Why, Where and How to use Semantic Annotation for Systems Interoperability

Yongxin Liao, Mario Lezoche, Hervé Panetto, Nacer Boudjlida

To cite this version:

Yongxin Liao, Mario Lezoche, Hervé Panetto, Nacer Boudjlida. Why, Where and How to use Semantic Annotation for Systems Interoperability. 1st UNITE Doctoral Symposium, Jun 2011, Bucarest, Romania. pp.71-78. hal-00597903

HAL Id: hal-00597903

https://hal.archives-ouvertes.fr/hal-00597903

Submitted on 2 Jun 2011

Why, Where and How to use Semantic Annotation for Systems Interoperability

Yongxin Liao[1], Mario Lezoche[1, 2], Hervé Panetto[1], Nacer Boudjlida[2]
[1]Research Centre for Automatic Control (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, gives a formal definition of semantic annotation method and presents a survey of current semantic annotation methods which include: languages and tools that can be used to develop ontology, the design of semantic annotation structure models and the corresponding applications. The survey presented in this paper will be the basis of our future research on models, semantics and architecture for systems interoperability.

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 (Agt, et al., 2010). It can semantically enhance XML-Schemas' information, which supports XML documents transformation (Köpke and Eder, 2010). It can describe web services in a semantic network, which is used for further discovery and composition (Talantikite, et al., 2009). It can support system modellers in reusing process models, detecting cross-process relations, facilitating change management and knowledge transfer (Bron, et al., 2007). Semantic annotation can be widely used in many fields. It can link specific resources according to its domain ontologies.

The main contribution of this paper is identifying three main components of semantic annotation, gives 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 provides the answers to *why* and *where* to use semantic annotation. Section 4 first presents an introduction to ontologies and semantic annoation structure models, and then discusses the usage of semantic annotations. Section 5 concludes this paper, together with some related work and potential extensions.

2. WHAT IS SEMANTIC ANNOTATION?

In this section, we first illustrate the types of annotations from different papers (section 2.1), and then propose a formal definition of semantic annotation together with its three main components (section 2.2).

2.1 Definition and Types of annotation

In Oxford Dictionary Online[1], 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 software programming, an annotation is represented as text comments embedded in the code to expand the program, which is being ignored when the program is running. 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.

[1]http://oxforddictionaries.com

Further, different annotation types are identified by the following papers: Bechhofer, et al. (2002) and Boudjlida, et al. (2006) 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 which is machine-readable. Similarly, three types of annotation are described in the research of Oren, et al. (2006): (i) *Informal annotation*: notes that are not machine-readable; (ii) *Formal annotation*: notes that are formally defined and machine-readable (but it does not use ontology terms); (iii) *Ontological annotation*: notes that use only formally defined ontological terms that are commonly accepted and understood.

Bechhofer, et al. (2002) further classified the annotation according to six possible uses that are not always clear and disjoint: (a) *Decoration*, comments on an object; (b) *Linking*, link anchors; (c) *Instances Identification*, strong assert that an object is an instance of a particular class. It may use a URI; (d) *Instance Reference*, less clear than instance identification, reference depending on background and world knowledge; (e) *Aboutness*, loose association of the object with a concept; (f) *Pertinence*, assertions about the concepts within an ontology without encoding that information.

According to the above classification, semantic annotation can be considered as a kind of formal metadata, which is machine and human readable. This will be further discussed in the following sections.

2.2 Semantic annotation

The term "Semantic Annotation" is described as "*the action and results of describing (part of) an electronic resource by means of metadata whose meaning is formally specified in an ontology*" (electronic resource can be text contents, images, video, services, etc.) by Fernández (2010). Talantikite, et al. (2009) introduced it 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 the research of Lin (2008), semantic annotation is concerned as "*an approach to link ontologies to the original information sources*". All above definitions from different papers show one thing in common: a semantic annotation is the process of linking electronic resource to a specific ontology. Ontology here is only one of the possible means to provide a formal semantic.

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 (Boyce, et al., 2007). It can be reused from existing ontologies or designed according to different requirements. (2) *Semantic Annotation Structure Model (SASM)*, which organizes the structure/schema of an annotation and describes the mappings between electronic resources and an ontology. (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 4: ontology (section 4.1), semantic annotation structure model (section 4.2) and application (section 4.3).

