Semantic annotation for knowledge explicitation in a product lifecycle management context: a survey
Yongxin Liao, Mario Lezoche, Hervé Panetto, Nacer Boudjlida, Eduardo Rocha Loures

To cite this version:
Yongxin Liao, Mario Lezoche, Hervé Panetto, Nacer Boudjlida, Eduardo Rocha Loures. Semantic annotation for knowledge explicitation in a product lifecycle management context: a survey. Computers in Industry, Elsevier, 2015, 71, pp.24-34. 10.1016/j.compind.2015.03.005 . hal-01123854

HAL Id: hal-01123854
https://hal.archives-ouvertes.fr/hal-01123854
Submitted on 5 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Semantic Annotation for Knowledge Explicitation in a Product Lifecycle Management Context: a Survey

Yongxin Liao\textsuperscript{a,b,c,i,*}, Mario Lezoche\textsuperscript{a,b,ii}, Hervé Panetto\textsuperscript{a,b,iii}, Nacer Boudjlida\textsuperscript{d,e,iv} and Eduardo Rocha Loures\textsuperscript{c,v}

\textsuperscript{a}Université de Lorraine, CRAN, UMR 7039, Boulevard des Aiguillettes B.P.70239, 54506 Vandoeuvre-lès-Nancy, France;
\textsuperscript{b}CNRS, CRAN, UMR 7039, France;
\textsuperscript{c}Pontifícia Universidade Católica do Paraná, PPGEPS, Rua Imaculada Conceição, 1155, Curitiba, Brazil;
\textsuperscript{d}Université de Lorraine, LORIA, UMR 7503, Boulevard des Aiguillettes B.P. 70239, 54506 Vandoeuvre-lès-Nancy, France;
\textsuperscript{e}CNRS, INRIA, LORIA, UMR 7503, France

\textsuperscript{i}yongxin.liao@pucpr.br, \textsuperscript{ii}mario.lezoche@univ-lorraine.fr, \textsuperscript{iii}herve.panetto@univ-lorraine.fr, \textsuperscript{iv}nacer.boudjlida@loria.fr, \textsuperscript{v}eduardo.loures@pucpr.br

\text{*The corresponding author. His phone number is +55 41 3271-1101

Abstract. Nowadays, the need for systems interoperability in or across enterprises has become more and more ubiquitous. Many research works have been carried out in the fields of information exchange, transformation, discovery and reuse. One of the main challenges in these researches is to overcome the semantic heterogeneity between enterprise applications along the life cycle of a product. As a possible solution to assist the semantic interoperability, the semantic annotation has gained many attentions and widely used in different domains. We collect a number of literature that applied semantic annotations on different objects, and classify them according to the subject being described in an enterprise architecture framework. In this paper, a detailed survey, especially from the formalization perspective, is presented to identify the existing drawbacks and to point out the possible research directions.

Keywords: Semantic Annotation; Knowledge Explicitation; Formalization; Semantic Interoperability; Product Lifecycle Management.

1. Introduction

In manufacturing enterprises, the Product Lifecycle Management (PLM) approach has been considered as an essential solution for improving the product competitive ability. It aims at providing a shared platform that brings together different enterprise applications
at each stage of a Product Life Cycle (PLC) in or across enterprises [1]. Although the main software companies are making efforts to offer a complete and integrated set of systems, most of them do not provide a coherent integration of the entire information system. This results in a kind of “tower of Babel”, where each application is considered as an island in the middle of the ocean of information, managed by stakeholders along the life cycle of a product.

Semantic interoperability is the ability to ensure that the exchanged information has got the same meaning considering the point of view of both the senders and the receivers [2]. In the context of a PLM, stakeholders have to work together on the exchanged information and make decisions based on it. They have different backgrounds, heterogeneous expertise, unique knowledge, particular needs and specific practices, which over increase the difficulty to achieve semantic interoperability [3]. The mutual understanding of the semantics that is embedded inside the exchanged information is the cornerstone in the quest for semantic interoperability. Being a way to realize this enrichment, the semantic explication [4] is not only just attaching the formal and shard terms between stakeholders to make semantics explicit, but also bringing the possibility to perform the semantic reasoning for some further operations.

The objective of this paper is to present a detailed survey of the collected semantic annotation literature, especially from the formalization perspective. The rest of this paper is organized as follows: Section 2 presents the definitions of annotation and semantic annotation. Section 3 illustrates and compares the semantic annotation researches being applied on different objects. Section 4 identifies the existing drawbacks and proposes the possible research directions. Section 5 concludes this paper and highlights our on-going and further research works.
2. Annotation and Semantic Annotation

The Oxford dictionary defines an annotation as “a note by way of explanation or comment added to a text or diagram”. It has special usages in different contexts. For example, in the software programming, an annotation is represented as a text comment embedded in codes to explain the program. In the mechanical drawing, an annotation is a snippet of text or symbol with specific meanings that illustrates the corresponding annotated part. In the commercial advertising, an annotation is usually used as a kind of footnote to detail some business restrictions.

In order to distinguish the semantic annotation from the other annotations, several kinds of classifications are proposed. Bechhofer et al. [5] categorized annotations into three types: the textual annotation, which adds notes and comments to an object; the link annotation, which extends the previous type of annotation by linking the object to an annotation content; the semantic annotation, which contains the human-readable as well as machine-readable information. Similarly, Oren et al. [6] proposed to classify annotations as: the informal annotation, which is expressed in an informal language and is not machine-readable; the formal annotation, which is machine-readable, but without any ontological terms; the ontological annotation, which is only composed of ontological terms that are commonly accepted and understood in a specific domain. These classifications identify two important features of a semantic annotation: (1) it is both human-readable and machine-readable, and (2) it contains a set of formal and shared terms that can exist for a community of human and/or machine agents.

