A method to build formal ontologies from texts

Jérôme Nobécourt

Université de Paris-Nord. Laboratoire d'Informatique de Paris-Nord (LIPN) - UPRES-A 7030 Av. J.B. Clément 93430 Villetaneuse (France), Jerome.Nobecourt@lipn.univ-paris13.fr tel: +33 (0)1 49 40 36 09

fax: +33 (0)1 48 26 07 12

Abstract. We present a method to build domain ontologies from texts, which originates in recent French works on knowledge acquisition. The method has been defined for an industrial application concerning the supervision of telecom networks. It uses the $Md_{\omega_{\varsigma}}$ modelling language, which allows further formalisation into CG or DL.

Introduction

We present a modelling method starting from texts and building a formal representation into description logics or conceptual graphs (DL and CG in what follows). This method uses the $Md_{\omega\varsigma}$ modelling language and a tool presented in [NB00]. It has been applied onto an industrial problem, the supervision of complex telecom networks¹. In this application, the domain knowledge have to be discovered from texts.

Therefore, a corpus has been build and linguistically analysed using the principles proposed by the French TIA² group. Terms with their description in natural language have been extracted. The method guides the identification of the conceptual primitives of the domain from the set of terms resulting from the linguistic analysis, as well as their modelling under $Md_{\omega\varsigma}$ schemata. All the schemata form a domain ontology that can be formalised in CG as well as in DL using some translation rules. The tool supporting $Md_{\omega \zeta}$ offers a graphical modelling by successive refinement from the text study to the formalisation, keeping the link to the texts.

The following section explains the method, on an example taken from the telecom application. Some general translation rules are explicited there. The place of $Md_{\omega\varsigma}$ is discussed in section 3 relatively to close works in knowledge engineering.

¹ Granted by CNET/CNRS cognisciences, Centre National d'Études des Télécommu-

² The "Terminologie et Intelligence Artificielle" group is a working group of the AFIA, the French Association for Artificial Intelligence.

2 The Method

2.1 The supervision domain

Supervision consists in making a real-time diagnosis of the incidents that happen in a complex telecom network. The diagnosis is based on the observation of the state of the network which is composed of equipments connected together, mainly beams and switches. When some incident happens, several indicators are positioned (lights, messages, curves...). These indicators give information on network's componants and traffic. Then, an operator provides a diagnosis concerning the type of the incident. With this diagnosis, he/she should perform some action to inhibit the consequences of this incident on the network. Our job concerned building an ontology to allow a description of every incident from the alarms appearing in the supervision room. Having no access to the site, we had to build the ontology from the various documents available. Nevertheless, an expert could validate our results.

2.2 Modelling phase

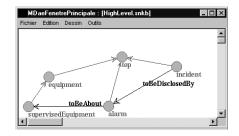
First, a corpus has to be defined relatively to the objective. A candidate-term extractor [Bou96] allows the automatic extraction and further expert validation of the main terms of the domain. This gives a list of conceptual primitives (CPs in what follows). During the modelling phase, the name given to the primitive should keep the link with the associate term. From this first and often very long list, the knowledge engineer looks for primitives that describe the main sub-domains of the ontology. These primitives structure the ontology. They are modelled as concepts and they form initial list of modelling CPs (LMCP). In the application, it is the case of "alarm", "incident" (figure 1).

Each primitive of LMCP is described in natural language. Their description is considered as a new document, from which a new list of CPs is created. The knowledge engineer looks for in this list:

- 1. CPs that express a relation between high-level concepts,
- 2. primitives already present in the initial list,
- 3. new primitives only present in this list.

In our application, the first case leads to define basic properties between high-level concepts, like "toBeDisclosedBy" or "toBeAbout" (figure 1). Cases 2 and 3 give new high-level concepts like "cause" (figure 2) and so allow to build up the LMCP.

