

Modeling a Vocabulary in an Object-Oriented Database*

Li-min Liu

NJIT

limin@homer.njit.edu

<http://hertz.njit.edu/~lxl2242>

Michael Halper

Kean College of New Jersey

mhalper@turbo.kean.edu

<http://www.kean.edu/~mhalper>

Huanying Gu

NJIT

helen@homer.njit.edu

<http://hertz.njit.edu/~hxg5423>

James Geller

NJIT

geller@homer.njit.edu

<http://hertz.njit.edu/~geller>

Yehoshua Perl

NJIT

perl@homer.njit.edu

Abstract

Controlled vocabularies have been used as the means for unifying disparate terminologies found within an application field. This unification leads to better administration of information and enhanced communication among various parties. Semantic networks have been shown to be excellent vehicles for modeling controlled vocabularies. However, they often lack the necessary access flexibility and robustness required by external agents such as intelligent information-locators and decision-support systems. In this paper, we describe the process of mapping an existing medical vocabulary based on a semantic network model into an Object-Oriented Database (OODB) system. We first consider two straightforward approaches to carrying out this task and describe their deficiencies. We then present a new approach which yields a very compact OODB schema for the representation of the vocabulary's entire hierarchy and inter-connectivity. We refer to the resulting OODB as the Object-Oriented Healthcare Vocabulary Repository (OOHVR), which is currently up and running in the context of ONTOS, a commercially available OODB system.

1 Introduction

Starting from the first generation of semantic networks [Bra79, Sow91, Woo85] and semantic data models [HM81, HK87], attempts have been made to computerize the semantics of natural language terms. While most of these attempts were limited to small domains or "toy" applications, there have been a number of notable exceptions such as Cyc [LG90] and WordNet [Mil95]. Another large semantics-based vocabulary called the Medical Entities Dictionary (MED) has been developed in the healthcare arena [CCHJ94]. From an application standpoint, controlled vocabularies alleviate software systems of the burden of

maintaining their own *ad hoc* vocabularies. A common, centralized vocabulary also facilitates communication among applications by eliminating costly and time-consuming translation tasks. From a user point of view, they can help standardize information processing among different organizations and thus reduce the overall cost of doing business.

In this paper, we describe the process of modeling and implementing a semantic network-based controlled vocabulary as an Object-Oriented Database (OODB) [KL89, ZM90]. We have chosen to focus on an existing medical vocabulary called the InterMED, an offshoot of the MED [CCHJ94]. One reason for our choice is the fact that the healthcare field is one where such vocabularies are becoming ubiquitous and are being exploited in a wide variety of settings. Also, the InterMED employs a semantic network model based on a variation of the original model of Woods [Woo85], and, because of this, the mapping described in this paper can be readily applied to other vocabularies in the medical field (e.g., UMLS [U. 96]) and other domains. We refer to the OODB obtained by this mapping as the Object-Oriented Healthcare Vocabulary Repository (OOHVR). At present, a version of the OOHVR is up and running as an ONTOS [Sol92] database.

There are a number of reasons why one would want to model a vocabulary in an OODB. First, in applications where external agents such as intelligent information-locators, decision-support systems, and end-user browsers are demanding the knowledge stored in the vocabulary, transparent and concurrent access to it is necessary. OODB systems provide the traditional access support of database systems and offer a "low impedance" pathway [ZM90] to the network, particularly at a time when more and more application programs are being built using object-oriented programming languages. The vocabulary can also be accessed declaratively using an SQL extension (like OSQL of ONTOS [ONT95]) or a "path" language such as XQL [KKS92]. In addition, the typical OODB system's repertoire of modeling constructs neatly captures those modeling features of semantic networks used to describe a typical controlled vocabulary. Thus, the object model of the vocabulary can be mapped directly from the semantic

*This research was (partially) done under a cooperative agreement between the National Institute of Standards and Technology Advanced Technology Program (under the HIIT contract #70NANB5H1011) and the Healthcare Open Systems and Trials, Inc. consortium, and the Center for Manufacturing Systems.

network into the OODB system without having to re-model it from scratch. Such a mapping is described in detail in this paper. Another important point is that ONTOS and other OODB systems are fully supported by commercial vendors, something which is rarely the case for semantic networks.

An additional benefit of using the OODB framework turns out to be a concise picture of the overall hierarchy and connectivity of the InterMED's 3,000 terms afforded by the OODB schema. In [GCH⁺96], we showed how this representation readily uncovered several inconsistencies and errors in the original modeling of the MED which contains 43,000 concepts. This enabled the designers to easily correct the mistakes and refine the MED's content. In general, we would say that the OODB schema provides an additional vehicle for comprehending the vocabulary. While the vocabulary's content is preserved at the instance level of the database, the schema provides an extra level of classification and summarization of the overall vocabulary structure.

There are a number of ways to map a semantic network into an OODB environment. We will examine some straightforward ways and observe their disadvantages. The approach that we developed results in an OODB schema for representing the InterMED that is very compact in its size. In fact, the ratio of nodes in the InterMED to object classes in the OOHVR is on the order of eighty. This ratio will grow much larger as the InterMED grows to encompass additional medical concepts.

Previously, an object-oriented framework has been used as a modeling vehicle for thesauri for (natural) language-to-language translation [Fis91, Fis93]. A terminology editor called TEDI was built in the same context as a tool for extracting relevant information from hypermedia documents [MR93]. The O₂ OODB system has been used to store portions of a general English dictionary based on a "feature structure" description of its entries [IMV93].

