Published in final edited form as:
Stud Health Technol Inform. 2001 ; 84(0 1): 161–165.

An Object-oriented Model for Representing Semantic Locality in the UMLS

Olivier Bodenreider
U. S. National Library of Medicine, Bethesda, MD, USA, olivier@nlm.nih.gov

Abstract

Several information models have been developed for the Unified Medical Language System (UMLS). While some models are term-oriented, a knowledge-oriented model is needed for representing semantic locality, i.e. the various semantic links among concepts. We propose an object-oriented model in which the semantic features of the UMLS are made available through four major classes for representing Metathesaurus concepts, semantic types, inter-concept relationships and Semantic Network relationships. Additional semantic methods for reducing the complexity of the hierarchical relationships represented in the UMLS are proposed. Implementation details are presented, as well as examples of use. The interest of this approach is discussed.

Keywords

Unified Medical Language System; Knowledge representation; Semantics; Object-oriented model

Introduction

The Unified Medical Language System® (UMLS®) is an extensive source of biomedical knowledge developed and maintained by the U.S. National Library of Medicine [1]. The UMLS Knowledge Sources include the Metathesaurus®, which provides a common structure for more than 50 biomedical vocabularies, and the Semantic Network, a high-level structure that defines and organizes the semantic types assigned to each Metathesaurus concept.

The representation of meaning in the UMLS allows users to define and explore the semantic space surrounding a given concept [2]. The various semantic links among concepts represent one of the organizing principles of the UMLS: semantic locality [3]. The dimensions of semantic locality include term information (synonymy, hypernymy, hyponymy), contextual information in a particular source, co-occurrence of terms in the medical literature, and the categorization of the concepts in a semantic network. Figure 1 shows a subset of the semantic space for the concept "Heart", based on the principles of semantic locality.

Address for correspondence. Olivier Bodenreider, National Library of Medicine, 8600 Rockville Pike (MS 43), Bethesda, MD 20894 - USA. olivier@nlm.nih.gov.

In slightly different terms, semantic locality is based on a combination of terminological knowledge (relationships among terms in source vocabularies), lexical knowledge (relationships derived from the lexical analysis of terms), symbolic knowledge (inter-concept relationships based on the meaning of the concepts) and statistical knowledge (relationships among concepts that co-occur in the medical literature).

Several strategies have been proposed to access UMLS data, based on different information models (relational or object-oriented [4]), different formalisms (directed acyclic graphs [5], conceptual graphs [6], ASN.1 [7]), and for different purposes, including knowledge representation and reasoning, terminology services [8], and vocabulary management [9].

The rationale for initiating this work was the following. The UMLS is designed to represent not only lexical and semantic information about the biomedical domain, but also virtually every bit of information present in the medical vocabularies it integrates. Semantic locality depends on the semantic information in the UMLS, but not on the detailed characteristics of the constituent vocabularies. Hence, the model representing semantic locality is knowledge-oriented rather than term-oriented and can be simpler than a comprehensive model of the UMLS. However, in order to perform common tasks on semantic spaces (e.g., building the graph of the ancestors of a given concepts, or selecting all the concepts related to a given concept through selected relationships), high level methods must be added to the information model.

An object-oriented model was preferred over the original relational model because it provides simpler, more flexible and extendable methods for utilizing the knowledge in applications, and offers both an additional level of abstraction above the UMLS distribution and some independence from its back-end implementation.

The Model

Our model is based on a minimal set of classes, properties and methods, as shown in Figure 2.

Classes and properties

There are four major classes.