Considering the essential of an ontology [7], which is a common agreement of a conceptualization of terms in a specific domain, different researchers have suggested many definitions of the semantic annotation related to an ontology. For example, Talantikite et al. [8] described it as “a semantic annotation is referent to an ontology”.
Lin [9] considered it as “an approach to link ontologies to the original information sources”. Kiryakov et al. [10] defined it as “a specific metadata generation and usage schema, aiming to enable new information access methods and to extend the existing ones”. To our knowledge, a semantic annotation can be considered as a means to perform the semantic enrichment of “something” by using a set of well formalized and commonly agreed terms from a specific domain, such as ontologies.

In our research, we mainly pay attention to two aspects of semantics that are made explicit through a semantic annotation: The domain semantics, which describes the context and the meaning of an annotated element in a specific domain; the structure semantics, which describes the interrelations between an annotated element and the other elements related to it. Taking into account these two aspects of semantics and the investigations that we have made in previous works [10][11], in the next section, we will discuss different semantic annotation methods inside the collected literature.

3. The Investigation of Semantic Annotation Researches

In the last decade, several surveys of semantic annotation researches have already been made with different focuses. Reeve and Han [13] presented a short survey about the classification and evaluation of six semantic annotation platforms. Uren et al. [14] reviewed and classified twenty seven semantic annotation systems according to the knowledge management requirements that they proposed. Mangold [15] presented a categorisation scheme for the classification of ten selected semantic search approaches and identified the open issues that are not addressed by those systems. Lautenbacher and Bauer [16] presented a survey to categorize and compare twenty one annotation approaches about semantic web services, grid workflows, and business process management. Hanbury [17] summarized five types of image annotation methods and then used it to analyse ten annotated image datasets. Dasiopoulou et al. [18] made a survey on
eight image and seven video annotation tools, from both functionality and interoperability perspectives, to highlight the issues of the communication, sharing and reuse of produced metadata. Oliveira and Rocha [19] introduced and briefly compared nineteen semantic annotation tools to show the challenge in the quest to fully automatic annotation. Joksimovic et al. [20] presented an empirical study on three ontology-based semantic annotators to discover the existed issues for the future development of those examined tools.

We can find that the surveys [13], [14], [15], [19] and [20] were mainly focusing on documents, as well as the surveys [17] and [18] paid major attention to images or videos. They analysed some existing annotation tools from both the functionality perspective ([14], [15], [17], [18] and [19]) and from the efficiency perspective ([13] and [20]). A number of self-defined requirements are used as the basis to compare the semantic annotation approaches in the surveys [14] and [16]. Depict efforts have been made by above-mentioned surveys, at least two shortcomings need to be noted: (1) most of the surveys concerning the approaches that applied semantic annotations on texts, images, or videos, and few surveys have addressed models. The survey [16] is the only one that concerned the annotation on a specific kind of model (e.g. workflows). However a more recent and detailed analysis is still required; (2) among these surveys, only three of them ([13], [17] and [19]) have taken into account and generally discussed the semantic annotation methods that are embedded behind those tools. Little attention has been paid to the in-depth study and comparison of the methods, especially from the formalization perspective. This section begins with an illustration of the collected and classified semantic annotation researches (Section 3.1). Based on the literature review, we will firstly analyse and compare the semantic annotation methods inside those researches.
Then, a more detailed discussion from the formalization perspective will be presented (Section 3.2).

3.1 The Illustration of Semantic Annotation Researches

With the supports of the ontologies from multiple domains and different levels, the semantic annotation approaches are widely studied and applied in diverse contexts. Concerning our research context (a PLM environment) and focus (dealing with semantic interoperability issue), the well-accepted enterprise architecture framework, named “Zachman Framework” [21], is employed as a template to assist the literature collection and classification. As it is shown in Figure 1, it presents two dimensional classification schemes for descriptive representations of an enterprise: the perspective dimension (row) and the abstraction dimension (column). For each perspective dimension in the framework, one example subject being described in the abstraction dimension is given. Based on the object of a semantic annotation research and the example subjects in the framework, the collected literature is classified into the corresponding grids. In this paper, for each annotation object, we illustrate seven collected literature as examples. Though the annotation objects of some researches might cross several subjects in the framework, in this paper, we only place them at the major subject that they worked on.

We surveyed 135 semantic annotation research works (including journal articles, conference papers, PhD theses and reports), fourteen of which we will illustrate and discuss in this paper. For each perspective dimension in the framework, we choose one subject as the object of the semantic enrichment. Moreover, for each chosen subject, we will present the analysis of two or three research works that applied semantic annotation on it as examples. To be more specific, this section is structured as follows: Concerning the three principal perspectives (Section 3.1.1), we illustrate the semantic annotation approaches about the “Process Models” for the Business Model Perspective (Section
3.1.1.1), about the “Data Models” for the System Model Perspective (Section 3.1.1.2), and about the “Computer-Aided Design Models” for the Technology Model Perspective (Section 3.1.1.3) respectively. Concerning the two additional perspectives, we take the approaches that applied semantic annotations on the “Texts” for the Scope Perspective (Section 3.1.2), and on the “Web Services” for the Detailed Representation Perspective (Section 3.1.3).

