

Semantic Commitment for Designing Ontologies: A Proposal

Bruno Bachimont¹, Antoine Isaac^{1,2}, and Raphaël Troncy^{1,3}

¹ Institut National de l'Audiovisuel, Direction de la Recherche
4, Av. de l'Europe - 94366 Bry-sur-Marne
{bbachimont,aisaac,rtroncy}@ina.fr & <http://www.ina.fr/>

² LaLICC, Université de Paris-Sorbonne,
<http://www.lalicc.paris4.sorbonne.fr>

³ INRIA Rhône-Alpes, Action EXMO,
<http://www.inrialpes.fr/exmo>

Abstract. The French institute INA is interested in ontologies in order to describe the content of audiovisual documents. Methodologies and tools for building such objects exist, but few propose complete guidelines to help the user to organize the key components of ontologies: subsumption hierarchies. This article proposes to use a methodology introducing a clear semantic commitment to normalize the meaning of the concepts. We have implemented this methodology in an editor, DOE, complementary to other existing tools, and used it to develop several ontologies.

1 Introduction

With the emergence of technical systems which exploit numerical contents, accessing and processing information are evolving at a fair rate. The French institute INA¹ has to manage large multimedia and audio-visual databases, a task that includes allowing an access as efficient as possible to the data stored. INA is thus greatly concerned with indexing – the core of its mission –, which implies dealing with ontologies to create relevant content description of the audio-visual documents.

While trying to use ontologies, one soon has to face the problem of the way they are designed, especially as regards to taxonomy structuration. Indeed, it is acknowledged that the taxonomies of domain concepts are key components of the built ontologies. Consequently, we searched for a methodological approach that gives guidelines to structure taxonomies (Section 2). We claim that none of these methodologies force the ontologist to explicit the real meanings of the concepts and consider thereafter a possible solution, using natural language. We detail the three steps of a methodology proposal (Section 3) and present a tool implementing it. We then conclude and outline further work (Section 4).

¹ INA (*Institut National de l'Audiovisuel*) has been archiving TV documents for 45 years and radio documents for 60 years. It stores more than 700 000 hours of broadcast programs (3 000 000 audio-visual documents) and some 2 000 000 images.

2 Which Methodology for Building Ontologies?

2.1 A Work Still in Progress

Many approaches (for a complete survey, the reader can refer to the OntoWeb Technical RoadMap²) have been reported to build ontologies, but few fully detail the steps needed to obtain and structure the taxonomies. For instance, Uschold and Grüninger methodology [7], METHONTOLOGY proposed by the LAI of Madrid [2], or researchers involved in the *On-To-Knowledge* project³ are rather interested in giving methodological outlines for the whole process of ontology engineering. They focus on the life cycle and the ordering of the general steps to develop these ontologies: identify the purpose of the information system, collect the relevant information for knowledge acquisition, evaluate the results, etc. Obviously, all these tasks are essential according to an “ontological engineering” point of view. However, the conceptualization step, in which the concepts and the relations between them are captured, has to be detailed.

For instance, METHONTOLOGY proposes to build the ontology at the knowledge level using a set of *intermediate representations*. Although the taxonomy is one of these representations, the methodology does not stress on the way to classify the concepts. The methodology introduced in the framework of *On-To-Knowledge* uses lexical pattern matching to extract some subsumption information from the answers given to informal competency questions [3]. It is an original way of considering the problem, but one may wonder whether this helping function could be generally applied. Finally, we may mention Nicola Guarino’s taxonomy cleaning method [4], which aims at removing wrong subsumption relation from concept hierarchies thanks to meta-properties defined by logical axioms. It is an interesting approach, yet only applicable after one has already built such a hierarchy: the first move remains to be done by the ontologist alone, which is not satisfying. That is the reason why we are eager to learn about whatever may follow from the merge attempt between this method and METHONTOLOGY [3].

2.2 Requirements for a Methodology Focusing on Natural Language

Among the methodologies evoked, few propose complete guidelines to help the user to organize the hierarchies. We claim here that, finally, none of these methodologies force the ontologist to explicit the real meanings of the concepts in the most natural way: using natural language (NL). Actually, some methodologies recommend using NL to explicit the meaning of the concepts inside comments or through documents surrounding the modeling process, but not in a principled way. The terms used to denote the concepts are still liable to multiple interpretations. This results in possible misunderstandings and consequently bad modeling and use of the ontology. We suggest then to follow an evolved version of methodological guidelines that were first proposed in [1].