Database technology has been exploited as a means for bringing persistence to knowledge-based systems. In [KMG95], the EXODUS object manager [CDRS89] is used as a subsystem of a frame representation system [KP95]. A storage model, based on techniques previously proposed for OODBs [VKC86], has been employed as the basis for storing Telos knowledge-bases on disk [MCP⁺94]. Both these efforts have sought to incorporate the database subsystem transparently, keeping it hidden from users. In contrast, we are directly employing a commercial OODB system for the representation of our vocabulary. After the mapping, the original semantic network in which the vocabulary was modeled is no longer needed. Users of the vocabulary, whether they be programmers or casual browsers, explicitly access the vocabulary through the standard mechanisms provided by the OODB system. Let us emphasize that it is not our goal to map *all* the modeling features of semantic network systems into the OODB environment. What we seek to do is represent controlled vocabularies, which ordinarily

require just a subset of such features. Note that the inheritance mechanism available in the semantic network is well mapped onto the OODB inheritance mechanism.

Besides the InterMED, the medical field has seen a number of standardized vocabularies such as SNOMED [Col82], ICD9-CM [Uni80], and MeSH [Nat]. A descriptive semantic network called Structured Meta Knowledge (SMK), employing a terminological knowledge-base, has been used to capture the semantics of patients' medical records [GGNR93].

The remainder of this paper is organized as follows. In Section 2, we give an overview of the InterMED. Section 3 covers the details of the mapping between the semantic network-based version of the InterMED and the OODB version (i.e., the OOHVR). In Section 4, we briefly discuss the ONTOS implementation of the OOHVR. For a full discussion of that implementation, see [LHG⁺96]. Section 5 contains the conclusions.

2 Description of the InterMED

In this section, we describe the InterMED, a controlled medical vocabulary modeled as a semantic network. The InterMED is the successor to the MED, which was developed and is presently in use at Columbia-Presbyterian Medical Center. It is being built as an inter-organizational vocabulary to be employed by various medical centers. Structurally, the InterMED is a semantic network whose nodes are medical concepts. Each node can have properties which are referred to as either attributes or relationships. An attribute is a property whose value is a primitive data type (such as a string). A relationship has as its value a reference to another concept in the network. One attribute common to all nodes is *name*, which holds a concept's associated *term* (or textual denotation). Another is *synonyms* which can hold alternate denotations aside from the primary one. Let us note that the InterMED exhibits nonredundancy, meaning that a given medical concept is represented by one and only one node. All synonyms are stored with that single node as a set-valued attribute. The choice regarding the primary denotation and the secondary denotations of a specific concept was made by the original designers of the vocabulary. An example of an InterMED relationship is *part-of* which links a concept to another concept of which it is definitionally a part.

The InterMED features a concept subsumption hierarchy—a directed acyclic graph (DAG) composed of concepts connected through super-concept (and sub-concept) links. This hierarchy acts as the property inheritance mechanism within the network. A sub-concept inherits all the properties of its superconcepts. For example, **Glucose Test** is a subconcept of (or, simply, "IS-A") **Test**, and therefore it inherits all of **Test**'s properties. In other words, the set of properties of **Glucose Test** is a superset of the properties of **Test**. A concept may have more than one parent concept. Also, the entire vocabulary hierarchy is rooted at a single concept called **Entity**. As we will discuss further below,

the hierarchy exhibits what we call sparse inheritance which influenced our design decisions.

The second purpose of the hierarchy is to support reasoning. Such a capability would be exploited, for example, by decision support systems that make subsumption-based inferences.

At present, the InterMED comprises about 3,000 medical concepts. This number is expected to increase into tens of thousands as the InterMED is extended over time to cover much of the current content of the MED. The concepts are linked by approximately 9,000 non-hierarchical (i.e., non-IS-A) relationships. The IS-A links total about 5,000.

Throughout this paper, a bold face font will be used for InterMED concepts. Properties of the InterMED will appear in italics and will be written strictly in lowercase letters. ONTOS (i.e., OOHVR schema) object classes will also be written in italics. These class names, however, will start with uppercase letters.

3 Representing a Semantic Network in an OODB

In this section, we describe the approach we used to represent the InterMED as an ONTOS database—a database we call the OOHVR. At first, we examine some straightforward schemes for carrying out such a mapping and discuss why these were ruled out. We then present our approach based on the pattern in which the properties are introduced within the semantic network.

3.1 OODB Modeling Alternatives

The InterMED is large collection of “concept” (or dictionary entity) nodes and links between these. One approach to constructing an OODB representation for the InterMED is to view all the nodes of the network uniformly: Everything is just a concept, so define a single object class *Concept* and make all nodes instances of it. All properties (i.e., attributes and relationships) defined with respect to concepts in the network would be made properties of this one class. This approach disregards the fact that different concepts may have highly disparate sets of properties, and these properties, in fact, carry much of the semantics of the network. For example, **Test** *measures* a **Substance** while **Heart Disease** *has-site* **Heart**. Clearly, it is absurd even to talk about the concept **Heart Disease** measuring something. However, folding all properties into a single class creates just such a potential in the OOHVR. Therefore, the single-class schema is conceptually incorrect. Besides, hiding the specific properties defined for a concept within the many more potential properties of the InterMED leads to an unnecessary overload of irrelevant information and, from a practical standpoint, is a waste of database storage.

As another shortcoming, consider what happens under this mapping to the InterMED’s concept subsumption hierarchy. To define a single class would mean “flattening” this

hierarchy and failing to exploit a fundamental aspect of object-oriented modeling.

