Experiences in Reusing Knowledge Sources

using Protégé and Prompt

M. Taboada (1), D. Martínez (2) and J. Mira (3)

(1) Department of Electronics and Computer Science. University of Santiago de Compostela. 15782 Santiago de Compostela. Spain. E-mail: chus@dec.usc.es

(2) Department of Applied Physics. University of Santiago de Compostela. Spain. E-mail: fadiego@usc.es

(3) Department of Artificial Intelligence. UNED. Madrid. Spain. E-mail: jmira@dia.uned.es

Author to contact: Maria Taboada. E-mail: chus@dec.usc.es


Telephone number: 34-981-563100

Fax number: 34-981-528012
Abstract

We study the general question of how ontologies and reference terminologies can be used to make development of knowledge bases more manageable, taking into account the methodologies and tools available nowadays. For this, we have carried out a case study on designing a knowledge base oriented to support a diagnosis-aid application in ophthalmology. Ideally, starting from a pre-existing domain ontology, development of knowledge bases is centred only on collecting specific knowledge for a particular application. In practice, this is a very time-consuming approach, as ontology repositories do not usually provide many information-seeking facilities. In addition, it is unlikely to find an ontology that includes all the required knowledge. Consequently, design of knowledge bases requires the combination and adaptation of one or more source ontologies. In this work, particular attention is paid to the proper merging of two ontologies using the tool PROMPT. Our study emphasizes the advantages of using PROMPT for merging ontologies containing closely related portions of knowledge, as well as some proposals for improvement. In a second step, our approach extends the evolving ontology, with a new component that holds both a meta-model representing a very simplified structure of a terminology system into Protégé-2000 and a set of constraints expressed using the Protégé Axiom Language. This set of constraints allows us to check the consistency and coherence of the imported information. Defining meta-classes in Protégé-2000 links this component to the rest of the models in the knowledge base. We report our experience in the reuse of several knowledge sources using Protégé-2000 and several of the plug-ins.
1. Introduction

*Ontologies* are explicit and formal representations of conceptualisations (Gruber, 1993). If these conceptualisations are specific for particular domains, then they are known as *domain ontologies* (van Heijst et al., 1997). Domain ontologies provide explicit and formal descriptions of concepts in a domain of discourse, their properties, relationships among concepts and axioms (Guarino, 1995). From the beginning of the Nineties, many ontologies have been designed in different domains, such as Mathematics (Gruber and Olsen, 1994), Medicine (Falasconi and Stefanelli, 1994) or Chemistry (Fernandez et al., 1999). Just a few years later, we can directly access various ontology libraries available in the Web. Some examples are the library of Ontolingua ontologies\(^1\), the library of DAML+OIL ontologies\(^2\), or the ON9.3 biomedical core ontology\(^3\). So, many researchers feel that access to ontology libraries would facilitate development of knowledge bases. Starting from pre-existing domain ontology, development of knowledge bases would be centred only on collecting specific knowledge (Musen, 1998). In addition, the use of a formal ontology would (1) clarify the semantics of representation, giving rise to knowledge bases endowed with a common and standard terminology, and (2) improve the sharing and reuse of the resulting knowledge bases (Chandrasekaran et al., 1999). In practice, this design approach is a very *time-consuming process*, as it implies a search for portions of required information and, in general, ontology repositories do not normally provide many information-seeking facilities (Li et al., 2000). Because of this, a lot of time is spent finding the required portions of

\(^1\) The library of Ontolingua ontologies. http://www.ksl.stanford.edu/software/ontolingua


\(^3\) The ON9.3 biomedical core ontology. http://ontology.ip.rm.cnr.it.
knowledge. On the other hand, it is unlikely to find an ontology that includes all the knowledge required for structuring a knowledge base. Consequently, its design requires the combination and adaptation of one or more ontologies (Uschold et al., 1998).

Reference terminologies are another type of knowledge that has been suggested as a candidate for reuse. In the health care domain and the biomedical sciences, large portions of terminological knowledge are available in electronic form in controlled terminology and classification systems, such as the Unified Medical Language System, UMLS\(^4\) (Lindberg et al., 1993). These systems supply standard knowledge sources that favour the later share and reuse of the resulting knowledge bases (Li et al., 2000; Schulz and Hahn, 2001). The idea of combining the massive information provided by unified terminology systems with the expressiveness of the knowledge representation languages was promoted by the ONIONS project (Gangemi et al., 1999). Currently, a lot of similar approaches can be found in the literature and we distinguish two types: (1) those focussing on converting large portions of data from a terminological system to some rigorous knowledge representation formalism, such as description logics (Schulz and Hahn, 2001), frame-based representation (Yeh et al., 2003) or an object-oriented database representation (Gu et al., 2000) and (2) those searching for smaller portions of information and further importing and reorganising it into the evolving knowledge base (Achour et al., 2001). For the first type of approaches, one of the most promising uses is in the domain of text understanding, where formal medical reasoning must be performed on massive data (Schulz and Hahn, 2001). On the other hand, the aim of the second type is to design knowledge bases in smaller domains and scale.

