Concept Maps: Theory, Methodology, Technology
Proc. of the First Int. Conference on Concept Mapping
A. J. Cañas, J. D. Novak, F. M. González, Eds.
Pamplona, Spain 2004

TWO-LAYERED APPROACH TO KNOWLEDGE REPRESENTATION USING CONCEPTUAL MAPS AND DESCRIPTION LOGICS

Hector Gómez-Gauchía, Belén Díaz-Agudo, Pedro González-Calero
Departamento de Sistemas Informáticos y Programación,
Facultad de Informática de la Universidad Complutense de Madrid
{hector,belend,pedro}@sip.ucm.es

Abstract. The knowledge adquisition phase in Knowledge-based systems is still an important bottleneck. Case-Based Reasoning is a paradigm to build knowledge based systems, where past stereotypical experiences are used as the main source of knowledge to solve current problems. We use a knowledge intensive approach to CBR where the involved general knowledge is formalized usign Description Logics (DLs). Our experience has taught us that DLs are too complex to be understood by domain users. We observed that, because this complexity, they try to avoid the use of the system. We developed a methodology that includes several aspects in order to break this blockage. One aspect is the use of a two-layered knowledge representation: a formal layer with DLs and a graphical layer based on CMAPS. We centered our work with CMAPS in the first stage of the knowledge engineering cycle, the conceptualization phase.

1 Introduction

The knowledge adquisition (KA) phase in knowledge-based systems is still an important bottleneck. Case-Based Reasoning (CBR) is a well known paradigm where past stereotypical experiences are used to solve current problems. This way the typical KA burden is reduced. The typical KA phase includes the conceptualization and formalization stages in the knowledge engineering cycle. It has been several approaches to ease the bottleneck. One of these approaches that has been successful is the graphical representation, such as conceptual maps (CMAPS). This type of representation is natural for the users. They have a similar kind of conceptual network to represent their knowledge in their minds.

There are other works that use CMAPS with CBR. One of the most relevant is described in (Leake&Wilson 2001). They have developed specific tools for aerospace design and stablished a framework for interactive case-based design support systems. They use CMAPS as the knowledge representation language for CBR. They have developed CBR tools specifically for this reprentation language.

On the contrary, our Knowledge Intensive CBR (KI-CBR) framework (Díaz-Agudo&González-Calero 2001; Díaz-Agudo 2002) is independent of the domain and it maybe applied to design or to any other task. The "knowledge intensive" qualifier is used when the specific knowledge described in the cases is complemented with an ontology containing general knowledge about the domain.

We[1] integrate this domain ontology with another ontology, called CBROnto, that has the knowledge about CBR itself, i.e.: case structure and the CBR processes (Problem Solving Methods that are generic and reusable) (Díaz-Agudo&González-Calero 2002). These methods include mainly similarity assessment for case retrieval and adaptation. Our KI-CBR system uses Description Logics (DLs) as the formalization language of all the knowledge, taking advantage of the charateristics of DLs for the reasoning tasks. In particular we use OWL (Bechhofer *et al.* 2004), a new standard that has recently reached a high relevance. The choice of OWL as representation language provides the additional advantage, that it is designed to work with inference engines like RACER (Haarslev & Möller 2003).

DLs are complex to be understood by domain users. We observed that, because this complexity, they try to avoid the use of the system. We developed a methodology that includes several aspects in order to break this blockage. One aspect is the use of a graphical layer, such as CMAPS. We centered our work with CMAPS in the first stage of the knowledge engineering cycle, the conceptualization phase.

There are other related works that uses two representation languages. One of these works is the effort of the RKF[2] (*Rapid Knowledge Formation*) development team, where they propose to develop a suite of tools, providing a framework to display and record the content of an evolving Knowledge Base. One of these tools is

[1] Supported by the Spanish Committee of Science & Technology (TIC2002-01961)
[2] http://www.rl.af.mil/tech/programs/rkf/

the MILK prototype. It implements a CMAPS version that communicates with Shaken, a Knowledge server developed by SRI International. MILK retrieves models from the server to be rendered as conceptual maps or verify operations that the users may perform on the models. This approach uses two different representations as ours: CMAPS and the Knowledge Server language. In our approach we keep the two languages acting as two different layers, the conceptualization and the formalization layer. Both are integrated in the methodology and the KI-CBR system.

