HAL
open science

Annotation of enterprise models for interoperability purposes

Nacer Boudjlida, Hervé Panetto

To cite this version:

Nacer Boudjlida, Hervé Panetto. Annotation of enterprise models for interoperability purposes. IEEE International Workshop on Advanced Information Systems for Enterprises, IWAISE'2008, Apr 2008, Constantine, Algeria. pp.11-17, 10.1109/IWAISE.2008.15 . hal-00232842

HAL Id: hal-00232842

https://hal.science/hal-00232842v1

Submitted on 2 Feb 2008

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.

Annotation of Enterprise Models for Interoperability Purposes

Nacer Boudjlida
LORIA – UMR 7503,
Nancy-Université, CNRS, INRIA
Nacer.Boudjlida@loria.fr

Hervé Panetto
CRAN – UMR 7039
Nancy Université, CNRS
Herve.Panetto@cran.uhp-nancy.fr

Abstract

The purpose of annotations is to describe the content of "something" and they may be considered as meta-data. They are used for a while for text books, articles, hypertext documents and so on. We explore their usage in semantic-based and model-based interoperability, with the aim to make explicit the meaning and the structure of given models (artefacts) to enable not only their understanding, but also their exchange (and their possible transformation) between collaborating actors (human or machine). We propose categories and types of annotations helpful to explicit the deep meaning of models and to ease their exchange.

1. Introduction

The purpose of annotations is to describe the content of "something" (we will call the annotated object) and therefore annotations may be considered as meta-data. They are used for a while for text books [16], articles, hypertext documents as well as in some communities like biologists. They may be provided under different forms, like links, paths, notes in and around a text, comments, highlights of important terms and definitions, numbered steps in a process, etc. They may serve various objectives. For instance, the current work on semantic annotations of Web resources and services is intended to serve for sophisticated Web resources retrieval, discovery and composition, as well as for reasoning [6, 8, 13]. In annotated texts [16], annotations can be the basis for information retrieval, for semantic query evaluation and so on. In our work, we consider annotations for enterprise systems and applications interoperability purposes. We are mainly concerned with enterprise interoperability based on enterprise models [19] which are an abstract representation of some enterprise perspectives. As such, an enterprise model may encompass various types of models (we will call enterprise model perspectives), such as an information model, a process model, a resource model, a decision model, etc. Models being artefacts expressed in a given notation (an enterprise modeling language), it is usually understandable to its only designer. Interoperability assumes, at least, that two "actors" can exchange and operate on an "object". Therefore they need to unambiguously interpret the exchanged object. In our work, "objects" are models or part of models enriched with annotations. We identify a variety of annotations that might be attached to models, together with the variety of their purposes [3, 4].

In addition, we distinguish between human readable annotations and machine readable ones, the latter being called *semantic annotations*. Furthermore, we consider that annotations are produced by an *annotation provider* and their aim is to render an annotated model understandable and/or processable by "someone else" (the human or the machine *annotation consumer*) than the model producer. We assume that any part or any fragment (that we call the *annotation grain*) within an enterprise model perspective may be annotated. We also assume that the *annotation grain* is identifiable and that multiple annotations can be attached to it. The purpose that is pursued in this work being annotation of enterprise models for interoperability [2], a requirement is that the interoperating actors (the *annotation provider* and the *consumer*) agree on the meaning of the variety of annotations that may be attached to an annotation grain (i.e. the semantics of the annotations themselves [1]), as well as on the form under which annotations are actually provided (the *annotation scheme*) i.e. an information structure dedicated to the actual provision of the annotation content.

We also propose an initial typology of annotations (decoration, linking, instance identification, etc.) with their proper meaning. The provided annotations can then serve for (i) improving the models readability, (ii) rendering explicit a model designer's decision and assumptions (iii) enabling the unambiguous exchange of models between collaborating agents (human and/or machines), (iv) Enabling the

transformation of models from one notation to another one, (iv) enabling the traceability of a model thanks to its annotation by the successive transformations that may be applied to some initial model, and ultimately (v) for applying formal techniques to analyze enterprise models with regard to formal properties (like soundness and completeness) and to infer further knowledge from the annotated models. This paper mainly deals with the items 1 to 4.

