

# Reusing the Mikrokosmos Ontology for Concept-based Multilingual Terminology Databases

Antonio Moreno, Chantal Pérez

Universidad de Málaga  
F. Filosofía y Letras, Campus de Teatinos  
29071 Málaga, Spain  
amo@uma.es

## Abstract

This paper reports work carried out within a multilingual terminology project (OncoTerm) in which the Mikrokosmos ( $\mu K$ ) ontology (Mahesh, 1996; Viegas et al 1999) has been used as a language independent conceptual structure to achieve a truly concept-based terminology database (termbase, for short). The original ontology, containing nearly 4,700 concepts and available in Lisp-like format (January 1997 version), was first converted into a set of tables in a relational database. A specific software tool was developed in order to edit and browse this resource. This tool has now been integrated within a termbase editor and released under the name of OntoTerm™. In this paper we focus on the suitability of the  $\mu K$  ontology for the representation of domain-specific knowledge and its associated lexical items.

## Introduction

The principal objective of the project *ONCOTERM* is the elaboration of a bilingual (English-Spanish) on-line information system specific to the medical subdomain of oncology. Its long-term objective is to deploy a wide range of information resources accessible through the World Wide Web to patients and their families, to professional medical translators, as well as to those who work in the public health system.

Within the framework of this project, two main resources have been developed:

1. a concept-based multilingual terminology database (termbase),
2. domain-specific corpora for both English and Spanish for information extraction.

The central component of the multilingual termbase is the representation of the conceptual structure of the domain of oncology, its relevant concepts, their defining characteristics and the relations holding between concepts. This conceptual structure serves as a link among terms in both languages.

## The need for domain knowledge representation

It is widely acknowledged that terminological work should be concept-based. Indeed, the onomasiological approach (i.e. starting from the concepts to arrive at their linguistic designation) is one of the main features in which lexicographical and terminographical entry compilation differ (Wright & Budin 1997: 328).

Systematic terminology management begins with the terminographer's acquisition of domain knowledge and the study and organization of the conceptual structure of that domain. The terminographer will then study the way in which concepts in the knowledge structure are represented by subject-specific lexical items, i.e. terms.

However, although the final results of terminology are *based* on knowledge, they are not systematically *encoded* as knowledge (Meyer, Eck & Skuce 1997). Thus, nearly all the subject-specific knowledge acquired by the terminographer is not explicitly represented in the final terminological product. This situation poses two main problems:

- Translators, technical writers, subject-field learners, information retrieval researchers, and even subject-field specialists, would benefit greatly from accessing those knowledge structures, to assist them in encoding, decoding or classifying specialised documentation.
- It hinders the reuse of valuable terminological products in other NLP applications.

One possible solution to these problems is the integration of knowledge representation techniques and formalisms borrowed from the field of Knowledge Engineering (Obermeier, 1989). Indeed, there is a long tradition of knowledge management and knowledge base construction in AI, which we could use for the construction of termbases, in which conceptual information is explicitly represented, thus acquiring the rank of *concept-based termbases* or *terminological knowledge bases* (in the sense used in Meyer, Eck & Skuce, 1997).

From the terminologist's standpoint, the representation system to be used will have to offer at least the following features to cater for their particular needs and working methods:

- Systematic representation of conceptual information associated to terms: concepts, their characteristics and the relations among concepts.
- An explicit and formal representation system that allows reuse, and is relatively simple to work with, as terminologists are *not* knowledge engineers.
- Facilitating the representation of multidimensionality in concepts (Bowker, 1997).
- Representation of interdisciplinary relations, as concepts in a domain are likely to be linked to others belonging to related domains.
- Reusability, guaranteed by constructing it as a stand-alone application, linked to one or more termbases, but independent in itself.

To date, only the Cogniterm project (Meyer et al., 1992) has used knowledge-based tools (CODE) for the representation of terminological information. Our approach is different in many ways, but especially in that it relies on a previously existent resource for knowledge representation, a full-fledged ontology of concepts, extending the information contained in this resource to meet our domain-specific needs.

