PROCEEDINGS

IEEE
Networking the World™

EMB

2000 IEEE EMBS INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY APPLICATIONS IN BIOMEDICINE, November 9-10, Key Bridge Marriott Hotel, Arlington, Virginia

ITAB-ITIS 2000

Joint meeting

Third IEEE EMBS International Conference on Information Technology Applications in Biomedicine (ITAB '00)

Third Workshop of the International Telemedical Information Society (ITIS '00)

Proceedings Editor-in-Chief
Swamy Laxminarayan

Co-Editors
Andy Marsh, Metin Akay, Ilias Iakovidis
Luis Kun, Christian Roux

Extracting a Partition from a Complex Schema of a Medical Terminology

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

[1]Health Informatics Dept.
UMDNJ
Newark, NJ 07103
guhy@umdnj.edu

[2]CIS Dept.
NJIT
Newark, NJ 07102
{perl, geller, kuo}@homer.njit.edu

[3]Math & CS Dept.
Kean University
Union, NJ 07083
mhalper@turbo.kean.edu

[4]Medical Informatics Dept.
Columbia University
New York, NY 10032
ciminoj@columbia.edu

Abstract

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. In this paper, we present a methodology for partitioning a medical terminology schema into more manageably sized fragments, that promote increased comprehension. The application of our methodology to the schema of a large, existing medical terminology, called the Medical Entities Dictionary, is presented.

Keywords: Terminology, Medical Entities Dictionary, Object-Oriented Database, Schema, Partition

1 Introduction

Controlled medical terminologies should play a major role in overcoming terminological differences in the design of computerized healthcare information systems and their integration. However, the size and complexity of the typical terminology—usually tens of thousands of concepts and a proportionate number of properties—can cause serious problems of comprehension for its users. This can greatly limit the effective design and utilization of terminologies.

In previous work, we have developed techniques for representing a controlled medical terminology as an object-oriented database (OODB) [1]. A major advantage of an OODB representation is its schema, which provides an important abstract view of the terminology. Using this view, one is able to obtain an understanding of the terminology's overarching structure. The schema can also be used as a means for effectively browsing and traversing the terminology [2] and gaining an orientation to its contents. Moreover, we have shown that the schema can be an important tool to uncover and correct errors that have been introduced into the system [2, 3].

We have applied our techniques to a number of terminologies, including the Medical Entities Dictionary (MED) [4]. The MED schema was derived by partitioning the concepts into classes such that all the concepts of a class share the same set of properties. Furthermore, in [2] we have developed an OODB schema which extends the Semantic Network of the Unified Medical Language System (UMLS) of the National Library of Medicine [5]. Their respective OODB schemas are very compact compared to the size of the original terminologies. An OODB schema of 124 classes captures the MED's approximately 56,000 concepts (1998 version)—approximately a 450-to-1 reduction [3]. In [6], we provide a more refined OODB schema for the MED consisting of 1,564 classes. A schema of 1,296 classes represents the over 500,000 concepts of the UMLS—a 385-to-1 reduction [2].

Even though the OODB schema is a compact abstraction that promotes enhanced understanding of terminologies, it can still be too large and difficult to comprehend. In this paper, we focus on the issue of reducing the complexity of a schema. We present both a theoretical paradigm and a methodology to aid in the comprehension of existing large terminology schemas.

0-7803-6449-X/00/$10.00 © 2000 IEEE

Our methodology is based on the division of a large terminology schema into disjoint meaningful, manageably sized parts. Thus, to comprehend the schema, the user can begin with the study of relatively self-contained small logical fragments and progress to study the interaction between such fragments. Importantly, each of the fragments should typically fit on a single computer screen. The methodology for finding such a forest hierarchy relies on an interaction between an expert and the computer. An expert is asked to refine the specialization hierarchy of a terminology schema based on his understanding according to the rules of disciplined modeling. The computer performs heavily computational algorithmic procedures. The resulting forest hierarchy represents a partition of the specialization hierarchy into trees, which functions as a skeleton of the schema and supports comprehension efforts. We applied our methodology to the MED's OODB schema. In [7], we presented a technique for partitioning a terminology's knowledge content.

2 Schema Partitioning

The specialization hierarchy of an OODB schema serves as the basis platform for property inheritance. It is the backbone of an OODB schema. Our partitioning concentrates on the specialization hierarchy of a schema which only contains the *subclass* relationships. Figure 2 shows a specialization hierarchy of the MED schema. We call it the *hierarchical (sub)schema of a schema.*

Previous research [8, 9] has identified two different kinds of specialization relationships, 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.