A second alternative to the mapping of the InterMED exists when the target OODB system supports some kind of run-time manifestation of object classes in, say, its data dictionary. The ONTOS OODB, supports such a feature as part of what it calls its meta-schema [ONT95]. The approach is as follows. In a vocabulary network, it is reasonable to view each concept (i.e., node) as a description of a general category or, if you prefer, *class* in the linguistic sense. For example, the node **Aspirin** denotes the general concept of a kind of medicine (namely, Aspirin), not some specific tablets or preparations composed of it. With this in mind, we could map every concept into an object class of its own in the OOHVR schema. Then, the OOHVR would consist solely of classes (represented, presumably, as objects in the data dictionary) and their associated properties, but it would have no instances of those classes.

There are two major shortcomings to this approach. First, the OOHVR schema would be enormous, and instead of aiding a user in the comprehension of the vocabulary it would be overwhelming and nearly incomprehensible itself. The second problem has to do with the representation of properties and their values. In ONTOS, the properties of a class are stored as independent objects apart from the run-time manifestation of their class within the data dictionary. Because these “property” objects are strictly intensional, there is no provision for associating actual values with them as required by the vocabulary’s structure. To overcome this problem, we could augment the meta-schema with additional constructs to associate a property with values for the different concepts that possess it. (As discussed, a property defined at one concept appears at all its descendants, too, via inheritance. Each of these descendants would have its own value for the property in question.) Alternatively, duplicate property objects for a single property could be used to make the connection between concepts and values. This would result in the need for a great deal of extra storage within the data dictionary, and would also require its reorganization. Such practices, while feasible, are not acceptable.

As a third modeling alternative, we could define the OOHVR schema to consist of two classes: *Concept* and *Property*. The instances of the former would represent the InterMED’s concepts themselves, of which there are about 3,000. The instances of the latter would represent the properties of the concepts. For example, the concept **Glucose** would be represented by an instance of the class *Concept* in the OOHVR. Its properties *name* (which, as noted, is a property of all concepts, and here has the value “Glucose”), *synonyms*, etc. would each be separate instances of the class *Property*. A pair of relationships (i.e., *has-properties/is-property-of*) defined between the two classes would allow concepts and their properties to be associated.

While this scheme does offer flexibility, it has two major drawbacks. First, in order to access or manipulate a concept,

it would have to be “joined” together from its constituent objects (one instance of *Concept*, and many instances of *Property*). For the sake of efficiency and ease of use, we prefer to represent one complete concept as a single object in the OOHVR. The second drawback is a proliferation of database objects resulting from the fact that each property would be an object itself. For example, the property *name* being associated with every InterMED concept would alone require about 3,000 instances of *Property*.

3.2 Initial OOHVR Schema

In the first two approaches described above, either the vocabulary has all instances or all class definitions. The strategy that we chose is situated somewhere in the range between these two extremes. Some nodes serve as the basis for the definitions of object classes in the OOHVR schema, while all nodes are mapped directly into instances of those classes.

The question is: Which nodes of the InterMED will actually guide the definition of classes and their associated properties? Because the purpose of an object class is, among other things, to define the properties of its instances, it is sensible to examine the nodes of the network that also function in this role. Ultimately, our principal task is to identify groups of nodes that share identical properties so that we can define the OOHVR schema’s classes.

It turns out that there are only 30 concepts of the InterMED that introduce properties. We will call these *property-introduction nodes*. The rest just inherit their properties from other concepts. Because only 30 out of the nearly 3,000 nodes introduce new properties, we call the InterMED’s subsumption hierarchy a *sparse inheritance hierarchy*. Vocabularies, in general, by their very nature tend to have sparse inheritance hierarchies. This situation is in sharp contrast to the subclass hierarchy of a typical OODB schema where at almost every class we expect to find the definition of new properties.

Let us note that, in the InterMED, for each property there is a unique concept *C* where the property is first introduced. By inheritance, this property is also defined exactly for all the descendant concepts of *C*. Overall, the InterMED contains only 58 distinct properties.

Due to the above observations, the InterMED can be rather compactly divided up based on the properties exhibited by its various nodes. Toward this end, we need the following.

Definition 1: Let *V* be a property-introduction node. Some of the descendants of *V* are themselves property-introduction nodes (for other properties). Such a node *W* is called a *direct-property-introduction descendant* of *V* if the upwardly directed subsumption paths (there can be more than one) from *W* to *V* do not contain any other property-introduction nodes.

Definition 2: An *area* is a set of nodes containing a property-introduction node *V* and all its descendants which are not direct-property-introduction descendants of *V* or

descendants of the latter. In other words, an area includes a property-introduction node and all its descendants down to but excluding its direct-property-introduction descendants.

Clearly, an area contains a single property-introduction node. (Its property-introduction descendants define new areas of their own.) The property-introduction node of an area will be called its root, and will be used to denote the area.