1. **Concept**. The UMLS Metathesaurus is organized by concept or meaning, which is a cluster of synonymous terms. Concepts are identified by a concept unique identifier (CUI) which is needed to instantiate Concept objects. A given term may have several meanings and belong to several concepts, which prevents a term from unambiguously instantiating a concept. Mapping text to UMLS concepts is necessary but must be kept distinct from instantiating Concept objects. Besides the CUI, Concept properties include the preferred name of the concept in the UMLS, a list of definitions, a list of sources, and the total frequency of occurrence in MEDLINE®. These properties are simply strings or numerical values rather than instances of other classes, since they are generally used only for illustrative purposes. In contrast, other properties such as sets of related concepts (e.g., parents,

children, siblings, etc.) or the set of semantic types are lists of instances of the Concept or SemType classes.

2. **SemType**. Semantic types are the nodes of the UMLS Semantic Network. They play a role in the Semantic Network equivalent to that of concepts in the Metathesaurus. Semantic types are identified by a unique identifier (TUI), but SemType objects may be instantiated either from a TUI or from a semantic type name, since no two semantic types share the same name. In addition, a property defines the semantic group (groupings of semantic types providing a broad categorization of the concept [10]) to which a given semantic type belongs. Technically, allowable relationships between semantic types in the Semantic Network (e.g., isa, treats) are also represented as SemType instances. The inverse name of a relationship can be queried.
3. **Relationships**. Inter-concept relationships defined in the Metathesaurus describe either symbolic knowledge or statistical knowledge. This class provides access to all relationships between two concepts, i.e. for a pair of CUIs or Concept instances. For a given type of relationship (e.g., child), detailed information about the nature of this relationship (e.g., isa, part_of), its sources and the frequency of co-occurrence in MEDLINE is provided when available.
4. **SemNet**. Semantic Network relationships (SNRs) are relationships defined between semantic types (STs), and the Semantic Network can be represented as a list of triplets (ST_1, SNR, ST_2). This class provides access to all relationships between two semantic types, i.e. for a pair of TUIs or SemType instances. Besides the two related semantic types represented as SemType instances, the only other property in this class is the list of relationships between these semantic types, provided as a list of SemType instances.

Additional classes were defined for more specialized purposes. The **ATX** class represents associated expressions, expression trees in which leaves are elementary concepts and nodes logical operators or main heading to subheading relationship indicators. The **COC** class offers several techniques for selecting the most important co-occurring concepts in MEDLINE, using the frequency of co-occurrence as a surrogate for the strength of the relationship. Finally, a **Term** class merely encapsulates calls to the UMLS Knowledge Source Server in order to provide for mapping terms to UMLS concepts using the traditional matching techniques (exact match, through the normalized string index, and approximate matching). All these additional classes link a term or a concept to a set of concepts represented as instances of the Concept class.

Methods

Methods defined for the major classes are essentially accessors, allowing users to get or set properties from a given instance. Several methods are systematically added to each class, allowing instances to format themselves to serve general purposes (e.g., write to a file or display as part of an HTML document). Additional methods were defined for certain classes for specific purposes. For example, the SemNet class has an *exist* method that tests for the existence of a given relationship between two semantic types.

In the Concept class, however, many more methods were defined, making this class substantially different from its counterpart in the UMLS relational model. Some methods were added for convenience, to group the values of several properties. For example, *anc1* retrieves all concepts in direct hierarchical relationship to a given concept, i.e. first-generation ancestors, whether the relationship comes from source vocabularies (*par* property, for ‘parent of’) or from the UMLS editors (*bro* property, for ‘broader than’). Similarly, *des1* combines all descendants of the first generation, i.e. linked by the ‘child of’ (*chd* property) or ‘narrower than’ (*nar* property) relationship in the Metathesaurus. Additionally, a *names* method fetches the preferred name for a concept in a given source, or all the names for this concept when called without arguments.

