Full exploitation of Product Lifecycle Management by integrating static and dynamic viewpoints

Dario Antonelli[1], Giulia Bruno[1], Antonia Schwichtenberg[2], Agostino Villa[1]

[1] Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy
[2] Ontoprise GmbH, An der RaumFabrik 33a, 76227 Karlsruhe, Germany

Abstract. Even if PLM offers a wide range of functionalities, they are currently not fully exploited by most of the companies, which use it mainly as a file manager. In this paper we aim at helping the full exploitation of PLM systems. To this aim, we propose a model of the product lifecycle management in the form of an ontology integrating both the static structure of product's data and the dynamic description of the related processes.

Keywords: PLM modeling, information integration, ontology, UML, IDEF0

1 Introduction

Initially, to control data proliferation in design activities, Product Data Management (PDM) systems were created, which allowed data integration and secured files exchange. Within PDM the focus was on managing and tracking the creation, updating and storage of all information related to a product (e.g., CAD models, FE simulations and administrative documents). Product Lifecycle Management (PLM) is the logical evolution of PDM systems. PLM integrates people, data, software, processes and business systems to provide a complete product information backbone [16]. However, the concept of PLM is not completely received by companies, which do not exploit the full potentiality of PLM systems but use them only for data management functionalities. Especially for small and medium enterprise the exploitation of PLM is a problem due both to the lack of business process models and strict roles separation.

The paper provides an approach to allow a better exploitation of PLM systems by integrating static and dynamic viewpoints of product lifecycle. Particularly, we propose a model to contemporarily represent the data structure of PLM, thus allowing interoperability and data exchange among different systems, and the dynamic processes of product lifecycle. The model is presented in the form of an ontology, due to the ontology capability of including both the static representation (in its structure) and the dynamic view (in its instances).

Ontologies support several useful features, such as to share common understanding of the structure of information among human or software agents, to enable reuse of domain knowledge, to separate domain knowledge from operational knowledge and to provide formal analysis of terms [15]. Ontologies have already been proposed for knowledge management in product design processes [1]. Ontologies have been devel-

oped for closed-loop PLM [12], for the Open Assembly Model [6], and for STEP (Standard for the Exchange of Product Model) [18]. A product design ontology that formalizes the functionality of shape processing methods in the design workflow is defined in [2], while an ontology to manage both the product and the PLM is proposed in [15]. A method for semantic integration of PLM objects based on an integrated ontology is described in [14] and the work in [17] uses ontologies for interoperability and present a model for using data of the entire life of the products as an input for the design and production of new products. Differently from the previous works, in which the ontologies were developed starting from general concepts of the product processes or from general models of PLM, we generated an ontology from the integration of the static structure of PLM and the dynamic description of the product lifecycle processes.

The rest of the paper is organized as follows. Section 2 describes the dynamic model of product lifecycle, while Section 3 provides the static structure of PLM. In Section 4 the derived ontology is presented. Finally, Section 5 draws conclusions and states future works.

2 Dynamic model of product lifecycle

The dynamic (or behavioral) model makes explicit the processes and the interactions among entities involved in the product lifecycle. IDEF0 [4] is selected in the present study as a formalism for the dynamic representation due to two reasons: (i) it is a well-established methodology in the industrial environment, thus companies had less problems representing their processes with this formalism, and (ii) it is suitable to our scope because it allows the separate identification of resources, files and tools inside activities. Furthermore it was demonstrated that IDEF0 models can effectively complement UML models when the objective is the system design instead of the software development [3].

Product lifecycle is a complex process and it is composed by several phases. We focused on the Beginning of Product Life (BPL), trying to generalize the very detailed process of the car body and chassis design employed by an automotive industry. Since this process is among the most long and complex, it represents a good starting point to build a general model. BPL has been divided in Concept Design, Preliminary Design, Detailed Design, and Development. For sake of shortness we report in Fig.1 the overall activity diagram (A0) and in Fig.2 the details of the first activity, i.e., the Concept Design (A1). The concept design is composed of activities oriented to the definition of the schematic design of product and process and to the set-up of general project guidelines. Care should be paid to the fact that the position of blocks in the diagram does not represent the temporal sequence of activities. As an example, in the investigated companies the economic feasibility study follows the engineering feasibility study (see Fig.2).