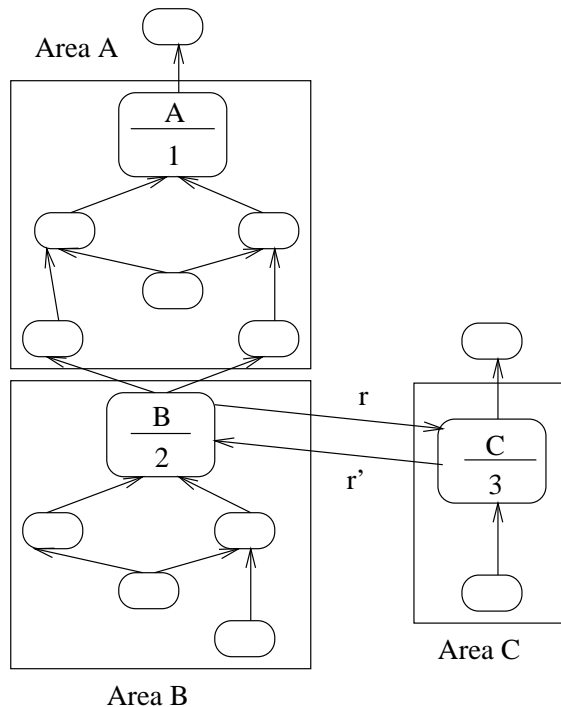


Figure 1: Three areas in a vocabulary

In Figure 1, we show three areas *A*, *B*, and *C* of a vocabulary. The nodes are represented as small rectangles with rounded edges, while the areas appear as large rectangles enclosing their respective nodes. Note that the root of area *A* (i.e., the node *A*) introduces the single attribute “1,” listed inside its rectangle. Area *A* extends down to, but excludes, node *B* which is a direct-property-introduction descendant of *A*. *B* defines the attribute “2” as well as the relationship *r* (drawn as a labeled arrow) and serves as the root of area *B*. Finally, area *C* has the root *C* which introduces attribute “3” and relationship *r'*, the converse of *r*. The IS-A links are drawn as unlabeled arrows directed from the sub-concept to the super-concept.

As a concrete example from the InterMED, the concept **Measurable Substance** introduces a new relationship *measured-by* and is thus the root of a new area. All descendant concepts between **Measurable Substance** and its direct-property-introduction descendants are in this “Measurable Substance” area. Examples of such concepts are **Color**, **Temperature**, **Specific Gravity**, **Viscosity**, **Blood Coagulation**, and **Optical Density**.

As we noted above, the InterMED has the concept **Entity**

as the root. **Entity** introduces a number of properties, and it therefore is the root of the “Entity” area. We will call this area the *root area* of the vocabulary. Because there are 30 property-introduction nodes in the InterMED, it is divided up into 30 areas. With respect to the overall size of the vocabulary—approximately 3,000 nodes—this is a very compact division. If all areas are mutually disjoint (i.e., no concept appears in more than one), then all nodes in an area will have the exact same properties (specifically those defined or inherited by its root). As such, areas will provide the partition we need to define the classes of the OOHVR. We will, in the remainder of this section, describe the OOHVR schema under the assumption that the areas of the vocabulary form a partition. In the succeeding subsection, we will discuss the additional complexity encountered when areas are not disjoint.

Under the assumption that areas are disjoint, we define the OOHVR schema as follows. For each area in the InterMED, we define an object class in the OOHVR whose instances will be exactly the concepts in that area, including its root. The class’s intrinsic properties are those defined by the area’s root. Because the extension of the class is precisely one area, we refer to it as an *area class*. Therefore, the OOHVR schema comprises area classes. The name of a class is formed by concatenating the name of the area’s root concept and “_Area.” So, the “Measurable Substance” area would have the corresponding class *Measurable_Substance_Area*. Its properties would include the relationship *measured-by*, among others.

Another issue that needs to be addressed is which area classes should be related via subclass relationships. Because the InterMED is singly rooted, each concept in the InterMED is a descendant of the **Entity** concept. Thus, the root of any area in the InterMED is a child of a node(s) in some other area(s). (The exception being **Entity** itself.) Thus, the root of an area has all the properties of its parents’ areas plus the properties that it intrinsically introduces. To capture this in our model, we place each area class corresponding to a root node in a subclass relationship with respect to the area class(es) of its parent(s). It should be noted that because a node (particularly a property-introduction node) may have more than one parent, the subclass hierarchy induced by this process is not necessarily a tree, as it may exhibit multiple inheritance. The class *Entity_Area* corresponding to the “Entity” area appears as the root of the OOHVR schema.

To illustrate this approach, we first show the result of mapping the three areas of Figure 1 into corresponding area classes and subclass relationships in Figure 2. Then in Figure 3, we show the entire OOHVR schema. Both figures were drawn using our OOdini-2 graphical notation which is based on a schema diagramming language presented previously in [HGPN93]. The pictures were produced with the OOdini-2 editor that is being built using the ObjectMaker Tool Development Kit of Mark-V. With OOdini-2, a class is represented as a rectangle, and a relationship, as a labeled

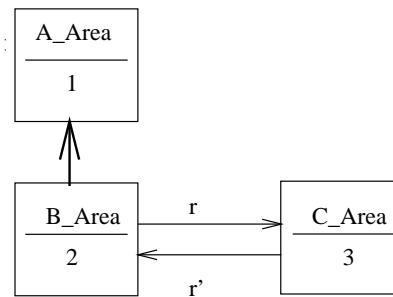


Figure 2: Areas classes corresponding to three areas in Figure 1

thin arrow. We denote a subclass relationship as a bold arrow directed upward from the subclass to its superclass. Attributes are listed inside their respective class rectangles beneath the class name. Let us emphasize again that the OODB schema produced by this mapping turns out to be very compact in terms of the number of classes, particularly when one considers that the InterMED contains thousands of concepts.

It is helpful to note that the InterMED’s concept subsumption hierarchy served as the basis for the mapping into the OOHVR schema. In fact, the mapping really constituted the identification of the property-introduction nodes and a “collapsing” of the inheritance paths between these concepts. Thus, the OOHVR schema can be seen as an abstraction of the property definitions and accompanying inheritance that occur within the InterMED. For this reason, we call this kind of schema for a sparse inheritance hierarchy a *network abstraction schema*.

