

Semantic Enrichment of Models to Assist Knowledge Management in a PLM Environment

Yongxin Liao^{1,2}, Mario Lezoche^{1,2}, Eduardo Loures³,
Hervé Panetto^{1,2}, and Nacer Boudjlida^{4,5}

¹ Université de Lorraine, CRAN, UMR 7039, Boulevard des Aiguillettes B.P.70239,
54506Vandoeuvre-lès-Nancy, France

² CNRS, CRAN, UMR 7039, France

³ Industrial and Systems Engineering, Pontifical Catholic University of Parana.,
ImaculadaConceicao 1155, Curitiba, Brazil

⁴ Université de Lorraine, LORIA, UMR 7503, Boulevard des Aiguillettes B.P. 70239,
54506Vandoeuvre-lès-Nancy, France

⁵ CNRS, INRIA, LORIA, UMR 7503, France

{yongxin.liao,mario.lezoche,herve.panetto}@univ-lorraine.fr,
eduardo.loures@pucpr.br, nacer.boujlida@loria.fr

Abstract. Product Lifecycle Management (PLM) has been considered as an essential concept for improving the product competitive ability in manufacturing enterprises. The PLM solution aims at providing a shared platform for facilitating the management of the knowledge related to any product development process in or across enterprises. However, facing with different standards, enterprise systems and stakeholders, enterprises still need to deal with interoperability issues between those collaborative information systems, encompassing their capability to find the right information during the whole Product Life Cycle (PLC). The objective of this paper is to cope with the major issue of semantic interoperability, by proposing a formalization of semantic annotations and a prototype for facilitating a coherent, complete and contextualized interoperability of knowledge between all enterprise systems and related stakeholders. An example of the instantiation of our method within a real application scenario in manufacturing domain is presented to demonstrate its applicability and use, both at the engineering and the exploitation phases.

Keywords: Semantic Interoperability, Semantic Annotation, BPMN, PLM, Knowledge Management.

1 Introduction

Nowadays, the need for interoperability in a Product Lifecycle Management (PLM) environment has become increasingly imperative. A PLM solution aims to bring together different enterprise systems that deal with the product-related knowledge at each stage and intend to facilitate the communications among all stakeholders [1]. However most of the enterprises have implemented only a few of these systems

without coping with any coherent integration of the entire information system. This has resulted in a kind of “tower of Babel”, where each application is considered as an island in the middle of the ocean of information. Of course, all of these stakeholders have their own background, unique knowledge, particular needs and specific practices. This “cloud of knowledge” is then over increasing the issue of interoperability, not only in the collaborations among the enterprise systems but also by the mutual understanding of the product-related knowledge between these stakeholders.

The objective of this paper is to cope with the issue of interoperability, especially the semantic interoperability, by proposing a formalization of semantic annotations and a prototype for facilitating a coherent, complete and contextualized interoperability of knowledge between all enterprise systems and related stakeholders. The remainder of this paper is organized as follows. Section 2 presents an overview of the background and identifies the semantic interoperability issues to limit the research scope and discusses the related works that made use of semantic annotations to meet particular needs. Section 3 illustrates eight formal definitions of semantic annotations, a conflict detection policy and a semantic annotation framework. A case study is presented in section 4 to demonstrate the applicability and the use of our solution. Section 5 concludes this paper and highlights future research directions.

2 Problems Statement and Related Works

The concept of the Product Life Cycle (PLC) has been introduced since the 1950s, it describes every phase a product of interest goes through, from the first initial requirement until it is retired and disposed [2]. In the early 1980s the problems of locating the required data and losing control of change process associated with these data became increasingly intense [3]. During the 1990s, the concept of Product Lifecycle Management (PLM) is proposed, which aims at providing a shared platform for facilitating the process of capturing, representing, organizing, retrieving and reusing the knowledge concerning the related product in or across enterprises. It should provide the integration strategies and technological supports to bring together all existing enterprise systems that deal with the product [4]. Therefore, product-related knowledge becomes one of the critical concepts in the PLM.