The method is therefore based on a process of successive refinements of the CPs initially selected. At each iteration, new schemata are created and the previous are updated; especially, the state and nature of a concept may change. The state indicates the progress of the primitive modelling. The nature specifies the origin and the role that the primitive plays in the modelling. There are two kinds of nature, a structural and a linguistic one. The structural nature explains the role



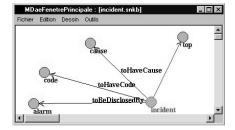


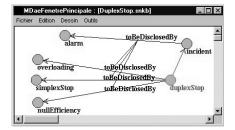
Fig. 1. The high-level of ontology

Fig. 2. Adding new concepts

of the concept in the modelling. The linguistic nature specifies the link of the primitive with the corpus.

An expert must as soon as possible validate the schema of the high-level ontology to stabilize the different sub-ontologies. Once a stable state is reached, the other primitives of the initial list are gradually modelled and added to the LMCP.

Let us suppose that the CPs on equipment, incidents and actions, are already modelled. We apply this process to the term "duplex stop incident" which occur in numerous technical documents. A primtive called "duplexStop", is associated to this term. Its description is the following: "A duplex stop incident is disclosed by one simplex stop alarm, one or more overloading alarms and at least three null efficiency alarms. The alarms of type simplex stop are alarm events on switches. The alarms of type overloading and the alarms of type null efficiency provide information on the state of the beams". This description includes many CPs are not yet modelled, among which "duplexStop" (incident of type duplex stop), "simplexStop" (simplex stop alarm), "overloading" (alarm of type overloading of beams), "nullEfficiency" (alarm of type null efficiency). The table 1 presents the LMCP before inserting these new primitives. Each primitive is describe according to its modelling as a concept or as a property, its level of concepts or its type of properties (Basic or Specialised).



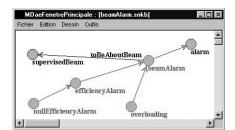


Fig. 3. Use of calculated property

Fig. 4. Example of grouping concept

Among the new CPs, some primitives specialise already existing concepts. For example "duplexStop" is kind of "incident" (figure 3). These primitives are modelled as specialising concepts. Their linguistic nature is terminological or pre-terminological depending on the fact that the primitives correspond more or less to domain terms. The description of these primitives lead to specialise some

Conceptual primitives (Concept)	Level	Conceptual primitives (Concept)	Level
alarm	1	$\operatorname{simplexStop}$	3
beam	3	$\operatorname{switchAlarm}$	2
cause	1	$\operatorname{supervisedEquipement}$	2
efficiencyAlarm	3	switch	3
incident	1		
Conceptual primitives (Property)	Туре	Conceptual primitives (Property)	Туре
toBeDisclosedBy	В	${ m toBeAboutBeam}$	S
${ m toBeAbout}$	В	${ m toBeAboutSwitch}$	S

Table 1. Example of conceptual primitives

properties. In our example, "duplexStop" is disclosed by three types of alarms which are modelled (figure 3) by three specialisations of "toBeDisclosedBy".

If some properties are shared by more than one concept, the knowledge engineer may add a grouping concept, like "beamAlarm" (figure 4) while its label does appear neither as a term, nor as a complex expression in the documents. Its linguistic nature is not terminological.

The final modelling gives:

- 1. the three properties "toBeDisclosedBy" have a terminological nature,
- 2. the concept "duplexStop" has a terminological nature,
- 3. the concepts "overloading", "nullEfficiencyAlarm" and "simplexAlarm" have a nature (structural, terminological).

2.3 Representation phase

The representation is the translation of the modelling schemata into an object formalism like CG or like DL. In this section, we present only a idea of the difficulties meet with this translation.

The translation needs to answer (at least) four questions:

What is a concept in the formalism?

The translation of a $Md_{\omega_{\varsigma}}$ concept must be clearly defined into the formalism. It is not obvious at all that a direct translation would be possible even if a primitive called *concept* is defined in the formalism. This concerns in particular the TopConcept.

What is a basic property?

