
PREFIX p4711: <http://www.my-company.com/amePLM/4711#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
INSERT {
     # Creating/retrieving the customer instance
     p4711:thisCustomer rdf:type amePLM:Customer.

     # Creating the new product instance for product 4711
     p4711:Product4711 rdf:type amePLM:Product.

     # Creating a new project instance
```

```
     p4711:thisProject rdf:type amePLM:Project;

     # Linking project instance with customer
     amePLM:HasCustomer p4711:thisCustomer;

     # Linking product and project
     amePLM:isProjectOf p4711:Product4711.

     # Adding some product characteristics coming from the RFQ form
     # Characteristics are product dependent.
     # Here we use the example length.
     p4711:thisPhysicalCharacteristicLen rdf:type amePLM:Length.

     # Creating unit for length; could also be retrieved by variable
     p4711:thisMMUnit rdf:type amePLM:Unit.

     # Creating instance for length 100
     p4711:thisLenValue rdf:type amePLM:Value;
            amePLM:value 100.

     # Setting the value
     p4711:thisPhysicalCharacteristicLen
            amePLM:HasUnit p4711:thisMMUnit;
            amePLM:CharacteristicValue p4711:thisLenValue.

     # Setting the characteristic for the product
     p4711:Product4711 amePLM:ProductCharacteristic
            p4711:thisPhysicalCharacteristicLen.      }
```

Integrating the information into the knowledge structure now allows us to asses this information and to compare it with requirements of previous projects. For the comparison we currently use a naïve approach by comparing the RFQ features of the current enquiry with the once of previous projects.

```
PREFIX amePLM: <http://www.amePLM.org/PLM#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT *
WHERE {
     # Find any project...
     ?project rdf:type amePLM:Project.

     # ...of products...
     ?project amePLM:IsProjectOf ?product .
```

```
# ...with length...
?length rdf:type amePLM:Length.
?product amePLM:ProductCharacteristic ?length.

# ...of characteristic value...
?valueInst rdf:type amePLM:Value.
?valueInst amePLM:value ?value.

# ... with average deviation of 10%...
# Here we put in the concrete value 100 to reduce query length.
# Usually an agent would query the value of the current project
# 4711 and use it for the query to retrieve similar projects.
Filter (xsd:integer(?length) > 100*0.9)
Filter (xsd:integer(?length) < 100*1.1)         }
```

Once the similar projects have been collected, their information gathered along the PLC is to be assessed. The following simplified query retrieves the documents containing information about a previous project with product 4710:

```
PREFIX amePLM: <http://www.amePLM.org/PLM#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT *
WHERE {
  # Find any project...
  ?project rdf:type amePLM:Project;
       # ...of products...
       amePLM:IsProjectOf p4710:Product4710 .

  # ...with product files...
  p4710:Product4710 amePLM:HasFile ?file.

  # ...and some details if they exist in the knowledge base.
  OPTIONAL {?file amePLM:Filename ?fileName.}
  OPTIONAL {?file amePLM:Filetype ?fileType.}
  OPTIONAL {?file amePLM:URL ?fileURL.}              }
```

This short set of over-simplified queries has the purpose of giving a brief overview on how the amePLM approach explores knowledge stored in the amePLM platform and how to apply simple heuristics using software agents.

5 Conclusions

In this paper we have described a semantic platform to structure the PLC knowledge of companies. The current version of this platform has been successfully applied in an industrial use-case and proven to be a feasible approach. It provides basic functionality necessary to support information needs within the PLC and it already integrates workflow, information workspace and other engineering modules with the information layer. However, it still needs improvements for extensibility and adoptability. One of the improvements will be the simplification of the integration of engineering modules (e.g. applications for CAx or simulation) and knowledge sources (e.g. product data management, customer feedback or failure statistics). Future works will also consider the integration of a semantic search engine into the platform.

Acknowledgments

The research presented in this paper is supported by the EU-FP7 research project on Advanced Platform for Manufacturing Engineering and Product Lifecycle Management (amePLM, contract number 285171).

References

1. Bradfield, D. J., Gao J. X.: A methodology to facilitate knowledge sharing in the new product development process. International Journal of Production Research 45, 1489-1504 (2007)
2. Lutters, E., Brinke, E. T., Streppel, T., Kals, H.: Information management and design & engineering processes. International Journal of Production Research 38, 4429-4444 (2000)
3. Young, R. I. M., Gunendran, A. G., Cutting-Decelle, A. F., Gruninger, M.: Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies. International Journal of Production Research 45, 1505-1519 (2007)
4. Panetto, H., Dassisti, M., Tursi, A.: ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. Journal Advanced Engineering Informatics 26, 334-348 (2012)
5. Matsokis, A.; Kiritsis, D.: An ontology-based approach for Product Lifecycle Management. Computers in Industry 61, 787-797 (2010)
6. M.B. Raza, T Kirkham, R. Harrison, Q Reul . Improving Manufacturing Efficiency at Ford Using Product Centred Knowledge, Digital Information Management, 2009. ICDIM 2009.
7. Xenia Fiorentini, Thomas Paviot, Virginie Fortineau, Jean-Louis Goblet, Samir Lamouri, Modeling nuclear power plants engineering data using ISO 15926, Industrial Engineering and Systems Management (IESM), Proceedings of 2013
8. J. Lentes. H. Eckstein N. Zimmermann, A platform for information and decision support in engineering, 22nd International Conference on Production Research (2013)
9. https://jena.apache.org/
10. Fielding, R. T. , Taylor, R. N.: Principled Design of the Modern Web Architecture, ACM Transactions on Internet Technology, Vol. 2, 115-150 (2002).
11. http://www.w3.org/TR/sparql11-query/
12. Antonelli, D., Bruno, G., Schwichtenberg, A., Villa, A.: Full exploitation of Product Lifecycle Management by integrating static and dynamic viewpoints. IFIP Advances in Information and Communication Technology 398 (part 2), 176-183 (2013).
13. Bruno, G., Villa, A.: The exploitation of an ontology-based model of PLM from a SME point of view. IFAC Proceedings Volumes (IFAC-PapersOnline), 1447-1452. (2013).