

Enterprise modelling and Ontology

N. Zouggar, D. Chen, B. Vallespir

IMS - LAPS/GRAI, Université de Bordeaux, CNRS

351, cours de la Libération 33405 Talence cedex, France

nabila.zouggar@laps.ims-bordeaux.fr

Abstract: Generally speaking, Enterprise model is used during the system engineering life cycle by other stakeholders rather than those who developed it. They do not necessarily know the context in which the model was built and quite often are not familiar with the language used for the modelling. This situation makes that the model loses its semantic during its exploitation and creates ambiguities and difficulties in its use.

We will propose in this paper a methodological approach to be followed in the creation of models which would make it possible to keep their semantics during their life cycle by the use of ontologies and semantic annotations. A simplified application example is presented to illustrate the use of the proposed approach.

1. INTRODUCTION

Enterprise modelling aims at representing the whole or part of an enterprise in the form of model which can be informal, semi formal or formal. This model is to be understood, used and re-used by different persons with different knowledge backgrounds. For that the semantic of the concepts used must be clear and without ambiguity. This paper presents the latest development of semantic enrichment of enterprise modelling using ontologies. First the relationship between the enterprise modelling and ontologies is discussed and possible redundancies and complementarities identified. Then the use of ontologies to enrich enterprise modelling languages and enterprise models is presented. We will also show the use of the semantic annotations based on ontology to trace the changes of an enterprise model during its life cycle. Finally an example of semantic annotation and traceability on GRAI decisional model will be presented. Future work and perspectives are discussed as part of conclusion.

2. ENTERPRISE MODELLING AND ONTOLOGY

Before giving the definition of enterprise modelling and ontologies, we will present the release element of this research which concerns the semantic of the models during their life cycle. The purpose is to give an automatic way to preserve and transmit the exact content and meaning wanted by the developer of a model.

Let us take the example of a model which is developed by the user U, which will be exploited by the user U1, and which will be compared with another model by the user U2, this model will lose part of its semantics v while advancing in time (Fig.1). V' and V'' represent the semantic loss of the model during its use and the goal of our work is to tend V' and V'' towards 0:

$$\{V', V''\} \approx 0$$

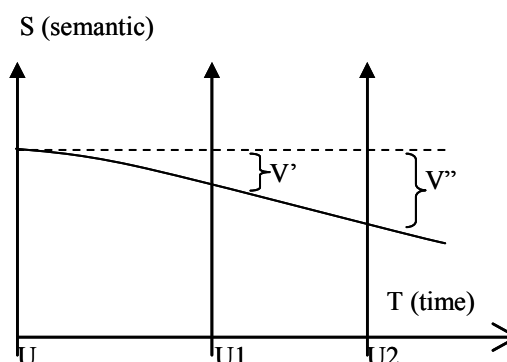


Fig. 1. Semantic evolution during model life cycle

2.1 Definition

Enterprise modelling aims to construct a model of whole or part of the enterprise, and generally of any organization, considered as a system, to explain the structure and the organization or to analyze their behaviour. The model must also be able to represent the particular point of view of an actor.

Several languages of enterprise modelling allow the construction and the exploitation of model (life cycle) according to steps which are often characterized by a level of abstraction (conceptual, organisational or technical).

Formalization degree of the models varies according to the languages used, it can be informal (such as natural language), semi-formal (such as language with graphic formalism) or formal (mathematical language). Most of time, the models

based on informal language are used to describe an existing situation while the models based on a formal language allow the checking of the properties fixed in a given project (Chapurlat et al., 1999). Concerning ontologies, several definitions exist. A commonly agreed definition of an ontology is: 'An ontology is an explicit and formal specification of a conceptualisation of a domain of interest' (Gruber, 1991). This definition stresses two key points: the conceptualisation is formal and hence permits reasoning by computer and a practical ontology is designed for some particular domain of interest.

Ontologies consist of concepts (also known as classes), relations (properties), instances and axioms. A more succinct definition of an ontology is as a 4-tuple $\langle C, R, I, A \rangle$, where C is a set of concepts, R a set of relations, I a set of instances and A a set of axioms (Davies et al., 2006).

The languages used for the construction of ontology may be classed as the enterprise models are: informal (understandable for the user but difficult to check the absence of redundancy or contradiction); semi formal (increased clarity and reduced ambiguity); formal (possibility to check redundancy and consistency) (Ushold et al., 1996).