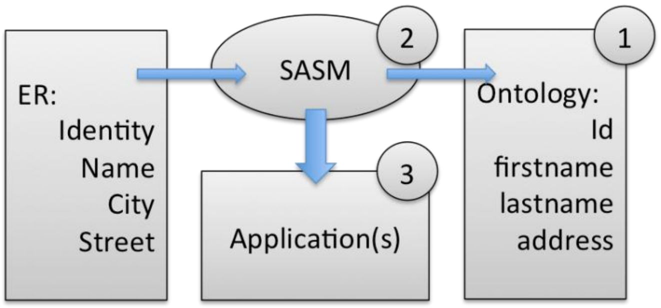

Fig. 1. Semantic Annotation components

The following definition formally defines a semantic annotation: a Semantic 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:
$\mathcal{O} = \{o_1, o_2, \dots, o_n\}$, 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 $\sigma_{o_j}: R_{o_j} \to (C^+)$ is a function which defines each relation name with its arity (Stumme and Maedche, 2001a).

Formally, $\mathcal{M} = \{m_x: \langle e_i, p_j \rangle |\ e_i\ \epsilon\ \mathcal{E}\ \times\ p_j\ \epsilon\ \mathcal{P}(\mathcal{O})\}$ and 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 $\mathcal{O}$.

A mapping $m_x(e_i, p_j)$ may represent three different kinds of semantic relations:

(1) $m_\sim(e_i, p_j)$ is a binary *equivalence* relation. If $m_\sim(e_i, p_j)$ then an electronic resource e_i is semantically equivalent to p_j, an element of the powerset $\mathcal{P}(\mathcal{O})$, in the context of an application $\mathcal{A}$.

(2) $m_\supset(e_i, p_j)$ is a binary relation stating that the semantic of an electronic resource e_i *subsumes* the semantic of an element p_j of the powerset $\mathcal{P}(\mathcal{O})$, in the context of an application $\mathcal{A}$.

(3) $m_\subset(e_i, p_j)$: is a binary relation stating that the semantic of an electronic resource e_i *is subsumed* by the semantic of an element p_j of the powerset $\mathcal{P}(\mathcal{O})$, in the context of an application $\mathcal{A}$.

$\mathcal{M}$ 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 (Ellis, 1996) or in the domain of ontology matching (Maedche and Staab, 2002), mapping (Doan et al, 2002), merging (Stumme and Maedche, 2001b), alignment (Noy and Musen, 2000).

In addition, Peng, et al. (2004) also gave a very simple definition of semantic annotation in their paper, which is $SA := (R, O)$, where R is set of resources and O is an ontology. Furthermore, Luong and Dieng-Kuntz (2007) 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 TO USE SEMANTIC ANNOTATION

A semantic annotation uses ontology objects for enriching resource's information that tells a computer the meanings and relations of the data terms. It can be used in many areas, such as Business Process Models, Web services, XML Schema, Strategic Data Models, Information Systems, etc. Several usages of semantic annotation are introduced:

Business Process Models: In the research of Lin (2008), 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). Born, et al. (2007) used semantic description of process artefacts to help a modeller in graphical modelling of business processes.

Web Services: Talantikite, et al. (2009) used a semantic annotation to represent web services as a semantic network. Based on the network and submitted requests, the composition algorithm produces the best composition plan. Patil, et al. (2004) proposed an annotation framework for semi-automatically marking up web service descriptions (WSDL files) with domain ontologies to help web services discovery and composition.

XML Schema: In the research of Köpke and Eder (2010), a path expression method is used to add annotation to XML-Schemas. Then they transform paths to ontology concepts and use them to create XML-Schema mappings that help XML document transformation.

Strategic Data Models: Diamantini and Potena (2008) presented a novel model that uses a mathematical ontology in semantic annotation to describe mathematical formulas in Data Warehouse schemas.

Information System: Agt, et al. (2010) used semantic annotations to help information system integration. They annotate the model/object at CIM (Computation Independent Model), PIM (Platform Independent Model) and PSM (Platform Specific Model) levels of the MDA approach (Mellor, et al. 2002; 2004), and then they discover some matching between model elements with respect to semantic process requirements.

