Semantic Annotation Model Definition for Systems Interoperability

Yongxin Liao[1], Mario Lezoche[1,2], Hervé Panetto[1], and Nacer Boudjlida[2]

[1] Centre de Recherche en Automatique de Nancy (CRAN), Nancy-University, CNRS, BP 70239,54506 Vandoeuvre-lès-Nancy Cedex, France
{Yongxin.Liao,Mario.Lezoche,Herve.Panetto}@cran.uhp-nancy.fr
[2] LORIA, Nancy-University, CNRS,BP 70239, 54506 Vandoeuvre-lès-Nancy Cedex, France
Nacer.Boudjlida@loria.fr

Abstract. Semantic annotation is one of the useful solutions to enrich target's (systems, models, meta-models, etc.) information. There are some papers which use semantic enrichment for different purposes (integration, composition, sharing and reuse, etc.) in several domains, but none of them provides a complete process of how to use semantic annotations. This paper identifies three main components of semantic annotation, proposes for it a formal definition and presents a survey of current semantic annotation methods. At the end, we present a simple case study to explain how our semantic annotation proposition can be applied.The survey presented in this paper will be the basis of our future research on models, semantics and architecture for enterprises systems interoperability during the product lifecycle.

Keywords: Semantic Annotation, Models, Ontology, Systems Interoperability.

1 Introduction

Nowadays, the need of systems collaboration across enterprises and through different domains has become more and more ubiquitous. But because the lack of standardized models or schemas, as well as semantic differences and inconsistencies problems, a series of research for data/model exchange, transformation, discovery and reuse are carried out in recent years. One of the main challenges in these researches is to overcome the gap among different data/model structures. Semantic annotation is not only just used for enriching the data/model's information, but also it can be one of the useful solutions for helping semi-automatic or even automatic systems interoperability.

Semantically annotating data/models can help to bridge the different knowledge representations. It can be used to discover matching between models elements, which helps information systems integration [1]. It can semantically enhance XML-Schemas' information, which supports XML documents transformation [2]. It can describe web services in a semantic network, which is used for further discovery and composition [3]. It can support system modellers in reusing process models, detecting cross-process relations, facilitating change management and knowledge transfer [4]. Semantic annotation can be widely used in many fields. It can link specific resources according to its domain ontologies.

R. Meersman, T. Dillon, and P. Herrero (Eds.): OTM 2011 Workshops, LNCS 7046, pp. 61–70, 2011.

The main contribution of this paper is identifying three main components of semantic annotation, giving a formal definition of semantic annotation and presenting a survey, based on the literature, of current semantic annotation methods that are applied for different purposes and domains. These annotation methods vary in their ontology (languages, tools and design), models and corresponding applications.

The remaining of this paper is organized as follows: Section 2 describes the definition of annotation and gives a formal definition of semantic annotations. Section 3 first provides the answers to why and where to use semantic annotation, then introduces the languages and tools that can be used to develop ontology, the design of semantic annotation structure models and the corresponding applications. Section 4 describes a simple case study example. Section 5 concludes this paper, together with some related work and potential extensions.

2 What Is Semantic Annotation?

In Oxford Dictionary Online, the word "annotation" is defined as "*a note by way of explanation or comment added to a text or diagram*". It is used to enrich target object's information, which can be in the forms of text descriptions, underlines, highlights, images, links, etc. Annotation has special meanings and usages in different fields. In java programming, annotation can be added on classes, methods, variables, parameters, etc., for example, JUnit[1] is a test framework that is based on java annotation. In mechanical drawing, an annotation is a snippet of text or symbols with specific meanings. In Library Management, an annotation is written in a set form (numbers, letters, etc.), which helps the classification of books.

Further, different annotation types are identified by [5] and [6]. They distinguished annotation as (i) *Textual annotation*: adding notes and comments to objects; (ii) *Link annotation*: linking objects to a readable content; (iii) *Semantic annotation*: that consists of semantic information (machine-readable). Similarly, three types of annotation are described in [7]: (i) *Informal annotation*: notes that are not machine-readable; (ii) *Formal annotation*: formally defined notes that are machine-readable (but it does not use ontology terms); (iii) *Ontological annotation*: notes that use only formally defined ontological terms (commonly accepted and understood).

According to the above classification, semantic annotation can be considered as a kind of formal metadata, which is machine and human readable.