In other cases, however, methods provide information that is not directly available in the UMLS and therefore constitutes some sort of an added value. For example, the SIB relationship defined in the UMLS retrieves the list of siblings of a given concept as defined in source vocabularies, i.e. the children of this concept’s parents. Let us assume that the *par* and *bro* properties are close in meaning and can be replaced by *anc1* for certain purposes, and that the same thing is true for *chd* and *nar*, replaced by *des1*. In this case, the notion of sibling can as well be extended from “children of the parents” to “children or narrower concepts of parents or broader concepts”, i.e. “first generation descendants of first generation ancestors”. We defined a *sibx* property for such an extended version of the siblings. Similarly, the *sib_bn* property of a concept retrieves the narrower concepts of its broader concepts.

Another reason for extending the set of methods applicable to Concept objects was to absorb some of the redundancy resulting from the way the UMLS is built. Due to differences in granularity among vocabularies, a hierarchical relationship may be defined directly between concepts C_1 and C_3 in some vocabulary while some finer-grained vocabulary may define C_1 parent of C_2 and C_2 parent of C_3. Though consistent, these relationship may appear unnecessarily redundant: assuming that their nature is the same, those coming from the finer-grained vocabulary are sufficient to infer the other one. Such redundancy may even be considered detrimental for display purposes, for example, or, more generally, when the goal is to simplify the representation. In graph theory parlance, the removal of such redundant links is called transitive reduction. For this reason, for each property or method related to hierarchical or hierarchy-based (siblings) relationships, we define an alternative method that has the same meaning but additionally performs a transitive reduction to the ancestors and descendants organized as a graph (methods with a *_tr* suffix).

Implementation

A prototype of the object-oriented model was implemented in Perl, using the object-oriented features available since version 5 of the language [11]. As shown in figure 3, the whole architecture classically consists of three layers: the UMLS classes described earlier, mediator classes, and a back-end. Therefore, a limited knowledge (limited to the first layer) is required to use this model in an application. Moreover, changes made to the back-end will not require the application’s code to be modified; mediator classes will make the changes transparent to the UMLS classes.

Most UMLS classes rely on data stored in a relational database, but data could generally be queried through the Knowledge Source Server (KSS), as is the case for the Term class. Having a local copy of the UMLS stored in a database allows for additional filtering of the data. For example, circular hierarchical relationship that would lead to cycles in the graph of concepts (and prevent performing the transitive reduction) may be removed from the database.

Mediator classes essentially contain predefined SQL statements or KSS calls used to retrieve a given property in a class. For example, the SQL statement “select STY from STYPE where TUI = ?” retrieves the name of a semantic type by its unique identifier. More complex statements are sometimes needed: for example, to instantiate a Relationship object from a pair of CUIs requires combined data from the MRREL and MRCOC tables.

Applications

This object-oriented model was used for the development of several UMLS-based applications at the National Library of Medicine. The *Restrict to MeSH* algorithm [12], a component of the Indexing Initiative prototype [13], helps find the MeSH descriptors closely associated with any UMLS concept. The *UMLS Semantic Navigator*[1], an experimental knowledge exploration tool, displays the semantic space surrounding an arbitrary UMLS concept, allowing users to navigate it. These two applications make heavy use of the graph data structure for representing hierarchical information from the Metathesaurus.

Using this model, we were able to rapidly develop a program for defining the “family” of a concept [14]. One part of the family consists of ancestors and descendants, siblings and “other relatives” (other related concepts), all already defined as related concepts in the UMLS and accessible through the corresponding property of the Concept class. Additionally, we used combinations of properties to define uncles (siblings of first-generation ancestors) or cousins (first-generation descendants of uncles). Figure 4 shows an example of Perl code for computing the unique identifiers for the uncles and the cousins of a given concept. The model can easily be extended through derived classes in order to serve specific purposes.

Discussion

This object-oriented, knowledge-oriented model quite obviously differs from term-oriented models and from the original relational model. Differences from other object-oriented models may be subtler. Gu and al. used an object-oriented database for representing the Metathesaurus and the Semantic Network as a unified system [4]. In contrast, we chose to keep the original structure of the UMLS, i.e. two distinct layers for the concepts and the semantic types. Instead of using a unified representation, we rather developed methods for exploring the semantic space from different perspectives, extending the set of relationships available in the UMLS. However, we use the same class to represent inter-concept relationships, whether symbolic or statistical.