In this paper, we study the general question of how ontologies and reference terminologies can be used to make development of knowledge bases more manageable, taking into account the methodologies and tools available nowadays. In this context, we address a number of relevant research questions. We consider the way in which the process of developing knowledge bases needs to be organised in order to reuse domain ontologies and reference terminologies. Basically, there are two ways of reusing ontologies. Ideally, they can be selected directly from an ontology library but, in most cases, they need to be combined and adapted from several ontologies. This second option is addressed in this paper. For this, we have carried out a case study on reusing medical domain ontologies in the development of a knowledge base for an application oriented towards medical diagnosis in ophthalmology. Regarding this subject, the main contribution of this paper is to show (1) that is possible to combine, group and adapt different portions of two or more domain ontologies in order to develop the core ontology of the knowledge base; (2) that this process must be divided into a number of activities, including knowledge extraction, ontology merging and alignment; (3) that each activity in this process can be simplified using tools oriented to locate similarities and differences between ontologies. For the purpose of our case study, a choice had to be made of the tools to be used. We have chosen the tool Protégé-2000 (Gennari et al., 2002), which is an ontology-design and knowledge acquisition tool, and the tool PROMPT (Noy and Musen, 2000), which is oriented to merge, extract and compare ontologies. The justification of this choice is further discussed in a later section. We also discuss the benefits of using these methodologies and tools, as well as the difficulties we have found.
In many approaches, reference terminologies are embedded in an evolving knowledge base. This is a consequence of transforming large portions of information directly from a terminological system to the evolving knowledge base. However, in our approach we started from a core ontology, which includes several models, such as a temporal model, a patient case model and a medical domain model. Initially, all these models were developed as independent models and later reconciled to form a coherent ontology. To avoid impairing the independence of these models and favouring the later reuse of the resulting knowledge base, our approach extends the core ontology with a new component, which holds (1) a meta-model representing a simplified structure of the UMLS into Protégé-2000 and (2) a set of constraints expressed using the Protégé Axiom Language (based on first order logic) oriented to check the consistency and coherence of the imported information during the development and maintenance of the knowledge base. Defining meta-classes in Protégé-2000 links this component to the rest of the models in the core ontology.

This paper is organised as follows. Firstly, we give some information about requirements of our knowledge base in Section 2. Then, the types of activities that should be performed during the core ontology design are analysed in Section 3. Section 4 contains an in-depth description of the component defining the terminological information in the knowledge base. In addition, in these two sections, we discuss the benefits and difficulties we have found when we used Protégé-2000 and PROMPT. Finally, in section 5, we conclude by discussing some implications of our work, taking account of other approaches and prospects of future work.
2. Requirements of the knowledge base

We started the development of the knowledge base by defining its purpose, level of formality, scope and initial strategy for acquisition of knowledge. The purpose of the knowledge base is to support a knowledge-based application oriented towards medical diagnosis in the ophthalmologic domain of conjunctivitis. The level of formality includes the modelling of the knowledge base using PROTÉGÉ-2000 and the implementation of the knowledge base using the development tool KAPPA-PC from IntelliCorp. In both tools, knowledge representation is based on frames, so the knowledge translation between them is relatively direct.

The scope of the knowledge base is the representation of knowledge related to the ophthalmologic problem known as conjunctivitis, which is an inflammation of the conjunctiva. It groups a number of diseases or disorders that mainly affect the conjunctiva. In most of the patients, the conjunctivitis remits on its own, but in some cases, it progresses and can cause serious ocular and extra-ocular complications. The strategy we have followed, for the efficient acquisition of knowledge, was centred on the revision of a text provided by a medical guideline, with the aid of a clinical expert. The conjunctivitis guideline of the American Academy of Ophthalmology (AAO) is directed towards the diagnosis and care of patients with conjunctivitis. The high quality of this handbook is guaranteed by its inclusion in the National Guideline Clearinghouse server. The handbook is a document of 25 pages and is structured in the following sections: orientation, background, prevention and early detection, care of the patient, references and appendix.

---

5 http://www.intellicorp.com
6 http://www.aao.org/aao/education/library/ppp.cfm
7 http://www.guideline.gov
We have concentrated on the modelling of the orientation and care of the patient. The latter includes results to be achieved, diagnosis and treatment. The selected sections include different types of informal knowledge: (1) mainly text; (2) lists of intentions, objectives and results to be achieved; (3) lists with large numbers of symptoms and signs to be taken into account when evaluating patients with conjunctivitis; (4) two tables, one including the clinical signs for the different types of conjunctivitis under consideration and the other for treatment options.