2.1 Semantic Annotation

The term "Semantic Annotation" is described as "*An annotation assigns to an entity, which is in the text, a link to its semantic description. A semantic annotation is referent to an ontology*" in [3]. Semantic annotation is concerned as "*an approach to link ontologies to the original information sources*" in [8]. These definitions from different papers show one thing in common: a semantic annotation is the process of linking electronic resource to a specific ontology. Electronic resource can be text contents, images, video, services, etc. Ontology here is only one of the possible means to provide a formal semantic.

[1] `http://www.junit.org/`

As it can be seen on Figure 1, the left side represents an Electronic Resource (ER) and on the right side, there are the three main components of semantic annotation: (1) *Ontology*, which defines the terms used to describe and represent a body of knowledge [21]. (2) *Semantic Annotation Structure Model (SASM)*, which organizes the structure/schema of an annotation and describes the mappings between electronic resources and one, or more, ontologies. (3) *Application*, which is designed to achieve the user's purposes (composition, sharing and reuse, integration, etc.) by using SASM. This figure also shows the three main steps on how to use semantic annotation, which is introduced in section 3.

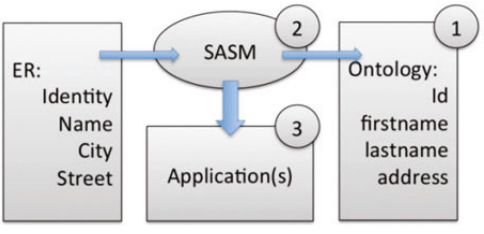

Fig. 1. Semantic Annotation components

2.2 Formal Definition of Semantic Annotation

The following definition formally defines a semantic annotation: a *S*emantic Annotation SA is a tuple $(\mathcal{M}, \mathcal{A})$ consisting of the SASM $\mathcal{M}$and an application $\mathcal{A}$.

$$SA := \{\mathcal{M}(\mathcal{E}, \mathcal{P}(\mathcal{O})), \mathcal{A}\}$$

Where:

$\mathcal{O} = \{o_1, o_2, \ldots, o_n\}$, is the set of ontology o_i that bring some meaning to any annotated element.

An *Ontology* $o_j \epsilon\, \mathcal{O}$ is a 4-tuple (C_{o_j}, is_a, R_{o_j}, σ_{o_j}), where C_{o_j} is a set of *concepts*, *is_a* is a partial order relation on C_{o_j}, $R_{o_j}$is a set of relation names, and $\sigma_{o_j}: R_{o_j} \rightarrow (C^+)$ is a function which defines each relation name with its arity [9].

Formally, $\mathcal{M} = \{m_x: \langle e_i, p_j \rangle |\ e_i \epsilon\, \mathcal{E} \times p_j \epsilon\, \mathcal{P}(\mathcal{O})\}$represents the set of relationships between an element e_i of the set of electronic resources $\mathcal{E}$ and an element p_j of the powerset of the ontology set $\mathcal{O}$.

$m_x(e_i, p_j)$ is a binary relationship that maps e_i to p_j in the context of an application $\mathcal{A}$. It may represent four different kinds of semantic relations:

(1) $m_{\sim}(e_i, p_j)$: stating that e_i is semantically *equivalent* to p_j.
(2) $m_{\supset}(e_i, p_j)$: stating that e_i *subsumes* the semantic of p_j.
(3) $m_{\subset}(e_i, p_j)$: stating that e_i *is subsumed by* the semantic of p_j.
(4) $m_{\cap}(e_i, p_j)$: stating that e_i *intersects* with the semantic of p_j.

$\mathcal{M}$ can be further extended, including also some additional parameters or constraints c_k, generally expressed using, in the worst case, natural language, or, better, a formal logical expression. $\mathcal{M}$ is then defined as $\mathcal{M} := \{m_x, c_k\}$.

The main issue, related to mappings such as in (2) and in (3), is being able to measure the semantic gap (2) or the over semantic (3), brought by the semantic annotation. Such measures have been studied by researchers in the domain of information retrieval [10] or in the domain of ontology merging [9], matching[11], mapping [12] and alignment[13].