## Ontologies in terminology

An ontology, in the sense used here, is a shared and common understanding of some domain that can be communicated across people and computers (Gruber, 1993; Guarino, 1995). The fact that ontologies can be shared and reused among different applications accounts for the interest these resources have aroused in recent years.

Following the already standard definition proposed by Gruber (1993), an ontology is understood as a formal, explicit specification of a shared conceptualization. "Conceptualization" here refers to an abstract model of some phenomenon in the world and the identification of the relevant concepts of that phenomenon. "Explicit" means that the type of concepts used and the constraints on their use are explicitly defined. "Formal" reflects the fact that the knowledge contained in the ontology should be represented by means of a formal representation language. Finally, "shared" accounts for the notion that an ontology should reflect consensual knowledge. Steve et al. (1998) distinguish three main types of ontologies:

- *Domain ontologies* concern specialized knowledge in a domain or subdomain. Most domain ontologies have been defined as part of a knowledge-based system development within larger projects.
- *Generic ontologies* concern general, foundational aspects of knowledge processes, part/whole structures, kinds of objects, etc.
- *Representation ontologies* specify the conceptualizations that underlie a knowledge representation formalism. A representation ontology is also considered a meta-ontology, as it defines "meta-level" categories.

Taken in isolation, none of these three types of ontologies can cover the knowledge representation needs of terminographers as described above. A domain ontology may be too restrictive to account for interdisciplinary concept relations; generic and representation ontologies may not contain concepts that are specific enough to represent domain-specific knowledge. What is required is an ontology that can combine characteristics taken from these three types. Such a compromise solution can be found in the Mikrokosmos concept of "situated ontology": a self-contained world model used as a computational resource created for solving a particular set of problems (Mahesh 1996).

### Reusing the Mikrokosmos ontology for the representation of the cancer domain

Mikrokosmos ( $\mu K$ ) is a Knowledge-Based Machine Translation (KBMT) system developed at New Mexico State University jointly with the US Department of Defense.  $\mu K$  is a large-scale MT system that has focused on translating Spanish and (recently) Chinese texts about company mergers and acquisitions into English (Viegas et al., 1999).

The ontology plays a central role in the  $\mu K$  MT system as it provides a language-independent knowledge source in the form of a highly structured set of concepts for representing word or phrase meaning for a source or target language. An analysis lexicon maps words in the lexicons to concepts in the ontology. An interlingual meaning representation (called TMR *Text Meaning Representation*) is then derived from the representations of word meanings

in the lexicons and from the representation of world knowledge in the ontology.

Although the  $\mu K$  ontology has been available for some years now<sup>1</sup>, to date there have been no other initiatives aiming to reuse this extremely valuable resource. This may be due to the fact that the original file is in Lisp-like format and it would take a system such as the original  $\mu K$  toolkit in order to take advantage of it. However, our approach has been to convert the original file into a relational database semi-automatically, by applying a set of regular expression replacements.

The E-R diagram shown in Figure 1 is a somewhat simplified version of the conceptual model we have used in order to capture (most of) the static knowledge encoded in the  $\mu K$  ontology, and therefore the one used for new ontology files created by OntoTerm (see below).

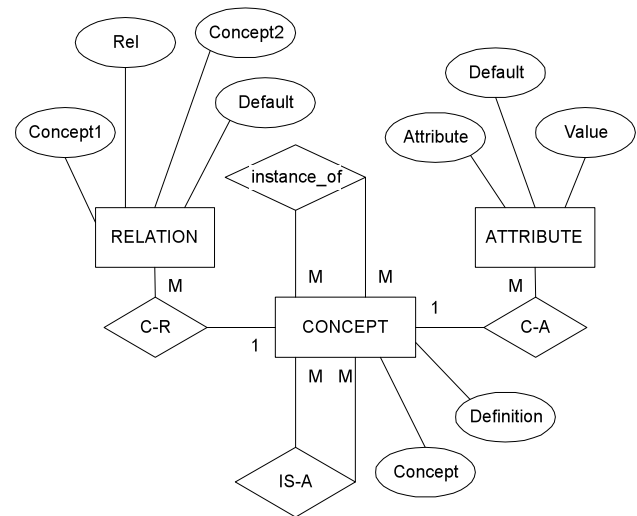


Figure 1: E-R diagram of ontology files

Needless to say, a relational database can hardly be said to serve any purpose by itself, as all editing operations would be virtually unmanageable. This is why we have developed a full-fledged application that allows coherent editing, easy browsing, and many added functions, some of which are described below.