Another concern of this work is the possible architecture and components of a software platform that supports the annotation of enterprise models. This platform is expected to include, in an integrated manner, facilities or services for the support of enterprise modeling, ontology management and annotations. However, due to space limitations, this topic is not considered in depth here.

The paper is structured as follows. We first define types, and services for annotating enterprise models (section 2). Then, in section 3, we elaborate on what is in an annotation and what are the purposes of annotations for interoperability. Section 3 discusses the required categories of annotations that enable model exchange and transformation. Concluding remarks are in section 5.

2. Annotations: Definition and Purposes

We introduce hereafter a typology of annotations, the link between the annotations and the ontologies together with requirements for annotations to be consistently first provided and second interpreted. Then we briefly discuss which kind of support is required to extend current enterprise modeling tools with annotation facilities.

2.1. Types and Semantics of Annotations

Different types of annotations may be distinguished; these include (i) *Textual annotations* that consist in added notes and comments to the annotated "object", (ii) *Link annotations* that extend the textual annotation notion: the annotation content is reachable through a provided link and (iii) *Semantic annotations:* while textual annotations and link annotations are primarily intended toward humans, semantic annotation content is some meaningful information intended to be human readable as well as machine readable and processable.

Further, annotations may appear as *informal* or *formal*: that means that the annotation expressions may range from natural language to annotations expressed in some sound and well-founded language (like First Order Logic, Description Logic, etc.). It is clear that the more we are close to the later types of language the more the machine-readability of the annotation can be performed. In addition to the annotation definition language, a common understanding of the provided annotations is required. Part of this common understanding may rely on the use of one or several ontologies, i.e. "a representation of a shared conceptualization of a particular domain" [18]. It means that the conceptualization has to be agreed by the authors of the annotation (the *annotation providers*) and by the ones who exploit the annotations (the *annotation consumers*). That also means that, for some types of annotations, the annotation contents are linked to concepts in some ontology. Additionally, in order for the annotation to be interpreted and processed consistently, *annotation consumers* need to understand the meaning of the variety of types of annotations that are provided to them [1]. Indeed, *"... meaningless data cannot acquire meaning by being tagged with meaningless metadata…"* [14]. As an illustration of types of annotations, [1] introduces some possible types, considering a Web resource: (i) *Decoration*: annotations are comments associated with the resource, (ii) *Linking*: annotations are links, (iii) *Instance Identification*: the annotated object is an instance of a given class and the annotation content may be a link to that class, (iii) *Aboutness*: no assertion is made about the existence of an instance of the annotated object, but there is a loose association with it and (iv) *Pertinence*: the target of the annotation may be of interest for the annotated object. The typology of the annotations can further be used by a search engine to classify query answers in accordance with the different types.

2.2. Services for annotations

Another matter concerns the way the annotations are actually provided. Since the content of some types of annotations relies on given ontologies, it seems clear that ontology services (like querying, match-making or browsing an ontology) have to be coupled with annotation services. Moreover, there is a progressive move from manual to automatic or semi-automatic annotation provision. In this framework, some existing annotation platforms use pattern-based and/or machine learning techniques to help in the actual provision of annotations (see [11] for a good survey on this topic and [12] for some freely available annotation platforms). However, to the best of our knowledge, no platform exists for the

annotation of models, except A* (http://leks-pub.iasi.cnr.it/Astar/) which works on a RDF representation of models.

3. Annotation content and purposes

A driving question is *how and for what purpose a model should be annotated.* In an enterprise, models are heterogeneous and are used for many purposes by many actors. In fact, models express enterprise knowledge but this knowledge belongs to the category of what cognitive psychologists have called declarative knowledge. Such explicit and factual knowledge constitutes the core of the enterprise competence. Modelers also formalize expertise, practical knowledge, often called procedural knowledge. While formal knowledge may be explicit, practical knowledge is, rather personal and tacit, being thus difficult to be expressed explicitly. Representing this knowledge in models is not obvious and we then postulate that annotations may help to make it visible and thus, exchangeable when applications must interoperate [13].