The choice of the formalization degree is done according to the use of ontology. Indeed, if the aim is the support of communication between people, then the representation of ontology may be informal since it is precise enough to capture the semantic of each one. If now ontology must be must by software tools then the semantics must be formal.

2.2 Complementarities and mapping

The definitions given previously enable us to say that there is a bond between enterprise modelling and ontologies. Initially we can affirm that both have as purpose to take part in the modelling of the enterprise. More particularly, research in ontology in enterprise domain mainly focuses on enterprise concepts identification and description; while research of enterprise modelling also deals with the concept definition (for example conceptual model of GRAI) but focuses on modelling language and model construction using the language. Thus we can tentatively say that a possible overlapping is the concepts identification (Fig.2).

However a deep analysis allow finding that the conceptual models developed in the enterprise modelling research are mainly informal ones and do not allow to capture precisely the semantics of the concepts. In the contrary, ontology technique used to describe enterprise concepts is more formal and thus allows better defining the semantics.

The difference stands also in the contents of the models. The enterprise model represents the structure or the operation of the enterprise whereas ontology organizes only the concepts used and the relations between them. In other words, ontologies in the domain of enterprise modelling such as TOVE (Gruninger et al., 2000) can be considered as enterprise ontology rather than enterprise model in the sense that there is no associate modelling language in ontology to allow building an enterprise model. Ontology technique is

useful to elaborate enterprise metamodel rather than developing enterprise modelling techniques and models. Thus the two approaches are complementary.

This difference can be beneficial for enterprise modelling; indeed the use of ontologies can mitigate the semantic deficit of the languages and the models that are primarily presented under their syntactic component. This situation is particularly highlighted when we seeks to exchange a model or to federate distinct modelling languages. The users are often in view of problem of comprehension due to the fact that the analysis suggested is based primarily on a syntactic analysis of the components; the semantics of the latter being not very explicit.

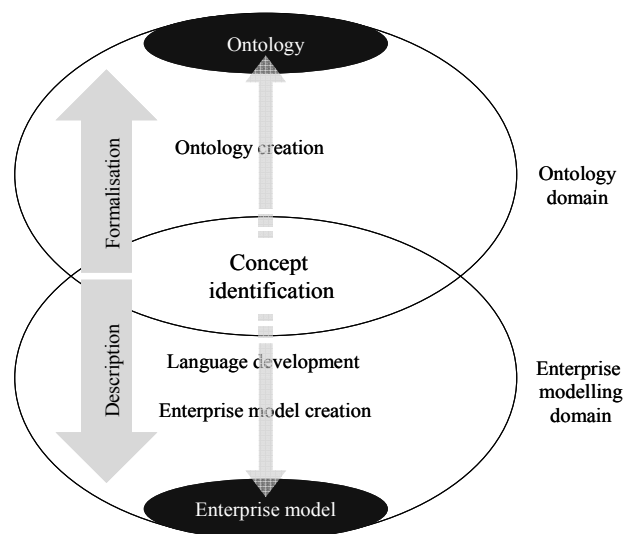


Fig. 2. Common element of enterprise modelling and ontologies.

We can distinguish two cases that induce the users in error: the existence of several terms for only one definition (synonyms) and the existence of a term corresponding to several concepts (polysemy). The use of an ontology ensures to avoid ambiguities by choosing for the first case a term that will be used and that will refer to the existing definition. For the second case we can manage ambiguity for instance by attentively define each concept indicated by the term by using only terms whose definition is consensual; it can be useful to give to these concepts labels without significance x_1, x_2 , etc. so as to be able to refer to it in a neutral way.

Ontologies can also be used within the framework of the model life cycle. An enterprise model evolving in time, it is necessary to trace its changes carried out to facilitate its understanding, its use and its re-use since all the history of the model can be consulted.

These two uses of ontologies for the enterprise modelling will be developed in the following part.

3. SEMANTIC ENRICHMENT OF ENTERPRISE MODELLING

In this section we present how semantic enrichment of enterprise modelling can be possible using ontology.

We propose to define semantic enrichment as being the automatic maintenance of the initial semantics of a model. As we explained previously (Fig. 1) we want to make tend v' and v'' towards 0. This operation can be possible by using the formal aspect of ontology. Indeed if we formalize the concepts used in modelling in the form of an ontology, the latter will be related to the model by semantic annotations, which will allow this pair {model, ontology} to evolve together in time, and thus to keep the initial semantics of the model through this ontology.