However, if one is still to use the concept subsumption hierarchy of the vocabulary in the other ways that it was intended (e.g., in order to reason with respect to it), then it is mandatory that it appear in its entirety within the OOHVR. This is accomplished by introducing two reflexive relationships at the root area class *Entity_Area*: *has_superconcepts* and *has_subconcepts*. These properties are defined as follows. In the InterMED, if **X IS-A Y**, then, in the OOHVR, the object corresponding to **Y** is a referent of **X** with respect to the *has_superconcepts* relationship; *has_subconcepts* is the converse. In other words, the hierarchy of concepts in the InterMED is represented in the OOHVR on the instance (object) level rather than at the schema level. The schema of the OOHVR provides a compact framework for the definition and inheritance of all the properties of the concepts in the vocabulary. It thus helps the user of the vocabulary comprehend the vocabulary’s overall structure.

3.3 Extended OOHVR Schema

In the InterMED, some concepts assume membership in more than one area, thus violating the disjointness condition. This multiple membership is due to the fact that each such concept is subsumed by multiple parents (or other ancestors) that

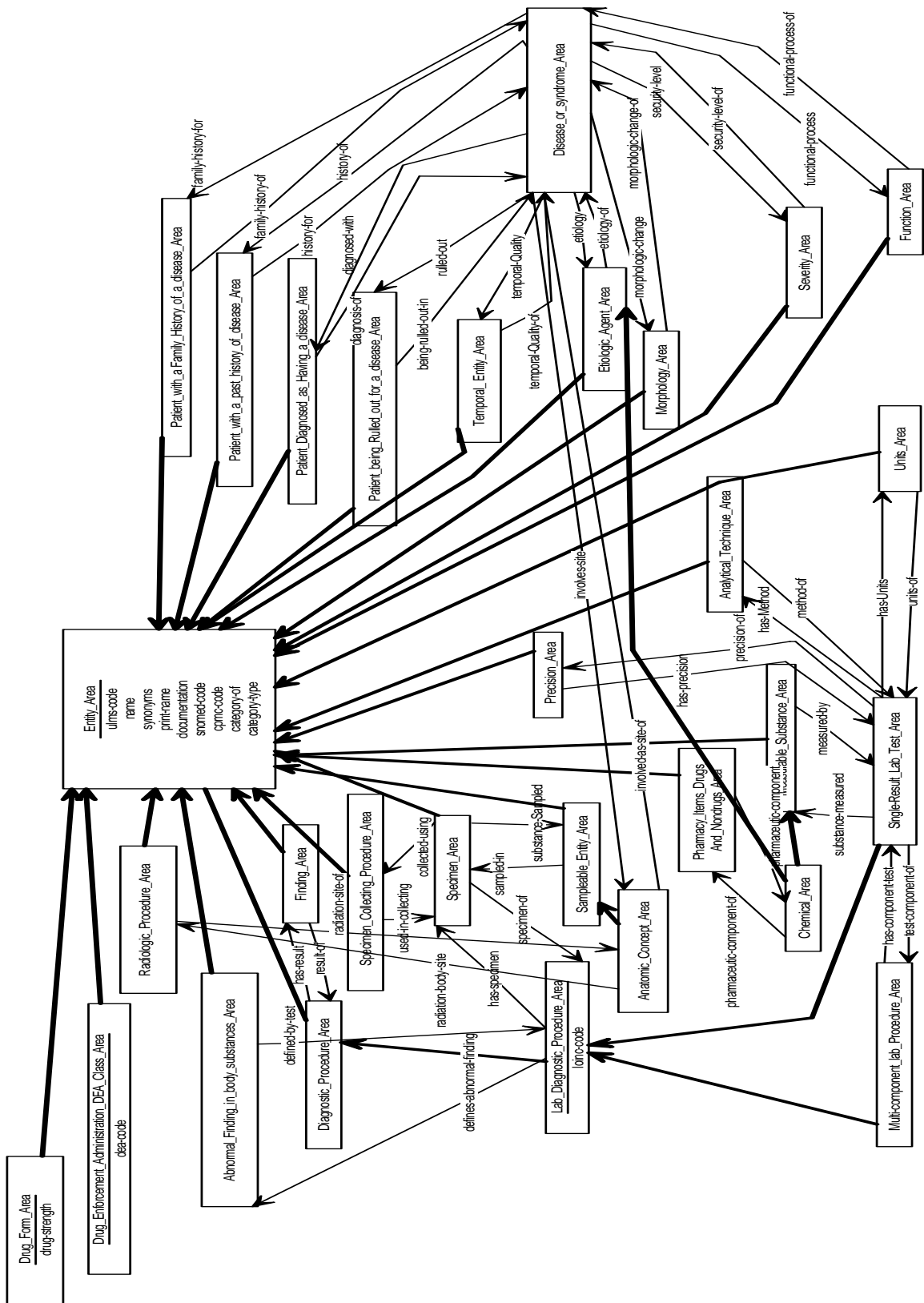


Figure 3: OOHVR Schema

reside in different areas. (Recall that this is possible because the vocabulary's concept subsumption hierarchy is a DAG, not a tree.) It should be noted that a node, say, **X** belonging to two (or more) areas cannot be a property-introduction node, otherwise it would be the root of some new area of its own. Instead, **X** would exhibit the combined properties from its multiple areas without introducing any properties that are new.