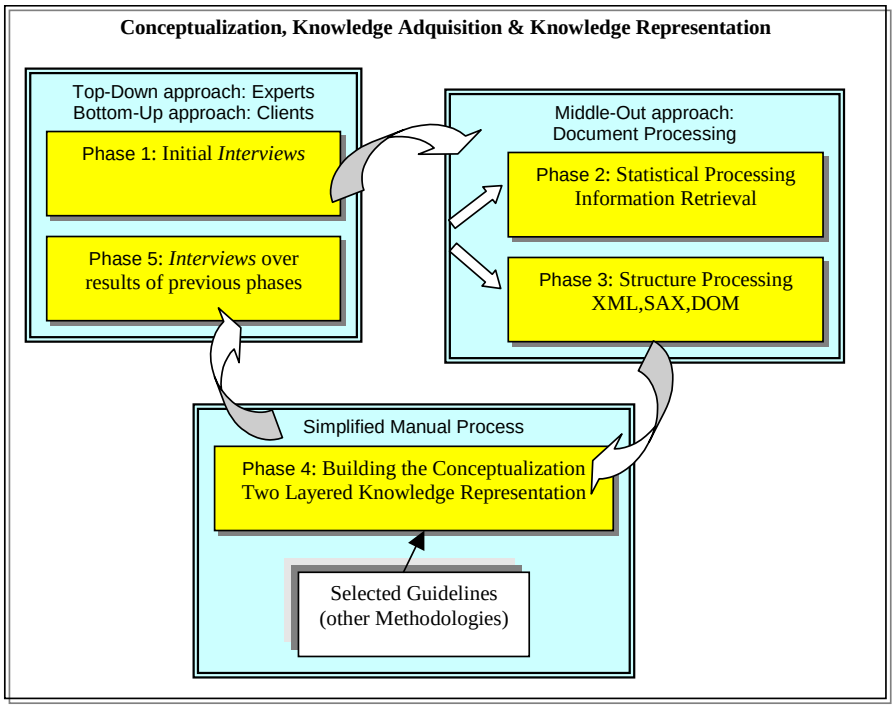


Fig. 1: Lightweight Ontology Methodology summary: Refinement Cycle

This article describes a two layered knowledge representation enbedded in a methodology to build lightweight ontologies. We have applied it in a project for a test site with good results. In order to understand the representation and its motivation we first describe briefly the methodology. Section 3 describes the first step of knowledge creation, i.e., how the domain ontology is acquired and represented using CMAPS. Section 4 introduces the basics of DLs as a knowledge representation technique, and Section 5 explains the second step of knowledge creation, i.e. how DLs are used to formalize the domain ontology. Section 6 deals with the transformation from one knowledge layer, the CMAP layer, into the other, the DLs layer. In Section 7 we describe shortly the test site, Golden Soft, a software producer for managent, and the results of a test experiment and Section 8 concludes the paper.

2 The methodology to build lightweight ontologies

To make the ontology of the KI-CBR system we reviewed, in (Gómez-Gauchía et al. 2004 a), some of the most representative methodologies to build ontologies, and we identified good guidelines that may be applied in the adequate phase of the proposed methodology, as it is shown in Fig. 1. The current methodologies to build ontologies follow a wide range of theories, approaches and scopes of their application. After doing the survey of methodologies and starting the conceptualization phase, we found several problems applying them:

1. The lack of understanding of the domain terms and the lack of experts.
2. The need to facilitate the ontology definition, using a formal representation language, by domain experts that are not computer experts. In this article we call them users.
3. The need to structure the whole process of guidelines, tasks and support materials.

To solve them we, as knowledge engineers, use theoretical paradigms already tested in other areas of research. We propose the following solutions to face up these problems:

- To obtain relevant concepts by processing written sources of knowledge, used as a guide for the next phases. We use Information Retrieval and Document Structure processing techniques.

- To support human communication through conceptual structures. We propose representing knowledge by means of two layers: a user layer with an easy graphical language, CMAPs and an internal layer with a formal representation language, Description Logics.
- To build up the ontology in an incremental manner. We define a refinement cycle. It is based on the three main conceptual strategies, top-down, bottom-up and middle-out. They are applied in different phases of the cycle, including a final phase with a manual task by the user.