### OntoTerm™: a concept-based multilingual terminology management system

OntoTerm is the application that we have developed in order to provide support to the framework we have just described. Its overall architecture may be said to stand on two primary modules: the Ontology Editor, where ontology construction is carried out, and the Termbase Editor, where lexical mappings and term description take place. A number of other tools facilitate browsing, navigating, and reporting the ontology.

A fundamental design principle has been to isolate the user from the application's internal operations. Thus, users can concentrate on the construction of the conceptual structures relevant for their domain, rather than on how this knowledge is encoded.

OntoTerm runs under MS Windows 9x/NT/2000,

<sup>1</sup> Detailed information the Mikrokosmos project can be found at <http://crl.nmsu.edu/Research/Projects/mikro/index.html>, as well as an on-line browsable version of the ontology.

requiring the usual hardware for this platform's applications. An evaluation version is now available for download from the OntoTerm web site (<http://www.ontoterm.com/>).

## The Ontology Editor

This can be said to be the main component of the application. The first step to create a new termbase is the construction of an ontology, as no term can exist without being mapped onto a previously defined concept. However, an ontology need not be fully developed in order to start terminological work, as the user may choose to perform both tasks concurrently, mapping terms in the Termbase Editor to concepts defined in the Ontology Editor as they are entered. This is possible because, although termbase files exist independently of ontology files, the former have information about the location of the latter and, conversely, an ontology has information about all termbases that are using it. Whenever a concept is added, deleted or modified in the Ontology Editor, all termbases assigned to this ontology are updated accordingly.

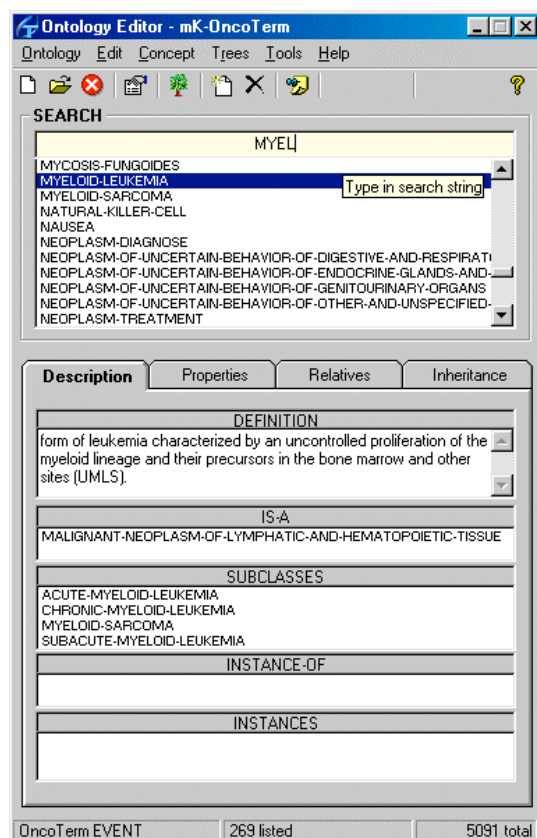


Figure 2: The Ontology Editor

Creating a new termbase consists simply of providing a name for the ontology file and a logical name for the new ontology. OntoTerm will automatically add the obligatory top-level nodes (ALL (EVENT, OBJECT, PROPERTY (ATTRIBUTE, RELATION))) plus some meta-level categories. Basic editing operations with concepts (add, define, delete, move) are all carried out using the graphical user interface. The system takes care that internal consistency is not broken by any of these operations; for example, moving a concept to a new location in the ontology

consists of two steps: first, a new IS-A link (the new location) must be added (by right-clicking on the IS-A link list-box (see figure 2) and selecting "Add IS-A link" from the pop-up menu), making the concept a child of two parent concepts; next the original IS-A link is removed (again, by right-clicking on the IS-A link list-box and selecting "Delete IS-A link"), thus no concept can ever become an "orphan". Consistency is guaranteed at all times by disabling options and performing cascade operations. For instance, deleting a concept will cause OntoTerm to cascade-delete all child concepts for which no other parent has been specified.