In this work, we concentrate solely on the representation of the static domain knowledge for diagnosis. This will allow the knowledge base to be reused to code the operational knowledge by some medical guideline representation language (Elkin et al., 2000) or using specialised algorithms (problem-solving methods) for the diagnostic task (Taboada et al., 2001). For this reason, in our knowledge base, the operational knowledge is not represented. Instead, each time we found expressions (that indicated objectives, intentions, or actions), we have opted for maintaining the elements of free text, exactly as they appear in the handbook. In this way, when the knowledge base is reused in the near future, it will be very easy to locate the elements that require operational representation.

3 Development of the domain ontology

As we have commented previously, the design of the core ontology frequently requires the inclusion of portions of knowledge from various domain ontologies. The manual inclusion of knowledge from different ontologies is very tedious (Uschold et al., 1998), as it can consist of (1) the analysis and incorporation of a necessary portion of a given ontology, (2)
the revision of other portions of the same ontology that are related to the selected portion and a decision as to whether or not they should be included, and (3) the combination, in some way, of various ontologies (or portions of them). Processes 1 and 2 together are known as *knowledge extraction*. Process 3 can be carried out using two different alternatives (Noy and Musen, 2000):

1. Integrating the ontologies, i.e. *merging various ontologies in a new ontology*. The integration of ontologies can be seen as a process consisting of (McGuinness et al., 2000):

   - Finding the points of intersection (similarities and/or differences) of the ontologies.
   - Associating semantically similar concepts through equivalence and subsumption relationships.
   - Checking of the results in order to obtain a consistent, coherent and non-redundant ontology.

2. Combining the ontologies in a consistent and coherent form, but separating them from each other. This alternative is sometimes known as *alignment of ontologies*.

Today, various tools exist that simplify the process of integrating ontologies. In general, these tools help the user to locate similarities or differences between ontologies. Examples are PROMPT (Noy and Musen, 2000), Chimaera (McGuinness et al., 2000) or ONION
(Mitra et al., 2000). To review and compare the most outstanding tools (Noy and Musen, 2002) can be consulted. We have selected the tool PROMPT for various reasons:

- It allows merging of various source ontologies into a single ontology. This facility reduces development and merging time, when compared to tools that only provide, as a result, the similarities and differences found in the source ontologies.
- It analyses the concepts, properties, restrictions on its values, and the relationships between concepts, with the objective of maintaining the evolving ontology coherent and consistent.
- It is an extension of PROTÉGÉ-2000.
- It presents the results as suggestions, allowing revision and later acceptance or rejection.

3.1 Activities in the development of the domain ontology

In Fig. 1, the activities carried out during the core ontology design in the domain of conjunctivitis are diagrammed. In this design, we have reused portions of knowledge provided by previously designed ontologies:

- The ontology that contains the semantic types defined in the UMLS Semantic Network. We have found this ontology in the server of DAML+OIL.\(^8\)
- The EON\(^9\) ontology, defining the general structure of clinical protocols.

\(^8\) http://www.daml.org/ontologies/218
\(^9\) http://www.smi.stanford.edu/projects/eon/
The set of activities carried out in this stage includes different types of operations, such as: knowledge extraction, translation of parts of the ontologies to a new framework and the integration of ontologies. Below, we review the different activities in the order in which they are carried out.

Activity 1: Development of an initial ontology (Conjunctivitis-1)

We begin with the DAML+OIL ontology that represents the semantic types of the UMLS Semantic Network. During the development of this activity, Protégé-2000 did not import DAML+OIL ontologies. In addition, initially we only needed a very small part of this DAML+OIL ontology. So, we decided to manually transfer only the required parts to Protégé. Nowadays, an alpha version of a DAML+OIL plug-in exists for Protégé-2000

10 http://www.ai.sri.com/daml/DAML+OIL-plugin/
Activity 2: Extraction of knowledge from the EON ontology using PROMPT

(Conjunctivitis-2)

EON permits the representation of the different types of knowledge found in guidelines and medical protocols. For this, it includes a set of models that contain the representation primitives (See Fig. 2).

![Fig. 2: Models of the EON ontology](image)

Initially, we plan to extract the hierarchy of concepts associated with the following models:

- **Time Entity**: Contains a set of classes that represent different types of temporal entities used by the other models.
- **Medical Domain Class**: includes portions of both the hierarchies of the Semantic Network (whose root concepts are **Entity** and **Event**). These concepts have been modelled as abstract entities, i.e. instances cannot be directly defined from them. These two hierarchies are the ones we have extracted from this model.
• *Case Entity*: defines the classes and attributes of patient data.