² <http://babage.dia.fi.upm.es/ontoweb/wp1/OntoRoadMap/index.html>

³ <http://www.ontoknowledge.org>

The first problem to face is the under-determination of meaning: every expression in language has its meaning contextually defined, since interpretation may vary according to the context (a specific application). Modeling will thus consist in choosing linguistic labels and associating with them a relevant and non-contextual semantics. The problem is then to determine which kind of semantics and how to use it in a *normalization* effort.

Second, defining a linguistic meaning is not sufficient to specify a system. A usual approach consists in associating a formal semantics with concepts. Formal semantics allows a mathematical modeling of the linguistic meaning as well as of the system behavior. The ontologist needs a semantics formal enough to efficiently specify computations, and yet close enough to the knowledge level to make these computations intelligible.

Finally, an ontology has to introduce knowledge primitives which will be the building blocks for programming a Knowledge-Based System (KBS). From this point of view, a label will be used in rules, or grammars, or inferences, to perform computation. The associated semantics is here a computational or operational one.

To sum up, a knowledge primitive has three semantic descriptions :

- a linguistic semantic description** that provides a human user with an unambiguous understanding of a term;
- a formal semantic description** that provides a human user with a mathematical and formal account of the previous level;
- a computational description** that makes explicit the intended behavior of the computer when handling with this primitive;

The first level is what a human being can understand, the third what a computer can perform, and the second the formal modeling establishing a mapping between the two: how to understand what the KBS is doing, how to specify what it should do.

3 Methodology

The three steps we propose (Fig. 1) consist in a *semantic normalization* of the terms introduced in the ontology, followed by a *formalization* of the meaning of the knowledge primitives obtained and an *operationalization* using knowledge representation languages. The two last steps are not very different from what can be found in other methodologies. The point is the way they are integrated in a process aimed at making ontology development and use easier.

3.1 First Step: Semantic Normalization

The first step of this methodology aims at reaching a semantic agreement about the meaning of the labels used for naming the concepts. Natural language is usually the best access to the knowledge of a domain. In INA, the archivists use a collection of textual documents that are delivered with TV programs. Hence, it

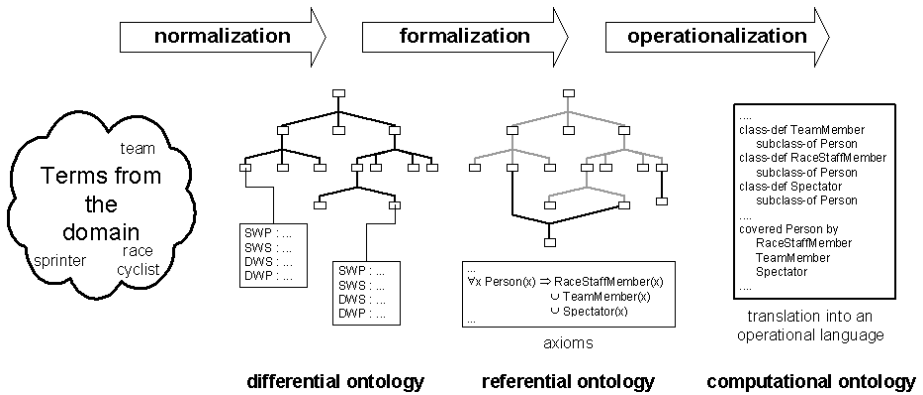


Fig. 1. The 3 steps of the differential methodology for building ontologies

seems natural to look for possible labels, candidates for future primitives, within these documents.

One of our ontologies deals with the field of cycling race, especially the *Tour de France* event. During the analysis of that domain we discovered, for instance, numerous terms referring to human beings who do not play obviously similar roles in a cycling race : *race cyclist*, *spectator*, *team manager*, *reporter*, *race supervisor*, *climber*, *wheeler*, *sprinter*...

After having extracted labels, the ontologist has to specify their meaning clearly, and therefore to use a relevant semantic theory. We are going to build a *differential ontology* which will turn these terms into *notions* based on *differential semantics* ([5]). Practically, the ontologist has to be able to express the similarities and differences of each notion with respect to its neighbors: its parent-notion and its siblings-notions. The result will be a taxonomy of notions, where the meaning of a node is given by the gathering of all similarities and differences attached to the notions found on the way from the root notion (the more generic) to this node.