Because OntoTerm considers an ontology as a reusable resource, i.e. a repository of structured knowledge that is able to support a number of lexical resources, concepts can be marked to belong to a specific project or subset. The user can then choose to list only those concepts belonging to a specific subset. The status bar at the bottom of the Ontology Editor has three panels that display, from left to right, the subset that the selected concept belongs to and the type of concept (OBJECT, EVENT, ATTRIBUTE or RELATION), the number of concepts listed (i.e. the selected subset) and the total number of concepts in the ontology (see figure 2).

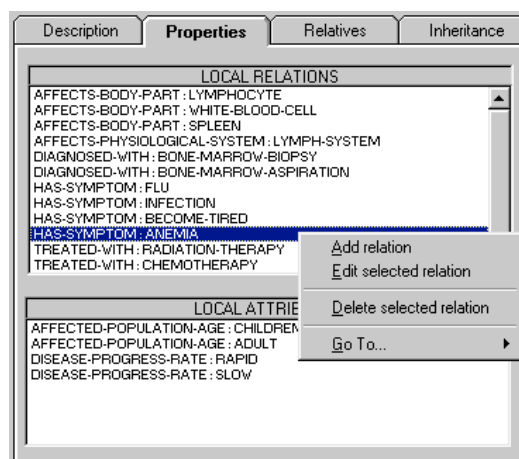


Figure 3: Local properties tab

Concepts are further defined by specifying their local properties (i.e. relations and attributes) in the "properties" tab (see figure 3). The other two tabs are used for showing the concept's ancestors and descendants ("Relatives" tab) and inherited properties (see figure 4). Inherited properties are shown together with the concept from which they are inherited, as well as the distance between the selected concept and the concept from which that property is inherited. Inheritance is usually turned off for faster browsing. The option buttons at the bottom allow the user to view inherited properties either in full mode or in override mode (i.e. properties with the same attribute:value pair will be overridden by those closer to the selected concept).

As figure 4 shows, properties are being inherited not only from concepts in the OncoTerm subset, but also from concepts further up in the hierarchy, i.e. the Mikrokosmos subset. In the following sections we describe what the relevant "ontology hooks" (i.e. parent concepts) have been for the OncoTerm subset.

Access to displayed concepts is made easy by using

context-sensitive menus (see figure 3). However, the main navigational tool is the “trees” feature. Apart from the global ontology tree, users can have any number of partial trees open at any one time, allowing them to concentrate on the section of the ontology relevant for the work at hand (see figure 6).

Description		Properties	Relatives	Inheritance
Attribute	Value	From	Distance	
AGENT	BACTERIA	DISEASE-EVE...	3	
AGENT	ENERGY	PHYSICAL-EV...	7	
AGENT	FORCE	PHYSICAL-EV...	7	
AGENT	PHYSICAL-OBJ...	PHYSICAL-EV...	7	
CAUSED-BY	CARCINOGEN	HAVE-NEOPLA...	2	
CAUSED-BY	TOBACCO	HAVE-NEOPLA...	2	
CAUSED-BY	X-RAY	HAVE-NEOPLA...	2	
DIRECTION-O...	NEGATIVE	CHANGE-STATE	6	
DIRECTION-O...	POSITIVE	CHANGE-STATE	6	
LOCATION	PLACE	PHYSICAL-EV...	7	
THEME	ANIMAL	DISEASE-EVE...	3	
THEME	OBJECT	CHANGE-STATE	6	
TREATED-WI...	MEDICAL-ARTI...	HAVE-NEOPLA...	2	
TREATED-WI...	PHARMACEUT...	HAVE-NEOPLA...	2	

Show:  Off  Override  Full  Distance

Figure 4: Inheritance tab

Another powerful browsing tool is the Ontology Navigator, which is a two-paned window with the full concept hierarchy on the left-hand side (in a tree-view control). Clicking on any of the nodes in the tree will cause OntoTerm to generate an HTML file containing all the information (both local and inherited) and display it on the built-in web browser (right-hand side pane), where all concepts and relations are actually hyperlinks to the HTML files that describe them. This tool can also be used to effectively publish the whole ontology in HTML format, during which process a framed index will also be automatically generated. All these files are stored on a separate directory named “OntologyName-publish”; publishing the ontology on the web is then straightforward, as the main index file is named “index.html”.