We carried out the extraction process of these models using PROMPT. This tool allows the semiautomatic extraction of portions of knowledge from a concept hierarchy modelled in PROTÉGÉ-2000. PROMPT allows the selection and extraction of classes, instances and attributes. It also provides the possibility of extracting an isolated concept or all of the knowledge related to a concept (subclasses, instances, or any other related concept). For example, we needed to extract the whole of the temporal model of EON and any other knowledge that was related to this model, in order to maintain its coherence. For this, we selected the concept *Time Entity* of the EON ontology and the option of copying this class, its subclasses, instances, and all related concepts. PROMPT automatically extracted the complete *Time Entity* hierarchy and a small part of the *Expression* hierarchy.

During the extraction operations that do not require the copying of all the related knowledge, PROMPT usually generates a list of suggestions of things that should be done to maintain the coherence of the resulting ontology. For example, during the extraction of part of the *Medical Domain Class* hierarchy, PROMPT showed us a list of suggestions that mainly included the copy of all the EON concepts which any concept in the extracted ontology was referred to. Faced with this list, we had to decide whether to accept or reject the suggestions about the copy of these concepts.
Activity 3: Merging of the ontologies resulting from the previous stages (Conjunctivitis-1 and Conjunctivitis-2)

When we selected the option of merging the two previously obtained ontologies, PROMPT automatically identified the classes that had the same name in both ontologies and recommended the merging of each pair of classes into only one class (Fig. 3). This facility is not important, if the two ontologies do not have names that are common to both of them.

![Fig. 3: First suggestions shown by PROMPT after merging the two initial ontologies](image)

However, in our case it was very useful, as the two original ontologies covered similar aspects of the same domain (medical) and contained generic concepts taken from the UMLS Semantic Network. Despite this, a total correlation does not exist between concepts, as both ontologies were developed independently. For example, the two ontologies exhibit different structures of the concept hierarchy and contain different descriptions of the same concepts.
For each pair of classes that have the same name in both ontologies, PROMPT generates a recommendation that suggests merging. During the merging, PROMPT can generate further recommendations on (1) selection of the resulting class (abstract or exact) or (2) copy a super-class from some other class that was already copied previously. On the other hand, PROMPT recommends copying of the set of classes that are not duplicated (by name) in the source ontologies.

When PROMPT identifies inconstancies in the merged ontology, it shows a set of conflicts and suggestions for their solution. In our case, it identified three types of conflicts:

1. *Multiple paths from a same concept.* This is a consequence of merging ontologies with different structures of the same concept hierarchy.

2. *Not defined concepts exist.* These are concepts missing from the evolving ontology, but referred to by other concepts already included in the merged ontology.

3. *Different defined types for a concept.* In one of the source ontology the concept is described as concrete and in another source ontology as abstract.

### 3.2 Experiences and difficulties using PROMPT

In this section, we discuss some aspects that, from our experience, could reduce ontology merging and extraction time using PROMPT.
Use of a synonym dictionary or equivalence relationships among concepts from different source ontologies

Firstly, PROMPT automatically identifies identical concepts from different sources for merging. In our case, this facility was very useful, as the two original ontologies covered similar aspects of the same domain and they contained generic concepts taken from the UMLS Semantic Network. Despite this, in some domains such as medicine, synonyms (identical concepts with different names) are very frequent. These are not detected by PROMPT and so it is necessary to manually identify them. In cases like this, the merging time could be reduced if PROMPT takes account of a thesaurus or some equivalence relationships among concepts.

Adding new operations for simplifying ontology merging and extraction

Secondly, PROMPT identifies dangling references and, for each of them, it makes a suggestion. The latter consists of copying a concept that does not exist. If you accept the suggestion, PROMPT automatically copies the concept. In this way, a lot of merging time is saved. But if you reject it, PROMPT does not facilitate any operation. In our case study, we identified two possible operations. Sometimes, we removed the property that made the dangling reference and other times we changed the restriction on the property values. So, the merging time could be reduced if PROMPT provides some operations such as these.
**Possibility of configuring operations specific to each application**

Thirdly, we have detected that we have made some operations repeatedly. For example, in our case, each time PROMPT identified the conflict ‘Multiple paths from a concept’, we selected the most specific path for the concept (This was possible as the original ontologies contained very similar structures of the concept hierarchy). An improved alternative would be to have the possibility of configuring operations when a conflict is detected. In this way, PROMPT would take account of the particular heuristics of each case.