The translation of basic properties must also receive a properly defined representation. It is necessary to specify whether their translation can modify the description of the concepts that use it.

How represent the specialisation?

The specialisation of properties must then be defined in the formalism. Again, it must be clear whether they may entail modifications of concept or property descriptions.

How translate calculed property?

Finally, case by case, each calculated property should be represented. The corresponding computation has to be represented by special-purpose operators, inferences rules or algorithms. Their translation is generally the most difficult problem. We have shown in [Nob00] that a chain of binary properties can sometimes be translated only into an inference mechanism.

These general guidelines have to be made explicit for each target formalism. We give a detailled description in [Nob99].

3 Discussion and Conclusion

3.1 Modelling from texts

Knowledge Acquisition from texts has a very important development with the current works in terminology and natural language processing (NLP). These works allow the acquisition of terminological knowledge by applying NLP-tools to a corpus. The first results lead to the definition of a "terminological knowledge base" [AGBCG95] [MSBE92]. Following this current trends in knowledge acquisition from texts, we argue that an ontology, i.e. the static knowledge on a domain, must be built from texts and that the link with texts must be kept: an ontology must be terminological. Apart the interest during the modelling, keeping the links to the texts eases the understandability and the maintenance of the ontology. The construction of an ontology has to be done according to three major steps: an informal terminological knowledge base (TKB afterwards), a semi-formal terminological ontology (SFTO) and finally a formal terminological ontology which is the representation of the SFTO into some formalism. The TKB corresponds to the linguistic level. It is built from the study of texts gathered in the domain, which are constituted into a corpus. This base gives a first list of conceptual primitives that correspond to terms, as notes or links expressed in natural language, related to concepts. This TKB is represented into a semantic network, with no automatic validation.

The SFTO corresponds to the description and the structuration of the conceptual primitives as concept or properties. The ontology is structured by properties linking the concepts together. The concepts and properties are described into a semi-formal language. It corresponds to the definition of [SPKR96] "An ontology is the hierarchical structuration of a set of terms of the domain". The syntax of the language must be clearly defined. A simple semantic verification must be feasible in order to make possible the translation into a formal language. However it has to leave a large freedom, to model without having troubles with complex inference mechanisms.

Finally, the formal representation of the terminological ontology is useful to validate the semi-formal one, to test it and to implement it.

In [Gua93] Guarino introduces an ontological level in which linguistics criteria allows the concepts and the properties to be structured, as the ontology. This level is the same as our modelling phase resulting into the SFTO.

[KP99] define the notion of a "conceptual ontology" between terminology and the operational ontology, which corresponds to the conceptualisation of the terms. Two phases are distinguished, the "ontologisation" and the "operationalisation". The ontologisation goes from the terminology to the conceptual ontology, it is realised in a language. It corresponds exactly to our modelling phase, the conceptual ontology corresponding to our SFTO. The operationalisation is the coding of the conceptual ontology into

a representation formalism, that is our representation phase that leads to a formal terminological ontology.

Among the most recent works in knowledge acquisition from texts, [BS99] present a method for building ontologies from a linguistic analysis of texts. They propose to apply some tools from natural language processing (terms extractors, tools for conceptual grouping, relations extractor...) onto a corpus constituted for an application. When the results of the tools have been sifted, it is possible to define what we called here the conceptual primitives of the domain. It is often necessary to return to the occurrences of the terms in the corpus. This approach is based on distributional and differential linguistic semantics. The distributional one argues that the meaning of a word can be obtained from its uses in the texts. The differential one argues that a word gets its meaning from its differences with other words, not from an hypothetical object of the world to which it would refer.

Then the informal definition must be normalised and formalised to fit into a formal ontology. The tool TERMINAE has been developed to support the successive steps from texts to formal ontology, via terminological and normalisation forms. It offers an entire traceability for modelling and keeps the links between the occurrences of a term and the conceptual primitive that formalises it.