### The Termbase Editor

The Termbase Editor is where lexical mappings and term description take place. Like the Ontology Editor, a new termbase is created just by providing a file name and a logical name for it. However, the Ontology Editor must be running and an ontology open in order to be able to create a new termbase, this forces the new termbase to be assigned to the current ontology, which becomes the termbase’s “working ontology”.

The conceptual schema used by OntoTerm’s termbases is based on the Reltef™ specification (Hardman, 2000) and implements the ISO 12620 data categories (ISO FDIS 12620), following the guidelines provided by the CLS Framework (Melby & Wright 2000; Wright, in press) for termbase design.

OntoTerm’s implementation of these standards has been geared towards making it easy for the terminologist to come to grips with this very comprehensive, but equally complex, framework for terminological work. OntoTerm accomplishes this by means of a very innovative user interface that guides the terminographer in the term description process in various ways.

According to the CLS Framework, a termbase consists of global information about itself (the header), a number of TermEntries (the body) and a set of SharedRefs and links (the backmatter). TermEntries correspond in fact with concepts, and therefore are taken to be the actual links with the concepts in the ontology, from which they acquire their ontological status. They also have their own internal structure which consists of a number of allowed data categories, just like any other data category. In fact, the CLS Framework treats all elements in the same way; they are differentiated by the type of value they take (text, picklist, date, etc.) and the set of data categories by which that element is liable to be further defined.

This obviously provides immense expressive power, but it also makes it very difficult for the terminographer to be aware of the possibilities granted and restrictions imposed by the framework.

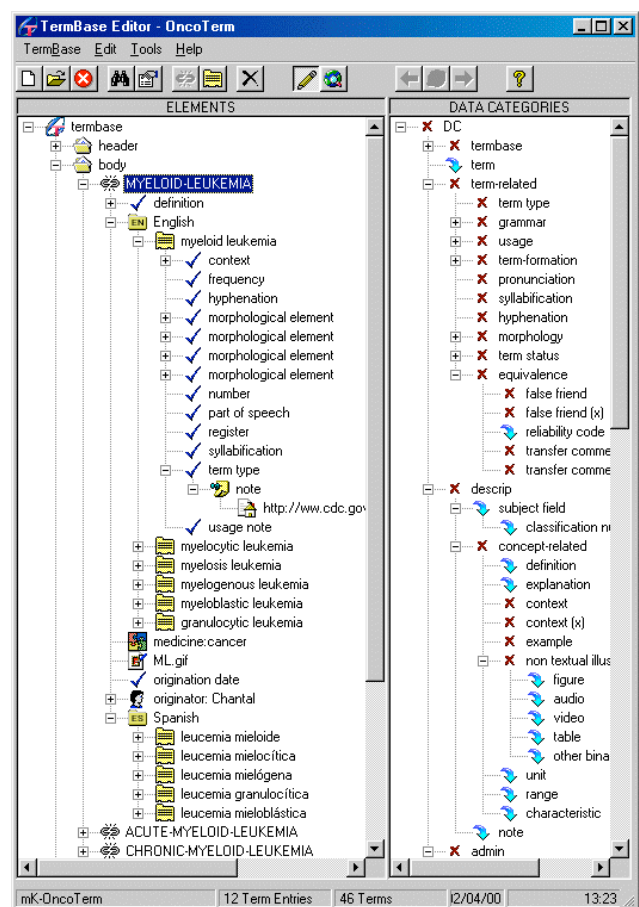


Figure 5: the Termbase Editor

Figure 5 may help to illustrate how OntoTerm manages to deploy this expressive power in a user-friendly way. The left tree shows the current structure of the termbase. Elements can be expanded or contracted as needed and different icons are assigned to each element type to make it easy for the user to identify them visually. The tree on the right-hand side is actually a hierarchical representation of the ISO 12620 data categories. Clicking on any element of the termbase (left tree) will cause the data categories tree to show those categories available to describe the selected element, marking them with a blue arrow pointing downwards, and barring access to disallowed categories (those with a red cross icon). Entering



information for any allowed category implies clicking the desired category on the right-hand side tree. This will bring up the edit frame at the bottom of the elements tree, which will show different controls depending on the type of data that this category selects (a text-box for free text data, a list-box for pick-list elements, a masked-edit box for date elements, or a slider for numeric range values). Thus, all editing and browsing functions are performed in this highly polymorphic component.