In the so-called DIKW Pyramid [5], referring to a hierarchical model for representing the structural relationships between Data, Information, Knowledge and Wisdom, knowledge is considered as the awareness of things that brings to its owner the capability of grasping the meaning from the information [6]. Knowledge is obtained through certain learning behaviors, in which, the external information from the real world is sublimated and becomes the awareness. In our research, we consider knowledge as a kind of invisible thing, which can only be captured by expressing it into multifarious forms of representations. In some of the literature about knowledge, researchers categorized it into two kinds: (1) tacit knowledge, which is highly personal, difficult to articulate and to formalize; (2) explicit knowledge which is easier to be expressed formally and systematically [7]. According to the main theses that Polanyi used to define the concept of knowledge, all knowledge is rooted in tacit knowledge

[8]. Therefore, we argue that knowledge is an internal awareness that is only explicit to its owner but remains tacit to external world.

In a PLM environment, we consider that all the relevant resources produced by different stakeholders through multifarious enterprise systems are Knowledge Representations (KRs). They act as the carriers of the stakeholders' knowledge to assist further collaboration activities. As the foundation of collaboration, interoperability signifies the ability of diverse entities being able to exchange and make use of those KRs, which is being categorized into five possible levels [9]: encoding level, lexical level, syntactic level, semantic level and semiotic level. While lexical and syntactic issues are now formally solved by many standards, enabling a seamless semantic interoperability remains a huge challenge [10]. The semantics that are contained in a KR is composed of two kinds: (1) explicit semantics, which is directly expressed in the KR; (2) implicit semantics, in opposite, which is hidden. Therefore enterprise systems and stakeholders, in order to cope with the semantic interoperability issue, need to overcome two important obstacles: (1) the implicit semantics that is necessary for understanding a KR is not being made explicit; (2) the lack of automatic semantic verification mechanism to guaranteed the correctness of explicit semantics in a KR.

Due to the essential of ontology, which is defined as a formal and shared understanding of some domains of interest, which specifies the concepts and the relationships that can exist for an agent or a community of agents [19], semantic annotation [11] is usually considered as a possible solution to deal with these two obstacles. Therefore, in our context, semantic enrichment is considered as the process of turning the implicit semantics explicit through the semantic annotations. Some questions are then emerging from the need for semantic enrichment in a PLM environment: What kinds of KR in a PLM environment need semantic enrichment? What kinds of ontology can be used to make the implicit semantics explicit from those KRs? What are the essential elements of a semantic annotation? What kind of policy is needed to detect the possible inconsistencies between semantic annotations and to identify possible mistakes among the annotated model elements?

The concept annotation is defined in the oxford dictionary as "a note by way of explanation or comment added to a text or diagram". Besides this basic meaning, a semantic annotation has two more important features: (1) the machines can read and process it; and (2) it contains a set of formal and shared terms for a certain domain [11]. Semantic annotations use formal knowledge, like ontologies, to capture annotator's knowledge and then they act as a knowledge carrier to enrich annotated object's semantics. Lots of efforts have been made to enrich the semantics of different KRs, such as, natural languages, images, web services and so on. Nevertheless, some shortcomings can be noted: (1) the formalization of semantic annotations is not the research focus in [14], [15], and [17], in which, it is only considered as a simple one-to-one link. Even if there is a specific formalization in [12], [13] and [16], it is difficult to be reused on other types of models but the studied ones; (2) some researches, such as [14] and [15], lack of mechanisms to perform inference to guarantee the consistency among semantic annotations; (3) the domain semantics of the annotated object is the only concern in [16] and [17], in which, the structure semantics (information that describes how the model should be built) is being ignored, or vice-versa. Based on the above literature analysis, in the context of the PLC, we focus our work on clearly identifying the essential of a semantic

annotation by proposing a formalization that can be used to enrich both domain and structure semantics of different types of models, and to facilitate and to assist knowledge management in a PLM environment.