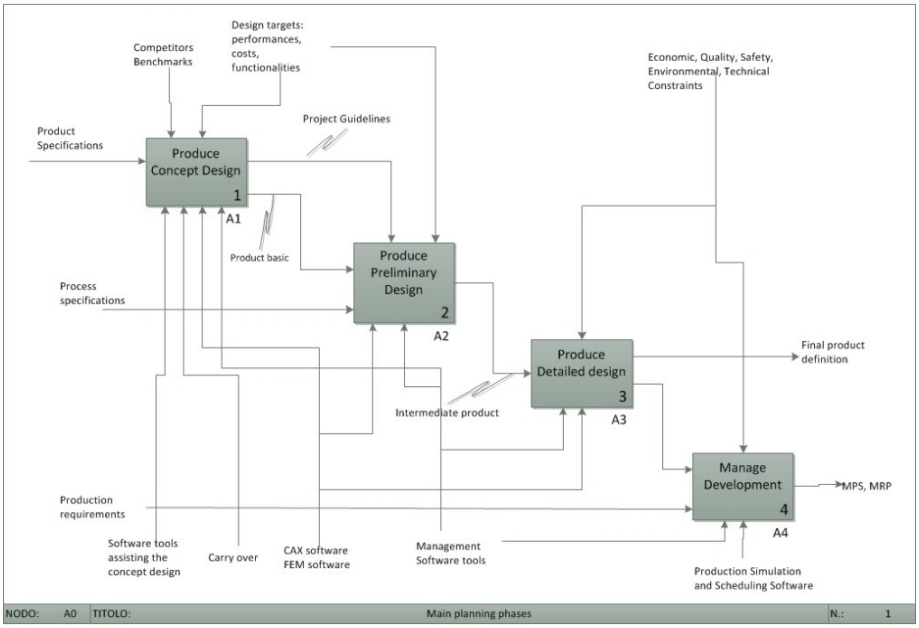


Fig. 1. IDEF0 diagram of product Beginning of Life

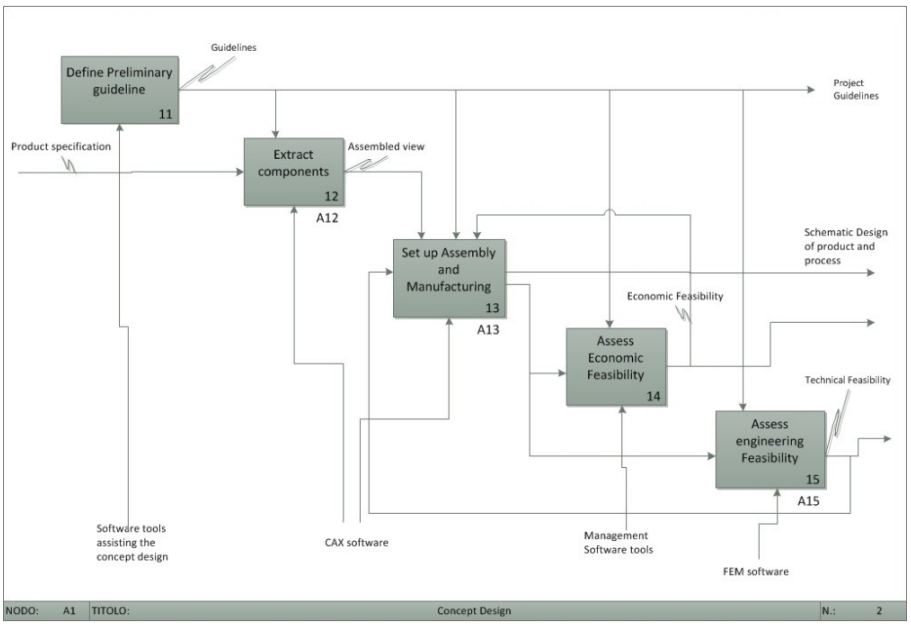


Fig. 2. IDEF0 diagram of Concept Design

3 Static model of PLM

The static (or structural) model aims at representing the structure of the system by using entities, attributes, relationships and operations. The most used formalism for structural models is the Unified Modeling Language (UML) class diagram [8], which has been already used to model PDM systems [5].

The UML class diagram of PLM is reported in Fig.3. The upper part of the diagram contains information about the product and its characteristics, while the most consistent part is focused on the product lifecycle management.