In (Gómez-Gauchía et al 2004 a) we discuss the first stages of the knowledge engineering approach, i.e. the conceptualization of the knowledge acquisition and the knowledge representation of a lightweight ontology. We have defined a pragmatic methodology as a complement to other classical methodologies. The methodology, shown in Fig. 1, has three main tasks, which group five phases of the cycle. These tasks are repeated in the refinement cycle until all the participants reach a consensus of the ontology semantics. The tasks are based on the source of information and its treatment:

1. *Interviews*, with the clients in a bottom-up style and with the experts in a top-down style. It includes phase one, the initial conceptualization, and phase five, the refinement of the ontology obtained in previous phases.
2. *Document Processing* of the written documentation to extract the most relevant terms. This is the middle-out strategy. Depending on the nature of documents, a statistical processing, phase two, or a document structure processing, phase three, is applied.
3. *Simplified Manual Process* to build the ontology in a two-layered representation, that takes, as basis the relevant concepts obtained in the previous tasks. This is phase four. The selected guidelines of the studied methodologies in the survey are applied in this task. The user only works with the informal graphical representation, leaving the formal layer internal to be used by the KI-CBR system.

The phase 4 is where the two-layered representation is applied. The solution was to define an incremental knowledge process with two cycling steps:

- *Conceptualization:* the user describes a piece of knowledge in a graphical language using CMAPS.
- *Formalization:* the system tries to automatically formalize the new portion of knowledge in a paradigm, such as DLs, that allows performing the verification and reasoning tasks that the KI-CBR system needed. This process is described in section 6 of this article.

The one-layered representation follows the *meaning triangle* (Ogden & Richards 1946). It makes a distinction between the symbol, the concept and the object, as depicted in Fig. 2a. For our particular application, the external symbol is the term or image that is drawn in the CMAP; the concept, inside the human agent, is what is named by the term; and the object, the thing in the real world, referred to by the concept. The theory behind the two-layered representation is the inclusion of the forth element, the *internal formal universe* (Hartley & Barnden 1997), into the meaning triangle. This gives the formal semantics needed to be able to reason with the meaning model drawn in Fig. 2b.

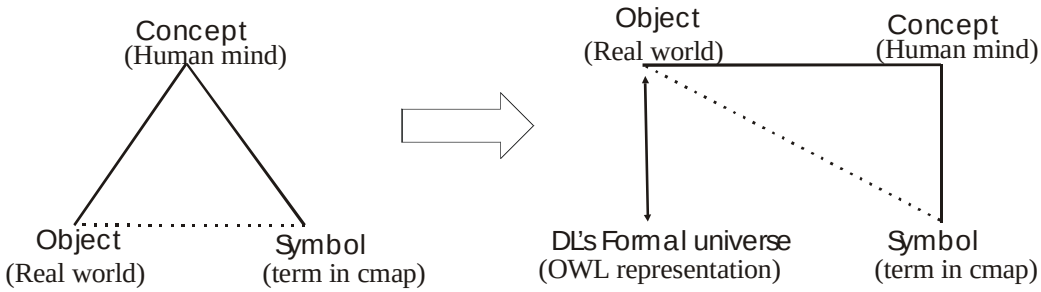


Fig. 2 a) The Meaning Triangle b) Formal Semantics included in the Meaning Triangle

3 The representation of our ontologies with Conceptual maps

As we have described, in our project we define and integrate two ontologies. To do it, we use the CmapTools system made by the Institute for Human and Machine Cognition. A small portion of them is depicted in Fig. 3:

- CBRonto (left) supplies the framework to represent the elements needed to deal with the CBR paradigm. It has around 200 terms, between concepts and relations.
- Domain ontology to represent the concepts of the application domain, the managment of companies in our project. It has few hundreds elements as well.

From the user point of view the main difficulty to deal with them is the so call *growth problem*: to be able to have a perspective of the relative location of an element in relation with the whole ontology. It is difficult even with a graphical editor, such as Protege that uses lists of topics in a hierarchy.

The main problem associated with Protege is that its *tree-like* representation of graphs does not allow a clear visualization of the structure of the knowledge. It is serious in our application because we use intensively multiple inheritance specially during the integration between the domain ontology and CBROnto (Díaz-Agudo&González-Calero 2002).