The question is how does this affect the mapping described in the previous section. To see the problem, let us assume that concept **X** resides in the two unrelated areas *A* and *B*. By "unrelated" we mean that neither *A*'s corresponding area class (i.e., *A_Area*) nor *B*'s (*B_Area*) is a descendant of the other in the OOHVR schema. **X**'s dual area membership implies that the object corresponding to it in the OOHVR must be an instance of both *A_Area* and *B_Area*. However, in the OODB paradigm (see, e.g., ONTOS [Sol92]), an object cannot be a "direct instance" of more than one class. Thus, we need to modify the mapping slightly in order to accommodate this scenario, which, as it happens, occurs infrequently within the InterMED.

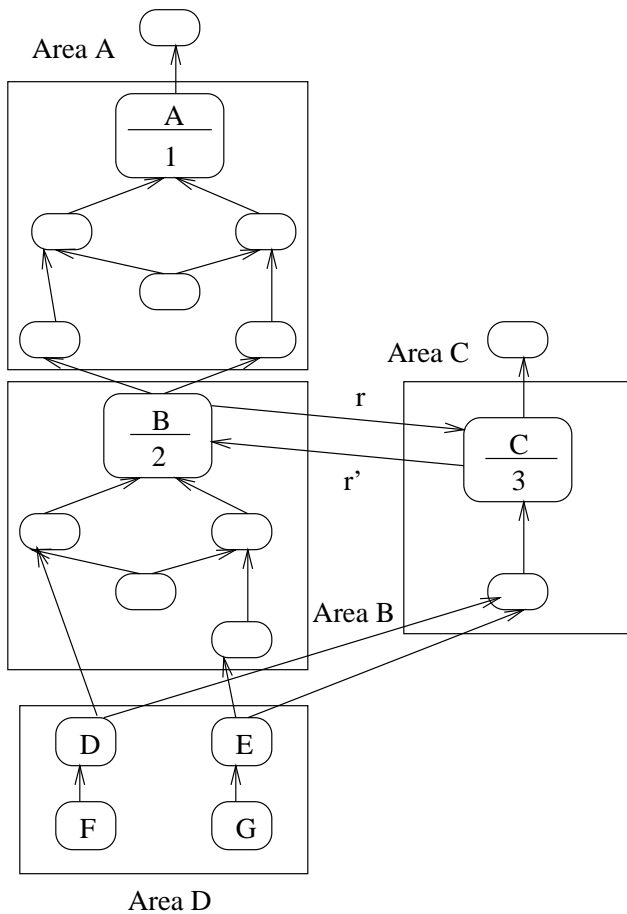


Figure 4: Expanded version of vocabulary from Figure 1

The problem described for **X** is true for any non-property-introduction node whose parents are in different areas or any

such node with an ancestor in that situation. In Figure 4, we expand the vocabulary pictured in Figure 1 to include four nodes that reside in both areas *B* and *C*. Nodes **D** and **E** both have parents in those areas, while **F** and **G** are in the areas by virtue of the fact that they are children of **D** and **E**, respectively. As a concrete example of this, the InterMED concept **Aspirin tablet preparations** resides in the two areas "Drug form" and "Pharmacy items, drugs, and non-drugs."

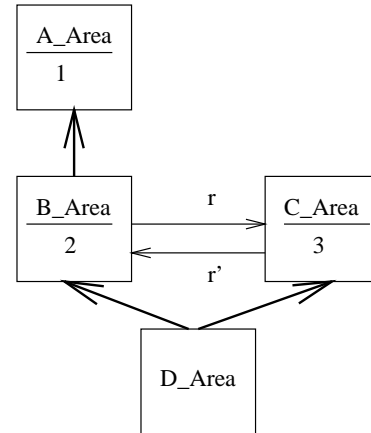


Figure 5: Areas classes corresponding to four areas in Figure 4

Our solution to the problem is to extend the notion of area and define the non-empty intersection of two areas as an area of its own, called an *intersection area*. As with all other areas in the vocabulary, a class is defined for it in the OOHVR schema. This new kind of class is referred to as an *intersection area class*. The concepts in the intersection area are made instances of this intersection area class, which does not introduce any new properties. Instead, the class gets its properties entirely via inheritance. The required set of properties is the union of exactly those properties of the two areas, the intersection of which the class models. Therefore, the intersection area class is defined as a subclass of the two area classes corresponding to those areas.

As shown in Figure 4, the intersection area might not have a root (i.e., a concept which is an ancestor of all others). If there exists a root **X**, then the corresponding intersection area class will naturally be denoted *X_Area*. Otherwise, the schema designer will have to select one of the concepts in the intersection which has parents in both areas as the name of the intersection area class. In Figure 5, we illustrate the result of the mapping that the intersection of the areas *B* and *C* (i.e., area *D*) from Figure 4 undergoes in the construction of the OOHVR schema. There, the name was chosen to be *D_Area*.

