DAFOE: AN ONTOLOGY BUILDING PLATFORM
From Texts or Thesauri

Sylvie Szulman, Jean Charlet
LIPN, Paris 13 University, Villetaneuse, France; *INSERM UMR_S 872, Eq. 20, Paris, France*
sylvie.szulman@lipn.univ-paris13.fr, Jean.Charlet@spim.jussieu.fr

Nathalie Aussenac-Gilles, Adeline Nazarenko
CNRS/IRIT, Universit Toulouse, Toulouse, France; *LIPN, Paris 13 University, Villetaneuse, France*
nathalie.aussenac-gilles@irit.fr, nazarenko@lipn.univ-paris13.fr

Eric Sardet, Valery Teguiak
LISI-ENSMA, CRITT-Informatique, Poitiers, France
eric.sardet@ensma.fr, teguiakh@ensma.fr

Keywords: Platform, Ontology, NLP, Ontology-Based Database

Abstract: This paper describes a new platform for building ontologies using many entries (texts, terminologies, thesauri or database). However, They may be build from scratch and bound with texts afterwards. After a description of the used data model, a meta-modelisation architecture is presented as well as the database implementation, which allows to manage large ontologies with large corpora.

1 INTRODUCTION

Although text-based ontology engineering gained much popularity in the last 10 years, very few ontology engineering platforms exploit the full potential of the connection between texts and ontologies.

We propose DAFOE[1], a new platform for building ontologies using different types of linguistic entries (text corpora, results of natural language processing tools, terminologies or thesauri). DAFOE supports knowledge structuring and conceptual modelling from these linguistic entries as well as ontology formalization. DAFOE outputs models with two main original features: an ontology articulated with a lexical component and a connection with the text or linguistic entry that motivated their definition.

The requirements of the platfom and its development focus 1) on integrating various kinds of tools currently used within a single modelling platform, 2) on guaranteeing persistence and traceability of the whole ontology building process, and 3) on developing the platform in an open source paradigm with possible plugin extensions.

This paper focuses on the terminological and knowledge representation in DAFOE: after the presentation of various ontology engineering environments from texts, we describe the data model of the DAFOE platform and the corresponding meta-modelling architecture that allows its database implementation.

2 TEXT-BASED ONTOLOGY ENGINEERING

2.1 Ontology Engineering Environments

There is a growing interest for ontologies and related tools, including Ontology Engineering Environments (OEEs). Since an early overview of tools supporting ontology engineering (Duineveld et al., 2000), several joint efforts provided extensive state-of-the-art overviews, like the review of OntoWeb thematic network (Gomez-Perez et al., 2002), the comparison of ontology editors (Denny, 2004) or the evaluation of OEEs at EON 2002 workshop (Sure and Angele, 2002).

In DAFOE perspective, we carried out a survey of about 15 OEEs. Many of them are pure ontology editors that support the development of formal ontologies (often represented using OWL) but do not assist the tasks of knowledge acquisition or structuring. Knowledge engineers are supposed to have a first of ontology draft before using such tools.

Among the tools that take text as input knowledge three major approaches can be identified.

[1] `http://dafoe4app.fr`

The first one aims at building an ontology automatically out of text analysis (Bozsak et al., 2002). An alternative approach is based on human-machine cooperation. Systems like ASIUM (Faure and Nedellec, 1999), OntoGen[2] or Caméléon (Aussenac-Gilles and Jacques, 2008) support parts of an incremental and interactive ontology development. A third approach, TERMINAE (Aussenac-Gilles et al., 2008), integrate results from several NLP tools like Text2Onto[3] but do not constrain their selection or combination. The result is more than an ontology, as long as lexical entries are associated to concepts. It is called a termino-ontological resource.

2.2 Original Features of DAFOE

The goal of DAFOE is both to extend the variety of human language technologies that can be used and to support scalable ontology engineering. It claims that there are several ways to get an ontology, and that tools and processes must be selected according to each ontology case-study. DAFOE will propose tools similar to those of Text2Onto, but human supervision will play a major role for selecting tools, validating their results and conceptualizing. Knowledge conceptualization requires that a human selects and organizes properly concepts and relations, but this process can be guided. The result of DAFOE will typically be a termino-ontological resource where the ontology is connected to a lexical component.

To sump-up, this survey of the state-of-the-art emphasizes the major motivations for developing the DAFOE platform:

- we want DAFOE to keep track of the linguistic items that justify the conceptual modelling; this would help not only to maintain and update the ontology, but also to produce richer ontologies associated with a lexical component. DAFOE will build termino-ontological resources.
- since we argue that extracting an ontology from texts is a complex process that requires several modelling steps and that the models produced at each stage are worth being stored, we want DAFOE to support that multi-step modelling process and to store the intermediary models as a modelling track.
- we want DAFOE to make it possible to use the links between those various models to come back from the ontology to the text that has been used to build it, to definitions and comments that justify the content of the ontology.
- we want DAFOE to support large-scale ontologies and lexical resources, which is a major argument towards a database implementation (Sec. **??**) and an important difference with systems like TERMINAE or Protégé.