<table>
<thead>
<tr>
<th>Scope (Contextual)</th>
<th>Function</th>
<th>Network Where</th>
<th>People</th>
<th>Time</th>
<th>Motivation Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>e.g. Texts. Such as [35], [36], [42], [43], [44], [45], [37]</td>
<td>e.g. Images, Videos or Audios. Such as [46], [47], [48], [49], [50], [51], [52]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Business Model (Conceptual) | e.g. Class Diagrams. Such as [90], [91], [92], [93], [94], [23], [95] | e.g. Process Models. Such as [57], [58], [59], [22], [9], [26], [25] |
| e.g. Business Logistics System | e.g. Work Flows. Such as [69], [70], [71], [72], [73], [74], [75] |
| e.g. Schedules. Such as [96], [97], [98], [99], [100], [101], [102] |
| e.g. Requirements. Such as [103], [104], [105], [106], [107], [23], [108] |

| System Model (Logical) | e.g. Logical Data Models. Such as [60], [61], [62], [30], [63], [21], [29] |
| e.g. Application Architecture |
| e.g. Distributed System Architecture |
| e.g. Human Interface Architecture |
| e.g. State and Control Diagrams. Such as [83], [84], [85], [86], [87], [88], [89] |

| Technology Model (Physical) | e.g. Physical Data Models |
| e.g. Technology Architecture |
| e.g. Presentation Architecture |

| Detailed Representations (Out-of-Context) | e.g. Web Services. Such as [41], [53], [40], [54], [8], [55], [56] |

<table>
<thead>
<tr>
<th>For Texts:</th>
<th>For Images, Videos or Audios:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[37] Ma et al. (2011a)</td>
<td>[52] Tang et al. (2011)</td>
</tr>
<tr>
<td>For Work Flows:</td>
<td>For Schedules:</td>
</tr>
<tr>
<td>[70] Cantalupo et al. (2005)</td>
<td>[97] Basra et al. (2005)</td>
</tr>
<tr>
<td>[73] Missier et al. (2010)</td>
<td>[100] Lee et al. (2006)</td>
</tr>
<tr>
<td>For State and Control Diagrams:</td>
<td>For Rules:</td>
</tr>
<tr>
<td>[84] Chen and Tu (2009)</td>
<td>[77] Nazarenko et al. (2011)</td>
</tr>
<tr>
<td>[85] Seo et al. (2009)</td>
<td>[78] Omrane et al. (2011)</td>
</tr>
<tr>
<td>[87] Rahman et al. (2012)</td>
<td>[80] Sainte Marie et al. (2011)</td>
</tr>
<tr>
<td>For Design Models:</td>
<td>For Web Services:</td>
</tr>
<tr>
<td>[70] Li (2012)</td>
<td>[71] Li (2012)</td>
</tr>
</tbody>
</table>

Figure 1. The Classification of the Collected Literature based on the Zachman Framework
3.1.1 Semantic Annotations for Models

Enterprise modelling is a process that focuses on capturing and representing knowledge from the perspectives of a system of interest. The interoperations among the systems not only require that models can be exchanged and operated on, but also demand an unambiguous understanding of the semantics inside those models.

3.1.1.1 Semantic Annotations for Different Kinds of Models in an Enterprise.

The Task Group 4 of the INTEROP project [22] investigated how annotations are able to contribute in making explicit the semantics of models in an enterprise for the purpose of enabling both semantic-based and model-based interoperability among collaborating actors. As it is shown in Figure 2, they proposed a general annotation schema for the annotation of all kinds of models. They assumed that any part within a model may be annotated and can be annotated with multiple annotations.

<table>
<thead>
<tr>
<th>General Annotation Schema:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annotation-Id</strong>: identifier of annotation</td>
</tr>
<tr>
<td><strong>Unformal Content</strong>: unformal comments</td>
</tr>
<tr>
<td><strong>Annotation Type</strong>: Type of annotation</td>
</tr>
<tr>
<td><strong>Ref2Ontology</strong>: URI of ontology concept</td>
</tr>
<tr>
<td><strong>Constraints</strong>: refer to ontology or meta-model</td>
</tr>
</tbody>
</table>

**Figure 2. The Semantic Annotation Schema from Boudjlida et al. (2007) [22]**

Liao [23] studied the issues of the interoperability, especially the semantic interoperability problems, and proposed a formal semantic annotation method to support the mutual understanding of the semantics inside the exchanged information within a Product Lifecycle Management (PLM) context. As can be seen from Figure 3, part of the semantic annotation structure model is illustrated. It can be used to assist the construction of semantic annotation schemas for various kinds of models along the life cycle of a product and to support the design and implementation of the semantic reasoning.
3.1.1.2 Semantic Annotations for Process Model

A process model is a "linear representation of a set of variables that define a theoretically meaningful sequence of related conditions and actions" [24]. As one of the critical components in the Business Model Perspective, it is usually used as a blueprint for the design of a workflow that coordinates activities, resources, and data according to the underlying processes [25].

Di Francescomarino [26] proposed some techniques to support the annotation of business process model with ontologies, which gives the possibility to perform reasoning for assisting designers and analysts in the management of their business process models.

On one hand, a semantic annotation is represented as the “textual annotations” in a business process diagram by using a “@” symbol with the name of the selected ontology class. On the other hand, an annotated element in process model has a corresponding ontology individual and is asserted to the selected class. Figure 4 shows the annotation schema that we summarized from that thesis.