In addition, [14] also gave a very simple definition of semantic annotation which is $SA := (R, O)$, where R is set of resources and O is an ontology. Furthermore, [15] defined it as $SA := \{R_A, C_A, P_A, L, T_A\}$. In this definition, R_A is a set of resources; C_A is a set of concept names; P_A is a set of property names; L is a set of literal values; and T_A is a set of triple (s, p, v), where $s \in R_A, p \in P_A, v \in (R_A \cup L)$. To the best of our knowledge, T_A in this definition is duplicated.

3 Why, Where and How to Use Semantic Annotation?

Semantic annotation uses ontology objects to enrich resource's information that tells a computer the meanings and relations of the data terms. It can be used to bridge the gap between modelsas additional information that helps description, discovery and composition.

Semantic Annotations are generally used in heterogeneous domains.We found several papers presenting different employments. [1] used semantic annotations to annotate the model/object at CIM, PIM and PSM levels of the MDA approach [17], which helps information system integration. In the research of [2], a path expression method is developed for adding annotations to XML-Schemas. Then they transform paths to ontology concepts and use them to create XML-Schema mappings that help XML document transformation. [3] used it to help matching and composition algorithm, which represents web services as a semantic network and produces the best composition plan. [4] used semantic description of process artefacts to help graphical modelling of business processes. In the research of [8], a semantic annotation framework is designed to manage the semantic heterogeneity of process model, to solve the discovery and sharing of process models problems in/between enterprise(s). In [16], an annotation framework is proposed for semi-automatically marking up web service descriptions (WSDL files) with domain ontologies to help web services discovery and composition. In the next three subsections we will present our proposed three semantic annotation components.

3.1 Step 1: Design or Select Ontology

Design or select an appropriate ontology for semantic annotations is the first step of the annotation process. Ontology has been actively studied for a long period of time, and there are many research works proposing ontology-engineering techniques, such as Ontolingua[2], F-logic[3], OWL[4], etc. We are not going to give, here, a complete

[2] http://www.ksl.stanford.edu/software/ontolingua/
[3] http://flora.sourceforge.net/
[4] http://www.w3.org/TR/owl-features/

overview of every ontology languages, but we provide a brief introduction toseveral examples.

[1] designed a SMM (Semantic Meta-model) to describe domain ontologies. Artefacts in ontology are castigated as DomainFunction and DomainObject. The relations (predicates) among Objects and Functions are defined as: *IsA*, *IsInputOf*, *IsOutputOf*, *Has*, etc. A RDF-like triple (e.g., *Tax Has TaxNumber*) is used as the statement in SMM.

[4] used two kinds of ontologies: sBPMN[5] ontology and a domain ontology. The first ontology is used to represent BPMN process models. The second ontology defines domain objects, states and actions according to objects lifecycle, which is used to provide the user advices during the modelling process.

[8] used Protégé OWL editor to design the ontology. In order to separately annotate meta-models (modelling language) and their process models, the author designs two ontologies: *General Process Ontology (GPO)* and *Domain Ontology*. The design of GPO is based on Bunge-Wand-Weber (BWW) Ontology [18]. The Domain ontology is formalized according to SCOR[6] specifications (Supply Chain Operations Reference-model).

3.2 Step 2: Design the Semantic Annotation Structure Model

The second component of a semantic annotation is SASM. It is the connection between electronic resources, applications and ontology concepts. A study in this direction is pursued by SAWSDL Working Group[7] that developed SAWSDL [19] (Semantic Annotation for Web Services Definition Language) which provides two kinds of extension attributes as follow: (i) *modelReference*, to describe the association between a WSDL or XML Schema component and a semantic model concept; (ii) *liftingSchemaMapping* and *loweringSchemaMapping*, to specify the mappings between semantic data and XML [20].

To be more specific, we abstract four structure models that are designed for different requirements. Figure 2 below gives an overview of these three SASMs: Model A is proposed to conceptually represent a web service from [3]; Model B is designed to annotate the business process model from [4]; and Model C is the annotation model for an activity element which is part of the Process Semantic Annotation Model (PSAM) from [8].

In order to compare above semantic annotation structure models, we identify five types for classifying the contents in SASM:

(1) *identity* of annotation;
(2) *reference to ontology concept*;
(3) *reference to element* (represent the relationship between element themselves);
(4) *text description*, the natural language definitions of annotation contents;
(5) *others* (extinction contents, such as: execution time, restriction, etc.).