3 Solution Proposal

Based on the research context, we consider the semantic annotation as a way that uses one or several ontology-based knowledge representations to make explicit the semantics of various target knowledge representations. In our case the chosen domain is the PLM. The components of the semantic annotation can be divided into three parts: Target Knowledge Representation (TKR), Ontology-based Knowledge Representation (OKR) and Semantic Annotation Structure Model (SASM). The interoperation process between enterprise systems and stakeholders not only requires that the models can be exchanged and operated on, but it also demands the unambiguous understandings of the exchanged models. Therefore, the necessary and sufficient semantics in those models must be made explicit. The knowledge that is being captured and represented inside the Ontology-based Knowledge Representation can be categorized into two aspects: the domain semantics one and the structure semantics one. The domain semantics aspect of any OKRs describes the context and the meaning of some objects in a certain domain [18]. The structure semantics aspect of any OKRs describes how to build the specific models. In our research, The OKRs spread both domain and structure aspects of knowledge. Because of the objective of this paper, we assume that all the OKRs that we need for semantic annotations are already collected, formalized and pre-processed. The SASM presents into eight definitions the formalization of the semantic relations between TKRs and OKRs.

Definition 1. Semantic annotation SA is defined as: $SA := \langle E, P, SR, MME, MR \rangle$ where: E is a set of elements from a TKR; P is a set of selected element sets from one or several OKRs, which makes explicit the domain semantics aspect of E . SR is a set of binary relations that describes the semantic relations between the elements in E and the elements in P . MME is a set of elements from one OKR, which makes explicit the structure semantics aspect of E . MR is a set of binary relations that describes the semantic relations between the elements in E and the elements in MME . Therefore, a semantic annotation $sa_x \in SA$ is defined as: $sa_x = \{(e_i, p_j, sr_k, mme_l, mr_k) | e_i \in E, p_j \in P, sr_k \in SR, mme_l \in MME, mr_k \in MR\}$.

Definition 2. E is a set of elements from a TKR. Each element $e_i \in E$ has its own domain and structure semantics. The responsibility of the two sets: P and MME is to make explicit the domain semantics and the structure semantics of E .

Definition 3. Let o_x represents an ontology, which can be generally defined as a 4-tuple: $\langle C_{o_x}, R_{o_x}, I_{o_x}, A_{o_x} \rangle$ where C_{o_x} is a set of concepts; R_{o_x} is a set of relationships; I_{o_x} is a set of instances; A_{o_x} is a set of axioms. Moreover, we assume that o_{all} is a set that contains all elements (concept, relation, instance or axiom) of o_x .

Definition 4. P is a set of selected element sets from one or several OKRs that makes explicit the domain semantics aspect of E . It can be considered as a subset of

the power set of ontology elements. P is defined as follow: $P \subseteq P(\bigcup_{o_x \in O} oall_{o_x})$, $\bigcup_{o_x \in O} oall_{o_x} = \{oe_{o_{xy}} | (\exists o_x)(o_x \in O \wedge oe_{o_{xy}} \in oall_{o_x})\}$.

Different from other semantic annotation methods, an e_i in a semantic annotation is not only annotated by one $oe_{o_{xy}}$, but also annotated by a set of selected mandatory ontology elements that are related to it. Given $p_j \in P$ and let use the defined p_j as the minimal set of ontology elements that is created by an annotator or a mechanism (such as semantic block identification [20]), who considers this shape of ontology elements contains the necessary and sufficient semantics for explaining the corresponding e_i .

Definition 5. SR is a set of binary relations that describes the domain semantic relations between $e_i \in E$ and $p_j \in P$. Given $sr_k \in SR$, $sr_k \subset E \times P$ is defined by the notation $e_i sr_k p_j$ meaning that $(e_i, p_j) \in sr_k$. Formally, sr_k is defined as: $sr_k = \{(e_i, p_j) | \text{the semantics of } e_i \text{ is in the relation } sr_k \text{ to the semantics of } p_j\}$, where E is the domain of sr_k , denoted by $dom sr_k$, and P is the range of sr_k , denoted by $ran sr_k$: $dom sr_k = \{e_i | (\exists e_i)(e_i \in E \wedge (e_i, p_j) \in sr_k)\}$, $ran sr_k = \{p_j | (\exists p_j)(p_j \in P \wedge (e_i, p_j) \in sr_k)\}$. To be more specific, we define the binary relations as: “is equivalent to” (sr_{\sim}), “subsumes” (sr_{\supset}), “is subsumed by” (sr_{\subset}), “intersects” (sr_{\cap}) and “is disjoint with” (sr_{\perp}). SR is defined with the following notations: $SR = \{sr_{\sim}, sr_{\supset}, sr_{\subset}, sr_{\cap}, sr_{\perp}\}$.