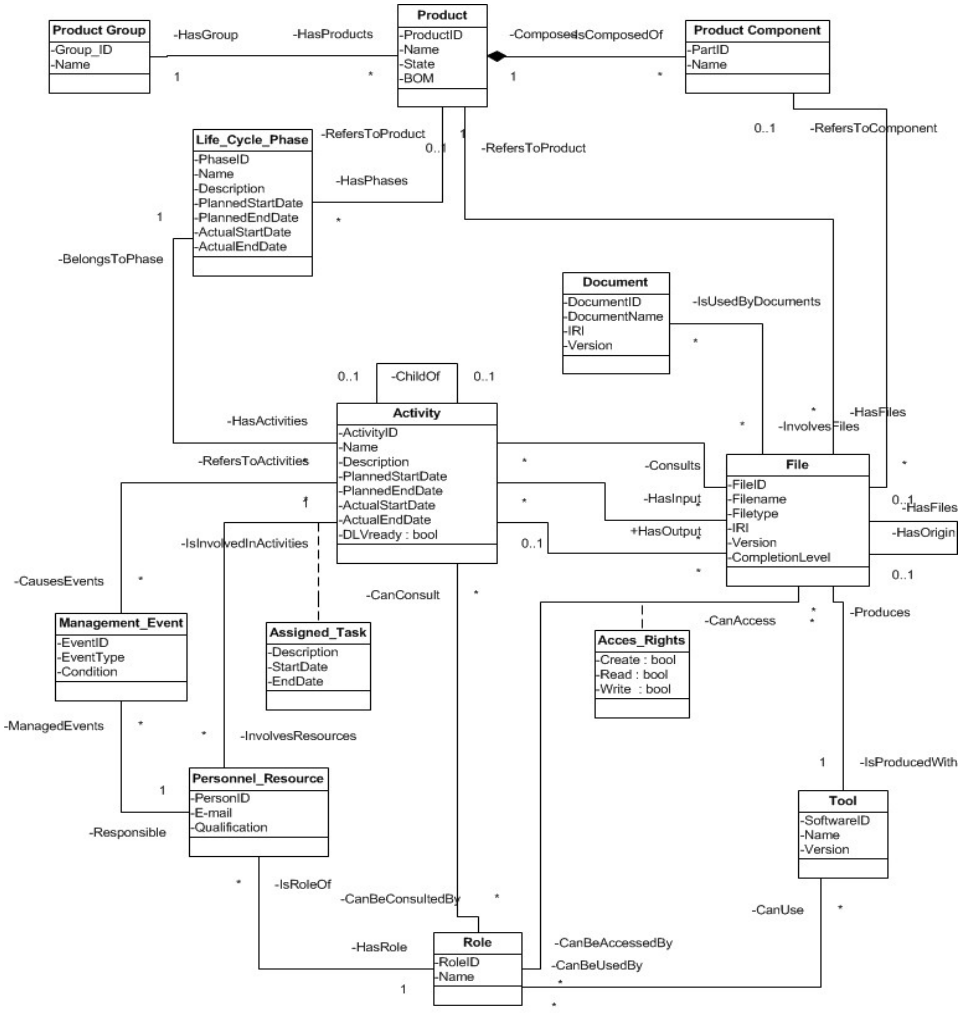


Fig. 3. UML class diagram of PLM

Since we are not interested in providing a detailed description of the product (for a more complete UML diagram of a product, the Core Product Model developed in [7] can be taken as a reference), we modeled the most important concepts of product in three classes. The class *Product* contains the information about products. It is associated to the *Product_Group* class, which contains information about the product group it belongs to. The composition association between the Product class and the *Product_Component* class represents the fact that each product can be made of several components, and each of them can have associated files that describe their state in the PLM.

The Product class is associated with the *Life_Cycle_Phase* class. The association between Life_Cycle_Phase and *Activity* means that each phase contains several activities. There are several kind of association between Activity and *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 or modified by the activity, and (iii) produce a new file as output. A file can be also connected to the product or product component it refers to. Furthermore, files or groups of files can be linked in a *Document* (e.g. a report). For each file is known the *Tool* exploited to produce it. To keep trace of the persons involved in the activities, the *Personnel_Resources* class was introduced, which include personal data, contacts, etc. For each person is specified the assigned task for each activity, to keep trace of the collaborative relationships among resources. Furthermore, to each person is assigned a *Role*, which specifies if a person can see an Activity, can use a Tool or can access to a File. In the last case, the access right can be specified as reading or writing by means of the associative class *Access_Rights*. Finally, in addition to the activities related to the product management, we also modeled the *Management_Event* class to considered all the events done to control the state of an activity (i.e., final approval of results, etc.).

4 PLM ontology

An ontology formally represents knowledge as a set of concepts, properties and relationships within a domain. Thus, it can be seen as a structural model. However, it is also possible to instantiate an ontology, i.e., assign real values to class properties and relationships. Thus, the ontology also includes information deriving from a behavioral model. Furthermore, an ontology offer additional functionalities, such as the possibility of making reasoning and logical inference among data. For these reasons we decided to resort to an ontology to provide a way to fully model a PLM system.