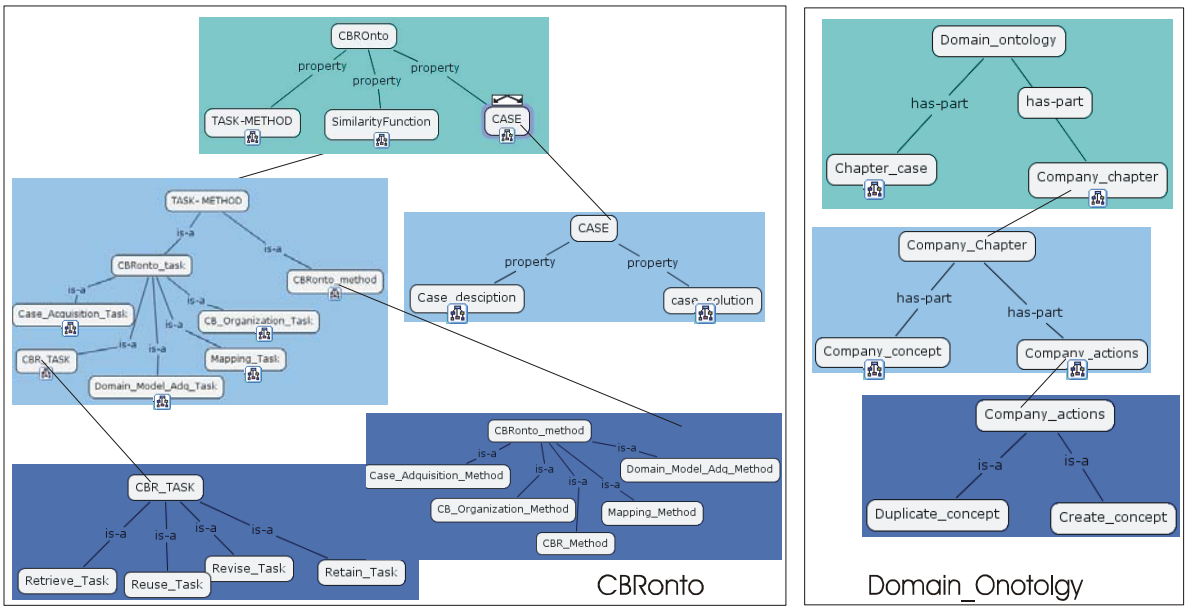


Fig 3. Ontologies of the Methodology applied to a KI-CBR system of Help-Desk

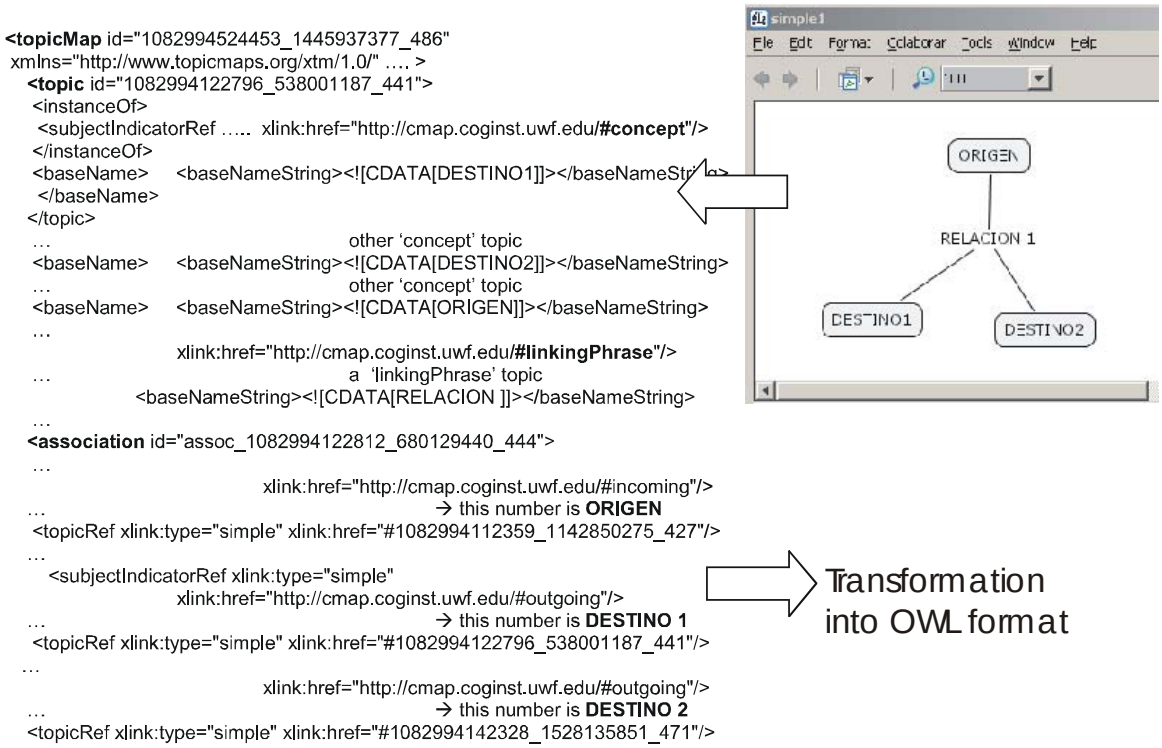
To deal with the problem in this phase, we facilitate the visual comprehension using CMAPS. We obtained possitive results using some restrictions in these maps:

- To create many small CMAPS with few concepts and relations each.
- To organize them in an ortogonal manner:
  - horizontally by levels of abstraction
  - vertically by subareas of the ontology.
- This is done by a code based on colours.

A different but important goal is that we must be able to formalize them into a DLs language. To obtain this we check the consintency of the map against its formal representation as we see in the next section.
The advantages of CMAPS representation are mainly congnitive. The users of the ontologies understand easily the ontology as a whole and they follow the relative position of each concept and relation in the whole ontology. The CMAPS graphical represention is a familiar format for non-thecnical users. A specific advantage of the CmapTools is that the internal representation languge is XTM, a small portion is in the Fig 4. The XTM is part of the ISO/IEC 13250 Topic Maps standard that enables to describe and navigate information objects (Biezunski & Newcomb 2001). The main disadvantage is that CMAPS don´t allow a formalization of the knowledge in order to check inconsinstencies and to reason with it. This is the motivator to use the DLs language.

4 The representation of our ontologies in Description Logics

DLs are considered one of the most important knowledge representation formalism unifying and giving a logical basis to the well known traditions of Frame-based systems, Semantic Networks, Object-Oriented




```
<topicMap id="1082994524453_1445937377_486"
 xmlns="http://www.topicmaps.org/xtm/1.0/" …. >
   <topic id="1082994122796_538001187_441">
    <instanceOf>
     <subjectIndicatorRef …..  xlink:href="http://cmap.coginst.uwf.edu/#concept"/>
    </instanceOf>
    <baseName>    <baseNameString><![CDATA[DESTINO1]]></baseNameString>
     </baseName>
   </topic>
    …                                    other 'concept' topic
    <baseName>    <baseNameString><![CDATA[DESTINO2]]></baseNameString>
    …                                    other 'concept' topic
    <baseName>    <baseNameString><![CDATA[ORIGEN]]></baseNameString>
    …
                  xlink:href="http://cmap.coginst.uwf.edu/#linkingPhrase"/>
    …                                    a  'linkingPhrase' topic
              <baseNameString><![CDATA[RELACION ]]></baseNameString>
    …
   <association id="assoc_1082994122812_680129440_444">
    …
                                  xlink:href="http://cmap.coginst.uwf.edu/#incoming"/>
   …                                     → this number is ORIGEN
    <topicRef xlink:type="simple" xlink:href="#1082994112359_1142850275_427"/>
   …
      <subjectIndicatorRef xlink:type="simple"
                  xlink:href="http://cmap.coginst.uwf.edu/#outgoing"/>
   …                                     → this number is DESTINO 1
    <topicRef xlink:type="simple" xlink:href="#1082994122796_538001187_441"/>
  …
                                  xlink:href="http://cmap.coginst.uwf.edu/#outgoing"/>
   …                                     → this number is DESTINO 2
   <topicRef xlink:type="simple" xlink:href="#1082994142328_1528135851_471"/>
```