Lin [9] proposed a semantic annotation framework to support the discovery and the sharing of process models in or between enterprises by reconciling the semantic
heterogeneity between process modelling languages (meta-model) and model contents. In that thesis, the meta-model is annotated by a general process ontology (GPO) and model contents are annotated by a domain ontology. A set of refined relations is also proposed to better describe the semantic relationships between elements in models and the concepts in ontologies. As it is shown in Figure 5, a process semantic annotation model (PSAM), which describes the process properties and annotation contents, was proposed to generate a common annotation schema for different kinds of process models.

\[
\text{PSAM} = (\text{AV}, \text{AR}, \text{AF}, \text{WP}, I, O, \Theta^{\text{pre}}, \Theta^{\text{pos}}, E, \text{PD})
\]

where
- \(\text{AV}\) is a set of activities
- \(\text{AR}\) is a set of actor-roles
- \(\text{AF}\) is a set of artifacts
- \(\text{WP}\) is asset of workflow patterns
- \(I\) is a set of input parameters
- \(O\) is a set of output parameters
- \(\Theta^{\text{pre}}\) is pre-conditions
- \(\Theta^{\text{pos}}\) is post-conditions
- \(E\) is a set of possible exceptions
- \(\text{PD}\) is a subset of the domain ontology concepts

Figure 5. The Semantic Annotation Schema from Lin (2008) [9]

3.1.1.3 Semantic Annotations for Data Models

A data model is “an effective technique to define the shareable semantics that are essential to the success of data communication in an integrated environment” [27]. Although the data modelling only represents a small phase in the System Model Perspective, it probably has more impact on the final results than the other phases in a whole enterprise systems development process [28].

Song et al. [29] investigated the issues of heterogeneous data systems and proposed a semantic information layer (SIL), which acts as a mediation tool among these systems to overcome gaps of data and semantic heterogeneity. This research focused on the development of an ontology-driven framework for supporting the extraction of ontologies from different databases and assisting the creation and management of the SIL.
Semantic annotations are automatically generated and be used as paths between the SIL and data schemas.

MOMIS project [30] proposed an annotation method to support the automatic and semi-automatic annotation on two or more data models that are extracted and converted from either structured or semi-structured data sources. Based on the annotations and the predefined semantic relationships, it generates a global schema to support the data integration between different data sources. In the annotation phase, the generic lexical database and a domain glossary are employed to provide human readable meanings for annotators. Semantic annotation is only considered as a kind of association between an element in data model and its text meanings.

### 3.1.1.4 Semantic Annotations for Computer-Aided Design Models

A Computer-Aided Design model can be considered as a simulated graphical representation of a product, which holds a complete depiction of information that is capable of supporting the manufacturing [31]. As one kind of models in the Technology Model Perspective, it acts as a final step of implementation or realization during the enterprise modelling phase.

Attene et al. [32] developed a semantic annotation system, named ShapeAnnotator, which is able to decompose a 3D shape into several interested features through a segmentation algorithm and to support the annotation of the selected features by connecting them to the corresponding individuals. These individuals are saved in a separated OWL [33] file with the imports of domain ontology. Figure 6 shows the annotation schema that we summarized from that research.

<table>
<thead>
<tr>
<th>Annotation Schema in the ShapeAnnotator:</th>
</tr>
</thead>
<tbody>
<tr>
<td>--Class: selected class in domain ontology</td>
</tr>
<tr>
<td>--ShannGeoContextURI: a URI refer to a multi-segmented mesh</td>
</tr>
<tr>
<td>--ShannSegmentID: an index of a segment in the multi-segmented mesh.</td>
</tr>
<tr>
<td>--Related Values: value is computed and added by the feature descriptors</td>
</tr>
</tbody>
</table>
Li [34] proposed an ontology-driven semantic annotation framework for CAD systems (OntoCAD), which provides product engineers with multiple engineering viewpoints of a product in its life cycle. In that thesis, as it is shown in Figure 7, an annotation data structure was proposed to formalize those annotations. A three layered ontology architecture knowledge base was proposed to capture, represent and manage multiple engineering viewpoint ontologies and to support the processing of querying and reasoning requests.

<table>
<thead>
<tr>
<th>Annotation data structure in the OntoCAD:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Anchor</strong>: the geometric elements that are being represented as OWL individuals;</td>
</tr>
<tr>
<td><strong>OWL properties</strong>: object property or data property in OWL;</td>
</tr>
<tr>
<td><strong>Content</strong>: OWL individuals or data values.</td>
</tr>
</tbody>
</table>

Figure 7. The Semantic Annotation Schema from Li (2012) [34]

3.1.2 Semantic Annotations for Texts

In the early stage of the enterprise modelling (the *Scope Perspective*), the graphical or text description of information is one of the most employed descriptive methods. The semantic enrichment of texts is mainly designed to help a machine to “understand” the meaning of the annotated texts and to support automated processes, such as information navigation. Of course, not limited to this, a large number of researches have been proposed.

Vargas-Vera et al. [35] presented the MnM, an ontology-based annotation tool, which integrates web browser, ontology editor and open APIs to provide both automatic and semi-automatic supports for the annotation of texts in web pages. It is able to extract information from web pages and then fills them into a pre-defined template. Further, a simple type-based validation was proposed to verify the correctness of contents that are filled into the template.
Popov et al. [36] developed a Knowledge and Information Management (KIM) platform, which is based on a KIM ontology and a massive knowledge base to automatic annotate, index, and retrieve documents. According to the hypothesis that Named Entities (NE), such as people and location that are referred by name, constitute the essential semantics in a document. The automatic semantic annotation is considered as the process of NE recognition and annotation. It provides for each extracted NE with two kinds of links: one link to the most specific class in KIM ontology to specify the named-entity type and the other link to the specific individual in knowledge base.