The ontology is formalized in ObjectLogic [9], a successor of F-logic [13]. This gives us the opportunity to add rules, to integrate easily other data sources and to query the data efficiently. The ontology model can also be exported to OWL without any lost. The ontology was developed with the OntoStudio software [10] and is reported in Fig.4.

The schema of the ontology is a direct representation of the structural model for PLM described in the UML diagram. The classes in the UML are concepts in the ontology with named relations between them having the cardinality restrictions and

the names as defined in the UML. For the attributes, it is possible to define ranges according to the XML Schema Definition [11]. So our model can be used together with any repository offering URIs for the documents stored there (e.g. by providing a REST interface).

Once the model has been developed, it is possible to fit it with the data providing from the behavioral models, i.e., to give real values to products, activities, files, resources, etc.. This forms the workflow within a concrete product lifecycle. Fig. 5 shows an example of an instantiation of our ontology for a Production Planning activity of a small industry.

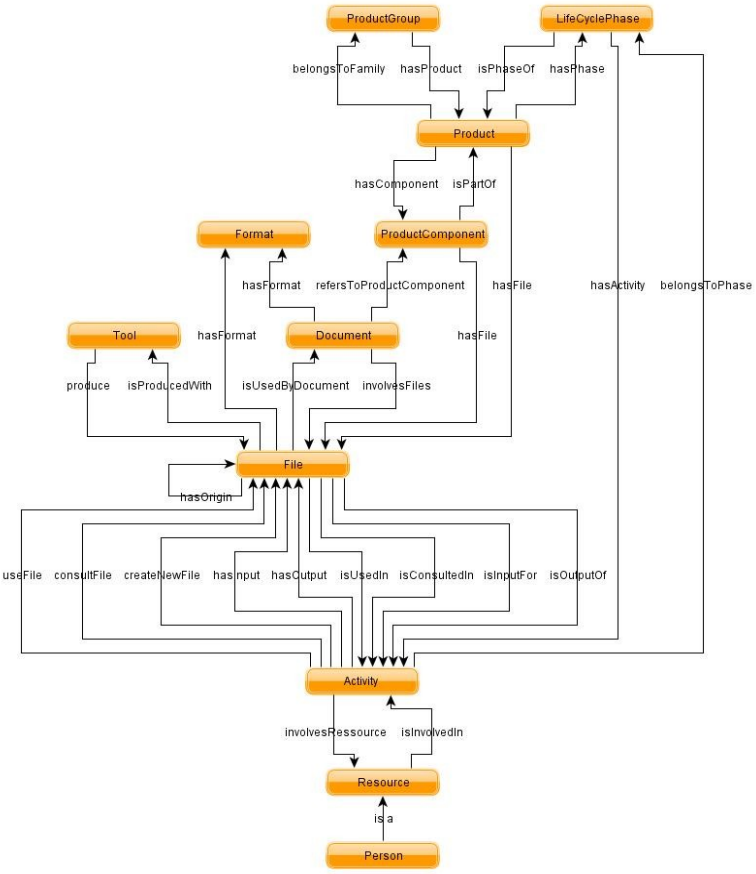


Fig. 4. Ontology of PLM

We would like to clarify that the proposed ontology aims at helping companies that would like to adopt a PLM system, and thus need to first produce a behavioral model of their process, and then to organize the information about product lifecycle among the entities and relations specified by the static UML class diagram of PLM. To have

a more complete ontology of the system, additional information arising from the specific PLM software chosen by the company should be also considered.