A designer of an OODB schema determines whether a given specialization relationship is *category-of* or *role-of* by whether the two connected classes are in the same context or not. However, this determination is not always easy. In spite of extensive research, e.g., [10, 11, 12], there is still no definition of "context" which is widely accepted. We are not trying to define the notion of context. Rather we are making the *a priori* assumption that contexts exist in human thinking, and we are trying to identify them.

We accept the situation that for some designers two classes are in the same context while for others they are in different contexts. We provide a theoretical paradigm for the existence of such assignments of classes to contexts that results in a forest subschema. Also, we are introducing a methodology for finding such a forest. In order to ensure that a forest hierarchical subschema can be identified, the assignment of classes to contexts must always satisfy three rules of *disciplined modeling.* For detail explanations, see [13].

First, we define a mathematical equivalence relation *equicontext* between classes. 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 a *category-of* descendant of one other 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 descendent of R. □

Theorem: Using disciplined modeling, a class has at most one superclass to which it has a *category-of* relationship. □ (For proof, see [13])

Such a theorem implies that the *category-of* hierarchy has a forest structure, i.e., it consists of one or more trees, serving as the backbones of the schema.

3 Finding a Forest Hierarchy

In this section, we will describe a methodology that identifies a forest structure subschema of a given schema based on our theoretical paradigm. Trees of the forest represent contexts which are each a logical subschema approximating all knowledge relevant to a specific subject area, further supporting the comprehension of the original schema. The methodology involves human-computer cooperation. The human domain expert is called upon to make some 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 require many computational steps. We will specify which parts are performed by a computer and which are performed by the human domain expert. The result of our methodology is a refinement of the specialization hierarchy of the terminology schema. Every subclass relationship becomes either a *category-of* or a *role-of*. The *category-of* relationships will form a forest.

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

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

Step 3: Identify roots of contexts. (Human)

The subschema is scanned top-down (see [13]) according to the order from **Step 2**. In this scanning, defining classes (roots) of contexts are identified. The decision should be made by the meaning and importance of the class in the terminology compared to its superclasses' meanings. These chosen classes start new contexts.

The subclass relationships from the root classes to their superclasses are changed to *role-of* relationships. This kind of *role-of* relationship is a *regular role-of*, where the relationship models a switch of context.

Step 4: Multiple superclasses. (Computer)

All classes with multiple non-*role-of* relationships to superclasses are listed in bottom-up order (see [13]).

Step 5: Identify major superclass. (Human)

For each class identified in **Step 4**, the expert needs to identify 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*.

In our experience, for most of the classes with multiple superclasses, an expert can easily determine which of the superclasses is the major one. There is a minority of cases where the decision about a major superclass of a given class is not easy.

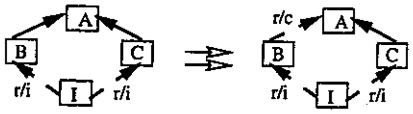

Figure 2: A diamond structure

In such cases, we try to distinguish which of the several superclasses, if any, should have a *category-of* relationship pointing to it, based on the partial context information we have already accumulated in our bottom-up processing. In [13], we give detail guidelines to deal with three different cases.

Step 6: Identify diamond structures. (Computer)

For each class I in the resulting list of **Step 4** and each pair of superclasses S_1 and S_2 of I, find a lowest common ancestor A of both S_1 and S_2. For each pair of such classes I and A, output the structure (represented by $< I, A >$) containing I, A, and all the classes which are descendants of A and ancestors of I. This is called a diamond or extended diamond structure.

Step 7: Resolve contradictions in the diamond structures. (Computer)

In order to fulfill **Rule 2** of disciplined modeling, each diamond or extended diamond structure must contain classes from more than one context. After executing the above steps, all the diamond structures already satisfy **Rule 2**. However, there is one case where we must artificially change additional *category-of* relationships to *role-of* relationships, in order to resolve a contradiction. In such a case, which we call a *contradictory diamond case*, the class I of the diamond structure $< I, A >$ is a *role-of/intersection* of its superclasses. All other classes in the diamond structure belong to one context (see Fig 2). Since the class I is the intersection of two superclasses B and C, they cannot both belong to the same context of their superclass A. Otherwise, since the intersection of a context with itself will result in the original context, the intersection class must belong to this common context. Thus, the classes B and C are also defined to be in separate contexts. The *category-of* relationship from either B or C to A is changed to *role-of*. We denote this kind of *role-of* as "*role-of/category-of*." It is represented by r/c in the figures.