The figure 3 shows how to go from the enterprise modelling that currently exists (State I) which consists to model an enterprise using a language. Semantics is not easily transmissible in this case, whereas in the second (State II) we can formalize this semantics in the form of an ontology.

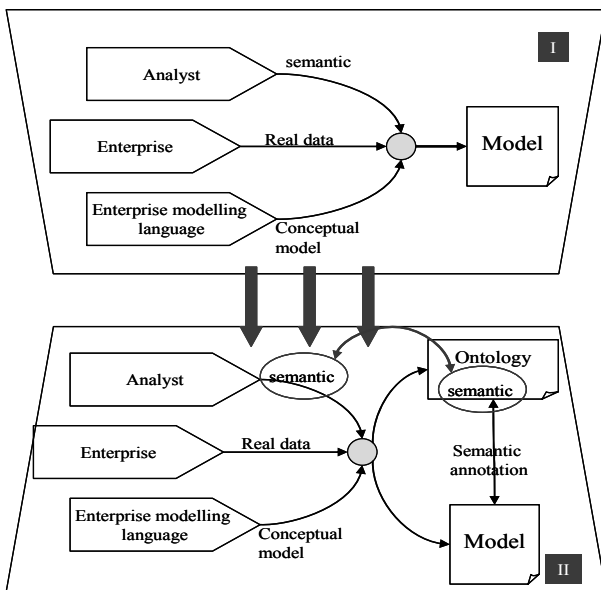


Fig. 3. Ontology as an answer to semantic formalization

There are two possibilities of enrichment, the first being to enrich only the model and the second one being to enrich the language and then the model. In both cases the step to be followed is to build ontology corresponding to the request, and then to use this ontology within the modelling process by semantic annotations.

3.1 Ontology creation

The steps to follow to create ontology are (Fridman Noy et al., 2001):

- Determine the field and scope of the ontology. Determine the degree of formalisation needed, choose an appropriate ontology language
- Study the possibility of using existing ontologies, to extend and refine them. Reuse existing ontologies can even constitute a requirement if our system needs to interact with other applications which already use specific ontologies or controlled vocabularies.

- Enumerate the significant terms in ontology. Indeed, it is useful to note in a list form all the terms to be treated or explained to a user, and the properties related to these terms.
- Define the classes and their hierarchy.
- Define the class properties (attributes) and their facets (values types, authorized values, number of values, etc).
- Create class instances in the hierarchy.

3.2 Semantic Annotation

The annotation is one of the most common forms of meta-data in the Web context, it is also graphic or textual information attached to a document and generally placed in this document.

The semantic annotation is a particular case of annotation because it refers to ontology. It can be made in the form of comments, of explanations note, questions or another type of external remark which can be attached to a document or a selected part of this document.

To perform an annotation it is necessary to proceed through the three following phases which are (Desmontils et al., 2002), (Hung, 2003):

- the location which consists in placing in the document the ontology concepts references that it contains. These elements are considered as meta-data,
- the instantiation which allows to give attributes values of the concepts using information present in the document,
- the enrichment which aims at adding information by the intermediary of concepts attributes which could not be given values in the preceding phase.

We note that in the first two steps, there are not information addition but rather localization and characterization of information already present. They are insertion steps. At the last step, the document is enriched by information which did not exist; it is a step of annotation formalized by meta-data.

The structure of a semantic annotation can be represented as follows (Boudjlida et al., 2006):

<Annotation

Annotation ID: identifier of the annotation

Type = indicates the "semantics of the annotation":

Unformal Content = textual description of the annotation

Ref2Ontology = reference (uri) to the ontology concept or part of it related to the current model concept

Constraints = may be written using OCL, with references to the ontology, if this one is represented by a UML class diagram

/>

The annotation type are:

- Decoration: annotations are comments associated with the resource;
- Link: annotations are links;
- Instance Identification: the annotated object (U#X) is an instance of a given class and the annotation content (Ref2Ontology) may be a link to that class (uri);
- Aboutness: no assertion is made about the existence of an instance of the concept, but there is a loose association with the concept;
- Pertinence: the target of the annotation may be of interest for the annotated object;
- Textual (human readable) description of the annotation content;
- Identification/location of the target of the annotation (example: link to an ontology): formal definition of the annotation content. The value of this formal definition depends on the types of the annotation. This part of the annotation scheme is intended to be machine readable and interpretable.