**4 Specialisation of the domain ontology**

Reuse information from reference terminology systems is very important in medicine, as a lot of terminology systems exist. Importing application-specific knowledge selectively from a terminology system enriches the knowledge base with standard common terminology. This step includes the following activities (Li et al., 2000): (1) searching the system for the information, (2) importing it from the system, and (3) integrating it with the evolving knowledge base.

The Unified Medical Language System (UMLS) is a system integrating many medical terminologies and coding systems. It consists of four knowledge sources that provide information about medical terminologies. We have used two of them: the Metathesaurus and the Semantic Network. The Metathesaurus contains terms, concepts, and relationships among them, drawn from different medical terminological systems. The Metathesaurus
represents each concept by a name, a concept unique identifier (CUI), a definition, a set of synonyms and a large amount of other information (Fig.4). Each concept in the Metathesaurus is assigned to one or more semantic types from the Semantic Network. The latter provides a classification of all the UMLS concepts. This network is structured in two hierarchies, where the initial types are Entity and Event. A name, a type unique identifier (TUI) and a definition represent each semantic type, which is related to at least one other type, through an is-a relationship, except Entity and Event, as they are the initial types in the two hierarchies that compose the network.

![Fig. 4: A UMLS concept and its semantic type, both imported to Protégé-2000](image)

### 4.1 Using meta-classes to loosely integrate reference terminology and ontology’s knowledge

In this section, we propose to loosely integrate the domain ontology with the imported terminological information through meta-classes. This way of looser integration is justified, as alignment is usually more suitable when the ontologies cover domains that are complementary to each other (Noy and Musen, 1999).
All medical concepts defined in the medical guideline text are formalised as concepts or instances of the *Medical Domain Class*. For example, the medical concept ‘*Discharge From Eye*’ is an indirect subclass of the *Medical Domain Class*. Each subclass of the *Medical Domain Class* belongs to one of these two meta-classes: *Metathesaurus Concept Metaclass* or *Semantic Type Metaclass*. They both contain a slot defining a relation to the corresponding terminological information about a Metathesaurus concept or about a Semantic Type. For example, the ‘*Discharge From Eye*’ class has been modelled as an instance of the *Metathesaurus Concept Metaclass* (Fig 5), as this concept was found in the UMLS Metathesaurus. As an instance of the *Metathesaurus Concept Metaclass*, it inherits the ‘*metathesaurus concept*’ slot. This slot represents a relation to the Metathesaurus concept ‘*Discharge from eye*’, which only describes terminological information (definition, cui, synonymous, etc).
Fig. 5: An Ontoviz screen showing the medical concept ‘Discharge From Eye’, which is an instance of the Metathesaurus Concept Metaclass and which is linked to the terminological information through the inherited slot ‘metathesaurus concept’.

In Fig. 6, on the left, we can see a Protégé-2000 screen showing a small part of the Medical Domain Class, including ‘Discharge From Eye’. In the middle, we can see some of the classes that are instances of the Metathesaurus Concept Metaclass, such as ‘Discharge From Eye’. On the right, the set of attributes that describes this class is shown. All of these attributes are essential properties needed to represent the knowledge required by the medical guideline text. The terminological information about this class is represented separately in the instance of the Metathesaurus Concept ‘Discharge from eye’ (Fig. 4 and 5). Both the ‘Discharge From Eye’ class and the ‘Discharge from eye’ instance are linked via the value of the slot Metathesaurus Concept (the bottom part on the right, Fig. 6), which is inherited from the Metathesaurus Concept Metaclass (Fig. 5). In this way, we are able to maintain different knowledge (terminology and essential properties of a concept) separately
through defining meta-classes. The advantages of this approach are commented in Section 4.4.

Fig. 6: A Protégé-2000 screen showing, on the left, a small part of the Medical Domain Class; in the middle, a set of classes belonging to the ‘Metathesaurus Concept Metaclass’; on the right, the class ‘Discharge From Eye’.

4.2 Importing from UMLS

In this work, we use the Protégé frame-based knowledge representation and other representation capabilities, such as the Protégé Axiom Language (PAL), to capture portions of
the knowledge of the two major components of the UMLS (the Metathesaurus and the Semantic Network) in a simplified and homogeneous way. In particular, we have extended the core ontology with a new component, which holds (1) a meta-model representing a simplified structure of the UMLS into Protégé-2000 and (2) a set of constraints expressed using the Protégé Axiom Language (based on first order logic) oriented to check the consistency and coherence of the imported information during the development and maintenance of the knowledge base (Fig. 7).