The notion of intersection area can be extended to encompass the intersection of three or more unrelated areas. In the InterMED, the "Acetaminophen/codeine tablet preparations" area is the intersection of three areas: "Pharmacy items (drugs and nondrugs)," "Drug enforcement administration (DEA) class," and "Drug form." Thus, the

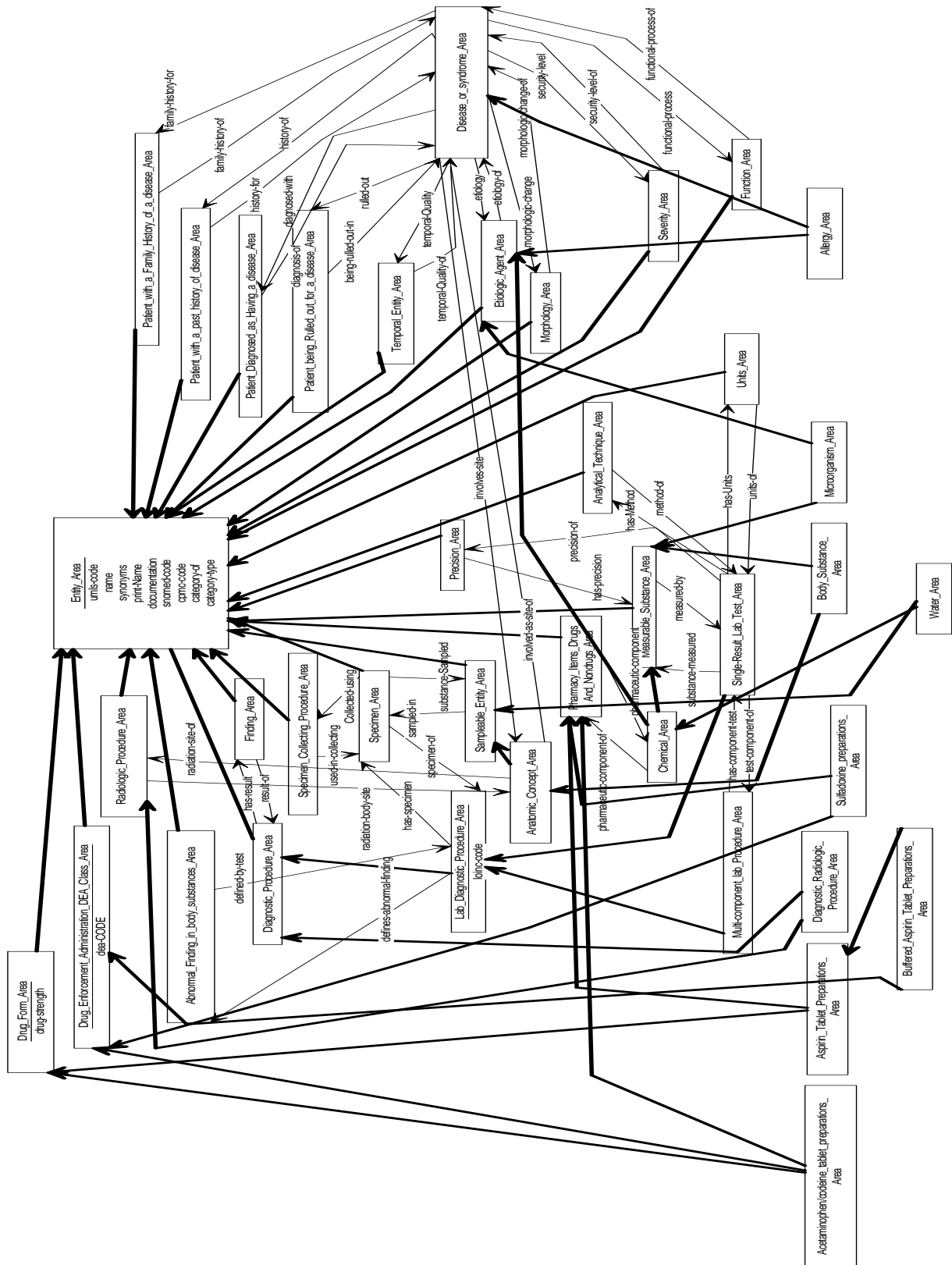


Figure 6: OOHVR schema including intersection area classes

intersection area class in this case has three parents in the OOHVR schema. It is also possible for an intersection area class to be a subclass of another intersection area class. For example, in Figure 6, area class *Buffered_Aspirin_Tablet-Preparations_Area* is a subclass of *Aspirin_Tablet-Preparations_Area* which is an intersection area class.

In Figure 6, we show the entire OOHVR schema, including all intersection area classes. The schema comprises a total of 39 area classes and 50 subclass relationships.

As we saw in [GCH⁺96], the OODB schema experiences a slow growth rate with respect to the size of the source vocabulary. For example, an OODB schema for the MED (which contains 43,000 concepts) required 90 classes, only about twice the number in the OOHVR's schema.

4 Implementation of the OOHVR

In the previous section, we described our mapping of the semantic network vocabulary into an OODB vocabulary. However, the InterMED is too large and the resulting OODB schema is too complex to consider creating the OOHVR by hand. Therefore, it is necessary to use a program to carry out this task. The program needs to perform two major functions: (1) the generation of the OOHVR schema (in this case, an ONTOS OODB schema), and (2) the actual populating of the OOHVR with the vocabulary concepts. We have built such a program, which we refer to as the *OOHVR generator*, that creates the OOVHR using ONTOS, a commercial OODB system. Due to lack of space, we omit a description of the program. See [LHG⁺96] for details.

5 Conclusion

Controlled vocabularies serve as excellent tools for the management of diverse terminologies within an application field. In this paper, we have described a representation of such a vocabulary in the context of an OODB system. While our discussions were centered around a medical vocabulary that we have implemented (the OOHVR), the techniques described are readily applicable to other vocabularies modeled with semantic networks. We would point out, however, that the framework presented herein is not currently suitable for building new vocabularies from scratch (i.e., without an existing source vocabulary to draw on). This issue is currently being investigated as part of our ongoing research.

In [LHG⁺96], we will present the architecture of a software system that can automatically transfer the contents of an existing semantic network-based vocabulary into an OODB. We have used this system to populate the OOHVR with the contents of the InterMED, an extensive, semantic medical vocabulary. Presently, the OOHVR is up and running within the context of the ONTOS OODB system.