We propose four principles to render explicit this information:

- The *similarity with parent* principle (or **SWP**): explicits why the notion inherits properties of the one that subsumes it;
- The *similarity with siblings* principle (or **SWS**): gives a semantic axis, a property – assuming exclusive values – allowing to compare the notion with its siblings.
- The *difference with siblings* principle (or **DWS**): precises here the property allowing to distinguish the notion from its siblings;
- The *difference with parent* principle (or **DWP**): explicits the difference allowing to distinguish the notion from its parent;

In the example given above, we can notice that terms like *climber*, *wheeler* and *sprinter* refer to *race cyclists* who are employed by teams. Actually, all

the people who usually attend the *Tour de France* do not play the same role. We can thereby gather these terms according to the role people play during the race. Thus, the notion **Person** can be specialized in three new notions – **Race Staff Member**, **Team Member** and **Spectator** – according to the differential principles given in Tab. 1.

Actually, all those principles do not have the same methodological status. First, we have noticed that the *SWP* and *SWS* principles are shared among the concepts from the same siblings. Second, the *DWP* principle has often proved to be the sum of the principles *SWS* and *DWS*: we give firstly a means to create a difference, and then we put it in a concrete form to finalize the concept definition.

→ <i>For all the following notions</i> SWP: he is a person SWS: a property precises why the person is present during the race
→ Race Staff Member DWS: he is accredited by the race management
→ Team Member DWS: he is employed by a team that takes part in the race
→ Spectator DWS: he is neither accredited by the race management, nor employed by a team that takes part in the race
→ <i>For all these notions</i> DWP: {SWS} + {DWS}

Table 1. The differential principles linked to the concepts directly specializing *Person*

3.2 Second Step: Knowledge Formalization

The ontological tree obtained in the first step allows to disambiguate the notions and to clarify their meanings for a domain-specific application. The transition to extensional semantics aims at linking the notions to a set of referents. The notions become *concepts* behaving as formal primitives and being part of a *referential ontology*. Each concept refers to a set of objects in the domain (its *extension*). Therefore, we can use the operations that exist for sets (*i.e.* union, intersection or complementary) in order to obtain new concepts.

The comparison of extensions allows to define an extensional inheritance relation between concepts: one is subsumed by another if and only if its extension is included in its parent’s extension. The subsumption relations of the differential ontology are still true in the referential ontology, but additional nodes may change the tree structure. For instance, **Climber** and **Wheeler** are exclusive notions, but the matching formal concepts can have extensions with common individuals. Typically, the race cyclist **Lance Armstrong** has these two skills.

Hence, we can define in the referential ontology – with a necessary and sufficient condition – a new concept `ClimberAndWheeler` to gather such individuals. Multiple inheritance is thereby possible.

Referential semantics allows to introduce new defined concepts but also definitions for existing concepts imported from the differential ontology. Also, the ontologist has to precise here the arity and domains of the relations. Relation signatures are defined by the means of cartesian product of concepts references. Finally, the ontologist can add some logical axioms in relation to relational algebra, part-whole reasoning, composition of relations, exhaustive partitions, etc [6]. For instance, `Race Staff Member`, `Team Member` and `Spectator` form a disjoint coverage of the concept `Person`.

3.3 Third Step: Towards a Computational Ontology

The third and last step of the methodology allows to equip the referential concepts with the possible computational operations available in a KBS: this is the *computational ontology*. The system uses an operational knowledge representation language which allows particular inferences. For a language based on the conceptual graph formalism, these inferences are graph operations (joint, projection, etc). For a language based on description logics, these inferences are mainly subsumption tests and classification.

The example below asserts that a `Person`, from the cycling point of view, is either a `Race Staff Member`, a `Team Member` or a `Spectator`. This assertion is written in the DAML+OIL language, an ontology language proposal for the Semantic Web.

```
<daml:Class rdf:about="#Person">
  <daml:disjointUnionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#RaceStaffMember"/>
    <daml:Class rdf:about="#TeamMember"/>
    <daml:Class rdf:about="#Spectator"/>
  </daml:disjointUnionOf>
</daml:Class>
```

3.4 Implementing the Methodology: The DOE Editor

*DOE*⁴ (*Differential Ontology Editor*) is a simple prototype that supports the three steps of the methodology detailed above. It is not intended to bring a direct competition with other existing environments (like *Protégé2000*, *OILed*, *OntoEdit* or *WebODE*). Rather, its purpose is to demonstrate by experimentation how taxonomy structuring can benefit from the methodology described in this paper.

During the first step, the ontologist can enter the definition of the notions according to our principles. The tool automatizes partly this task, following the

⁴ The tool is available for free at <http://opales.ina.fr/public/>.

observations made in Section 3.1. As an illustration, the Fig. 2 shows the interface recalling our *Race Staff Member* example. For the second step, it imports the taxonomies built in the previous step and allows the ontologist to specialize existing concepts and relations, as well as specify the arity and domains of the relations. Here the editor is able to make some consistency checking (propagation of the arity all along the hierarchy – if specified – and inheritance of domains). The last step is implemented by exporting the referential ontology into commonly-used KR languages (DAML-OIL, RDFS). This export mechanism also allows to refine the ontologies built, using the features supported by other editors.