The Termbase Editor has a large number of features whose proper description would exceed the limits of this paper. To name some of the most important:

- Cross-references can be specified graphically by clicking the elements in the tree and defining a descriptive label. Data categories that have cross-references as values (e.g. homonyms and synonyms) are specified likewise, only no label is required.
- It allows the inclusion of multimedia elements such as graphics, audio, video or any other binary data file.
- It has a browse mode in which it generates HTML pages on-the-fly that integrate both ontological and terminological information (with a number of options). These pages, just like any isolated multimedia element, can be viewed using the built-in web browser.
- The HTML Report Generator permits publishing the termbase in full or a subset of it.

### Integrating domain-specific terms in the $\mu K$ ontology

Aided by the editing and browsing facilities described above, we set out to enlarge and adapt the conceptual information contained in the  $\mu K$  ontology. From the beginning, it was our intention not to modify or delete existing concepts, unless that seemed strictly necessary, but rather to find the “ontology hooks” relevant to our specific domain (cancer), together with those necessary for the description of these concepts. From these “ontology hooks” we would then enlarge the ontology by integrating our domain-specific concepts and enriching the system of relations and properties already present in  $\mu K$ . As in the  $\mu K$  project, we have tried to follow a set of guidelines for the inclusion of new concepts and for the specification of relations among them, which are very much in accordance with those of  $\mu K$ .

The integration of cancer concepts in  $\mu K$  ontology started by completing and enlarging the section of the hierarchy of one of the concepts already present in the ontology: DISEASE-EVENT. This concept was originally a child of LIVING-EVENT and, to enrich its inherited relations and attributes, we made it a child of CHANGE-STATE-ANIMAL. Within the DISEASE-EVENT branch, we kept all the  $\mu K$  concepts already present, such as COUGH, HAVE-VIRUS, HAVE-FEVER and HAVE-CANCER, as they proved to be very useful for the description of other concepts. HAVE-CANCER was changed into the more subject-specific HAVE-NEOPLASM. In this point of the ontology we included a very comprehensive hierarchical classification of neoplasm types and subtypes extracted from the ICD-9 CM (*International Classification of Diseases*) issued by the *World Health Organization*. Figure 6 illustrates the DISEASE EVENT branch of the ontology and some of the domain-specific concepts added.

Other concepts added in this section of the ontology correspond to other DISEASE-EVENTS not included in the ontology but essential for the representation of relations pertaining to cancer type concepts. Some of these concepts are ANAPLASIA, FLU, ANEMIA, ANOREXIA, CONSTIPATION, INFECTION, NAUSEA, and DIARRHOEA, all of them very relevant as they identify SYMPTOMS of a particular disease or SIDE-EFFECTS of a TREATMENT.