In short, semantic annotation can be considered as a semantically enrichment of models or data, which may be widely used for many purposes. In business process models and Information system, it can be used to bridge the gap between two models. In Web service and Strategic Date Models, it can be used as additional information that helps description, discovery and composition. To the best of our knowledge, the path expression method in XML Schema will lead to lose information in Schema (e.g. restrictions of max-occur/min-occur, sequence or choice of elements, etc.), which still needs to be improved.

4. HOW TO USE SEMANTIC ANNOTATION

In this section, we present an introduction to the three main components of semantic annotation: the languages and tools which can be used in designing ontology; semantic annotation model's structure and mappings; and the applications of semantic annotation.

4.1 Introduction to Ontology

Designing 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. We are not going to give, here, a complete overview of every ontology languages, but we provide a brief introduction to the three more representative languages. We will show also some simple examples and typical development tools.

Ontolingua was developed by KSL (Knowledge Systems Lab, Stanford University) (Fikes, et al, 1997). It is an extension of KIF[2] (Knowledge Interchange Format) through adding frame-based representation and translation functionalities. But because of the newly development of semantic web ontology, Ontolingua is not frequently used recently. Figure 2-a) shows a simple Ontolingua example from Mizoguchi (2003). Ontolingua Server[3] provides an editor, which can be used to browse, create, edit, modify, and use Ontolingua ontologies.

F-Logic was presented by Michael Kifer (Stony Brook University) and Georg Lausen (University of Mannheim) (Kifer and Lausen, 1995). It is an object-oriented language that is frequently used for Semantic Web. It also can map straightforward to most frequent ontological constructs. Figure 2-b) shows a simple F-logic example from Liao, et al.

[2]http://www-ksl.stanford.edu/knowledge-sharing/kif/
[3]http://www.ksl.stanford.edu/software/ontolingua/

(2010). Flora2 [4] is an F-logic ontology development application, which extends F-logic with HiLog and Transaction Logic.

OWL (Web Ontology Language) was developed by World Wide Web Consortium, which shares many characteristics with RDF [5] (Resource Description Framework) and RDF Schema (Horrocks, et al., 2003). It is written using the XML syntax, and contains three sublanguages: OWL Lite, OWL DL and OWL Full. OWL is considered as a standard language for ontology representation for semantic web. Figure 2-c) shows a simple OWL example from OWL Guide[6]. Protégé[7] ontology editor is a Java-based tool that can export ontology into formats such as OWL, RDF and XML Schema. *OntoStudio*[8] supports the modelling of RDF(S), OWL and Object-Logic with possible transformation between them.

```
a) Ontolingua
(define-class Tutoring-objective (?t-obj)
"Attributes are also represented as slots"
:def (and (individual ?t-obj)
     (value-type ?t-objTuroring.policy Policy))
:axiom-def (subclass-partition Tutoring-objective
     (setof Transfer-ofknowledge Remedy)))
```

```
b) F-logic
General class information:
     person[name*=>string, children*=>person].
Database facts:
     John:person[name->'John Doe', children-> {Bob, Mary}].
Mary:person[name->'Mary Doe', ciildren->{Alice}]
Deductive rule:
     ?X:human:- ?X:person
Query:
     ?X:person[name->?Y, children->Mary]
```

```
c) OWL
Class and Individuals:
     <owl:Classrdf:ID="Wine">
<rdfs:subClassOfrdf:resource="&food;PotableLiquid"/>
<rdfs:labelxml:lang="en">wine</rdfs:label>
<rdfs:labelxml:lang="fr">vin</rdfs:label>
</owl:Class>
Properties:
<owl:ObjectPropertyrdf:ID="hasWineDescriptor">
<rdfs:domainrdf:resource="#Wine" />
<rdfs:rangerdf:resource="#WineDescriptor" />
     </owl:ObjectProperty>
```

Fig. 2. Examples of Ontolingua (a), F-logic (b) and OWL (c).

The design methods of ontology for annotations have their own purposes and structures.