The information imported from UMLS is represented as indirect instances of the *UMLS Mapping* class (Fig. 8), which is separated from the *Medical Domain Class*. The *UMLS Mapping* class models the UMLS information that we have considered relevant for our particular case. We have defined three subclasses for *UMLS Mapping*: *Metathesaurus Concept*, *Semantic Type* and *Semantic Relation*. Each instance of these subclasses has a unique identifier, each Metathesaurus concept has one or more semantic types and each semantic type has an ‘is-a’ relationship with another semantic type. All concepts imported from the UMLS are modelled as instances of some of these three subclasses. For example, the concept *Discharge from eye* imported from the Metathesaurus (Fig. 4) is represented as an instance of the *Metathesaurus Concept* class. A much more complete meta-model, described using Entity Relationship Diagrams, for these two knowledge sources of UMLS can be found in (De Keizer and Abu-Hanna, 2000).
Fig. 7: Importing terminological information and consistency checking into Protégé-2000

Fig. 8: An Ontoviz screen showing the representation of a very simplified meta-model of the Metathesaurus and the Semantic Network in Protégé-2000
Complementary to the representation of the terminological information in Protégé, a formal counterpart (based on first order logic) is required to enhance expressivity and to support consistency after importing information, as the latter may be erroneously stored into the terminological system or the import process may introduce some mistakes. The following PAL constraints show some illustrative examples of the type of required consistency checking.

**Example 1: Two different UMLS concepts cannot have the same identifier.** An identifier can be conceived as a slot holding 1) a string representing the identifier or 2) an instance of a class, where the class represents the identifier. We chose the second option as it allows us to assemble the whole set of knowledge base identifiers into only one class (named *Unique Identifier*). As we can see in Fig. 9, we have distinguished three types of unique identifiers: CUI, TUI and Case_ID. All of them have been modelled as subclasses of the *Unique Identifier* class. According to this type of modelling, we require the definition of only one PAL constraint to check all knowledge base identifiers. The PAL constraint in Fig. 9 expresses that two different identifiers cannot hold the same value. This PAL constraint has been added to the definition of the *Unique Identifier* class. In this way, it is inherited by all subclasses of the *Unique Identifier* class, so it must be verified by all of them.

**Example 2: Each UMLS concept has assigned a unique identifier.** In Fig. 10, a PAL constraint allows us checking whether each semantic type has assigned a not-null identifier.

**Example 3: Each UMLS concept is represented by a string starting with a “C”.**
Example 4: Each UMLS concept has assigned, at least, a not-null semantic type. We have defined the ‘semantic type’ slot of the Metathesaurus Concept class (Fig. 8) as a ‘required’ instance of the Semantic Type class. That is, the value of the ‘semantic type’ slot cannot be null. If the slot does not verify this facet, Protégé outlines it in red in order to notify this violation. But, with the aim of detecting all possible inconsistencies of the knowledge base, we would require to revise each concept individually. A quicker way of detecting all of them in only one step is by defining slot attachments with PAL constraints.

Fig. 9: A PAL constraint expressing ‘An identifier is unique’, that is, two different identifiers cannot have the same value for the slot ‘ui’.
4.3 Integrating all knowledge in a new ontology using meta-classes and PAL constraints

During the import process, the two Semantic Network hierarchies are merged with specific concepts (linked to Metathesaurus concepts). The main strategy followed in this work was to preserve the ‘is-a’ relationships between each Metathesaurus concept and some semantic type. In some cases, we required the addition of new concepts (those not found in the UMLS). In other cases, we decided to add new hierarchies among classes linked to Metathesaurus concepts. As a result, sometimes, the semantic type assigned to a Metathesaurus concept is not the same than the direct super-class of the linked Medical Domain Class concept. However, the semantic type is always a super-class of the linked concept (although this may be indirect). For example, in Fig. 11 Discharge From Eye is a direct subclass of Eye Symptom, whereas Sign or Symptom is the semantic type assigned to
Discharge from eye (Fig. 4). Although, in some cases similar to this one, we designed new levels in the hierarchy in Protégé, we always preserved the ‘is-a’ relationship between a Metathesaurus concept and the semantic type which it belong to. The PAL constraint, expressing that ‘is-a’ relationships between Metathesaurus concepts and semantic types must be preserved, is shown in Fig. 12.

![Diagram showing 'is-a' relationship between 'Discharge From Eye' and 'Sign or Symptom']

*Fig. 11: A small part of the Medical Domain Class and the UMLS original ‘is-a’ relationship between ‘Discharge From Eye’ and ‘Sign or Symptom’*
Fig. 12: On the right, a PAL constraint expressing that ‘is-a’ relationships between Metathesaurus concepts and semantic types are preserved.

4.4 Experiences and lessons learned

The Protégé frame-based knowledge representation is very intuitive and its structure is easy to understand. We have used a lot of Protégé features and resources, such as meta-classes and PAL constraints. These features provide a more flexible and powerful means of representing knowledge than traditional frame-based techniques.