The “is equivalent to” relation on two sets E and P denotes the fact that the related two elements from the domain and the range are semantically equivalent. Relation $sr_{\sim} \subset E \times P$ is defined as: $sr_{\sim} = \{(e_i, p_j) | \text{the semantics of } e_i \text{ and the semantics of } p_j \text{ are equivalent}\}$.

The “subsumes” relation on two sets E and P denotes the fact that the semantics of the related element from the domain is more general than the one of the related element from the range. Relation $sr_{\supset} \subset E \times P$ is defined as: $sr_{\supset} = \{(e_i, p_j) | \text{the semantics of } e_i \text{ is more general than the semantics of } p_j\}$.

Conversely, the “is subsumed by” relation on two sets E and P denotes the fact that the semantics of related elements from the domain is less general than the semantics of the related elements from the range. Relation $sr_{\subset} \subset E \times P$ is defined as: $sr_{\subset} = \{(e_i, p_j) | \text{the semantics of } e_i \text{ is less general than the semantics of } p_j\}$.

The “intersects” relation on two sets E and P denotes the fact that the related two elements from the domain and the range have only a part of common semantics. In our case, we specify that this relation doesn’t cover the binary relations of sr_{\sim} , sr_{\supset} and sr_{\subset} . Relation $sr_{\cap} \subset E \times P$ is defined as: $sr_{\cap} = \{(e_i, p_j) | e_i \text{ and } p_j \text{ have some common semantics}\}$, $sr_{\cap} \cap sr_{\sim} = \emptyset$, $sr_{\cap} \cap sr_{\supset} = \emptyset$, $sr_{\cap} \cap sr_{\subset} = \emptyset$.

The “is disjoint with” relation on two sets E and P denotes that the related two elements from the domain and the range have not common semantics. Relation $sr_{\perp} \subset E \times P$ is defined as: $sr_{\perp} = \{(e_i, p_j) | e_i \text{ and } p_j \text{ have not common semantics}\}$.

Definition 6. A meta-model is a model that specifies the concepts, relationships and rules to model a model [21]. Given a meta-model mm_x , it can be generally defined as

a 3-tuple: $\langle C_{mm_x}, R_{mm_x}, RU_{mm_x} \rangle$ where C_{mm_x} is a set of concepts; R_{mm_x} is a set of relationships; RU_{mm_x} a set of rules.

Definition 7. *MME* is a set of elements from one OKR, which makes explicit the structure semantics aspect of E . Each element $mme_l \in MME$ is defined as: $MME = \{mme_l | mme_l \in C_{mmo_x} \cup R_{mmo_x} \cup I_{mmo_x} \cup A_{mmo_x}\}$. An e_i in a semantic annotation is annotated by one mme_l .

Definition 8. *MR* is a set of binary relations that describes the structure semantic relations between $e_i \in E$ and $mme_l \in MME$. Given $mr_k \in MR$, $mr_k \subset E \times MME$ is defined by the notation $e_i mr_k mme_l$, meaning that $(e_i, mme_l) \in mr_k$. Formally, mr_k is defined as: $mr_k = \{(e_i, mme_l) | e_i \text{ is in the relation } mr_k \text{ to } mme_l\}$ where E is the domain of mr_k , denotes as $dom\ mr_k$, and MME is the range of mr_k , denotes as $ran\ mr_k$: $dom\ mr_k = \{e_i | (\exists e_i)(e_i \in E \wedge (e_i, mme_l) \in mr_k)\}$. $ran\ mr_k = \{mme_l | (\exists mme_l)(mme_l \in MME \wedge (e_i, mme_l) \in mr_k)\}$.

