How to Partition a Complex Schema of a Medical Terminology

Huanying (Helen) Gu,[1] Yehoshua Perl,[2] Michael Halper,[3]
James Geller,[2] Feng-shen Kuo,[2] James J. Cimino[4]

[1]Dept. of Health Informatics, University of Medicine & Dentistry of NJ, Newark, NJ 07103
[2]CIS Dept., New Jersey Institute of Technology, Newark, NJ 07102
[3]Mathematics & Computer Science Dept., Kean University, Union, NJ 07083
[4]Dept. of Medical Informatics, Columbia University, New York, NY 10032

Introduction: Controlled medical terminologies are increasingly becoming strategic components of various healthcare enterprises. However, the typical medical terminology can be difficult to exploit due to its extensive size and high density. The schema of a medical terminology offered by an object-oriented database representation provides an abstract view of a terminology. Thus, the schema enhances terminology comprehensibility, presentation, and usability. However, terminology schemas themselves can be large and unwieldy. We present a theoretical paradigm and a methodology for partitioning a medical terminology schema into more manageably sized fragments that promote increased comprehension.

Method: The specialization hierarchy of an OODB schema is the schema's backbone. Our partitioning methodology concentrates on the specialization hierarchy which is composed of *subclass* relationships, each connecting a subclass to a superclass. Previous research has identified two variants of the subclass relationship, namely, *category-of* and *role-of*. *Category-of* (*role-of*) is a specialization relationship used in the case where both the superclass and the subclass are in the same (different) context.

Our methodology identifies a forest structured subschema of a given terminology schema. The trees of this forest represent contexts, each of which is a logical subschema approximating all knowledge relevant to a specific subject area. This partition into well focused contexts helps to support comprehension of the original schema.

The methodology involves human-computer cooperation. A domain expert is called upon to make judgment decisions based on an understanding of the medical knowledge, while the computer provides results of algorithmic procedures for tasks which do not involve complex intuitive decisions but might be computationally intensive. The result of this cooperation is a refinement of the specialization hierarchy of the terminology schema. Every original subclass relationship becomes either a *category-of* or a *role-of*. The *category-of* relationships will form a forest.

The methodology has a sound theoretical basis, defined in terms of a set of three formal rules. The first of these utilizes the notion of *equicontext* relation which is defined as follows: A pair of classes belongs to the equicontext relation if both classes belong to the same context.

Rule 1: The equicontext relation partitions the classes of a schema into disjoint contexts. □

Rule 2: Two classes which are *category-of* specializations of the same superclass can neither be *category-of* descendants of one another nor have a common *category-of* descendant class. □

Rule 3: For each context, there exists one class R which is the *major* (or defining) class for the context such that every class in the context is a descendant of R. □

The human-computer interaction is guaranteed to yield a forest structured schema as long as the above rules are satisfied, as stated by a theorem:

Theorem: When **Rules 1–3** are satisfied, a class has at most one superclass to which it has a *category-of* relationship. □

The human-computer interaction is specified in the following as a sequence of 8 steps.

Step 1: (Computer) All attributes and relationships other than subclass relationships are removed from the OODB schema.

Step 2: (Computer) Resulting subschema from **Step 1** is arranged in topological sort order.

Step 3: (Expert) The subschema is scanned top-down according to the order from **Step 2** to identify defining classes (roots) of contexts. These chosen classes start new contexts. The subclass relationships from the root classes to their superclasses are changed to *role-of* relationships.

Step 4: (Computer) All classes with multiple non-*role-of* relationships to superclasses are listed in bottom-up order.

Step 5: (Expert) For each class identified in **Step 4**, the expert identifies at most one superclass which is in the same context as the class in order to conform to **Rule 2**. The subclass relationship to this superclass will be defined as a *category-of* relationship while all other subclass relationships of the class are defined as *role-of*'s.

Step 6: (Computer) Identify diamond structures.

Step 7: (Computer) In order to satisfy **Rule 2**, each diamond structure must contain classes from more than one context. Resolve any contradictions in the diamond structures, if necessary.

Step 8: (Computer) After all subclass relationships are refined as either *category-of* or *role-of*, a forest hierarchy of *category-of* relationships is obtained by deleting all *role-of*'s.

Results: We applied our methodology to the large and complex MED OODB schema which consists of 124 classes and 190 subclass relationships. The whole MED schema is partitioned into 48 trees.