Ma et al. [37] proposed a framework to support the semantic reasoning on both domain and linguistic information that are embedded in the annotations of texts. It uses two ontologies: (1) a domain ontology to provide semantic labels (domain knowledge), and (2) a language ontology to give text model (linguistic knowledge). For the former one, as it is shown in Figure 8, a semantic annotation assertion is defined. For the latter one, it is represented as a set of OWL axioms and SWRL [38] rules, which contributes to bridge the inference constraints.

Semantic Annotation is a tuple $<\text{tf}, \text{ot}, \text{at}>$ where
- $\text{tf}$ is a set of text fragment;
- $\text{at}$ is a set of semantic labels;
- $\text{ot}$ is a set of relations between $\text{tf}$ and $\text{ot}$.

Figure 8. The Semantic Annotation Schema from Ma et al. (2011) [37]

3.1.3 Semantic Annotations for Web Services

A web service is “a software system designed to support interoperable machine-to-machine interaction over a network” [39]. In the Detailed Representation Perspective, adding semantic annotations to a web service is mainly for supporting the automatic verification of certain tasks, which must be executed before or during invocation of corresponding services [40].
Talantikite et al. [8] proposed to use semantic annotations to assist the creation of an inter-connected service network. This network is then processed by a composition algorithm and is used to discover an appropriate composition services plan for answering the corresponding user requests. As can be seen from Figure 9, they proposed a kind of annotation schema, in which the inputs and outputs are used for the similarity measurement. The exec-time and All-Resources are quality criteria for the evaluation of the best composition plan.

<table>
<thead>
<tr>
<th>Semantic Annotation of Web Service (WS):</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- Sid: the identifier of a WS</td>
</tr>
<tr>
<td>-- Sname: the name of a WS</td>
</tr>
<tr>
<td>-- inputs: the input of a WS</td>
</tr>
<tr>
<td>-- outputs: the output of a WS</td>
</tr>
<tr>
<td>-- exec-time: the execution time of a WS</td>
</tr>
<tr>
<td>-- All-Resources: the required resources</td>
</tr>
<tr>
<td>-- Bindings: the used protocol</td>
</tr>
<tr>
<td>-- Service: the URI of a WS</td>
</tr>
</tbody>
</table>

Figure 9. The Semantic Annotation Schema from Talantikite, et al. (2009) [8]

Patil et al. [41] proposed the MWSAF, a framework for semi-automatically annotating web services with domain ontologies, to help web services discovery and composition. A semantic annotation is simply used as a “is a” association to link one element in a service and one concept in an ontology based on the linguistic similarity and structure similarity matching.

In order to simplify and standardize the complex semantic annotation methods for web services, the Semantic Annotation for WSDL and XML Schema (SAWSDL) was proposed [40]. It aims to add semantics to web services by providing extension attributes. As it is shown in Figure 10, the SAWSDL extensions can be classified into two kinds. The Model Reference (sawsdl:modelReference), which describes an association from a WSDL component or a XML Schema component to a concept in semantic models. The Schema Mapping (sawsdl:liftingSchemaMapping and sawsdl:loweringSchemaMapping),
which specifies how an instance data in an XML Schema maps to a semantic data in a semantic model.

Figure 10. The Semantic Annotation Schema from Kopecký et al. (2007) [40]

3.2 The Comparison of Semantic Annotation Researches

3.2.1 The Comparison: from the General Point of View

As it is illustrated in Table 1, an overall comparison of the above-mentioned examples is presented. It firstly uses five columns to answer the questions about “What”, “Why” and “How” to perform the semantic annotation. And then it uses four columns to describe the four major factors that are important for the formalization of a semantic annotation. More specifically, each column in this table is introduced as follows:

1. The column “Application Domains” answers the question about “What to annotate?”. It describes the object of semantic annotations, which relies on the context of researches (e.g. Process Models).

2. The column “Usages of Annotation” answers the question about “Why to annotate?”. We classified the reasons into the following three groups:
   a) Group 1. The annotations are used to make explicit the implicit semantics of annotated elements and to improve their understandability.
   b) Group 2. The annotations are used to identify the common semantics among the annotated elements that from different sources and to support similarity matching operations (e.g. transformation).
   c) Group 3. The annotations are used to attach machine-readable semantics to the annotated elements and to obtain semantic reasoning supports (e.g. inference).

3. The column “Ways of Annotation” answers part of the question about “How to
It describes how semantic annotations are being added to the annotated elements. The contents of this column can be “Manual”, “Semi-automatic” or “Automatic”.

(4) The column “Semantic Browsers” answers part of the question about “How to annotate?”. It describes what browser is used to view the semantic models. In the case of automatic annotation, this column is omitted.

(5) The column “Employed Ontologies” answers part of the question about “How to annotate?”. It describes what ontologies are used in the corresponding research.

(6) The column “SA verification” describes whether there is a mechanism to verify the correctness of existing semantic annotations.

(7) The column “SA Schema” describes whether there is a semantic annotation schema in the corresponding research. In this column, the simple “is a” association is not considered as a schema.

(8) The column “SA Independency” describes how semantic annotations are attached to the annotated elements. They can be embedded as references (e.g. URI) in an annotation object or stored independent from it.