The variety of purposes of annotations for interoperability has been listed in the section 1. We elaborate, in this section, on the content of the annotations.

3.1. Sources and targets of annotations

The following links between an annotated object and the content of the annotation are identified to be useful:

- *From the artefact (the model) to a (even informal) definition*: it expresses the natural meaning of the term used for naming the artefact. The meaning may be provided by available ontologies, by the modelling language as well as by some lexicons or terminology, such as http://wordnet.princeton.edu/ or http://wikipedia.org/. .
- *From the artefact to the application domain (and conversely)*: in some situations, it is necessary to link a concept in a model to the part(s) of the application domain that concept represents. This type of link might be, for instance, a reference to a piece of text in the description of the application domain, references to business rules, etc.
- *From the artefact to one or several ontologies and/or to the meta-model of the artefact*: this type of link relates a concept in a model to another one in the ontology or in the meta-model of the modeling language (see section 4).

Another matter of consideration is the *granularity* of an annotated object: it may range from the whole model (*gross grain*) to any of its component like an "entity", a relationship, a pre-condition of a process, etc. (*fine grain*). We assume that (i) any component (grain) of a model is uniquely identifiable, for instance, thanks to a Uniform Resource Identifier (URI) and (ii) conversely, any reference in the target of a link is also identifiable.

Therefore, the actual establishment of the links, between the annotated object and the corresponding concept in the ontology or in the meta-model, gives rise to the following questions that typically fall under the domains of model matching and formal specification of the results of the matching [7]:

1. *How to automatically or semi-automatically identify and locate the right concepts and/or the relationships in the ontology*, starting from the concepts and/or the relationships in the model under annotation? This problem concerns at the same time concepts (usually modelled as classes) as well as relationships between concepts (in the experiments we had, while modelling and annotating, that identification was not made automatically: it has been performed by "visual and manual" browsing of the ontologies and the meta-models).

2. *How to actually express the content of the annotation*, especially when there is not an exact match between a concept in the model and a concept in the ontology (or in the meta-model)? Indeed, the following situations may arise:

a) There is an *exact match* between a concept in the model and a concept in the ontology;

b) A concept in the model is *more specific* or *more general* than a concept in the ontology;

c) A concept in the model can be *part of* a concept found in the ontology (and vice versa);

e) A concept in a model is an *instance of* a concept found in the ontology (and vice versa)

f) A concept in a model *matches* (*exactly, partly*) not a single concept, but possibly more than one concept and relationship (and conversely).

g) *No concept in the ontology is satisfactory,* as compared to a concept in the model;

Excluding the last item in the preceding list, the other items must be part of the content of the annotation as expressions that use the concepts of the ontology or the ones of the meta-model. The last item may require extending the ontology or the meta-model, adopting another accurate ontology.

3.2. Annotation scheme

The preceding considerations, about annotation types, content and so on, conducts to the definition and to the adoption of a common scheme for the provision of the annotations. This scheme includes the following elements:

1. *Identification* of the annotation;
2. *Annotation type* (link, aboutness, any newly identified and defined type, etc.);
3. *Textual* (human readable) description of the annotation content;
4. *Location of the target* of the annotation.
5. *Formal definition of the annotation content*: when applicable, expression of complementary information, like the type of relationship that holds between the annotated object and the target of the annotation (exact/partial match, more/less general, etc.).

For our concern, we defined and used a concrete structured scheme that encompasses attributes corresponding to the above-listed elements. Further, the annotation scheme may be extended with additional elements or attributes. In practice, the annotation scheme may be encoded using XML, or RDF and RDFS, or be SAWSDL-like [20].

4. Model annotations for interoperability

In this section, we discuss the interoperability problem in a heterogeneous context together with the types of annotations that we feel necessary to achieve exchange and transformation of models within that environment, considering a simple definition of the interoperability as being at the same time *connectivity* (i.e. ability to exchange "things" like models, data, business rules, etc.) and *cooperation* (i.e. jointly carry activities to achieve a goal).