4. ENTERPRISE MODEL TRACEABILITY

Another contribution of ontology technique to enterprise modelling is to develop enterprise model traceability (Sarraipa et al., 2007).

Model traceability is a characteristic in which the model is clearly linked to their source and to the outputs created during the model life cycle (Balasubramaniam 1997). This life cycle is composed of 5 steps: problem definition, model formulation, model solution, model interpretation and model maintenance. When going from one step to another, the model changes according to an operation O, as model transformation, model evolution/versioning.

Other operations can be defined and used on model to map, translate or check information. In all these cases we have a starting model M1 and the resulting model M2. Between these two models it will have new information or a loss of information according to the operation carried out.

Model traceability is represented by a relationship between model entities that have changed as consequence of a specific model operation in a backward and forward direction. The traceability will make possible to follow a model evolution (Sarraipa et al., 2007).

A possible scenario (Fig. 4) is when a model experienced some operations in its life-cycle (e.g. Model Operation N; Model Operation N+1).

The “Entity δ” is the result of “Entity α” in Model N-1, applied to a Model Operation N-1 (Entity δ= Model Operation N (Entity α)). As consequence “Entity x” is the result of “Entity δ” in Model N, applied to a Model Operation N (Entity x = Model Operation N (Entity δ)). Entity δ has one Model Traceability Instance represented in the Model Traceability Ontology. As a result, the Entity δ will have one

annotation in the Model N that will have a link (Instance Identification) to the mentioned Instance in the Model Traceability Ontology.

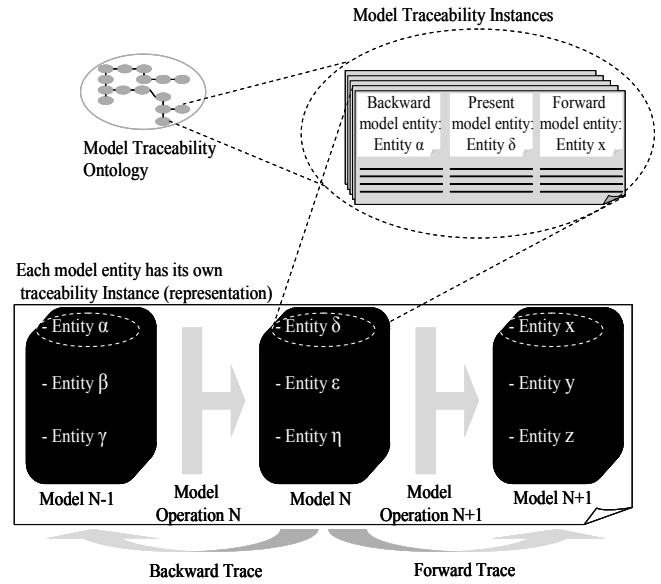


Fig. 4. Model Traceability Scenario

Model traceability Ontology

The Model Traceability Ontology aims at representing the information that should be gathered for use in the model traceability process. Figure 5 shows its structure (Sarraipa et al., 2007).

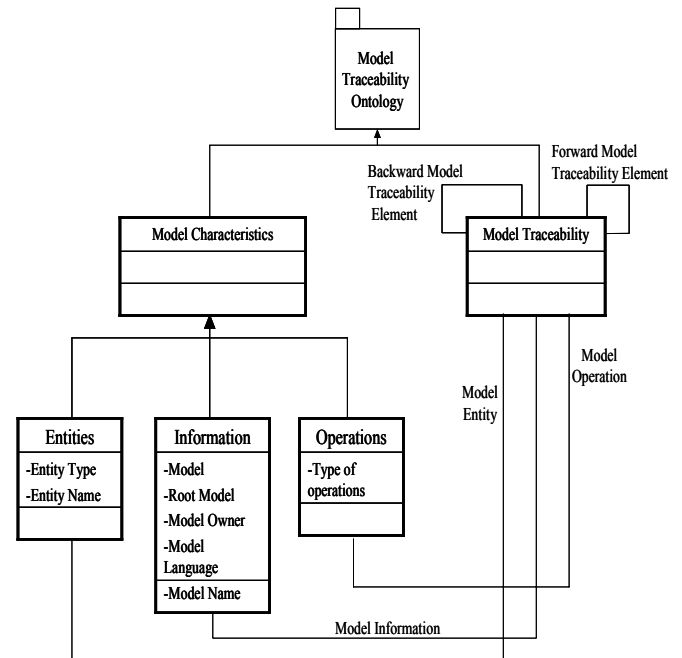


Fig. 5. Model Traceability Ontology structure

The ontology represents two classes: Model Characteristics and Model Traceability.