In this paper, we defined this binary relation as: “is instance of” (mr_{io}). *MR* is defined as follow notation: $MR = \{mr_{io}\}$.

This formalization not only describes the semantic relations between TKR and OKRs, but also acts as the foundation for inconsistency detection and mistake identification. To cope with this objective, SAs are further classified into two types: (1) initial SA, which directly annotates an element in TKR by an annotator; (2) inferred SA, which is suggested to annotate an element in TKR through the inference of the related element’s SA and certain rules. Based on the outcomes, three types of possible results can be identified: result (a) expresses that sa_x and sa_y are consistent with each other; result (b) expresses that there is a possible inconsistency between sa_x and sa_y ; result (c) expresses that there is an inconsistency between sa_x and sa_y . These results not only detect the inconsistencies (or possible inconsistencies) between two SAs, but then can also be used to identify the possible mistakes among those annotated elements in TKR. In order to apply the above formalization of semantic annotation into a PLC context, a semantic annotation framework (SAF) is proposed. There are four main modules in the SAF: the Semantic Annotation and Processing Agent (SAPA), the OKR Creation and Management (CM) module, the Knowledge Cloud (KC) module and the Reasoning Engine (RE) module. Some processes, that describe a PLC, represent the use of various kinds of enterprise systems (ESs) to manage their own TKRs. KC is the core of the SAF, which is composed of a set of OKRs that are collected, formalized and generated by the CM module. RE is an external call pattern-matching search engine that is in charge of performing inferences to answer different reasoning requests according to some predefined rules. The SAPA acts as a mediator to support the communications between ESs in different processes of the PLC and three other modules in SAF.

4 Case Study

The proposed solution for semantic enrichment of models through semantic annotations is illustrated by a case study that deals with the semantic interoperability requirement of a process model. This PLC scenario is provided by an educational

production site: AIPL¹, in which, MEGA² is used to design the initial manufacturing process models and to provide product engineers with a global view of the production phases in the PLC. A segment of the manufacturing processes in the AIPL product lifecycle is chosen as the context of this case study. There are five main processes: 1) The bar cutting process; 2) The base turning process; 3) The disc cutting process; 4) The part sticking process; 5) The product assembling process. The knowledge related to the above processes is formalized into the AIPL product ontology. Furthermore, two more OKRs are also integrated into the Knowledge Cloud: MSDL Ontology, which describes the manufacturing capability at the supplier level, process level and machine level of abstraction and the BPMN Ontology [22]

The prototype of the semantic annotation tool, SAP-KM (Semantic Annotation Plugin for Knowledge Management), has been developed as a plugin in the MEGA modeling application. It supports the semantic enrichment of process models and shows the possibility of applying this proposed solution into the reality. To be more specific we take the sub process “Usinage” with its input “Rondelle” and output “Palet” in this process model as an example. The “Usinage” is annotated by sa_1 that states that its domain semantics is more general than the semantics of p_1 and it is an instance of mme_1 . The “Rondelle” is annotated by sa_2 that states that its domain semantics is more general than the semantics of p_2 and it is an instance of mme_2 . After the reasoning based on these two initials semantic annotations and rules (SWRL Rules created in the preprocessing phase), the inference engine Pellet and predefined policies in the prototype suggest two inferred SAs. The “Rondelle” is suggested to be annotated with sa_3 that states that its domain semantics is more general than the semantics of p_3 and it is an instance of mme_3 . The “Palet” is suggested to be annotated by sa_4 that states that its domain semantics is more general than the semantics of p_4 and it is an instance of mme_4 . Because both sa_2 (initial) and sa_3 (inferred) are SAs of the “Rondelle”, based on the list of possible results of inconsistency detection between two SAs the semantics similarity comparison outcome $p_2 \perp p_3$ signifies a possible inconsistency between sa_2 and sa_3 .

5 Conclusion

Because of the initial design objective of the modeling languages and the diversity of expressions of models, the inaccurate and inconsistent semantics among the model elements are difficult to be identified. With the supports of semantics enrichment solutions, this paper shows the possibility of using formal semantic annotations to enrich the enterprise models in a PLM environment for facilitating semantic interoperability issue during the knowledge management process. A semantic annotation framework is presented to describe and generate interactions among its four main modules. A prototype annotation tool, SAP-KM, is designed and implemented to instantiate the formal semantic annotations and to demonstrate its applicability and

¹ AIPL <http://www.aip-primeca.net/> - Atelier Inter-Établissements de Productique Lorrain.