[4]http://flora.sourceforge.net/
[5]http://www.w3.org/RDF/
[6]http://www.w3.org/TR/2004/REC-owl-guide-20040210/
[7]http://protege.stanford.edu/
[8]http://www.ontoprise.de/en/products/ontostudio/

Lin (2008) 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 (Bunge 1977; Wand and Weber, 1993). GPO contains nine main concepts: *Activity*, *Artifact*, *Actor-role*, *Input*, *Output*, *Precondition*, *Postcondition*, *Exception* and *WorkflowPattern*. Relations between above concepts are *has_actor-role*, *has_artifact*, *has_subActiviy*, *has_input*, *has_output*, *related_to*, *has_precondition*, *has_postcondition*, *has_exception*, *handled_by* (e.g. *Activity* uses *has_actor-role* relation to link *Actor-role*). The Domain ontology is formalized according to SCOR [9] specifications (Supply Chain Operations Reference-model).

Agt, et al. (2010) designed a semantic meta-model (SMM) 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*, *IsListOf*, *IsEquivalentTo*, etc. A RDF-like triple (e.g., *Tax Has TaxNumber*) is used as the statement in SMM.

Born, et al. (2007) used two kinds of ontologies: sBPMN[10] 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. More details of above ontologies can be found in references.

4.2 Introduction to Semantic Annotation Structure Model

The second component of a semantic annotation is SASM. It is the connection between electronic resources and ontology concepts. A study in this direction is pursued by SAWSDL Working Group [11] that developed SAWSDL (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 *loweringSchema-Mapping*, to specify the mappings between semantic data and XML (Martin, et al., 2007; Kopecký, et al. 2007).

To be more specific, we analyse four SASMs that are designed for different requirements. Figure 3 below gives an overview of these four SASMs: Model A is the annotation schema for enterprise models from Boudjlida and Panetto (2007); Model B is designed to annotate the business process model from Born, et al. (2007); Model C is proposed to conceptually represent a web service from Talantikite, et al. (2009); and Model D is the annotation model for an activity element which is part of the Process Semantic Annotation Model (PSAM) from Lin (2008).

[9]http://supply-chain.org/
[10]http://www.ip-super.org
[11]http://www.w3.org/2002/ws/sawsdl/#Introduction

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

(1) *identity* of annotation (e.g. id, name, etc.);
(2) *reference to ontology concept* (e.g. element Customer has a reference "same_as" which is referenced to ontology concept Buyer);
(3) *reference to element* (represent the relationship between element themselves. e.g. element manufacture has a reference "has_input" which is referenced to element material);
(4) *text description*, the natural language definitions of annotation contents;
(5) *others* (extinction contents, such as: execution time, restriction, annotation types, etc.). The classification results of each SASM are described by linking model contents to type numbers

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 an example, Lin (2008) adds "has_Actor−role" to denote the relationship between activity element and actor-role element; Boudjlida and Panetto (2007) added "Informal Content" for explaining the intent of the annotation; Talantikite, et al. (2009) added "exec-time" into SASM to record the execution time of a web service request. In the rest of this section, the discussion is focused on the design of *reference to ontology concepts*.

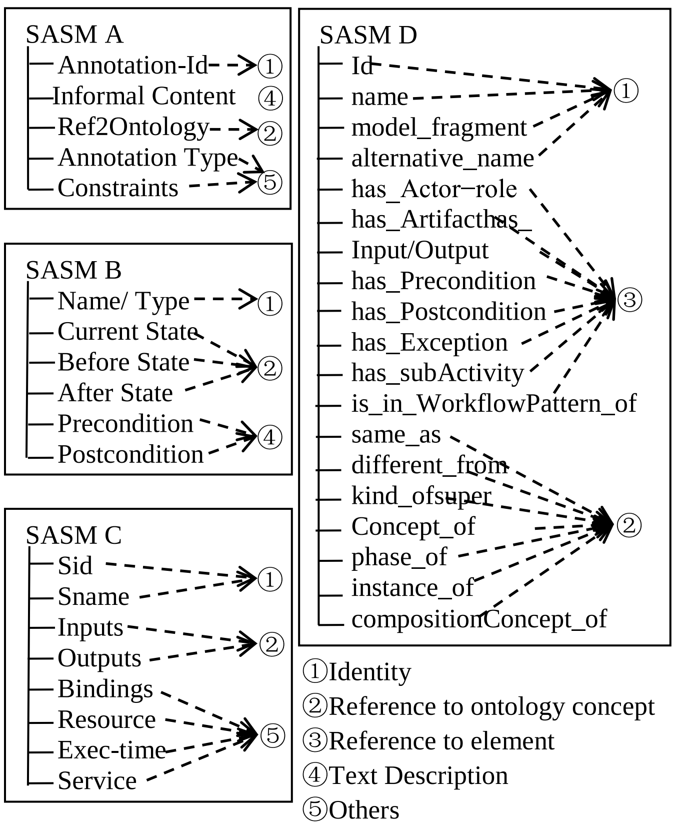

Fig. 3. Semantic Annotation Structure Model Examples.