Fig 4. A conceptual map: graphical representation in CmapTools and in XTM format

representations, Semantic data models, and Type systems. They are characterized by its expressiveness and clearly defined semantics. DLs capture the meaning of the data by concentrating on entities, which are grouped into classes or concepts. The entities are associated by relationships. This intuition is shared by formalisms such as semantic data models, semantic networks or frame systems. More important than the DLs representational characteristics are its reasoning mechanisms. The most important characteristic is the checking of inconsistencies and the organization of the concepts on a taxonomy that the system automatically builds from the concept definitions. This is possible because of the clear and precise semantic of concept definitions that avoid the user to put the concepts in the correct place of the hierarchy. This task is done manually in other systems, such as frame systems, which provide inheritance but not classification.

DL reasoning mechanisms are based on three main functions. The first one is subsumption, to determine whether a description -concept or relation- is more general than another. The second one is instance recognition, to determine the concepts that an individual satisfies and the relations that a tuple of individuals satisfies. The third one is contradiction detection, both for descriptions and assertions about individuals. Subsumption supports classification, i.e., the ability of automatically classifying a new description within a semi-lattice of previously classified descriptions. Instance recognition supports completion, i.e., the ability of drawing logical consequences of assertions about individuals, based on those descriptions they are recognized as instances of.

DLs are first-order logic predicate calculus with ideas from semantic networks that allow hierarchical representation of classes and instantiations of terms and their relationships, called TBox, the terminological box, and assertions over them, called ABox, the assertional box. To build the lightweight ontology is only necessary the TBox.

Regarding KI CBR, our main contribution is the definition of a domain-independent architecture to help in the integration of ontologies for CBR applications. The core of this architecture is CBROnto, an ontology incorporating the common CBR terminology that is used to guide the domain ontologies integration. It is done in a subsequent phase, after the domain modelling phase: the CBR application designer relates the specific domain knowledge with the CBROnto terms. This integration between the CBR terminology and the domain knowledge, allows reusing generic problem solving methods (PSMs) that are defined using the CBR terminology. DLs classification allows that PSMs have access and reason with the domain terms that are classified below the CBR terms.

We use DLs reasoning mechanisms and deductive inferences based on subsumption and instance recognition. Subsumption determines if a term is or not more general than another, and instance recognition finds all the concepts that an individual satisfies. Furthermore, completion mechanisms perform logical consequences like inheritance, combination of restrictions, restriction propagation, contradiction detection, and incoherent term detection. These mechanisms will be used during the ontology integration, the case representation and, in general, as the base for all the CBR processes.

The description of the intensive use of the DLs reasoning mechanisms to help the CBR processes is out of the scoper of this paper. We refer the interested reader to (Díaz-Agudo 2002).We formalize our ontologies, Domain and CBROnto, using OWL as internal representation. For the reasoning tasks we loaded it into RACER

```
<owl:Class rdf:ID="DESTINO1"/>
<owl:Class rdf:ID="ORIGEN1">
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:someValuesFrom>
    <owl:Class rdf:ID="DESTINO2"/>
   </owl:someValuesFrom>
   <owl:onProperty>
    <owl:ObjectProperty rdf:ID="RELACION1"/>
   </owl:onProperty>
  </owl:Restriction>
 </rdfs:subClassOf>
 <rdfs:subClassOf>
  <owl:Restriction>
   <owl:someValuesFrom rdf:resource="#DESTINO1"/>
   <owl:onProperty rdf:resource="#RELACION1"/>
  </owl:Restriction>
 </rdfs:subClassOf>
</owl:Class>
```

Fig. 5. The example transformed into OWL

(Haarslev & Möller 2003). RACER is a DL reasoner that reads OWL.

5 The transformation of our ontologies from CMAPS to Description Logics

The transformation from CMAPS to DLs takes into account the different expressiveness of both formats for our specific problem. We process only the elements that affect to the representation in our system. We describe the process using a simple example. The transformation process is associated to the updating process, that maybe online or offline. To allow the inclusion of existing CMAPS, there is an offline load process of the map into the DL layer. The online process is when the user modifies the ontologies with a graphical interface, like the CMAPS program.