[1]umlsks.nlm.nih.gov → Resources → Semantic Navigator

Although this model has not been used outside its development environment, it has proven to be usable for different purposes (information retrieval, visualization and navigation), in the context of application development. Other properties such as sharability and inter-operability need to be evaluated.

References

1. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993; 32(4):281–291. [PubMed: 8412823]
2. McCray AT, Nelson SJ. The representation of meaning in the UMLS. Methods Inf Med. 1995; 34(1–2):193–201. [PubMed: 9082131]
3. Nelson SJ, Tuttle MS, Cole WG, Sherertz DD, Sperzel WD, Erlbaum MS, et al. From meaning to term: semantic locality in the UMLS Metathesaurus. Proc Annu Symp Comput Appl Med Care. 1991:209–213. [PubMed: 1807589]
4. Gu H, Perl Y, Geller J, Halper M, Liu LM, Cimino JJ. Representing the UMLS as an object-oriented database: modeling issues and advantages. J Am Med Inform Assoc. 2000; 7(1):66–80. [PubMed: 10641964]
5. Bodenreider O, Burgun A, Botti G, Fieschi M, Le Beux P, Kohler F. Evaluation of the Unified Medical Language System as a medical knowledge source. J Am Med Inform Assoc. 1998; 5(1): 76–87. [PubMed: 9452987]
6. Volot F, Joubert M, Fieschi M. Review of biomedical knowledge and data representation with conceptual graphs. Methods Inf Med. 1998; 37(1):86–96. [PubMed: 9550852]
7. McCray AT, Divita G. ASN.1: defining a grammar for the UMLS knowledge sources. Proc Annu Symp Comput Appl Med Care. 1995:868–872. [PubMed: 8563416]
8. Hogarth MA, Gertz M, Gorin FA. Terminology query language: A server interface for concept-oriented terminology systems. Proc AMIA Symp. 2000; (20 Suppl):349–353. [PubMed: 11079903]
9. Gu H, Cimino JJ, Halper M, Geller J, Perl Y. Utilizing OODB schema modeling for vocabulary management. Proc AMIA Annu Fall Symp. 1996:274–278. [PubMed: 8947671]
10. McCray AT, Burgun A, Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. MEDINFO. 2001 (submitted).
11. Conway, D. Object Oriented Perl. Greenwich, CT: Manning; 2000.
12. Bodenreider O, Nelson SJ, Hole WT, Chang HF. Beyond synonymy: exploiting the UMLS semantics in mapping vocabularies. Proc AMIA Symp. 1998:815–819. [PubMed: 9929332]
13. Aronson AR, Bodenreider O, Chang HF, Humphrey SM, Mork JG, Nelson SJ, et al. The NLM indexing initiative. Proc AMIA Symp. 2000; (20 Suppl):17–21. [PubMed: 11079836]
14. Burgun A, Bodenreider O. Methods for exploring the semantics of the relationships among co-occurring concepts. MEDINFO. 2001 (submitted).