As can be seen from above figure, *reference to ontology concepts* in model A is just a conceptual reference without meanings. Model B describes the references with meaning of states of objects (current, before and after). Model C uses inputs and outputs to represent the relationships. Model D gives more meanings to references like same_as, kind_of, phase_of, etc. Further, one to one mapping is not the only mapping type in SASM. For example, in Model C, there can be more than one input, which means the mapping between model content and ontology concept is one to many. Here, we analyses "*reference to ontology concepts*" according to mapping types and definitions of mappings.

Mappings are separated into two levels in the research of Lin (2008): meta-model level and model level. In the meta-model level, mapping direction is from ontology to model contents. The mappings are defined as: *Atomic Construct* (one to one. e.g. Activity is mapped to Task), *Enumerated Construct* (one to many. e.g. Artifact is mapped to Information or Material) and *Composed Construct* (one to combination. e.g. Workflow Pattern is mapped to a combination of Flow and Decision Point). In model level, semantic relationships are: *Synonym* (same_as, alternative_name), *Polysemy* (different_from), *Instance* (instance_of), *Hyponym* (superConcept_of), *Meronym* (part_of, member_of, phase_of and partial Effect_of), *Holonym* (composition Concept_of) *and Hypernym* (kind_of). (e.g. *Meronym*: *Airline* member_of *Air Alliance*). Agt, et al. (2010) described five mapping types in their work: *single representation* (one model element to one ontology concept), *containment* (one model element to multiple ontology concepts), *compositions* (multiple model elements to one ontology concepts), *multiple* and *alternative representation* (the mappings with AND and OR/XOR operators). Table 2 shows the comparison and classification of the mappings from Agt, et al. (2010) and Lin (2008). In order to classify those mappings, we assume the mapping direction in the table is from a model element to an ontology concept.

Table.1. Mappings from Model to Ontology

Types	Lin (2008)	Lin (2008)	Agt, et al.(2010)
1 to 1	Atomic Construct	Instance Synonym Polysemy Hyponym Hypernym	Single represent
1 to n			Containment, Multiple Alternative
n to 1	Enumerated Construct Composed Construct	Meronym Holonym	Composition

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. Figure 4 shows the mapping types and semantic relationships for each kind of mapping. *1 to 1* means one element is annotated by one ontology concept. Semantic relationships can be: equal_to, similar_to, etc. *1 to n* means one element is annotated by the composition/aggregation of several ontology concepts. Semantic relationships can be: contains, has, etc. *n to 1* means the composition/aggregation of several elements are annotated by one ontology concept. Semantic relationships can be: part_of, member_of, etc. One element can have several semantic relationships, but for each relationship, they belong to one mapping type.

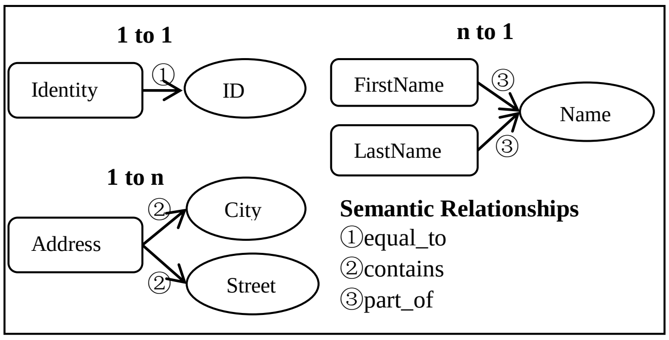

Fig. 4. Mapping types and semantic relationships

Since the structure and semantic relationships of SASM are designed, we should consider how to implement the annotation process. The annotation process can be performed manually, semi-automatically or automatically (Reeve and Han, 2005). In the research of Lin (2008), mapping is manually linking the process models to ontology. In the work of Patil, et al. (2004), mapping is semi-automatically computed. They developed algorithms to 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.