² MEGA <http://www.mega.fr/en>

usability of our semantic enrichment solution. The further development of this prototype will be concentrated on the iterated suggestion and real-time verification.

References

1. Ameri, F., Dutta, D.: Product lifecycle management: closing the knowledge loops. *Comput.-Aided Des. Appl.* 2, 577–590 (2005)
2. Rink, D.R., Swan, J.E.: Product life cycle research: A literature review. *J. Bus. Res.* 7, 219–242 (1979)
3. CIMdata: PDM to PLM: Growth of an Industry (2003)
4. Ball, A., Ding, L., Patel, M.: An approach to accessing product data across system and software revisions. *Adv. Eng. Informatics* 22, 222–235 (2008)
5. Zeleny, M.: Management support systems: towards integrated knowledge management. *Human Systems Management* 7, 59–70 (1980)
6. Ackoff, R.L.: From Data to Wisdom. *J. Applies Syst. Anal.* 16, 3–9 (1989)
7. Ropo, A., Parviainen, J.: Leadership and bodily knowledge in expert organizations:: epistemological rethinking. *Scand. J. Manag.* 17, 1–18 (2001)
8. Polanyi, M.: *The tacit dimension*. Peter Smith, Gloucester (1983)
9. Euzenat, J.: Towards a principled approach to semantic interoperability. In: *IJCAI 2001, Workshop on Ontology and Information Sharing*, Seattle, USA, pp. 19–25 (2001)
10. Panetto, H.: Towards a classification framework for interoperability of enterprise applications. *Int. J. Comput. Integr. Manuf.* 20, 727–740 (2007)
11. Boudjlida, N., Panetto, H.: Annotation of enterprise models for interoperability purposes. In: *IEEE International Workshop on Advanced Information Systems for Enterprises, IWAISE 2008*, Constantine, Algeria, pp. 11–17 (2008)
12. Lin, Y.: *Semantic annotation for process models: Facilitating process knowledge management via semantic interoperability*, PhD Thesis (2008)
13. Di Francescomarino, C.: *Semantic annotation of business process models*, PhD Thesis (2011)
14. Attene, M., Robbiano, F., Spagnuolo, M., Falcidieno, B.: Characterization of 3D shape parts for semantic annotation. *Comput.-Aided Des.* 41, 756–763 (2009)
15. Li, C.: *Ontology-Driven Semantic Annotations for Multiple Engineering Viewpoints in Computer Aided Design*, PhD Thesis (2012)
16. Bergamaschi, S., Beneventano, D., Po, L., Sorrentino, S.: Automatic Normalization and Annotation for Discovering Semantic Mappings. In: Ceri, S., Brambilla, M. (eds.) *Search Computing II. LNCS*, vol. 6585, pp. 85–100. Springer, Heidelberg (2011)
17. Song, F., Zacharewicz, G., Chen, D.: An ontology-driven framework towards building enterprise semantic information layer. *Adv. Eng. Informatics* 27, 38–50 (2012)
18. Maedche, A., Staab, S.: Measuring Similarity between Ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) *EKAW 2002. LNCS (LNAI)*, vol. 2473, pp. 251–263. Springer, Heidelberg (2002)
19. Gruber, T.R.: A translation approach to portable ontology specifications. *Knowl. Acquis.* 5, 199–220 (1993)
20. Lezoche, M., Yahia, E., Aubry, A., Panetto, H., Zdravković, M.: Conceptualising and structuring semantics in cooperative enterprise information systems models. *Comput. Ind.* 63, 775–787 (2012)
21. OMG: *Meta Object Facility Core Specification*, <http://www.omg.org/spec/MOF/2.4.1/>
22. Ghidini, C., Rospocher, M., Serafini, L.: *A formalisation of BPMN in description logics*. Technical report (2008)