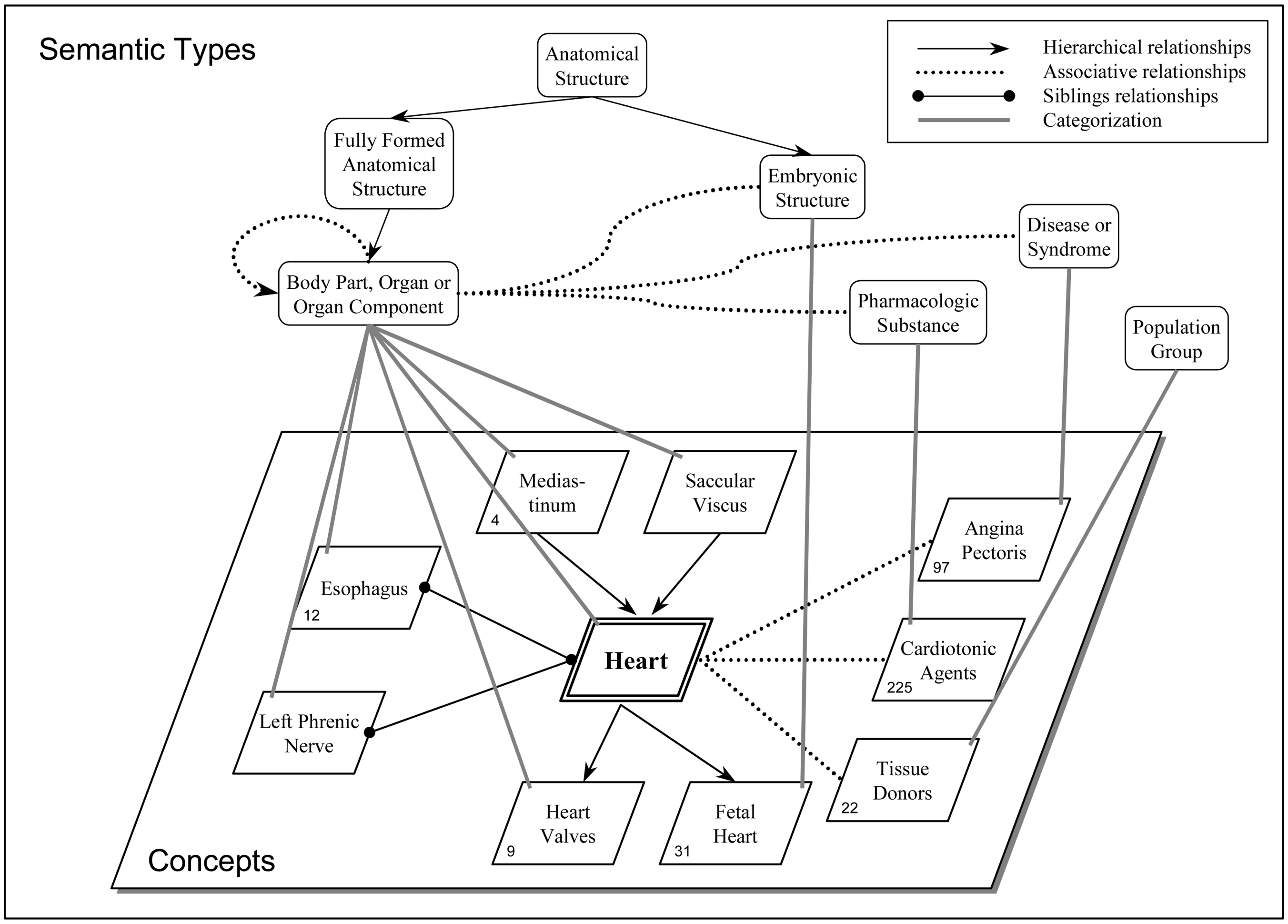

Figure 1.
Semantic space for the concept “Heart” (partial representation). Numbers refer to the frequency of co-occurrence in MEDLINE between “Heart” and other concepts, when available.