5.1 The offline process: loading existing conceptual maps

We use as basis two facts: CmapTools is able to export, as in Fig.4, the map to the XTM language and the DLs language that uses the reasoning engine is OWL. Since both languages are XML based we use a parser to perform the load process. We use the sort example in CMAPS and XTM of Fig. 4 and its equivalent in OWL of Fig. 5 to describe the relevant elements that are needed for the transformation. We don't need the graphical information of the XTM map; we only use its conceptual elements:

- The *topic* tag that defines the labels and their role e.g.: `<topic id="10829922796_538001187_441">`. They have a unique identifier used to refer to them later in the map. They are instances of two classes:
  - *Concept*, that refers to the concept itself.
  - *LinkingPhrase*, that defines the relationship among concepts.
- The *association* tag that defines the specific relations of a *linkingPhrase* with the *concepts*. The id is the same one of the *linkingPhrase*, e.g.: `<association id="assoc_12_6809440_444">`. It has the role specification of the concepts in the relation to define the direction of the map:
  - *Incoming* concepts are the origin
  - *Outgoing* concepts are the target

We generate the OWL formalization using the same structure that uses the RACER reasoning engine. Following the example in the Fig. 4 and 5 we have:

- A relation in CMAP, i.e.: a *LinkingPhrase* topic, is generated as a property in OWL:

  e.g. `<owl:ObjectProperty rdf:ID="RELACION1"/>`

- A concept in CMAP, i.e.: a *concept* topic, is generated as a class in OWL

  e.g. `<owl:Class rdf:ID="ORIGEN1">`
- A relation in CMAP, i.e.: an *association* tag, is generated in OWL following this process:
  - To connect a relation with the concepts that are part of it, an anonymous class is created with `<rdfs:subClassOf>` inside the origin concept. This way the subclass inherits from the concept. The declaration of the connection is made with a restriction `<owl:Restriction>` with two elements: the target concept, that is a class e.g. `<owl:Class rdf:ID="DESTINO2"/>` and the relation, that is a property, e.g. `<owl:onProperty rdf:resource="#RELACION1"/>`.
  - A conflict appeared because RACER allows unary and binary predicates, and, on the contrary, CMAPS allow relations many to many. This is solved by the creation of one anonymous class for each pair of concepts related. In the Fig. 5 shows the complete example in OWL format.

5.2 Online updates

The are two types of online updates:
- The *terminological updates* with the Tbox of the DLs. The user with the graphical interface makes them in the CMAP layer. In this layer there is not distinction between primitive and defined concepts.
- The *assertive updates* with the Abox. These are restrictions over the *instances*, the cases in the Case Base of our project. This is made in the DLs layer because there is no representation of instances in CMAPS. This is done with an OWL editor, such as Protege 2.0.

In this manner, the transformation problem is centered in the *terminological updates*. These are the steps we take:
1. The user introduces a change in a concept or in a relation (CMAPS layer)
2. The system checks with the DLs reasoning engine (RACER in `*auto-realize*` mode)
3. If it is inconsistent with the current ontology: it displays a message
4. If it is consistent: it includes the modification into the DLs layer
5. The cycle 1 to 4 continues until all the updates are performed.

6 Results

As a test site, we are developing a project for a software company. The company is called GoldenSoft in Spain. It needed a computer support for a very demanding human work of the Department of Client Attention, or, the so-called, Help-Desk. GoldenSoft is one of the main local software companies developing software for the management of small and medium size companies. It sells five complex integrated applications for invoicing, accounting, payroll, teller machines and taxes. The Help-Desk supports the phone calls with questions about these applications. One of the keys of its success is this Help-Desk department with ten technicians that are trained during a year before they are ready to answer calls. They attend two hundred calls a day about complex problems of several kind of topics, mainly program errors, application use, legal or computer user related issues. Some of these questions take even hours to be solved.

In (Gómez-Gauchía et al. 2004 b) we have described the knowledge elicitation phase associated to this project. Although we have tried to automate the process, the experts still has two remaining hard duties:
1. Revise and refine general knowledge about the domain that is automatically extracted from the company documentation.
2. Integrate the domain terminology within the CBROnto terminology in order to allow the reusable methods included in CBROnto can work properly with the domain terminology.