4.1. Annotations and heterogeneity

Figure 1 illustrates the variety of heterogeneity that may be encountered between two "actors" that wish to interoperate. Every interoperating actor may use a proper notation for models (M1 and M2) and consequently, a proper meta-model (MM1 and MM2), a proper ontology (O1 and O2), and possibly, a proper annotation scheme (A1, A2). This is clearly the worst situation where, for connectivity and cooperation, mappings and transformations are required everywhere: between annotation schemes, models, meta-models and ontologies. However, as stated before, a reasonable assumption is that the interoperating actors have a common agreement, at least, on the annotation scheme, on the semantics of the variety of the annotations they use, and they share a common ontology. Besides the already defined classification of annotations (into decoration, linkage, instance identification and so on), we recognize at least three complementary annotations that may be associated with a grain within a model (see Figure 1).

4.1.1. Structural annotations: they express what a modeling construct has been used to build the annotated grain; they refer to a given meta-model. This meta-model may be a *specific meta-model*: usually it is the meta-model underlying the notations (or languages) that are used for expressing a model. The meta-model may be a *generic one*: it is a meta-model that subsumes some given meta-models (for example, the Unified Enterprise Modeling Language (UEML) meta-model [5] is a representative of such generic meta-models; it subsumes some of existing modeling methods). When a *specific meta-model* is used, the structural annotations are usually "naturally" inherited from the modeling constructs that are offered by a modeling method. For example, using the *Process* concept in an enterprise modeling method, an artefact named *Proc_P_01* is added to a model: *Proc_P_01* is implicitly an instance of *Process* in the meaning of the method that is used. When a *generic meta-model* is used in a structural annotation, the interpretation of the annotation by an interoperating actor *Act1*, requires from *Act1* either the knowledge of the *generic meta-model*, or the availability of mappings between the *generic meta-model* and the *Act1's* specific meta-model (this mapping is represented as dotted lines in Figure 1).

Example: Consider an annotated grain X modeled as a *Decision Centre* in GRAI and assume that X has to be sent to a collaborating actor who uses *MEGA modeling concepts* (MEGA Process, Mega International, http://www.mega.com.). Assume also that UEML 1.0 [5] is used as a generic meta-model. An existing mapping states that a GRAI Decision Centre is *more specific* than a UEML Activity (i.e. the UEML Activity concept *subsumes* the GRAI Decision Centre one: GRAI-Decision Centre $\subset$ UEML-Activity) and that a MEGA Activity is *more general* than a UEML Activity (i.e. the UEML Activity concept *is subsumed by* the MEGA Activity one: UEML-Activity $\subset$ MEGA-Activity). Therefore, at the model level, the grain X can be unambiguously understood, as a MEGA Activity without any knowledge about the GRAI concepts.

4.1.2. Lexical/Terminological annotations: they express, generally at a "*surface semantic level*", what the annotated grain represents in the application domain; these annotations concern the names that are associated with the constructed artefacts (example: *Proc_P_01*) and they usually refer to a commonly

agreed definition of terms. The definition may simply be part of a thesaurus or a taxonomy. But, preferably, the definition of terms may come from one or several ontologies, and therefore the annotation content will refer to that ontology(ies). Back to the example: assume that, in the modeller mind, as an intended meaning, *Proc_P_01* is a planning activity. The lexical annotation may explicit this meaning thanks to a link between *Proc_P_01* and the concept of *"Plan Supply Chain"* in SCOR, the Supply-Chain Operations Reference-model [10] used here as an ontology.

In addition, ideally, interoperation should respect some principles like:

a) The autonomy principle: the consequence of this principle is that an interoperating actor has not to change its way of working, the notations or the vocabulary that the actor is familiar with, the possible naming rules, etc. Lexical notations contribute to the satisfaction of this principle.

b) The privacy principle: according to this principle, even in a cooperative process, a partner may hide all or part of its business rules and processes from the partner he/she collaborates with: behaviour annotations are especially concerned with this principle and they may be hidden from the annotation consumer.