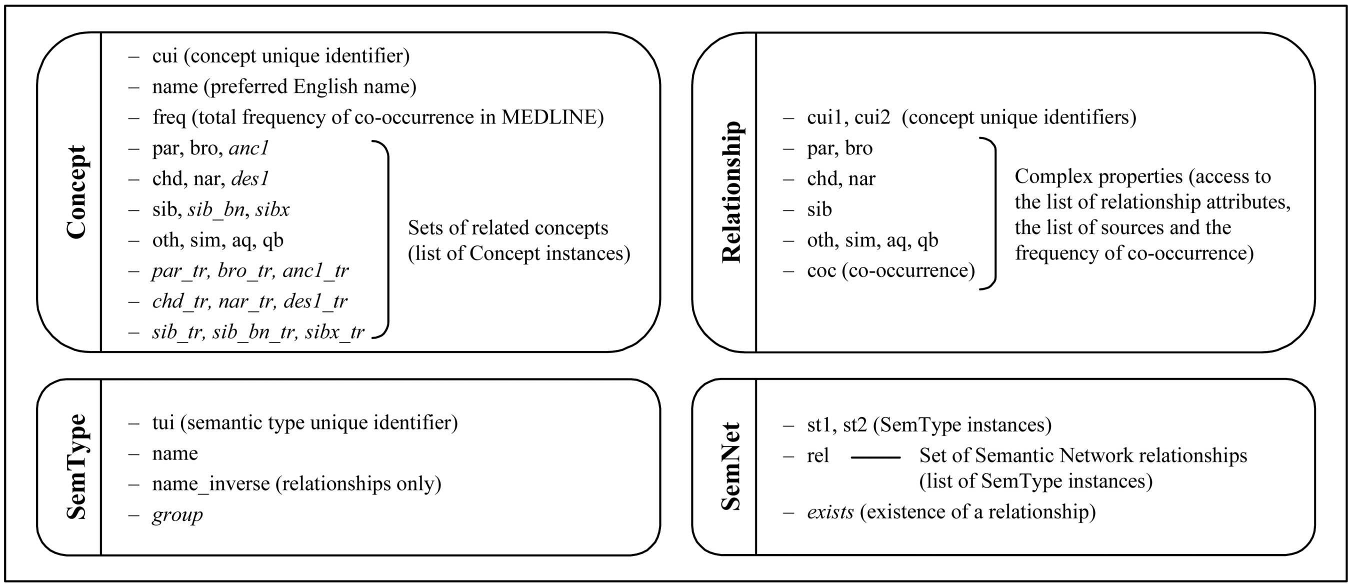
Concept
– cui (concept unique identifier)
– name (preferred English name)
– freq (total frequency of co-occurrence in MEDLINE)
– par, bro, *anc1*
– chd, nar, *des1*
– sib, *sib_bn*, *sibx*
– oth, sim, aq, qb
– *par_tr, bro_tr, anc1_tr*
– *chd_tr, nar_tr, des1_tr*
– *sib_tr, sib_bn_tr, sibx_tr*
Sets of related concepts (list of Concept instances)
Relationship
– cui1, cui2 (concept unique identifiers)
– par, bro
– chd, nar
– sib
– oth, sim, aq, qb
– coc (co-occurrence)
Complex properties (access to the list of relationship attributes, the list of sources and the frequency of co-occurrence)
SemType
– tui (semantic type unique identifier)
– name
– name_inverse (relationships only)
– *group*
SemNet
– st1, st2 (SemType instances)
– rel
Set of Semantic Network relationships (list of SemType instances)
– *exists* (existence of a relationship)

Figure 2.

The four major UMLS classes, with their properties and method (properties that are not directly available in the UMLS are displayed in italic).