The Model Characteristics class represents the generic information of a specific model (Class: Information); the entities that can be part of the model (Class: Entities) and the operations that a model can suffer in its life cycle (Class: operations).

The Entities class represents model elements such as concept; relation attribute or constraint.

The Operations class is related to the operations that are applied between each stage of a model life cycle.

The Information class allows keeping the relevant information for the model traceability such as model name, model owner, model language or model objective/purpose. The property root model was defined in order to specify if a model is the first one in the life-cycle (meta-model).

5. EXAMPLE OF APPLICATION

To illustrate the proposed methodology, an application to the case study “TelCo Company” is outlined. Firstly, the decisional GRAI grid is presented, and then model transformations have been carried out using the ‘dysfunctions detecting rules’ of GRAI method. Lastly, the model traceability instance and the annotation used for the model traceability are presented. TelCo is a telecommunication retailer. The company operates through a network of owned and franchised shops throughout the country, replicating the business model that has been established as the undisputed leader of technology retailers in Greece and in many other countries of Eastern Europe and Central Asia. Cooperating closely with the producers, TelCo also collaborates with operators of cellular telephony networks for its cell-phone operation. “Singular Enterprise” (SEn) software was implemented in the company’s headquarters to control the daily workflow of the company, including all financial activities, the movement of products and services to and from its suppliers and customers, as well as the daily retail sales.

The decisional GRAI grid is composed of several modelling constructs: decisional level, decision centre, decision frame, function and information (Vallespir et al., 2007):

- Decisional level: it is linked to the time horizon and period of the decision, i.e. strategic, tactic and operational.
- Decision centre: it is conceptually defined as the intersection between a function and a decision level.
- Function: it is the grouping of decision activities concurrent to a common goal.
- Decision frame: it is composed by objective, decision variables, constraints and criteria.
- Information: information needed to make a decision.

The transformations to be carried out in the grid will be made according to the ‘dysfunction detecting rule’ of GRAI method. For this case it is concerned with the decision centre (IT/20) and the horizon of the level of decision (50), where two dysfunctions are detected:

- Each elementary function of control must have a decision centre on each hierarchical level.
- The horizon must be longer than the duration of the physical activities of production controlled by the decision centres of the level.

For model traceability, initially the “model traceability instance” is identified. In the presented case there are two of them (Fig.6), the others will be determined according to the transformations on GRAI grid. The first entity which was transformed is (IT/20), the second is (50).

Backward model entity: - IT 20	Present model entity: - IT 20	Forward model entity: - IT 20
∅	To plan Investments in IT system	∅
Backward model entity: - 50	Present model entity: - 50	Forward model entity: - 50
∅	H = 1 week	∅

Fig. 6. Model traceability instances

Annotations are used to trace the changes of these entities. For that, it is proposed that the Edinburgh Enterprise Ontology server is the host for the model traceability ontology and the uri related to that ontology is <http://www.aiai.ed.ac.uk>.

The annotation refers to the model traceability ontology, where the model traceability instances are formalized. These annotations give access to all the transformations carried out on the model and thus a traceability of the model. For instance, the (IT/20) annotation would have as value for the “Annotation Type = Model_traceability_1” which is its Instance identification. Thus, it should be possible to access to the entire information related to this specific annotation traceability, for instance as it is illustrated in the text box in appendix A.

6. CONCLUSION

Ontologies have an obvious interest in the development of the enterprise modelling and that while bringing a degree of formalization that is missed in enterprise model. That is initially possible by using the semantic annotations that bind the concepts used in a language or model with an ontology that already exists or that should be built. The two approaches possible of semantic enrichment which exists, namely model enrichment and language enrichment are to be exploited to found the best approach on technical and financial point of view. Also, ontologies allow to follow-up a model, therefore a traceability that makes possible to store any change that has occurred during its life cycle.