[5] http://www.ip-super.org
[6] http://supply-chain.org/
[7] http://www.w3.org/2002/ws/sawsdl/#Introduction

We can easily find that the basic components of SASMs are: *identity of annotation* and *reference to ontology concepts*; *reference to element*, *text description* and *others* are added for different usages. As examples, [3] added "exec-time" into SASM to record the execution time of a web service request and used "inputs" and "outputs" to represent the relationships between processes; [4] described the references with meaning of states of objects (current, before and after); [8] adds "has_Actor−role" to denote the relationship between activity element and actor-role element.

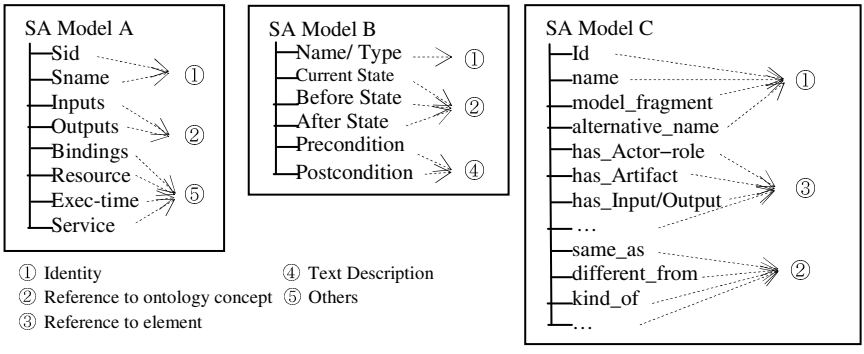

Fig. 2. Semantic Annotation Structure Model Examples

Furthermore, one to one mapping is not the only mapping type in SASM. For example, in Model A, there can be more than one input, which means the mapping between model content and ontology concept is one to many. Here, we analyses the mappings of "*reference to ontology concepts*".

Mappings are separated into two levels in the research of [8]: meta-model level and model level. In the meta-model level the mappings are defined as: *Atomic Construct*, *Enumerated Construct* and *Composed Construct*. In model level, semantic relationships are: *Synonym*, *Polysemy*, *Instance, Hyponym*, *Meronym*, *Holonymand Hypernym*. Five mapping types are described in [1]: *single representation*, *containment*, *compositions*, *multiple* and *alternative representation*.

In our opinions, there are three high level mapping types: *1 to 1* mapping, *1 to n* mapping and *n to 1* mapping (n to n is a combination of 1 to n and n to 1). For each of the mapping, we can design different semantic relationships for further usages. In *1 to 1* mapping semantic relationships can be like: "equal_to", "similar_to", etc. In *1 to n* they can be as: "contains", "has", etc. In *n to 1* they can be as: "part_of", "member_of", etc. One element can have several semantic relationships, but for each relationship, they belong to one mapping type.

Since the structure and semantic relationships are designed, we should consider how to implement the annotation process. It can be performed manually, semi-automatically or automatically. In the research of [8], mapping is manually linking the process models to ontology. [16] developed algorithms to semi-automatically match and annotate WSDL files with relevant ontologies. Automatic mapping is, for the moment, restricted to some simple cases because of the impossibility to completely explicit knowledge from the different models.

3.3 Step 3: Develop the Application

Once SASM step is accomplished, designers can begin to design the application to achieve their purpose (composition, sharing and reuse, integration, etc.). In several different domains semantic annotation methods are used to resolve particular problems.

In the domain of web services, [3] presented a matching algorithm to process the "input" and "output" (SASM model C, Figure 3) of elements, and builds a semantic network. This network is explored by a composition algorithm, which automatically finds a composite service to satisfy the request;

In business process modelling domain, [4] designed name-base and process context-base matchmaking functionalities to help user annotating process models. Name-base matching uses string distance metrics method for the matching between business process models and domain ontology. Process context-base matching uses the lifecycle (state before, state after, etc.) in domain ontology for suggesting the next activity during modelling.

[8] developed a prototype Process Semantic Annotation tool (Pro-SEAT), which is used to describe the relationship between process models and ontologies. They use Metis[8] as a modelling environment integrating Protégé OWL API to provide an ontology browser. Ontologies are stored on an ontology server, which can be loaded by annotators. The output of the annotation is an OWL instance file that is used to support the process knowledge query, discovery and navigation from users.

Indeed, there are many tools and technologies that enable designing applications in semantic annotation. The selections are always depending on the design of SASM and ontology. In any case, all three components of semantic annotation are closely related.