4.1.3. Behaviour annotations: they are an *in-depth expression* of the role played by the annotated grain. They can be viewed as the explicit definition of the business logic, the procedures, the rules and the policies that govern the annotated object. The purpose of such a type of annotation is, for example, to ensure that two interoperating processes have the same behaviour (then one can be used instead of the other), or to enforce a defined behaviour from a cooperating process that is sub-contracted, out-sourced or delegated to a collaborating partner. This type of annotation may be provided under various forms (textual descriptions like references to regulations and policies that govern the behaviour of the artefact, enumeration of steps within a process, etc.) and at various levels of details. When this type of annotations is used for exchanging models between interoperating partners, the level of their details depends on the existing privacy constraints.

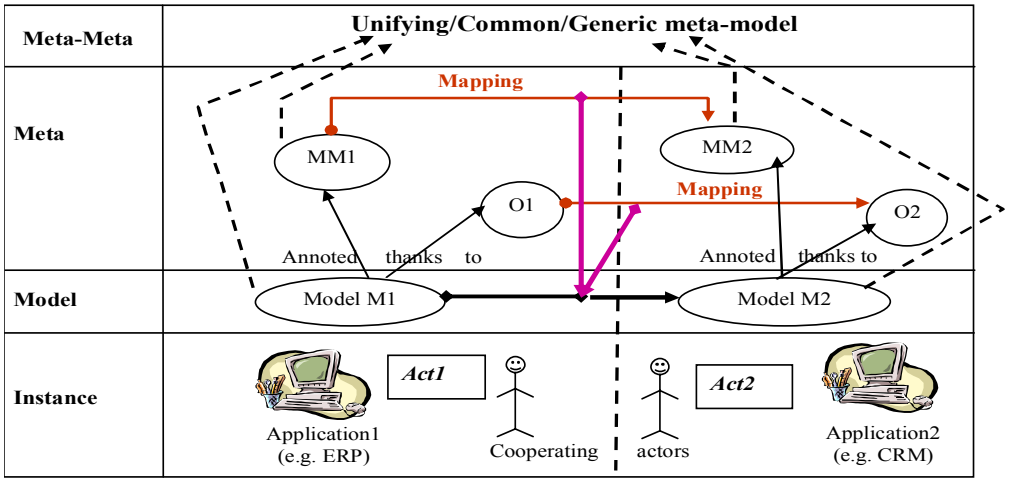

Figure 1. Model exchange and annotation scenario

For example, the work reported in [10] is a first step toward this aim: it is based on a general process ontology and it uses a goal-oriented approach thanks to the annotation of processes or sub-processes with their intended goals (however, the level of details of the annotations do not indicate how the goals can be attained).

Back to the *Proc_P_01* example: when receiving the annotated object, an interoperating partner knows its purpose (*planning activity*) thanks to the lexical annotation and, at the enactment level, the behaviour annotations will govern the actual planning activity. However, if the ontology that is used for expressing the lexical annotations encompasses a commonly agreed or a "standard" behaviour, this type of annotation is redundant and therefore useless.

4.2. Annotations for Model Exchange

We feel that the preceding types of complementary annotations help the annotation purposes that are concerned in this work: annotation for model exchange, for model transformation and for

model traceability. In addition, these purposes are intimately related, since an exchanged model that is notated in a notation *N1* may be mapped/transformed into a model expressed in a notation *N2*. Further, a *trace* of the model history might be preserved thanks to the expression, as annotations, of the transformation that is applied to the model. Let us elaborate more on model exchange and transformation.

For connectivity purposes, interoperating actors obviously need to exchange a variety of "things". In our context, they are supposed to exchange models or model fragments. In Figure 1, assume that the model *M1* is to be sent by the actor *Act1* to the cooperating actor *Act2*. In a *homogeneous context*, *Act1* and *Act2* use the same notations for their models as well as the same ontology (in figure 1, this means that *MM1* and *O1* are same as *MM2* and *O2*, respectively or, equivalently, that mappings are identity isomorphism). Therefore, fitting *M1* with the only *lexical annotations* (and possibly with *behavioural annotations*) is sufficient for *Act2* to interpret the received model.