These two ways to exploit ontologies in enterprise modelling allow a better use, comprehension and exchange of the models consequently allows the interoperability of these models.

It is quite obvious that this research topic is only at its beginning, first results should be soon assured and that is thanks to work and collaboration that were carried out within the framework of INTEROP Network of Excellence (Interoperability Research for Networked Enterprises Applications and Software, n° 508011) and ATHENA integrated project (Advanced Technologies for interoperability of Heterogeneous Enterprise Networks and their Applications, n° 507849).

REFERENCES

- Balasubramaniam R. (1997). Representing and reasoning with traceability in model life cycle management. *Annals of Operations Research*, Volume 75, Number 0, pp. 123-145.
- Boudjlida N and al. (2006). A practical experiment on semantic enrichment of enterprise models in a homogeneous environment. INTEROP Deliverable DTG4.1
- Chapurlat V., Larnac M., Lamine E. and Magnier J. (1999). Definition of a formal analysis framework for existing enterprise modelling approaches. Proceeding of ICIMS-NOE conference, Life cycle approaches to production systems: management, control, supervision (ASI), Louvain, Belgique.
- Davies J., Studer R. And Warren P. (2006). *Semantic web technologies. Trends and research in ontology-based systems.* Wiley edition. England.
- Desmontils E., Jacquin C. (2002). Annotation sur le web: notes de lecture. Journées de l'AS Web sémantique. <http://www.lalic.paris4.sorbonne.fr/stic/>.
- Fridman Noy N., McGuinness D-L. (2001). *Ontology development 101: A guide to creating your first ontology.* Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.
- Gruber T, 2001. What is an ontology? Summary statement of Gruber's definition of ontology. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>.
- Gruninger M., Atefi K., and Fox M.S. (2000). Ontologies to support process integration in enterprise engineering. *Computational and Mathematical Organization Theory*, Volume 6, Number 4, pp. 381-394.
- Hung H-H. (2003). Développement d'un outil efficace pour annoter les documents. PhD thesis. Institut de la francophonie pour l'informatique. Vietnam.
- Sarraipa J., Zouggar N., Chen D and Jardim-Goncalves R. (2007). Annotation for enterprise information management traceability. Proceeding of ASME 2007 International Design Engineering Technical Conferences

& Computers and Information in Engineering Conference. Las Vegas, Nevada, USA.

Uschold M., Gruninger M. (1996). *Ontologies: principles, methods and applications.* The Knowledge Engineering Review. Volume.11, Number.2, pp. 93-155

Vallespir.B and Doumeingts.G. (2007). *The GRAI Method. Part 1: Global Modelling.* INTEROP tutorial.

Appendix A. IT20 INSTANCE DATA EXAMPLE

```
<Model_Traceability rdf:ID="Model_Traceability_1">
  <model_information>
    <Information rdf:ID="Singular_Enterprise">
      <Model_owner
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Telco Company</Model_owner>
      <Model_name
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Singular Enterprise</Model_name>
      <Root_model
rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >true</Root_model>
      <Model_language
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >GRAI</Model_language>
      <Model_objective
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >â€œSingular Enterpriseâ€œ (SEn) software was
implemented in the companyâ€™s headquarters to control the
daily workflow of the company, including all financial
activities, the movement of goods to and from its suppliers and
customers as well as the daily retail sales.</Model_objective>
      <Model_version
rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
      >1.0</Model_version>
    </Information>
  </model_information>
  <model_entity rdf:resource="#Entities_2"/>
  <forward_model_traceability_element>
    <Model_Traceability rdf:ID="Model_Traceability_7">
      <model_entity rdf:resource="#Entities_2"/>
      <model_information
rdf:resource="#Singular_Enterprise"/>
      <backward_model_traceability_element
rdf:resource="#Model_Traceability_1"/>
      <model_operation>
        <Operations rdf:ID="Operations_5">
          <Type_of_operations
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >transformation</Type_of_operations>
        </Operations></model_operation>
      </Model_Traceability>
    </forward_model_traceability_element>
    <model_operation rdf:resource="#Operations_5"/>
  </Model_Traceability>
</Entities rdf:ID="Entities_2">
  <Entity_type
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >concept</Entity_type>
  <Entity_name
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >IT/20</Entity_name>
</Entities>
```