3 DATA MODEL

DAFOE data model has to take into account various ontology building strategies, whatever information source (texts, terminologies, thesauri or human expertise) is used. Moreover, DAFOE must evolve to offer new functionalities in satisfaction to new needs, which means that plugin extensions will be supported and that the data model allows these extensions.

3.1 Overall Architecture

The data model is based on a valid methodology for building ontologies from texts, which has inspired tools such as TERMINAE (Aussenac-Gilles et al., 2008) or Text2Onto (Cimiano and Volker, 2005). This methodology takes into account the whole process of "transforming" textual data into ontologies and split it into different phases, which correspond to various input levels if one wants to start with a thesauri rather than text, for instance. This methodologies relies on two main ideas: 1/ textual data are an important information source to build ontologies, especially if the ontology is to be used to annotate textual documents but 2/ textual data cannot be mapped directly into an ontology and the transformation must be mediated.

The data model is therefore structured into four layers. Each one corresponds to a specific methodological step.

3.2 Corpora Layer

The corpora layer is useful for the knowledge engineer willing to build an ontology from text. He/she can build a working corpus by selecting different source documents and browse that corpus, either as plain documents or as segmented ones. In the data model the corpus is represented as a sequence of sentences, each one having a unique identifier.

3.3 Terminological Layer

The terminological layer gives a view over the domain specific lexicon of the corpus. It gathers the terms of the domain and their relationships. Terminological knowledge is traditionally produced by NLP tools such as term extractors applied on the working

[2] http://ontogen.ijs.si/

[3] http://ontoware.org/projects/text2onto/

corpus. An alternative approach would consist in importing a preexisting terminology of the domain. The underlying assumption is threefold: text analysis can extract term candidates that are relevant for a given domain, those terms are likely to be turned into ontology concepts and the distribution of these terms reflects their semantics (Harris, 1968).

3.4 Termino-Conceptual Layer

This layer represents a semantic structure of unambiguous termino-concepts and termino-conceptual relations. The knowledge engineer may build that layer by importing a preexisting termino-conceptual resource such as a thesaurus or out of the analysis of the terminological layer. In that case, he/she analyses the meaning of terms and relations that appear at the terminological layer with respect to each other and by looking at their occurrences. He/she clusters terms and relationships that have the same meaning, distinguishes the various meanings of ambiguous terms, compares the contexts in which they are used. He/she thus defines non ambiguous termino-concepts (TC) and termino-conceptual relations (RTC) holding between CTs.

The termino-conceptual layer is pivotal for transforming linguistic elements into conceptual ones and tracing the ontology back to the linguistics. This traceability improves ontology readability and maintenance. The terms and terminological relations that are connected to termino-conceptual elements are said to be ”conceptualised”.

3.5 Ontology Layer

The ontology data model allows to formalize TCs and RTCs in a formal language equivalent at OWL-DL. Concepts are described as classes, individuals as instances of classes, properties between classes as object properties and properties between a class and a value as data properties or attributes. An automatic process will translate TCs and RTCs into formal concepts in a hierarchy with inherited properties as usual subsumption in description language. This translation exploits the structure of the semantic network represented in the termino-conteptual layer and the differential criteria associated with TCs and RTCs.

4 META-MODELLING ARCHITECTURE AND DATABASE IMPLEMENTATION

DAFOE platform is intended both to support large volumes of data and to provide a variety of ontology engineering methods. As such a diversity can not be managed in a unique and static model, we adopted for DAFOE platform an extended Ontology-Based Database (OntoDB) architecture that supports model management. In this section, we present the original OntoDB model, we show how it is extended to cope with DAFOE specificities, we illustrate the resulting architecture with a simple example and we give some implementation details.

4.1 ONTOLOGY-BASED DATABASE (ONTODB)

The OntoDB model has two characteristics: (1) ontology and data are stored in the database and support the same processes (insert, update,...), (2) every piece of data is associated with an ontological element that defines its meaning. The OntoDB architecture is quite similar to the layers defined in the MOF[4] architecture, which consists of three levels of modelling: the M1 level allows to represent the models, the M2 level represents the meta-models and the M3 level represents the meta meta model (instances are represented in M0). An important feature provided by OntoDB is its capability to support model evolution and data intensive applications (H. Dehainsala, 2007). This is one reason why we have proposed an OntoDB-like architecture for DAFOE.