Meta-classes favour the modular representation of the knowledge base. With this divided into smaller modules according to knowledge content (UMLS knowledge, Medical Domain Class, Time Entity, Case Entity), it is easier to read, understand and revise it. In addition,
meta-classes provide a good means of separating the UMLS terminological knowledge from the definitional knowledge associated with guideline text. There are several reasons that justify the use of meta-classes together with the separation of the information imported from the UMLS:

1. Using meta-classes, we are able to classify concepts from different sources (Metathesaurus, Semantic Network and new concepts from scratch) in the same medical domain hierarchy. At the same time, we conserve the origin of the information.
2. In this way, we can easily distinguish between UMLS terminological information and essential properties of a class or instance.
3. A user can access the definition of the guideline without the UMLS descriptions. From the reuse point of view, this approach is more accurate, as potentially it does not overwhelm the user with excessive information.
4. The use of meta-classes simplifies the maintaining of the knowledge base: When the information contained in the UMLS evolves over a period of time, the use of meta-classes will be very useful for re-interpreting the terminological information alone, with no changes required in the Medical Domain Class.

One purpose of PAL is to express constraints about a knowledge base. PAL provides a means of modelling sentences that are always true. We have included PAL sentences in our knowledge base for both constraining its information and verifying its correctness. In particular, as we commented in the previous section, PAL sentences allowed us to automatically check the two following aspects: (1) was UMLS information imported correctly? (2) Was UMLS information integrated with the knowledge base appropriately?
PAL constraints are defined as independent elements in the knowledge base and they can be added to the definition of classes through the class constraint pane. The Protégé facility for including the PAL constraints inside the definition of classes improves readability and maintenance of the knowledge base. But you often change PAL constraints in development time. As a result, you need to remove or add them each time you modify. A feature that could reduce development time is the possibility of automatically detecting the classes defined in each PAL constraint and asking the user whether these should be added to the respective classes. Another important Protégé characteristic is the checking of the PAL constraints, although the debugging process could provide more help to the user.

We have also used the Protégé facility for automatically generating forms that allow us to enter the instances of every class. By entering instances on case-data we can check, in a simple and quick way, if the knowledge base is modelled as intended. This facility is necessary for improving the knowledge base model. However, after entering a lot of instances on case-data, these are mixed with instances relative to knowledge and it is more difficult to read, understand and revise the knowledge base.

In connection with UMLS, initially we searched the Metathesaurus for concepts using the web based interaction tools provided by the UMLS Knowledge Source Server (UMLSKS). Later, we used the UMLS plug-in, with the aim of comparing both the search and import processes between these two alternatives.

With regard to the search process, the UMLS plug-in is very intuitive and the visualization of results is very well organized. It provides two screens for visualizing the results: (1) a
screen for showing and navigating the set of results obtained during the search of a text, and (2) a set of sub-screens for displaying the UMLS information of the selected term on another screen. This way of displaying the information is more useful from the point of view of the knowledge base design, as the relevant information is located more quickly than using the tool provided by UMLSKS. In particular, we have found this way of displaying the ordered list of search results and the narrow tree of a concept, to be very useful.

Importing the UMLS information to a knowledge base in Protégé is quick, as you only need to select the concept to be imported and the place (class or instance) in the knowledge where the information will be imported. The plug-in automatically creates all slots (broader, cui, narrower, related slot, semantic type and synonym) and fills them with the search results. In addition, you can directly import not only the searched concept, but also any concept in the narrow tree. However, the UMLS plug-in does not take into account any UMLS meta-model, not even a very simplified one. Instead, it represents the UMLS structure defining a meta-class whose slots all are strings. We think that a description of the UMLS structure, at least simplified, is necessary in order to facilitate automated reasoning, such as consistency checking. On the other hand, there is no means of selecting the information to be imported. It could be of interest to have the possibility of importing only some slots.

The previously mentioned differences in the definition of slots between a Protégé project and the UMLS tab prevent us from importing the UMLS information directly into our knowledge base. One alternative is to use the UMLS plug-in for (1) searching for and
importing the needed knowledge in a separate Protégé project and (2) merging this new project with the knowledge base, by using PROMPT. This approach requires the carrying out of several activities with PROMPT: merging the meta-classes relative to UMLS information, merging the identical slots that define the UMLS information, and copying of the classes and instances containing the UMLS information from the two original projects. Later, the information of the differently defined slots must be moved. Therefore, this alternative is only useful when you expect to import a large amount of information during a work-session.