Acknowledgments

We thank Jim Cimino from CPMC for his help in the interaction between the InterMED and the OOHVR, and for his continuous feedback on this research. We also thank Rajashekar Rao and Jignesh Dhruv for providing the schema figures using OOdini-2.

References

- [Bra79] R. J. Brachman. On the epistemological status of semantic networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, Inc., New York, NY, 1979.
- [CCHJ94] J. Cimino, P. Clayton, G. Hripcsak, and S. Johnson. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *Journal of the American Medical Informatics Association*, 1(1):35–50, 1994.
- [CDRS89] M. J. Carey, D. J. DeWitt, J. E. Richardson, and E. J. Shekita. Storage management for objects in EXODUS. In W. Kim and F. H. Lochovsky, editors, *Object-Oriented Concepts, Databases, and Applications*, pages 341–369. ACM Press, New York, NY, 1989.
- [Col82] College of American Pathologists, Skokie, IL. *Systematized Nomenclature of Medicine*, second edition, 1982.
- [Fis91] D. H. Fischer. Consistency rules and triggers for thesauri. *Int. Classif.*, 18(4):212–225, 1991.
- [Fis93] D. H. Fischer. Consistency rules and triggers for multilingual terminology. In *Proc. TKE'93, Terminology and Knowledge Engineering*, pages 333–342, 1993.
- [GCH⁺96] H. Gu, J. Cimino, M. Halper, J. Geller, and Y. Perl. Utilizing OODB schema modeling for vocabulary management. To appear in *Proc. 1996 AMIA Annual Fall Symposium*.
- [GGNR93] C. A. Goble, A. J. Glowinski, W. A. Nolan, and A. L. Rector. A descriptive semantic formalism for medicine. In *Proc. 9th ICDE*, pages 624–631, Vienna, Austria, 1993.
- [HGPN93] M. Halper, J. Geller, Y. Perl, and E. J. Neuhold. A graphical schema representation for object-oriented databases. In R. Cooper, editor, *Interfaces to Database Systems*, pages 282–307. Springer-Verlag, London, 1993.
- [HK87] R. Hull and R. King. Semantic database modeling: Survey, applications, and research issues. *ACM Computing Surveys*, 19(3):201–260, September 1987.
- [HM81] M. Hammer and D. McLeod. Database description with SDM: A semantic database model. *ACM Transactions on Database Systems*, 6(3):351–386, 1981.
- [IMV93] N. Ide, J. Le Maitre, and J. Véronis. Outline of a model for lexical databases. *Information Processing and Management*, 29(2):159–186, 1993.

- [KKS92] M. Kifer, W. Kim, and Y. Sagiv. Querying object-oriented databases. In *Proc. 1992 ACM SIGMOD Conference on Management of Data*, San Diego, CA, June 1992.
- [KL89] W. Kim and F. H. Lochovsky, editors. *Object-Oriented Concepts, Databases, and Applications*. ACM Press, New York, NY, 1989.
- [KMG95] P. D. Karp, K. Myers, and T. Gruber. The generic frame protocol. In *Proc. IJCAI-95*, pages 768–774, Montreal, Canada, 1995.
- [KP95] P. D. Karp and S. M. Paley. Knowledge representation in the large. In *Proc. IJCAI-95*, pages 751–758, Montreal, Canada, 1995.
- [LG90] D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley Publishing Co., Inc., Reading, MA, 1990.
- [LHG⁺96] L. Liu, M. Halper, H. Gu, J. Geller, and Y. Perl. Controlled vocabularies in OODBs: Modeling issues and implementation. In preparation, 1996.
- [MCP⁺94] J. Mylopoulos, V. Chaudhri, D. Plexousakis, A. Shrufi, and T. Topaloglou. Building knowledge base management systems: A progress report. Technical Report DKBS-TR-94-4, Department of Computer Science, University of Toronto, 1994.
- [Mil95] G. A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [MR93] W. Möhr and L. Rostek. TEDI: An object-oriented terminology editor. In *Proc. TKE'93, Terminology and Knowledge Engineering*, pages 363–374, 1993.
- [Nat] National Library of Medicine, Bethesda, MD. *Medical Subject Headings*. Updated annually.
- [ONT95] ONTOS, Inc. Lowell, MA. *ONTOS DB 3.1 Reference Manual*, 1995.
- [Sol92] V. Soloviev. An overview of three commercial object-oriented database management systems: ONTOS, ObjectStore, and O₂. *SIGMOD Record*, 21(1):93–104, March 1992.
- [Sow91] J. F. Sowa. *Principles of Semantic Networks, Explorations in the Representation of Knowledge*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1991.
- [U.96] U. S. Department of Health and Human Services, National Institutes of Health, National Library of Medicine. *Unified Medical Language System*, 1996.
- [Uni80] United States National Center for Health Statistics, Washington, DC. *International Classification of Diseases: Ninth Revision, with Clinical Modifications*, 1980.
- [VKC86] P. Valduriez, S. Khoshafian, and G. Copeland. Implementation techniques of complex objects. In *Proc. VLDB '86*, pages 101–109, Kyoto, Japan, August 1986.
- [Woo85] W. A. Woods. What's in a link: Foundations for semantic networks. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, pages 218–241. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1985.
- [ZM90] S. B. Zdonik and D. Maier, editors. *Readings in Object-Oriented Database Systems*. Morgan Kaufmann Publishers, Inc., San Mateo, CA, 1990.