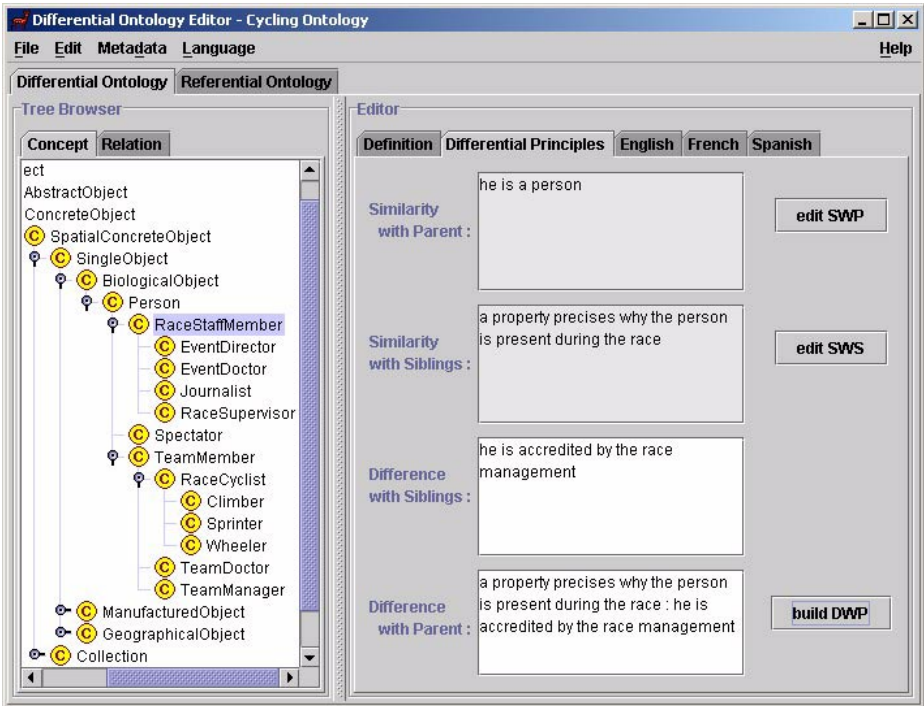


Fig. 2. The differential principles bound to the notion *Race Staff Member* in the *DOE* tool

4 Conclusion and Future Work

We have briefly evoked some methodologies for building ontologies but we have noticed a weakness: nothing forces the ontologist to assign a clear meaning to

concepts, the comments remaining mostly informal. We have proposed guidelines, mainly based on linguistics recommendations (using differential semantics) to explicit the linguistic meaning of the knowledge primitives of the ontology. The proposed methodology follows three steps: normalization, formalization and operationalization. We have implemented this methodology in an edition tool prototype, *DOE*, and several quite important ontologies have already been built within it.

For the future, we plan to better integrate our solution in a more complete ontology engineering process. Prior to the first step of the methodology, we could use the results of terminological extraction tools to get candidate-concepts and discover candidate-relations. We should also develop import mechanisms to reuse ontologies developed with other tools.

References

1. Bouaud, J., Bachimont, B., Charlet, J., Zweigenbaum, P.: Methodological principles for structuring an ontology. In *IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995.
2. Fernández, M., Gómez-Pérez, A. and Juristo, N.: METHONTOLOGY: From Ontological Art Towards Ontological Engineering. In *AAAI97 Spring Symposium Series on Ontological Engineering*, 33-40, Stanford, California, 1997.
3. In Gómez-Pérez, A. (editor): Notes for SIG on Enterprise Standard Ontology Environment. Second Ontoweb Meeting, Amsterdam, December 2001.
4. Guarino, N. and Welty, C.: Evaluating Ontological Decisions with OntoClean. In *Communications of the ACM*, 45(2): 61-65.
5. Rastier, F., Cavazza, M. and Abeillé, A.: *Sémantique pour l'analyse*. Masson, Paris, 1994.
6. Staab, S. and Maedche, A.: Ontology Engineering beyond the Modeling of Concepts and Relations. In *14th European Conference on Artificial Intelligence (ECAI'00), Workshop on Applications of Ontologies and Problem-Solving Methods*, Berlin, Germany, 2000.
7. Uschold, M. and Grüninger, M.: Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, (2), 93-155, 1996.