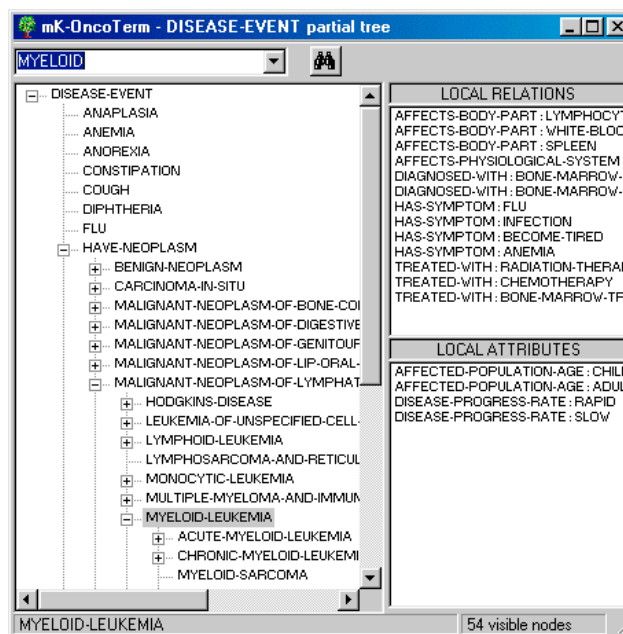


Figure 6: DISEASE-EVENT partial tree

The concept TREATMENT has also been added to the ontology as a subject-specific concept, included as a child of the  $\mu K$  concept MEDICAL-SERVE. Child concepts of NEOPLASM-TREATMENT are, among others, different types of BONE-MARROW-TRANSPLANT, different types of cancer therapy, such as CRYOTHERAPY, DIFFERENTIATION-THERAPY, LASER-THERAPY, NEUTRON-BEAM-THERAPY, and different types of CHEMOTHERAPY, RADIATION THERAPY, and other SUPPORTIVE-TREATMENTS.

Many other  $\mu K$  concepts were useful as parents of our cancer-related concepts. Thus, for example, knowledge related to drugs used in cancer treatment has been included under the  $\mu K$  concept PHARMACEUTICAL-MATERIAL. Different types of professionals who care for cancer patients (medical oncologist, oncologic surgeon, pediatric surgeon, radiation oncologist, etc.) have been included under the generic  $\mu K$  concept ONCOLOGIST, a child concept of DOCTOR (child, in turn, of MEDICAL-ROLE). In some other cases, finding the appropriate “ontology hook” to fit the domain-specific information was not so straightforward. For example, knowledge relating to B-CELLS and T-CELLS had to be included in the ontology, as they play an important role in the body’s immune response. To include such specific concepts, we had to add a number of other in-between concepts, such as LYMPHOBLAST and LYMPHOCYTE and further complete other parts of the ontology, including concepts such as IMMUNE-SYSTEM and LYMPH-SYSTEM and their subordinates.

One feature of the  $\mu K$  ontology which makes it particularly useful for the representation of our cancer

concepts is the possibility of including RELATIONS -and their INVERSE-RELATIONS- and ATTRIBUTES (divided into LITERAL-ATTRIBUTES and SCALAR-ATTRIBUTES) as concepts. This has allowed us to use the rich classification of relations already included in the ontology, which has been complemented by some others essential for the conceptual characterization of cancer-related terms. The domain-specific relations and attributes added so far to the ontology include the following:

DISEASE-EVENT RELATION	INVERSE-DISEASE-EVENT-RELATION
ADMINISTERED-BY	USED-FOR-ADMINISTERING
AFFECTS-BODY-PART	BODY-PART-AFFECTED-BY
AFFECTS-PHYSIOLOGICAL-SYSTEM	PHYSIOL.-SYSTEM-AFFECTED-BY
DIAGNOSED-WITH	DIAGNOSTIC-TEST- FOR
HAS-RISK-FACTOR	RISK-FACTOR-FOR
HAS-SIDE-EFFECT	SIDE-EFFECT-OF
HAS-SYMPTOM	SYMPTOM-FOR
TREATED-WITH	TREATMENT-FOR

LITERAL-DISEASE-EVENT-ATTRIBUTE	SCALAR-DISEASE-EVENT-ATTRIBUTE
AFFECTED-POPULATION-AGE	RECOVERY-RATE
DISEASE-PROGRESS-RATE	RELATIVE-FREQUENCY-OF-OCCURENCE
PROGNOSIS	

These relations and attributes are used, for example, to specify that one particular type of cancer, ACUTE-MYELOID-LEUKEMIA is characterized by the following local relations:

AFFECTS-BODY-PART:[WHITE-BLOOD-CELL; SPLEEN]  
 AFFECTS-PHYSIOLOGICAL-SYSTEM:LYMPH-SYSTEM  
 DIAGNOSED-WITH:BONE-MARROW-BIOPSY  
 HAS-RISK-FACTOR:[NUCLEAR-ENERGY; CHEMOTHERAPY; BENZENE]  
 HAS-SYMPTOM:[HAVE-FEVER; ANEMIA; BECOME-TIRED]  
 TREATED-WITH:[CHEMOTHERAPY;BONE-MARROW-TRANSPLANT]

and the following local attributes:

AFFECTED-POPULATION-AGE:ADULT  
 DISEASE-PROGRESS-RATE:RAPID

It must be noticed that both slots and fillers of relations are concepts themselves (with their own properties), which gives an idea of the complexity involved in the process of structuring and describing the conceptual map of the ontology, as well as its great expressive power and consistency.