In a heterogeneous context, the cooperating actors may use different notations for their models (for example, *M1* is notated according to the meta-model *MM1* and *M2* according to the meta-model *MM2*) and they may refer to different ontologies (for example, *Act1* uses an ontology *O1*, while *Act2* uses an ontology *O2*, different from *O1*). According to the *autonomy principle*, *Act2* is not required to have any knowledge about the notations used by *Act1*. So, what are the types of annotations that have to be attached to a model or a fragment of a model *M1* to enable Act2 to unambiguously understand that model or that fragment of model (we will below use the term *model* as a synonym for *model or fragment of a model)*.? Consider the variety of situations that can occur (Table1):

1) When *Act1* and *Act2* refer to different ontologies: a mapping must exist (or be discovered) between both the ontologies, and lexical annotations *must* be attached to *M1*.

2) When *Act1* and *Act2* use different notations to express their models, and therefore different meta-models then, in addition to a mapping between the meta-models *MM1* and *MM2* that must exist (or be discovered), (a) Structural annotations *must* be attached to *M1* to enable the possible transformation into *M2*, (b) Lexical annotations must be attached also to M1 to enable Act2 to consistently understand the meaning of M1 as intended by Act1, with reference to the ontology used in the annotation, (c) Behaviour annotations are *not mandatory*; if this type of annotation is also attached to M1, it will give *Act2* a deeper understanding of what *M1* is expected to perform.

3) Similarly, when *Act1* and *Act2* have an agreement about a common *generic meta-model*,

a) Mappings between the specific meta-models and the generic one are inherently available;

b) Structural annotations, referring to the generic meta-model, *must* be provided;

c) Lexical annotations *must* also be;

d) Behaviour annotations are *not mandatory*.

	Ontology	**Specific Meta-model**	**Generic Meta-Model**	**Types of required annotations**
***Act1* and *Act2* use**	Same one	Same one		- Lexical annotations (mandatory); - Behaviour ones are optional
Act1	O1	-	-	- Mapping between O1 and O2 must exist or be discovered. - Lexical annotations are mandatory; - Behaviour ones are optional
Act2	O2	-	-	
Act1	-	MM1	None	- Structural and Lexical annotations are mandatory; - Behaviour ones are optional
Act2	-	MM2	None	
Act1	-	MM1	GMM	- Mapping between every specific meta-model (MM1, MM2) and the generic one (GMM); - Structural annotations must exist, with reference to the common GMM; - Lexical annotations are mandatory; - Behaviour ones are optional.
Act2	-	MM2	GMM	

Table 1. Variety of Cases and Required Annotations and Mappings

5. Concluding remarks

As concluding remarks, we stress the facts that annotations have a "syntax", i.e. the form under which the annotation is provided (the annotation scheme), which is understandable by the interoperable actors. The annotation "syntax" must be extendible to cover the needs of the variety of purposes, i.e. new properties may be added to the annotation scheme.

Annotations have a "semantics", i.e. a variety of annotations can be attached to an annotated object and every annotation in that variety has a proper meaning. That meaning must be shared by the interoperable actors. Annotations may serve for different purposes and the variety of annotation types depends from their purpose (their intended usage). Finally, the provision of a support for annotating models is technically feasible: fragments of technology exist today and they mainly need to be made interoperable. And the availability of such a support is clearly an important factor for the scalability of the approach we propose for semantic and model-based interoperability.

However, an important feature is missing: it is the one that permits the automatic or the semi-automatic provision of the annotations. Our current activity deals with the integration of that variety of services to support model annotations together with the exploration of paths toward the support for the annotation provision.

6. References

[1] S. Bechhofer, L. Carr, C.A. Goble, S. Kampa and T. Miles-Board. The Semantics of Semantic Annotation. *In proceedings of CoopIS/DOA/ODBASE*, p. 1152-1167, Irvine, California, USA, October 30 – November 1, 2002, LNCS# 2519.