Step 8: Get a forest hierarchy. (Computer)

After all subclass relationships are refined as either *category-of* or *role-of* relationships, a forest hi-

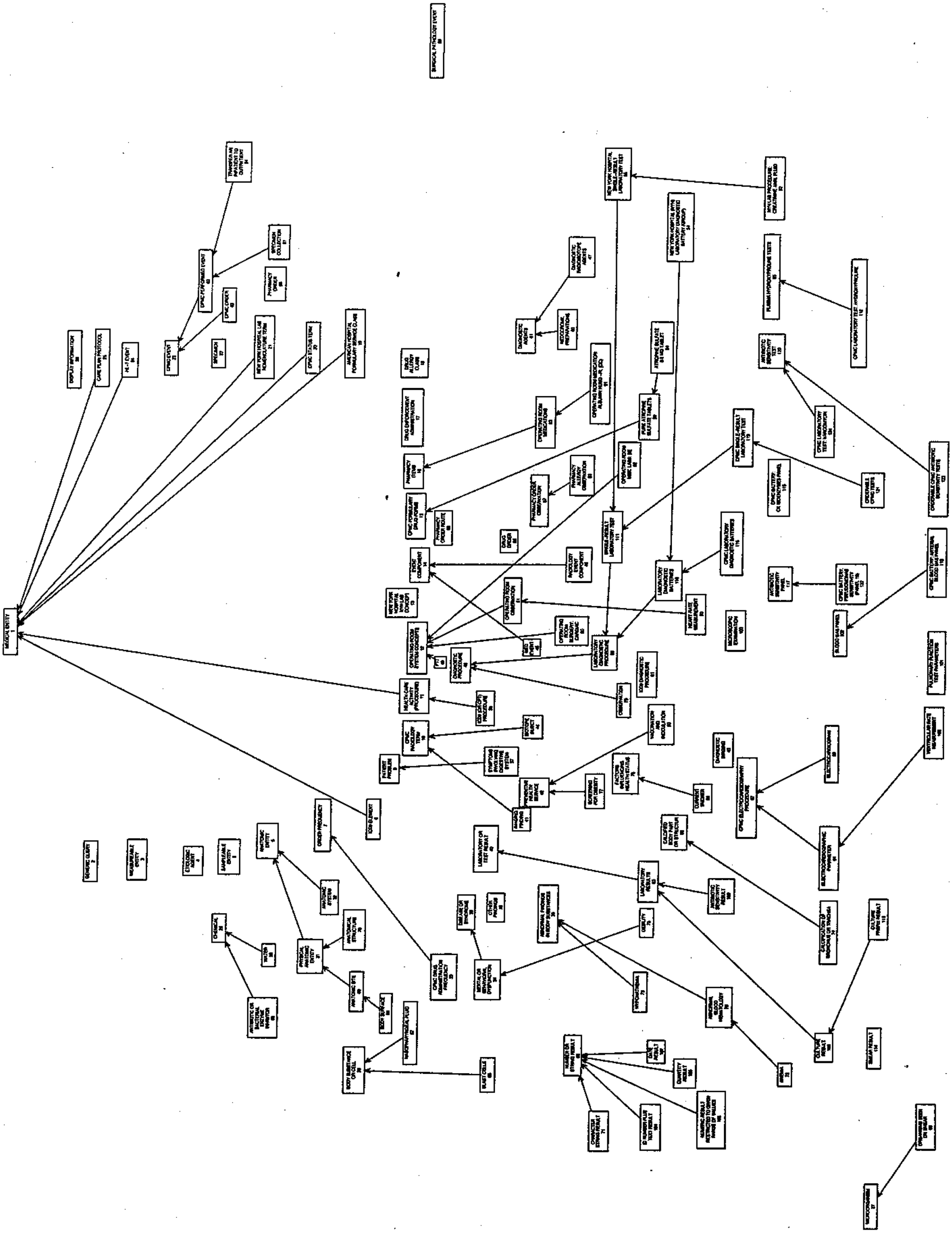

Figure 3: The forest structure of the MED schema after applying the methodology