We distinguish between two different stages of the test. During the first one we convince the experts of using Protégé, a DLs graphical editor to help them in the domain knowledge refinement and integration within CBROnto. During the second one we provide with a CMAP based interface to help their tasks. We have performed two polling processes applied to 10 domain experts. We have compared the opinion polls results and they clearly indicate that CMAPS facilitate the visual comprehension of the model. An informal productivity test has been performed showing a reasonable improvement, but a more formal test is needed.

7 Conclusions and further work

Ontology modeling deals with the question on how to describe in a declarative and reusable way the domain information of an application, its relevant vocabulary, and how to constrain the use the data, by understanding what can be drawn from it. In this paper we have described a two layered approach embedded in a methodology

to ontology modeling. The first layer is based on CMAPS to facilitate the visual comprehension of the model and help the experts to define medium to large number of concepts and relations. The second layer is based on DLs. In particular, DLs well founded semantics facilitate the analysis of knowledge, avoidance of incompleteness, inconsistencies and redundant knowledge. We tested this approach in a project with aceptable results.

As a further work we plan to give a software support to the methodology in order to manage the complete cycle of refinenment. It includes authoring, collaboration, validation and versioning tools. Also it will be carry out formal tests with metric polls to evaluate the improvements.

References

Baader, F., Kuesters, R., & Wolter, F, (2002). Extensions to description logics. In F. Baader, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2002.

Biezunski, M., Newcomb, S.R.(2001) XML Topic Maps: Finding Aids for the Web *IEEE Multimedia*, April-June 2001. pp. 104-108

Bechhofer, S., van Harmelen, F., Hendler, J, Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, A., (2004). OWL Web Ontology Language Reference, http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Borgida, A. (1996) On the relative expressiveness of description logics and first order logics. *Artificial Intelligence*, 82:353-367.

Borgida A., & Brachman, R. J. (2002) Conceptual modelling with description logics. In F. Baader, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2002.

Calvanese, D., Lenzerini, M., & Nardi, D..(1998) Description logics for conceptual data modeling. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*. Kluwer.

Díaz Agudo, B.(2002) Una aproximación ontológica al desarrollo de sistemas de razonamiento basado en casos. *PhD Thesis, Dep. Sistemas Informáticos y Programación, U. Complutense*, October 2002.

Díaz-Agudo, B., González-Calero, P.A.(2001) A Declarative Similarity Framework for Knowledge Intensive CBR. In Aha, D., Watson, I., (Eds.): Case-Based Reasoning Research and Development (Procs. of the 4th International Conference on Case-Based Reasoning, ICCBR 2001), Lecture Notes in Artificial Intelligence, 2080, Springer, 2001.

Díaz-Agudo, B., González-Calero, P.A. (2002) CBROnto: a Task/Method ontology for CBR. In (Eds.): *The 15th International FLAIRS 2002 Conference (Special Track on Case-Based Reasoning).* AAAI Press.

Donini, F., (2002) Complexity of reasoning. In F. Baader, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2002.

Gómez Gauchía, H., Díaz Agudo, B., González Calero, P.A. (2004 a) Towards a pragmatic methodology to build lightweight ontologies: a case study. *In Procs. of the IADIS International Conference, Applied Computing 2004, Lisboa, Portugal* ISBN: 972-98947-3-6.

Gómez Gauchía, H., Díaz Agudo, B., González Calero, P.A. (2004 b) A case study of structure processing to generate a case base. *In 7th European Conference in Case Based Reasoning, Madrid, España, Septiembre 2004* Springer-Verlag LNCS/LNAI

Haarslev, V., & Möller, R., (2003). RACER User s Guide and Reference Manual Version 1.7.7, Concordia University and Univ. of Appl. Sciences in Wedel

Hartley, R.T. and Barnden, J.A (1997). Semantic Networks: Visualizations of Knowledge. Trends in Cognitive Science, 1(5), pp 169-175, 1997.

Leake, D. B., Wilson, D. C. (2001) A Case-Based Framework for Interactive Capture and Reuse of Design Knowledge. *Applied. Intelligence. 14*(1): 77-94.

Ogden, C.K, Richards, I.A. (1923) *The Meaning of Meaning: A Study of the Influence of Language Upon Thought and of the Science of Symbolism*, 8th ed., Harcourt Brace, New York, 1946.