4.2 DAFOE ARCHITECTURE: A MODEL-BASED DATABASE (MBDB)

Since corpora and extracted terminologies are generally large, we opted to store corresponding data in a database and the choice of a MOF architecture led us to adopt an OntoDB architecture. The integration of DAFOE data model into OntoDB is not trivial, however. The main difficulty stems from the fact that an OntoDB architecture relies on a single model, while the DAFOE approach is based on three different models: a Terminological Model (TM), a Termino-Conceptual Model (TCM) and an Ontological Model (OM). Hence, the idea is to represent each of DAFOE

[4] http://www.omg.org/mof/

three models in a separate OntoDB based architecture. More generally, this integration requires an effort of identification and distribution of DAFOE data according to a MOF architecture.

4.2.1 Data Distribution

The data exploited in DAFOE for building ontologies from texts is distributed according to the three abstraction levels of MOF. The particularity of this approach is that it provides, for each part (data, model and meta-schema) of the managed information layers (terminology, termino-conceptual and ontological), a meta-modelling level supporting the evolution of those part structures. It is represented by the MOF syntax (Mi/Mi-1), which means that the information model Mi-1 is represented as instances of the information (meta) model Mi.

4.2.2 Model Transformation

Even if the integration of DAFOE into OntoDB solves the problem of model evolution thanks the various abstraction levels defined by MOF, the problem of transition from one modelling layer to another remains open. To solve this problem, we propose to use model transformation mechanisms to enhance the DAFOE architecture.

5 CONCLUSION

DAFOE is a new platform to assist a knowledge engineer throughout the ontology building process. It allows him/her to integrate domain specific knowledge sources (text corpora, terminologies, thesauri or human expertise) and to define a formal ontology. The strength of DAFOE approach is i) a precise definition of the various steps by which one can design a formal ontology; ii) a data model guaranteeing persistence and traceability of the whole ontologies building process; iii) the supply of flexible methodological guidelines that support the knowledge engineer without constraint; iv) an architecture based on the MOF model and plugins adaptability to ensure extensibility of the model and processes around a core tool; v) the specification of various modelling strategies based on different input/output of the platform; vi) the final production of an ontology which is associated to a terminological component.

A prototype of the DAFOE platform is under implementation. We use Model Driven Engineering and the EXPRESS language.

REFERENCES

Aussenac-Gilles, N., Despres, S., and Szulman, S. (2008). *Bridging the Gap between Text and Knowledge: Selected Contributions to Ontology learning from Text*, chapter The Terminae Method and Platform for Ontology Engineering from Texts. IOS Press.

Aussenac-Gilles, N. and Jacques, M.-P. (2008). Designing and evaluating patterns for relation acquisition from texts with caméléon. *Terminology, special issue on Pattern-based Approaches to Semantic Relation Extraction*, 14:45–73.

Bozsak, E., Ehrig, M., Handschuh, S., Hotho, A., Maedche, A., Motik, B., Oberle, D., Schmitz, C., Staab, S., Stojanovic, L., Stojanovic, N., Studer, R., Stumme, G., Y. Sure, J. T., Volz, R., and Zacharias, V. (2002). Kaon - towards a large scale semantic web. In *Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web 2002), Aix-en-Provence, France*, volume 2455 of LNCS, pages 304–313. Springer Verlag.

Cimiano, P. and Volker, J. (2005). Text2onto - a framework for ontology learning and data-driven change discovery. In Montoyo, A., Munoz, R., and Metais, E., editors, *Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB)*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238, Alicante, Spain. Springer.

Denny, M. (2004). Updated ontology editor survey results (table). available at http://www.xml.com/pub/a/2004/07/14/onto.html.

Duineveld, A., Stoter, R., Weiden, M., Kenepa, B., and Benjamins, V. (2000). Wondertools? a comparative study of ontological engineering tools. *International Journal of Human-Computer Studies*, 6(52):1111–1133.

Faure, D. and Nedellec, C. (1999). Knowledge acquisition of predicate argument structures from technical texts using machine learning: the system asium. In Fensel, D. and Studer, R., editors, *Proceedings of the 11th International Conference on Knowledge Engineering and Knowledge Management*, pages 329 – 334. Springer-Verlag, LNAI.

Gomez-Perez, A., Angele, J., Fernandez-Lopez, M., Christophides, V., Stutt, A., and Sure, Y. (2002). A survey on ontology tools. ontoweb deliverable 1.3. Technical report, Universidad Politecnica de Madrid.

H. Dehainsala, G. Pierra, L. B. (2007). Ontodb : An ontology-based database for intensive applications. pages 497–508.

Harris, Z. (1968). *Mathematical Structures of Language*. Interscience Publishers.

Sure, Y. and Angele, J., editors (2002). *Proceedings of the First International Workshop on Evaluation of Ontology based Tools (EON 2002), Siguenza, Spain*, volume 62 of *CEUR Workshop Proceedings*. CEUR-WS Publication. available at http://CEUR-WS.org/Vol-62/.