4.3 Introduction to Application

Once the semantic annotation structure model is defined, designers can begin to design the application to achieve their purpose (composition, sharing and reuse, integration, etc.). Several applications of semantic annotation are introduced as follow:
Talantikite, et al. (2008) designed an application, which uses a matching algorithm to process the "input" and "output" (SASM model C, Figure 3) of elements, and builds a semantic network for web services. This semantic network is explored by a composition algorithm, which automatically finds a composite service to satisfy the request. Authors implement a prototype in java, which includes: Pellet [12] Reasoner (matching algorithm), RSsw (Réseau Sémantique des Services Web), Request and Composor (returns an optimal composite service for requesters).

Lin (2008) developed a prototype Process Semantic Annotation tool (Pro-SEAT) to describe the relationship between process models and ontologies. They use Metis[13] as a modelling environment integrating Protégé OWL API to provide the OWL ontology browser. Ontologies (GPO, Domain ontology, etc.) are stored on an ontology server, which can be loaded by annotators. The output of the annotation is an OWL instance file, which is used by a knowledge repository service to support the process knowledge query, discovery and navigation from users.

Born, et al. (2007) used Tensegrity Graph Framework[14] as environment to support graphical design functions. Name-base and Process Context-base matchmaking functionalities are designed to help user annotating process models. Name-base matching uses string distance metrics method for the matching between business process models and domain ontology, and it supports the user for specifying or refining the process. Process Context-base matching uses the lifecycle (state before, state after, etc.) in domain ontology for suggesting the next activity during modelling.

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

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, Semantic Annotation Structure Model and Application. In addition, a formal definition of semantic annotation is proposed. It contributes to better understand what a semantic annotation is and then contributes to a common reference model. How to use 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

[12]http://clarkparsia.com/pellet/
[13]http://www.troux.com/

[14]http://www.tensegrity-software.com/

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

Agt, H., Bauhoff, G., Kutsche, R., Milanovic, N., Widiker, J. (2010). Semantic Annotation and Conflict Analysis for Information System Integration. In: Proceedings of the 3rd Workshop on Model-Driven Tool & Process Integration. 7-18

Bechhofer, S., Carr, L., Goble, C., Kampa, S. and Miles-Board, T. (2002). The Semantics of Semantic Annotation. In: Proceedings of the 1st International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. 1151-1167.

Born, M., Dorr, F., Weber, I. (2007). User-Friendly Semantic Annotation in Business Process Modeling. In: Proceedings of the 2007 international conference on Web information systems engineering.

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. (2006). A practical experiment on semantic enrichment of enterprise models in a homogeneous environment. INTEROP NoE Deliverable DTG4.1. INTEROP NoE IST 508011. http://www.interop-vlab.eu

Boudjlida, N., Panetto, H. (2007). Enterprise semantic modelling for interoperability. In: Proceedings of the 12th IEEE conference on emerging technologies and factory automation, Patras, Greece. 847–854.

Boyce, S., Pahl, C. (2007). Developing Domain Ontologies for Course Content. Educational Technology & Society 275-288.

Bunge, M. (1977): Treatise on Basic Philosophy (Vol 3): Ontology I: The Furniture of the World. D. Reidel Publishing Company, first edition

Diamantini, C., Potena, D. (2008). Semantic enrichment of strategic datacubes. In: Proceedings of the ACM 11th International Workshop on Data Warehousing and OLAP. 81-88.

Ding, G., Xu, N. (2010). Automatic semantic annotation of images based on web data. In: Proceedings of the 6th international conference of Information Assurance and Security. 317-322

Doan, A., Madhavan, J., Domingos, P., Halevy, A.Y. (2002). Learning to map between ontologies on the semantic web. In Proceedings of the World Wide Web conference. 662-673.

Ellis, D. (1996). The Dilemma of Measurement in Information Retrieval Research. Journal of the American Society for Information. Vol. 47, N° 1. 23-36.