4 A Simple Case Study

To explain how our semantic annotation proposition can be applied, we give a simple example to illustrate the annotation process between Sage X3[9], an Enterprise Resource Planning (ERP) model and Flexnet[10], a Manufacturing Execution System (MES) model focused on Bill of Materials concept.

As can be seen in Figure 3, the ONTO-PDM product ontology, based on Tursi work [22], is used by both systems as the annotation source. In general cases two systems use different ontologies and the solution can be: mapping between each other; mapping with a reference ontology. Our solution provides SASM and its two main contents are "①identity of annotation", "②reference to ontology concept". The annotate process is performedmanually by a domain expert. Element "BillOfMaterialsID" in Sage X3 model is annotated as semantic annotation SA_1, which is referenced to the ontology concept "MaterialEntity"; Element "BomNumber" in Flexnet model is annotated as semantic annotation SA_2, which is referenced to the same ontology concept "MaterialEntity".

[8] http://www.troux.com/
[9] http://www.sage.com
[10] http://www.apriso.com/products/flexnet_production/

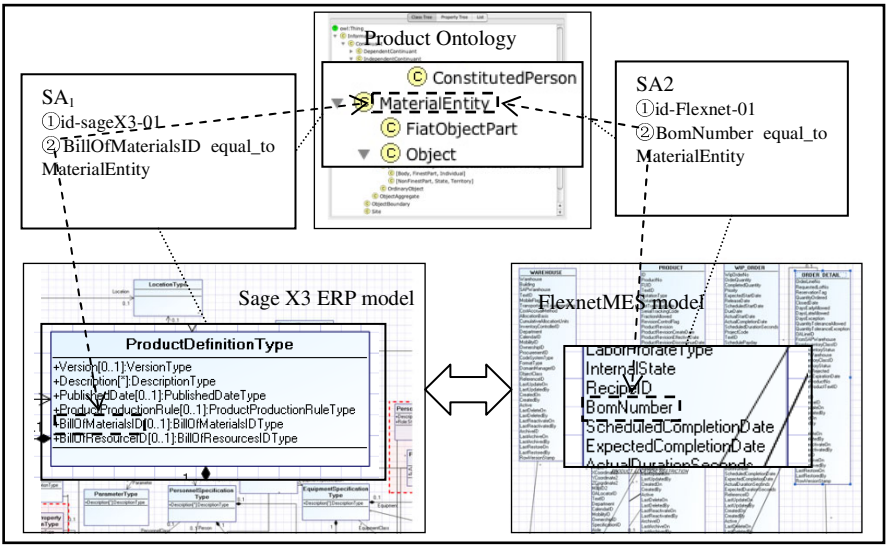

Fig. 3. Semantic Annotation between Sage X3 and Flexnet

Semantic Annotation SA_1 can be formally described as $SA_1 = \{m_{\sim}(ProductDefinitionType.BillOfMaterialsID, MaterialEntity), id_1, A\}$ (id_1 is the identity parameter that refers to"id-sageX3-01"). Similarly, SA_2 can be formally described as $SA_2 = \{m_{\sim}(WIP\_Order.BomNumber, MaterialEntity), id_2, A\}$ .Through the comparisonof SA_1 and SA_2, application A can match the two different models concepts. However, the design of our semantic annotation model definition still needs to be further developed. We will discuss the possible problems and future work in the last section.

5 Conclusions

In this paper, a brief survey of semantic annotation in different domains is presented. We identify three main components of semantic annotations that are Ontology, SASM and Application. In addition, a formal definition of semantic annotation is proposed. It contributes to better understand what a semantic annotation is and proposes a common reference model. But, how using semantic annotation? There are still many problems can be further discussed during the annotation process. For example, how to optimize ontology and an annotated model? How to solve the inconsistency or conflicts during the mapping? How to add consistent semantic on models in different levels of a system? How to achieve semi-automatic or automatic annotation?

We are currently investigating how semantic annotations can help collaborative actors (organizations, design teams, system developers, etc.) in co-designing, sharing, exchanging, aligning and transforming models. In particular, this research work will be based on general systems with several kinds of interactions. We can have interoperation between systems that with different versions (during many years, systems may have been modified or updated). We can also have systems with same

functions but used by different enterprises. Semantic annotations can bridge this knowledge gap and identify differences in models, in schemas, etc. In some case, interoperation is a process between a set of related systems throughout a product lifecycle (Marketing, Design, Manufacture, Service, etc.), and semantic annotations can influence the existing foundations and techniques which supports models reuse, semantic alignment and transformation, etc. Above all, our research work will focus on designing, and reusing appropriate ontologies in relationship with a formal semantic annotation structure model.