(9) The column “Aspects of Semantics” describes which aspects of semantics (structure and/or domain) of an annotated element are being made explicit in the corresponding research.
### Table 1. The Comparison of Semantic Annotation Researches

<table>
<thead>
<tr>
<th>Name of Author(s)</th>
<th>Application Domains</th>
<th>Usages of Annotation</th>
<th>Ways of Annotation</th>
<th>Semantic Browser</th>
<th>Employed Ontologies</th>
<th>SA Verification</th>
<th>SA Schema</th>
<th>SA Independency</th>
<th>Aspects of Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boudjlida et al. (2007) [22]</td>
<td>Models in an Enterprise</td>
<td>Group 1, 2 and 3</td>
<td>Manual</td>
<td>No Specify</td>
<td>Domain</td>
<td>No</td>
<td>Yes</td>
<td>Embedding URIs</td>
<td>Both</td>
</tr>
<tr>
<td>Liao [23]</td>
<td>Models in an Enterprise</td>
<td>Group 1, 2 and 3</td>
<td>Manual</td>
<td>Ontology tree view</td>
<td>Domain Meta-model</td>
<td>Yes</td>
<td>Yes</td>
<td>Independent</td>
<td>Both</td>
</tr>
<tr>
<td>Di Francescomarino (2011) [26]</td>
<td>Process Models</td>
<td>Group 2 and 3</td>
<td>Semi-automatic</td>
<td>No Specify</td>
<td>BPMN and BPO</td>
<td>Yes</td>
<td>Yes</td>
<td>Independent</td>
<td>Both</td>
</tr>
<tr>
<td>Lin (2008) [9]</td>
<td>Process Models</td>
<td>Group 1, 2 and 3</td>
<td>Manual</td>
<td>Ontology tree view</td>
<td>GPO, Goal and Domain</td>
<td>No</td>
<td>Yes</td>
<td>Independent</td>
<td>Both</td>
</tr>
<tr>
<td>Song et al. (2013) [29]</td>
<td>Data Models</td>
<td>Group 2 and 3</td>
<td>Automatic</td>
<td>Domain</td>
<td>No</td>
<td>No</td>
<td>Independent</td>
<td>Domain Semantics</td>
<td></td>
</tr>
<tr>
<td>Li (2012) [34]</td>
<td>CAD Models</td>
<td>Group 1, 2 and 3</td>
<td>Manual</td>
<td>Ontology tree view</td>
<td>STEP</td>
<td>No</td>
<td>Yes</td>
<td>Independent</td>
<td>Domain Semantics</td>
</tr>
<tr>
<td>Popov et al. (2003) [36]</td>
<td>Texts</td>
<td>Group 1 and 3</td>
<td>Automatic</td>
<td>KIM Knowledge base</td>
<td>No</td>
<td>No</td>
<td>Embedding URIs</td>
<td>Domain Semantics</td>
<td></td>
</tr>
<tr>
<td>Ma et al. (2011) [37]</td>
<td>Texts</td>
<td>Group 2 and 3</td>
<td>Automatic</td>
<td>Domain Language</td>
<td>Yes</td>
<td>Yes</td>
<td>Independent</td>
<td>Both</td>
<td></td>
</tr>
</tbody>
</table>
This comparison emphasizes several key information in a semantic annotation research, including: (1) Most of the semantic annotation researches focused on using semantic annotations to support the usage Group 3. (2) In the cases of manual and semi-automatic annotation, most of the researches provided ontology tree view as the semantic browser. (3) The verification mechanism is not taken into account by most of the semantic annotation researches. (4) Various kinds of semantic annotation schemas have been proposed by different researches. (5) In the case of semantic annotation for Web Services and Texts, semantic annotations are always embedded inside the annotation objects. To the contrary, concerning Models, semantic annotations are always independent. (6) Less than half of the researches take into account the structure semantics.

Furthermore, we discovered that there exists two extremes of the semantic annotation researches: (1) The researches that focus on developing an appropriate knowledge base, which has high-coverage of semantics, for example, the researches [35] and [36]. (2) The researches that focus on discovering a suitable semantic annotation structure model and related mechanisms, which has high-adaptability to different knowledge bases, for example, the researches [22], [23] and [40]. The challenges for the first direction are mainly the completeness and multiplicity of semantic models. The challenges for the second direction are mainly the applicability and tolerance of annotation models and related mechanisms. According to our research interest (the analysis of semantic annotation methods), in the next section, we will present a more detailed comparison from the formalization perspective.

3.2.2 The Comparison: from the Formalization Perspective

Totally, night semantic annotation schemas are discovered from the compared researches (Table 1). As it is shown in Figure 11, the elements inside those schemas are categorized into six types as follows:
(1) Element type ①, which contains the identifier of the annotated element.

(2) Element type ②, which contains the domain semantics.

(3) Element type ③, which contains the structure semantics.

(4) Element type ④, which contains the relations between the annotated element (that identified by ①) and its domain or structure semantics (② or ③).

(5) Element type ⑤, which contains some specific properties that are associated to the annotated element. It does not describe the semantics of the annotated element.