 $Md_{\omega_{\varsigma}}$ comes into the normalisation phase of this method. The study and the preliminary extraction of the conceptual primitives in $Md_{\omega_{\varsigma}}$ can be realised during the phase of linguistic study of TERMINAE. The modelling language of $Md_{\omega_{\varsigma}}$ can then help to define the modelling forms of TERMINAE, or even replace them by schemata. It allows a very smooth beginning of structuration and makes the numerous backtracks easier. Moreover, $Md_{\omega_{\varsigma}}$ proposes translation rules towards DL and CG, it so gives some translation abilities to other formal systems that TERMINAE.

3.2 Conclusion

We have presented a method for acquiring knowledge from texts. It aims to keep the link between words in text and conceptual primitives in model. Thus, understanding and maintaining the model but also its formalisation would be easier.

We have illustrated the modelling steps, from the texts to a semi-formal terminological ontology in Md_{ω_5} with examples taken from an industrial application. The translation rules, from the semi-formal ontology to a formal one in CG and DL, may seem obvious as Md_{ω_5} is also semantic network-like. We have shown in [Nob00] that a chain of binary properties can sometimes be translated only into an inference mechanism.

The language and method are supported by a graphical Java tool in which a domain ontology can be conceived. The schemata in this paper come from the user interface. All the operations on the schemata are graphical. The tool checks the syntax and the semantic rules. It can be interfaced with CG and DL implementations. The translation rules from Md_{ω_s} to the formalism are then applied by dialoguing with the user.

References

[AGBCG95] N. Aussenac-Gilles, D. Bourigault, A. Condamines, and C. Gros. How can knowledge acquisition benefit from terminology? In *Proceedings of KAW'95*, 1995.

- [Bou96] D. Bourigault. Lexter, a natural language processing tool for terminology extraction. In *Proceedings of the 7th EURALEX International Congress*, Goteborg, 1996.
- [BS99] B. Biébow and S. Szulman. Terminae: A linguistics-based tool for building of a domain ontology. In D. Fensel and R. Studer, editors, *Proceedings of the 11th European Workshop (EKAW'99)*, LNAI, 1621, pages 49–66. Springer-Verlag, 1999.
- [Gua93] N. Guarino. The ontological level. In Proceedings of 16th Wittgenstein Symposium, 1993.
- [KP99] G. Kassel and S. Perpette. Cooperative ontology construction needs to carefully articulate terms, notions and objets. In V.R. Benjamins, D. Fensel, and A.G. Pérez, editors, Proceedings of International Workshop on Ontological Engineering on the Global Information Infrastructure, pages 57-70, 1999.
- [MSBE92] I. Meyer, D. Skuce, L. Browker, and K Eck. Towards a new generation of terminological resources: An experiment in building a terminological knowledge base. In 13th International Conference on Computational Linguistics (COLING), Nantes, France, 1992.
- [NB00] J. Nobécourt and B. Biébow. Md_{ω_s} : a modelling language to build a formal ontology in either description logics or conceptual graphs. In R. Dieng and O. Corby, editors, *Proceedings of the 12th International Conference*, EKAW'2000, LNAI 1937, pages 57–64. Springer-Verlag, 2000.
- [Nob99] J. Nobécourt. De la modélisation à la représentation : $Md_{\omega_{\varsigma}}$ un exemple de langage de modélisation multi-formalismes. PhD thesis, Université Paris-Nord, Villetaneuse, France, 1999.
- [Nob00] J. Nobécourt. Représenter la notion de propriété dans les graphes conceptuels et les logiques de description. In J. Charlet, M. Zacklad, G. Kassel, and D. Bourigault, editors, *Ingénierie des connaissances : Evolutions récentes et nouveaux défis*, pages 177-194. Eyrolles, 2000.
- [SPKR96] B. Swartout, R. Patil, K. Knight, and T. Russ. Toward distributed use of large-scale ontologies. In Proceedings of Knowledge Acquisition Workshop (KAW'96), 1996.