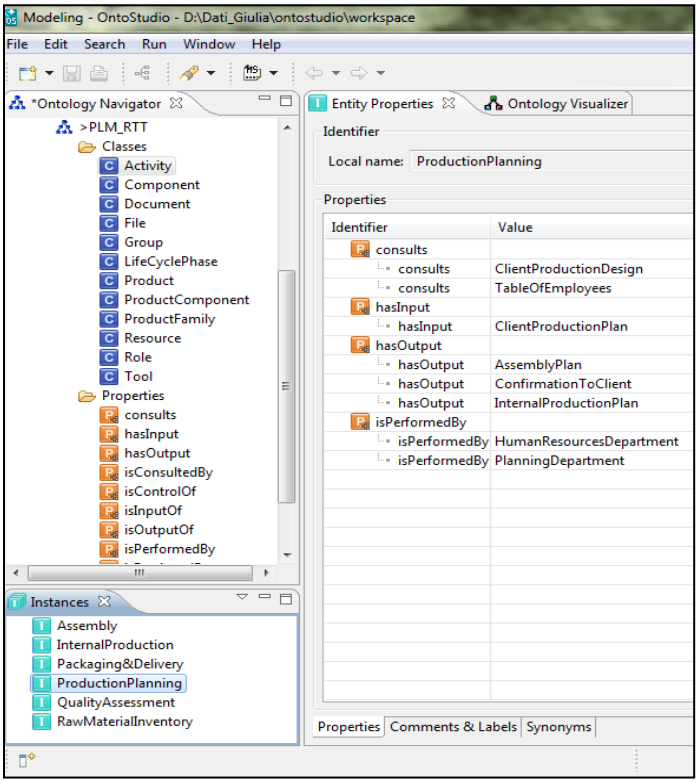


Fig. 5. Example of an instantiation of the PLM ontology for a Production Planning activity

5 Conclusions

In this paper we describe an approach to full exploit PLM systems by integrating static and dynamic viewpoints. Particularly, we propose a model of PLM in the form of an ontology, due to the ontology capability of including both the static view (in its structure) and the dynamic view (in its instances). We are currently working on applying our approach to some industrial cases to show how this general model can improve the product lifecycle management.

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).

References

1. Brandt, S.C. , Morbach, J., Miatidis, M., Theißen, M., Jarke,M., Marquardt, W.: An ontology-based approach to knowledge management in design processes. Computers and Chemical Engineering, 32, 320–342 (2007)
2. Catalano, C.E., Camossi, E., Ferrandes, R., Cheutet, V., Sevilmis, N.: A product design ontology for enhancing shape processing in design workflows. J Intell Manuf, 20, 553-567 (2009)
3. Kima, C.-H., Westonb, R.H., Hodgsonb, A., Leea, K.-H.: The complementary use of IDEF and UML modelling approaches. Computers in Industry, 50, 35–56 (2003)
4. Chen Y.-Q., Hu, J., Mo P.: The Development of The Lifecycle Function Model By IDEF0 For Construction Projects, International Conference on Wireless Communications, Networking and Mobile Computing, 1-4 (2008)
5. Eynard, B., Gallet, T., Nowak, P., Roucoules, L.: UML based specifications of PDM product structure and workflow, Computers in Industry, 55, 301-316, (2004)
6. Fiorentini, X., Gambino, I., Liang, V.-C., Foufou, S., Rachuri,S., Bock, C., Mani, M.: Towards an Ontology for Open Assembly model. Proceeding of the International Conference on Product Lifecycle Management, 445-456 (2007)
7. Foufou, S., Fenves, S.J., Bock, C., Rachuri, S., Sriram, R.D. (2005) A Core Product Model for PLM with an illustrative XML implementation. International Conference on Product Lifecycle Management.
8. Fowler, M., Scott, K.: UML distilled, Addison-Wesley (2000)
9. http://www.ontoprise.de/fileadmin/user_upload/Publications_EN/ObjectLogic_Tutorial.pdf
10. http://www.ontoprise.de/de/produkte/ontostudio/
11. http://www.w3.org/2001/XMLSchema#
12. Jun, H.-B., Kiritsis, D., Xirouchakis, P. A primitive ontology model for product lifecycle meta data in the closed-loop PLM. Enterprise Interoperability II: New Challenges and Approaches, Springer Verlag London, 729–740 (2007)
13. Kifer, M, Lausen, G., Wu, J.: Logical foundations of objectoriented and frame-based languages. J. ACM, 42(4),741–843 (1995)
14. Kwak, J.-A., Yong, H.-S.: An Approach to Ontology-Based Semantic Integration for PLM Object, IEEE International Workshop on Semantic Computing and Applications, 19-26 (2008)
15. Matsokis, A., Kiritsis, D.: An ontology-based approach for Product Lifecycle Management. Computers in Industry, 61, 787–797 (2010)
16. Stark, J.: Product lifecycle management: 21st century paradigm for product realization. Springer-Verlag, London (2005)
17. Suh, S.-H., Shin, S.-J., Yoon, J.-S., Um, J.-M.: UbiDM: a new paradigm for product design and manufacturing via ubiquitous computing technology, International Journal of Computer Integrated Manufacturing 21 (5) 540–549 (2008).
18. Wang, Q., Peng, W., Yu, X.: Ontology based geometry recognition for STEP, IEEE International Symposium on Industrial Electronics, 1686-1691 (2010)