(6) Element type ⑥, which contains some specific properties that are associated to the semantic annotation itself.
General Annotation Schema:

--Annotation-Id: identifier of annotation
--Unformal Content: unformal comments
--Annotation Type: Type of annotation
--Ref2Ontology: URI of ontology concept
--Constraints: refer to ontology or meta-model

(a)

SA:=(E, P, SR, MME, MR)
where
E: a set of elements from the annotation object
P: a set of elements from PLC-related ontologies
MME: a set of elements from meta-model ontologies
SR: a set of binary relations from E to P
MR: a set of binary relations from E to MME

(b)

Three assertions of a BPD instance:
-- BPD instance
-- Three types of assertions
  BPM-type assertions
  BPM-structural assertions
  BPM-semantic assertions
-- Class of Ontology

(c)

PSAM=(AV,AR,AF,WP,I,O, ⊓pre, ⊓pos,E,PD)
where
AV is a set of activities
AR is a set of actor-roles
AF is a set of artifacts
WP is asset of workflow patterns
I is a set of input parameters
O is a set of output parameters
⊓pre is a set of pre-conditions
⊓pos is a set of post-conditions
E is a set of possible exceptions
PD is a subset of the domain ontology concepts

(d)

Annotation Schema in the shapeAnnotator:
--Class: selected class in domain ontology
--ShannGeoContextURI: a URI refer to a multi-segmented mesh
--ShannSegmentID: an index of a segment in the multi-segmented mesh
--Related Values: value is computed and added by the feature descriptors

(e)

Annotation Schema in the shapeAnnotator:

(f)

Semantic Annotation of Web Service (WS):
-- Sid: the identifier of a WS
-- Sname: the name of a WS
-- inputs: the input of a WS
-- outputs: the output of a WS
-- exec-time: the execution time of a WS
-- All-Resources: the required resources
-- Bindings: the used protocol
-- Service: the URI of a WS

(h)

Extension attributes in SAWSDL:
-- sawsdl:modelReference
-- sawsdl:liftingSchemaMapping
-- sawsdl:loweringSchemaMapping

(i)

Combining the element type with the column “SA independency” in Table 1, we can discover that this type of elements only exist in (b), (c), (d), (e), (f), (g) and (h). They belong to the researches that store semantic annotation independently. To the contrary, for those researches which embed references in the annotation objects, their schemas, such as (a) and (i), do not contain this type of elements.
All the schemas contain element type ②. Besides (a) and (i) which express this type of elements as URIs, the rest of them use ontology concepts. It is normally used to make explicit the domain semantics of the annotated elements. However, in (h), it is used to express the domain semantics of the annotated elements’ inputs and outputs.

Based on the observation of those schemas, only (a), (b) and (c) contain the element type ③. However, after analysis, we discover that the structure semantics are also taken into account by (d), (e) and (g). In (d), besides element PD, the rest of the elements in the schema are generated based on a meta-model ontology, named GPO. After mapping a meta-model to GPO, the corresponding model element is converted as an individual of the mapped class in GPO. In (e) and (g), structure semantics are not directly presented in the schema. Instead, the former one expressed it through the topological relations between two features (e.g. adjacency). The latter one represented it as a text model (language ontology) with pre-defined axioms.

Outwardly, besides (b), (c), (f) and (g), the rest of the schemas do not contain the element type ④, which defines the semantic relationship as a simple assertion or a link between an annotated element and its domain or structure semantics. In (b), there are two types of relations. The SR is a set of binary relations that describes the semantic relationships from the E to the P. The MR is a set of binary relations, which describes the semantic relationships from the E to the MME. In (c), relations are represented as three types of assertions. The “BPM-type assertions” assert an instance to a class of the BPMN ontology. The “BPM-semantic assertions” assert an instance to a class of a domain ontology. The “BPM-structure assertion” is used to describe the relations between two instances. In (f), the definitions of the “owl:ObjectProperty” and the “owl:DatatypeProperty” from OWL are
refined. The former one denotes the annotation content is an individual and the latter one denotes the annotation content is a data value. In (g), relations are classified into four kinds. The “sa:Concept” states the tf (text fragment) is annotated by an class. The “sa:Role” states the tf is annotated by a property. The “sa:Individual” states the tf is annotated by an individual. The “sa:Ind-Con” states the tf is an individual, as well as the annotation content is the class that the tf belongs to. Although element type ④ does not directly appear in (d), the semantic relationships are represented as OWL properties, such as Synonym, Polysemy, Hypernym, Hyponym, Meronym, Holonym, and Instance.

Element type ⑤ and ⑥ are usually used as a kind of additional elements of a semantic annotation to fulfil some particular requirements. For example, in (h), the “exec-time” is a property of the annotated element, which is used to record the execution time of a web service request. In (g) “Annotation-Id” is a property that is associated to the annotation itself, which is used to record the value of the identifier of that annotation.

4 Existing Drawbacks and Possible Research Directions

As discussed in previous sections, we found that despite lots of efforts have been made in semantic annotation researches, at least, three existing drawbacks can be noted.

The formalization of semantic annotations is not the focus in some of above-mentioned researches ([29], [30], [35], [36] and [41]), where it is only considered as a kind of “is a” association between an annotated element and an ontology concept. Meanwhile, some specific schemas are proposed by some of the rest ([8], [9], [26], [32], [34] and [37]). However, these schemas are difficult to be reused in other researches but the studied ones. There exists a kind of general schemas in the research [22], [23] and [40]. The research [22]
presented a general annotation schema, which still needs to be further formalized in more details. The research [23] acted as a successor of the previous research, and detailed the semantic annotation through a number of formal definitions. The research [40], although it provides to its user a large degree of freedom, it does not contain any semantic relationships and additional conventions.