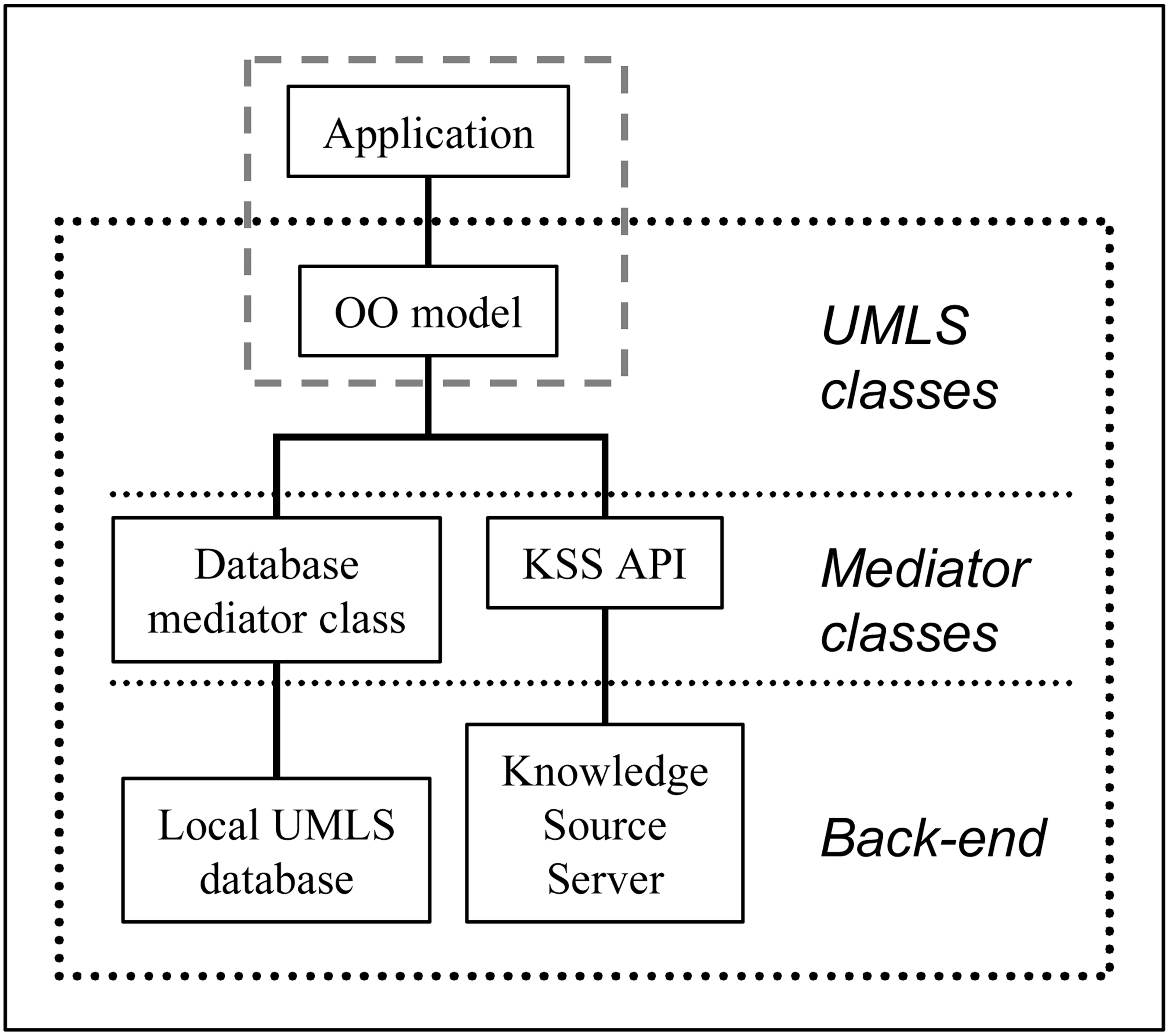

Figure 3.
Architecture of the model

```
use UMLS::CON::Concept;

my $c = UMLS::CON::Concept->new('C0005400');
my %seen = ();
my %ancestors = ();
my %uncles = ();
my %cousins = ();
foreach my $anc ($c->anc1_tr) {
    $ancestors{$anc->cui}++;
    foreach my $uncle ($anc->sibx_tr) {
        next if $seen{$uncle->cui};
        # remove from the uncles
        # those that are ancestors themselves
        next if $ancestors{$uncle->cui};
        $seen{$uncle->cui}++;
        $uncles{$uncle->cui}++;
        foreach my $cousin ($uncle->des1_tr) {
            $cousins{$cousin->cui}++;
        }
    }
}
```

Figure 4.
Example of code (Concept class)