## Conclusion

Our experience shows that if we want to achieve a formal encoding of specialized knowledge that can then be reused for other NL applications, it is not enough to build a domain-specific ontology, because specific concepts do hold a number of relevant relations with general-knowledge concepts, which must be expressed and accounted for.

The availability an existing, comprehensive, general ontology such as  $\mu K$  has been key to achieving a complete encoding of knowledge related to such a complex and far-reaching domain as cancer, its aetiology, morphology and treatment. The overheads of building a customized software tool and of converting the available Lisp-like source code into a customized relational format have proved to be cost-effective, as they easily compensate for

the difficulty and complexity of building a general-purpose ontology from scratch.

## Acknowledgements

This research has been partly carried out within the framework of the project *OncoTerm: System of oncological information and resources*, funded by the Spanish Ministry of Education (DGICYT) under code number PB98-1342.

## References

- Bowker, L. (1997) Multidimensional Classification of Concepts and Terms. In S. E. Wright & G. Budin (Eds.) (pp. 133--143).
- Gruber, T.R. (1993). A Translation Approach to Portable Ontologies. *Knowledge Acquisition*, 5(2), 199--220.
- Guarino, N. (1995) "Formal Ontology, Concept Analysis and Knowledge Representation", *International Journal of Human and Computer Studies*. 43 (5/6), 625--640.
- Hardman, D. (2000) The Reltef™ Specification. <http://www.ttt.org/clsframe/reltef/>
- ISO FDIS 12620:1995. Terminology--computer applications--data categories. Geneva: ISO.
- Mahesh, K. & Nirenburg, S. (1995). A Situated Ontology for Practical NLP. In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing. International Joint Conference on Artificial Intelligence (IJCAI-95), August 1995. Montreal, Canada.
- Mahesh, K. (1996). Ontology Development for Machine Translation: Ideology and Methodology. NMSU. Computing Research Laboratory. Technical Report MCCS-96-292. New Mexico.
- Melby, A.K. & Wright, S.E. (2000). The CLS Framework. <http://www.ttt.org/clsframe/index.html>
- Meyer, I., Skuce, D., Bowker, L. & Eck, K. (1992). Towards a New Generation of Terminological Resources: An Experiment in Building a Terminological Knowledge Base. In COLING-92: Proceedings of the 15th International Conference on Computational Linguistics (pp.129--138).
- Meyer, I., K. Eck & Skuce, D. (1997). Systematic Concept Analysis within a Knowledge-Based Approach to Terminology. In Wright, S. & G. Budin (Eds.) (pp. 98--118).
- Obermeier, K.K. (1989). Natural Language Processing Technologies in Artificial Intelligence. Chichester, West Sussex: Ellis Horwood Limited
- Steve, G, Gangemi, A. & Pisanelli, D. (1998). Integrating Medical Terminologies with ONIONS Methodology. Document available in <http://saussure.irmkant.rm.cnr.it>.
- Viegas, E., Mahesh, K., Nirenburg, S. & Beale, S. (1999). Semantics in Action. In P. Saint-Dizier, (Ed.) *Predicative Forms in Natural Language and Lexical Knowledge Bases* (pp. 171--204). Dordrecht, the Netherlands: Kluwer Academic Publishers.
- Wright, S.E. & Budin, G. (Eds.) (1997). *Handbook of Terminology Management*, Vol. 1. Amsterdam: John Benjamins Publishing Company.
- Wright, S.E. (in press) "Data Categories for Terminology Management", in S.E. Wright & G. Budin (Eds.) *Handbook of Terminology Management*. Vol. II. Amsterdam/Philadelphia: John Benjamins Publishing Company.