References

1. Agt, H., Bauhoff, G., Kutsche, R., Milanovic, N., Widiker, J.: Semantic Annotation and Conflict Analysis for Information System Integration. In: Proceedings of the 3rd Workshop on Model-Driven Tool & Process Integration, pp. 7–18 (2010)
2. Köpke, J., Eder, J.: Semantic Annotation of XML-Schema for Document Transformations. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 219–228. Springer, Heidelberg (2010)
3. Talantikite, H.N., Aïssani, D., Boudjlida, N.: Semantic annotations for web services discovery and composition. Computer Standards & Interfaces 31(6), 1108–1117 (2009)
4. Born, M., Dörr, F., Weber, I.: User-Friendly Semantic Annotation in Business Process Modeling. In: Weske, M., Hacid, M.-S., Godart, C. (eds.) WISE Workshops 2007. LNCS, vol. 4832, pp. 260–271. Springer, Heidelberg (2007)
5. Bechhofer, S., Carr, L., Goble, C., Kampa, S., Miles-Board, T.: The Semantics of Semantic Annotation. In: Proceedings of the 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, pp. 1151–1167 (2002)
6. Boudjlida, N., Dong, C., Baïna, S., Panetto, H., Krogstie, J., Hahn, A., Hausmann, K., Tomás, J.V., Poler, R., Abián, M.Á., Núñez, M.J., Zouggar, N., Diamantini, C., Tinella, S.: A practical experiment on semantic enrichment of enterprise models in a homogeneous environment. INTEROP NoE Deliverable DTG4.1. INTEROP NoE IST 508011 (2006), doi:http://www.interop-vlab.eu
7. Oren, E., HinnerkMöller, K., Scerri, S., Handschuh, S., Sintek, M.: What are Semantic Annotations? Technical report, DERI Galway (2006)
8. Lin, Y.: Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway (2008)
9. Stumme, G., Maedche, A.: Ontology Merging for Federated Ontologies on the Semantic Web. In: Proceedings of the International Workshop for Foundations of Models for Information Integration (2001)
10. Ellis, D.: The Dilemma of Measurement in Information Retrieval Research. Journal of the American Society for Information 47(1), 23–36 (1996)
11. 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)
12. Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y.: Learning to map between ontologies on the semantic web. In: Proceedings of the World Wide Web Conference, pp. 662–673 (2002)

13. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: Proceedings of the 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence, pp. 450–455 (2000)
14. Peng, W., Baowen, X., Jianjiang, L., Dazhou, K., Yanhui, L.: A Novel Approach to Semantic Annotation Based on Multi-ontologies. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, vol. 3, pp. 1452–1457 (2004)
15. Luong, P., Dieng-Kuntz, R.: A Rule-based Approach for Semantic Annotation Evolution. Computational Intelligence 23(3), 320–338 (2007)
16. Patil, A., Oundhakar, S., Sheth, A., Verma, K.: Meteor-S Web Service annotation framework. In: Proceedings of the 13th International Conference on the World Wide Web, pp. 553–562 (2004)
17. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: Model-Driven Architecture. In: Bruel, J.-M., Bellahsène, Z. (eds.) OOIS 2002. LNCS, vol. 2426, pp. 290–297. Springer, Heidelberg (2002)
18. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information System Journal 3(4), 217–237 (1993)
19. Kopecký, J., Vitvar, T., Bournez, C., Farrell, J.: SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Computing 11(6), 60–67 (2007)
20. Martin, D., Paolucci, M., Wagner, M.: Towards Semantic Annotations of Web Services: OWL-S from the SAWSDL Perspective. In: Proceedings of the OWL-S Experiences and Future Developments Workshop at ESWC 2007 (2007)
21. Boyce, S., Pahl, C.: Developing Domain Ontologies for Course Content. Educational Technology & Society, 275–288 (2007)
22. Tursi, A., Panetto, H., Morel, G., Dassisti, M.: Ontological approach for Products-Centric Information System Interoperability in Networked Manufacturing Enterprises. IFAC Annual Reviews in Control 33(2), 238–245 (2009)