5 Conclusions

Nowadays, knowledge base design is viewed as a process of enlarging domain ontology with specific knowledge of a particular application (Musen, 1998). The prompt benefits of this approach are knowledge bases endowed with a common and standard terminology. However, we argue that ontology-based development is a labour intensive and time consuming process. Firstly, it is unlikely to find a domain ontology that contains all the knowledge required for structuring a specific knowledge base. As a result, the knowledge base core must be designed by the combination and adaptation of various ontologies. So, this process should not be viewed as a one-stage process. In general, it includes several activities, where each one is focused on a particular operation, such as: ontology development, extraction, merging or comparing. Our study provides a practical example of how the tool PROMPT can be used for facilitating these operations.
Secondly, importing knowledge from unified terminology systems, when they are available, enriches knowledge bases, improving their reuse and share. This step includes the selective importation of information and its integration with the knowledge base core ontology. Our application provides a practical scenario of how unified terminology systems can be used for designing knowledge bases. Many other examples exist in both medical and biological domains. Converting large portions of data from a terminological system to some rigorous knowledge representation formalism was the aim of most approaches found in the literature. Two examples are the terminological knowledge base designed in the clinical domain of human anatomy and pathology (Schulz and Hahn, 2001) and the knowledge base modelling the Gene Ontology in Protégé (Yeh et al., 2003). However, the knowledge covered in our application is more sophisticated, as it is oriented to provide sound medical reasoning. In this way, our application was focussed on searching a terminological system for smaller portions of information and importing (and reorganizing) them to an evolving knowledge base.

An application closely related to our work is a clinical decision support system designed in the domain of blood transfusion (Achour et al., 2001). The authors did not only develop the application, but they also implemented a knowledge acquisition tool for navigating and importing knowledge from UMLS. They did not reuse ontologies explicitly, although they needed to extract relationships between semantic types and concepts. As a consequence, they incorporated the required portions of the Semantic Network ontology manually. In the design of the knowledge base for human anatomy and pathology (Schulz and Hahn, 2001), the authors also remodelled the top-level concepts of the Semantic Network ontology manually. We started, on the contrary, from a core ontology, which included, among other
conceptualisations, the Semantic Network ontology. Our design approach was based on (1) analysing the required types of knowledge for structuring the knowledge base, (2) reusing and merging the ontologies providing the needed structure and (3) importing the terminological knowledge. In any case, tools oriented to integrate ontologies, such as PROMPT, can be used at any time in development of knowledge bases.

Another important difference is that our application incorporates more sophisticated medical knowledge, incorporating broader descriptions of symptoms, signs and pathologies than those provided by a terminological system. These descriptions are necessary for representing all knowledge described in the guideline text and some of them have been incorporated reusing conceptualisations from other ontologies. So, one of the main contributions of our study is the integration between knowledge imported from a reference terminology system and the reused ontologies. For that, we have considered a very simplified ontology of the UMLS structure and imported the terminological information as instances of this ontology. In addition, we have aligned the UMLS ontology and the knowledge base core ontology by establishing links between them, through meta-classes. This method of looser integration is justified, as alignment is usually more suitable when the ontologies cover domains that are complementary to each other (Noy and Musen, 1999). The advantages of this type of integration were commented in the previous section.

As a result of our study, we can conclude that, in order to incorporate information from remote knowledge sources (such as UMLS) into a knowledge base, it is necessary to have available an ontology providing (1) a complete and unambiguous description of the structure of the terminological system and (2) a set of logical constraints to support
consistency during import of terminological information. Studies, such as (Schulz and Hahn, 2001), (Yeh et al., 2003) or (Gu et al., 2000), have collected empirical evidence for the lack of logical consistency of the UMLS Metathesaurus, so it is always necessary to check the imported information. In our current application, the UMLS structure is represented in a very simplified way, by only checking unique codes, terminological cycles and the relationships between Metathesaurus concepts and semantic types. In the future, we plan to extend the ontology with the aim of supporting a complete and unambiguous description of the structure of UMLS. For that, we plan to translate the proposal of representation described in (De Keizer and Abu-Hanna, 2000), using Entity Relationship Diagrams, to Protégé-2000 and PAL.

The last comment is the need for developing a new generation of tools for improving the search process in reference terminology systems. Currently, the search is a very time-consuming process and it is not error free, as many concepts are not defined and, therefore, it is difficult to know whether the search is successful or not. In addition, sound medical reasoning usually requires highly specialized concepts, such as mild mucus discharge or conjunctival scarring beginning in medical fornices, which are not covered by the system. In these cases, it is necessary to plan some guidelines in order to reduce search time. We think that the use of natural language processing-based techniques could substantially reduce the search time.

Acknowledgments
We thank Samson Tu for providing us the EON ontology and Natasha Noy for her thoughtful comments on this work. We are also grateful to anonymous reviewers for their insightful feedback. This work has been supported by grant from the Xunta de Galicia, through the research project PGIDT01-PXI20608PR.
References