Fernández, N. (2010). Semantic Annotation Introduction, [online] (Updated 14 Oct 2010) Available at <http://www.it.uc3m.es/labgimi/teoria/Module2/SA-Intro.pdf>

Fikes, R., Farquhar, A., Rice, J. (1997). Tools for Assembling Modular Ontologies in Ontolingua. In: Proceedings of the 14th national conference on artificial intelligence and 9th conference on Innovative applications of artificial intelligence. 436-441.

Horrocks, I., Patel-Schneider, P.F., Harmelen, F.V. (2003) From SHIQ and RDF to OWL: the making of a Web Ontology Language. Journal of Web Semantics. Vol 1, N° 1. 7-26.

Irfanullah, I., Aslam, N., Loo, J., Loomes, M., Roohullah, R. (2010). In: Proceedings of the IEEE international symposium on Signal Processing and Information Technology. 491-495

Lin, Y. (2008). Semantic Annotation for Process Models: Facilitating Process Knowledge Management via Semantic Interoperability. PhD thesis, Norwegian University of Science and Technology, Trondheim, Norway.

Liao, Y., Romain, D., J. Berre, A. (2010). Model-driven Rule-based Mediation in XML Data Exchange. In: Proceedings of the 1st International Workshop on Model-Driven Interoperability. 89-97

Luong, P., Dieng-Kuntz, R. (2007). A Rule-based Approach for Semantic Annotation Evolution. In Computational Intelligence. Vol. 23, Issue 3, 320–338

Kifer, M., Lausen, G., Wu, J. (1995). Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the ACM. Vol.42, N°4. 741-843.

Kopecký, J., Vitvar, T., Bournez, C., Farrell, J. (2007). SAWSDL: Semantic Annotations for WSDL and XML Schema. IEEE Internet Computing. Vol.11, N° 6. 60-67

Köpke, J., Eder, J. (2010). Semantic Annotation of XML-Schema for Document Transformations. In: Proceedings of the OTM Workshops. 5th International Workshop on Enterprise Integration, Interoperability and Networking. Lecture Notes in Computer Science, Vol. 6428. 219-228.

Maedche, A., Staab, S. (2002). Measuring Similarity between Ontologies. In: Proceeding of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web. 251-263.

Mellor, S.J., Scott, K., Uhl, A., Weise, D. (2002). Model-Driven Architecture. In: Proceedings of the Workshop at the 8th International Conference on Object-Oriented Information Systems. 290-297.

Mellor S.J., Kendall S., Uhl A. and Weise D. (2004). Model Driven Architecture, Addison-Wesley Pub Co.

Martin, D., Paolucci M., Wagner, M. (2007). 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.

Mizoguchi, R. (2003). Tutorial on Ontological Engineering: Part 2: Ontology Development, Tools and Languages. New Generation Comput.

Noy, N.F., Musen, M.A. (2000). 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. 450-455.

Oren, E., Hinnerk Möller, K., Scerri, S., Handschuh, S., Sintek, M.(2006). What are Semantic Annotations?. Technical report, DERI Galway

Patil, A., Oundhakar, S., Sheth, A., Verma, K. (2004). Meteor-S Web Service annotation framework. In: Proceedings of the 13th International Conference on the World Wide Web. 553-562.

Peng, W., Baowen, X., Jianjiang, L., Dazhou, K., Yanhui, L. (2004). A Novel Approach to Semantic Annotation Based on Multi-ontologies. In: Proceedings of the third International Conference on Machine Learning and Cybernetics. Vol. 3. 1452 - 1457

Reeve, L.H., Han, H. (2005). Survey of semantic annotation platforms. In: Proceedings of the ACM Symposium on Applied Computing. 1634-1638

Stumme, G., Maedche, A. (2001a). Ontology Merging for Federated Ontologies on the Semantic Web. In: Proceedings of the International Workshop for Foundations of Models for Information Integration.

Stumme, G., Maedche, A. (2001b). FCA-MERGE: Bottom-Up Merging of Ontologies. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence. Seattle, Washington, USA. 225-234.

Talantikite, H.N., Aïssani, D., Boudjlida, N. (2009). Semantic annotations for web services discovery and composition. Computer Standards & Interfaces. Vol. 31, N°6. 1108-1117.

Wand, Y., Weber, R. (1993). On the ontological expressiveness of information systems analysis and design grammars. Information System Journal Vol 3. N°4. 217-237.