Making explicit the domain semantics is the only concern in some of above-mentioned researches ([8], [29], [30], [34], [35], [36] and [41]), where the structure semantics is ignored. The advantages of making explicit the structure semantics have been acquired by the rest of them ([9], [22], [23], [26], [32] and [37]). In the research [9], it is used as a mediator for the reconciliation of various process modelling language constructs. In the research [22], it is used to express modelling construct and support models transformations. In the research [26], it is used to support the verification of modelling constraints. In the research [32], it is used to support the automatic computation of relations between features in a model. In the research [37], it is used to support the creation of text model and conserve linguistic knowledge. Among above-mentioned usages, the structure semantics and domain semantics are used separately. The research [23] is the only one that proposed to combine both aspects of semantics in the reasoning phase. However, this proposition needs the hypothesis that the interconnections between domain and meta-model ontologies are already prepared.

In the cases of automatic or semi-automatic annotation, semantic annotations are usually suggested by some similarity measurements methods ([26], [29], [30] and [41]) or training corpus([35], [36] and [37]). The correctness verification of semantic annotations is only taken into account by the researches [26] and [37]. In the research [26], four axioms were proposed to prevent erroneous annotations according to the types of concepts. In the research [37], two SWRL rules were designed to report missing and erroneous annotations
on a noun compound. However, these two researches only focus on the verification of one annotation on an annotated element. In the case of manual annotation, the research [23] designed three mechanisms to detect conflicts between two semantic annotations and to identify possible mistakes in an annotated model. Nonetheless, the proposed mechanisms relied on the hypothesis that the semantic similarity between two objects can be measured.

Therefore, taking into account all above considerations, three possible research directions for a future semantic annotation research are identified as follows:

(1) **The novel application of semantic annotations.** According to the classification of collected semantic annotation literature, as it is shown in Figure 1, more research efforts are needed to apply the semantic annotations on the subjects being described in the empty grids.

(2) **The standardization of the semantic annotation process.** The standardization of the essential procedures to apply the semantic annotation, which can be easily adopted by other semantic annotation researches. For the semantic annotation structure model, it is supposed to contain, at least, the element type ①, ② and/or ③, and ④.

(3) **The maintaining of annotation consistency.** Along with the versioning of annotated objects and the evolution of ontologies, there remains a promising challenge to carry out future researches in maintaining the consistency of semantic annotations.

5. **Conclusion**

In a PLM environment, various kinds of representations are used to capture and describe the knowledge related to a product along its life cycle. During the collaboration, a mutual understanding of the semantics inside these shared and exchanged knowledge representations
is the foundation to achieve the semantic interoperability. In this paper, we present a survey on a number of collected semantic annotation literature and provide a detailed comparison and discussion, especially from the formalization perspective. Based on this survey, several existing drawbacks and possible research directions are identified. In our on-going and future research works, we intend to enrich this survey with a more complete analysis from different perspectives, such as implementation and performance perspectives.

Acknowledgement

The authors would like to thank the financial support provided by the Charles Hermite Research Federation (CNRS FR 3198), the “Région Lorraine” local government, and the Pontifícia Universidade Católica do Paraná (PUCPR).

References


Biography

Yongxin Liao is the corresponding author. He is a Post-Doc researcher at the Graduate Program in Production Engineering and Systems (PPGEPS), Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil. He obtained his PhD degree in Automatic from the University of Lorraine in 2013. He obtained his Bachelor degree in Software Engineering from the Harbin Institute of Technology in 2008, and his Master degrees in Enterprise Computing and Engineering & Software Engineering from the University of Bordeaux 1 & the Harbin Institute of Technology in 2010. He specializes in production engineering, Semantic Web, data structuring, algorithm design, and programming.

Mario Lezoche is an assistant professor at the University of Lorraine, where he teaches object-oriented software engineering and database development. He conducts his research on Enterprise Model Interoperability at the Research Centre for Automatic Control (CRAN), Joint Research Unit with CNRS. He graduated at the Roma TRE University in Computer Science Engineering. He received his PhD degree in Computer Science Engineering in 2009. He has good experience in Semantic Web research and in models and semantics for systems interoperability. He is presently working on a conceptualization approach for enterprise information systems interoperability.

Eduardo Rocha Loures is an assistant professor at the Graduate Program in Production Engineering and Systems (PPGEPS), Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil. He received his Bachelor degree in Electrical Engineering from the Federal Technological University of Paraná, his Master degree in Informatics from the Pontifical Catholic University of Paraná, and his PhD in Industrial Systems from the University of Toulouse. He was a Post-Doc researcher at the Research Centre for Automatic Control (CRAN). His research interests include automated manufacturing systems, business process management, process aware information systems, and enterprise interoperability assessment.
Hervé Panetto is a full Professor of Enterprise Information Systems at the University of Lorraine where he teaches Information Systems modeling and development at the School of Engineering in Information Technology and conducts research at the CRAN. His is working on information systems modeling for enterprise applications and processes interoperability, with applications in enterprise modeling, manufacturing processes modeling and furniture data modeling. He is author or co-author of more than 80 papers in the field of automation engineering, enterprise modeling and enterprise systems integration and interoperability. He is currently Chair of the IFAC Technical Committee 5.3 “Enterprise Integration and Networking”.

Nacer Boudjlida is a full Professor at the University of Lorraine and a researcher at the Lorraine Research Laboratory in Computer Science and its Applications (LORIA), Joint Research Unit with CNRS. As a professor, his lectures include the various facets of databases (DBMS, database design, distribution, architecture, administration and so on). As a researcher, his domain of interest includes process modeling and support, distributed and cooperative computing, semantic-based mediated architectures. He authored two books on databases and numerous papers on software systems integration and interoperability, on software process modeling and support.