[2] N. Boudjlida. *"Semantic Annotations for Interoperability"*. Invited talk at the International Research School "Ontologies, a Smart Way Toward Interoperability", Paris, France, IAE, 13 April 2006.

[3] N. Boudjlida & al. A practical experiment on semantic enrichment of enterprise models in a homogeneous environment. *INTEROP Deliverable DTG4.1*. June 2006.

[4] C. Diamantini and N. Boudjlida. *About Semantic Enrichment of Strategic Data Models as Part of Enterprise Models*. In Proceedings of the 4th Business Process Modelling Workshops, p. 348-359, Vienna, Austria, Sept. 2006, LNCS# 4103, Springer-Verlag, June 2006.

[5] G. Berio, K. Benali, N. Boudjlida and M. Petit. *A Unified Enterprise Modelling Language for Enhanced Interoperability of Enterprise Models*. In *Proceedings of the 11th IFAC Symposium on Information Control Problems in Manufacturing,* Salvador, Brazil, April 2004.

[6] S. Hamri, N. Boudjlida and M. Boufaïda. *An approach for Building an OWL Ontology for Workflow Interoperability*. Proceedings of the 3rd Intern'l Conf. on Interoperability of Enterprise Software and Applications I-ESA'07, Madeira, Portugal, March 2007, p. 357-364. Springer-Verlag, ISBN 978-1-84628-857-9.

[7] M. Huza, M. Harzallah and F. Trichet. OntoMas: a Tutoring System Dedicated to Ontology Matching. *Proceedings* of I-ESA'07, Madeira, Portugal, March 2007, p. 377-388. Springer-Verlag, ISBN 978-1-84628-857-9.

[8] A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic Annotation, Indexing, and Retrieval. *Elsevier's Journal of Web Semantics*, Vol. 2(1), 2005.

[10] Yun Lin, Arne Sølvberg. *Goal Annotation of Process Models for Semantic Enrichment of Process Knowledge*. In proceedings of the 19th International CAiSE'2007, Springer-Verlag, Trondheim, Norway, June 2007.

[11] L. Reeve and H. Han. Survey of Semantic Annotation Platforms. *In Proceddings of the ACM Symposium on Applied Computing,* Santa-Fe, USA, March 13-17, 2005.

[12] Semantic Web: Annotation and Authoring Tools. http://annotation.semanticweb.org/tools/

[13] L. E. Whitman and H. Panetto H. The missing link: culture and language barriers to interoperability. *IFAC Annual Reviews in Control*. 30/2, 233-241, Elsevier, December 2006, ISSN: 1367-5788

[14] K. Sivashanmugam, K. Verma A. Sheth and J. Miller, Adding Semantic to Web Services Standards, *1rst Intern'l Conference on Web Services,* Las Vegas, June 2003, p. 395-401

[15] John F. Sowa. *Ontology, Metadata, and Semiotics*. Presented at ICCS'2000 in Darmstadt, Germany, on August 14, 2000. LNAI#1867, Springer-Verlag, Berlin, 2000, pp. 55-81. www.jfsowa.com/ontology/ontometa.htm#sec1

[16] Supply-Chain Operations Reference-model Version 7, The Supply-Chain Council, 2005, http://www.supply-chain.org.

[17] Y. Toussaint. *Semantic Annotation of Texts*. Invited talk at the 2nd Intern'l workshop on Enterprises and Networked Enterprises Interoperability, Vienna,Sept. 2006. http://www.loria.fr/~nacer/ENEI06/10-Invited-Talk.pdf.

[18] M. Uschold and M. Gruninger. *Ontologies: Principles Methods and Applications*. Knowledge Engineering Review, 11(2), 1996.

[19] F. Vernadat. *Enterprise Modeling: Objectives, Constructs & Ontologies,* Tutorial at the 1st EMOI workshop at CAiSE 2004, Riga, Latvia, June 2004.

[20] W3C. Semantic annotations for WSDL and XML schema. August 2007, www.w3.org/TR/sawsdl.