Converting an Integrated Hospital Formulary into an Object-Oriented Database Representation

Huanying (Helen) Gu, Li-min Liu, Michael Halper[1], James Geller, Yehoshua Perl
CIS Dept., NJIT, Newark, NJ 07102
[1]Mathematics & Computer Science Dept., Kean University, Union, NJ 07083

Controlled Medical Vocabularies (CMVs) have proven to be extremely useful in their support of the tasks of information sharing and integration, communication among various software applications, and decision support. Modeling a CMV as an Object-Oriented Database (OODB) provides additional benefits such as increased support for vocabulary comprehension and flexible access. In this paper, we describe the process of modeling and converting an existing integrated hospital formulary (i.e., set of pharmacological concepts) into an equivalent OODB representation, which, in general, we refer to as an Object-Oriented Healthcare Vocabulary Repository (OOHVR). The source for our example OOHVR is a formulary provided by the Connecticut Healthcare Research and Education Foundation (CHREF). Utilizing this source formulary together with the semantic hierarchy composed of major and minor drug classes defined as part of the National Drug Code (NDC) directory, we constructed a CMV that was eventually converted into its OOHVR form (the CHREF-OOHVR). The actual conversion step was carried out automatically by a program, called the OOHVR Generator, that we have developed. At present, the CHREF-OOHVR is running on top of ONTOS, a commercial OODB management system, and is accessible on the Web.

INTRODUCTION

Controlled Medical Vocabularies (CMVs) [1]—systems of concepts that unify and consolidate disparate terminologies in the medical domain—have proven to be extremely important tools in many application areas. CMVs facilitate and enable information sharing, communication among programs, automated reasoning about medical knowledge, and the development of decision-support systems [2].

In previous work [3–5], we have introduced a methodology for representing CMVs as Object-Oriented Databases (OODBs). In general, we call such an OODB representation of a CMV an Object-Oriented Healthcare Vocabulary Repository (OOHVR). A major advantage of this OOHVR representation is the two-dimensional view of the source CMV that it affords. At the first level, called the schema level, a user is provided with an abstract view that aids in the comprehension of the overall structure and layout of the vocabulary. This view, in fact, has been utilized by the principal designer of the Medical Entities Dictionary (MED) [6] to uncover some errors in the MED's original modeling and improve its design [3]. At the second level, called the concept level, users can directly access objects which denote concepts and, in this way, obtain any desired detailed medical knowledge.

To complement our modeling methodology, we have developed a type of interface based on the two-dimensional view of the CMV. The interface also employs what we call *analogical forms* capturing pictorial diagrams through a form based interface [7]. With this interface, users can traverse the CMV at either the schema level or the concept level and readily switch among the two views. We have made a number of examples of this kind of interface available on the Web [7,8].

In this paper, we demonstrate the use of our OODB methodology by applying it to an integrated hospital formulary (i.e., collection of pharmacological concepts). The source formulary was supplied by the Connecticut Healthcare Research and Education Foundation (CHREF). As such, we call the resultant OODB the CHREF-OOHVR. The details of the CHREF-OOHVR's OODB model, the processing of creating it (which is done automatically by a program we have developed), and the CHREF-OOHVR's Web-based interface are described herein.

The formulary as originally provided to us did not meet the requirements of a CMV as laid out in [1] and was therefore not initially amenable to our conversion approach. Most important among its deficiencies was the fact that its concepts were arranged in a "flat" form. That is, there was no concept subsumption ("IS-A") hierarchy. This precluded such tasks as reasoning directly against the set of concepts.

To address this problem, we introduced an additional modeling step into our methodology that utilized the National Drug Code (NDC) system and its hierarchy of major and minor drug classifications. After imposing this hierarchical knowledge on the underlying formulary, we obtained a system of concepts that satisfied the characteristics of a CMV. At that point, we were able to apply our methodology. Details of all the steps are discussed below.

The rest of this paper is organized as follows. In the next section, we discuss the source formulary and how the NDC's hierarchy was imposed on it. We then describe the OODB schema that models the revised formulary. After that, the process of automatically generating the CHREF-OOHVR by our program called the *OOHVR Generator* is presented, followed by a discussion of the CHREF-OOHVR's Web Interface. The last section contains the conclusions.

BACKGROUND

In this section, we describe the source material for the CHREF-OOHVR. As noted, its basis is an integrated hospital formulary that was delivered to us as a table in a text file. The pharmacological concepts of the formulary each exhibit a common set of properties. These are: Hosp_ID, Drug_Code, NDC, Drug_Description, Dose_Unit, Unit_Cost, Bill_No, and Form_Status. In particular, let us point out the property NDC, a unique eleven character code to which other data elements are keyed. Another point that should be noted is the fact that the formulary does *not* contain any explicit hierarchical knowledge. That is, no notions of subconcept and superconcept are expressed. To rectify this situation, we consulted the National Drug Code (NDC) system.

The NDC system [9] was intended to provide a standard way of identifying drug products, using a number with three component parts: the labeler/vendor, product, and trade package size. The first part, the labeler code, is assigned by the FDA. A labeler is any firm that manufactures or distributes a drug product. The second part, the product code, identifies a specific strength, dosage form, and formulation for a particular labeler. The third part, the package code identifies trade package sizes. Both the product and package codes are assigned by the labeler. Together, these components form a unique number for every strength, dosage form, and package size of every drug. NDC serves as a universal product identifier for prescription drugs and a few selected OTC products.

An important element for our purpose is NDC's drug classification hierarchy. As we will discuss below, we will utilize this in the conversion of the given formulary into a CMV network representation and then into an OOHVR. This will result in a formulary that is more accurate, comprehensive, and easy-to-use in a variety of applications.

OBJECT MODEL

To model the CHREF source formulary as an OOHVR, there needs to be some hierarchical structure. Unfortunately, as we noted above, no hierarchical information is visible in the source formulary. To overcome this problem, we will integrate the CHREF source formulary and NDC's drug classification scheme in order to construct a semantic-network based CMV. After that, we will be able to map the CMV into its representation as an OOHVR. In this section, we describe that process. In the next subsection, we start off by imposing a hierarchical structure on the formulary. We then map the resultant CMV into an OODB schema.

Drug Classification Network

The NDC directory contains 6 text files, one of them being the Drug Class File which contains drug class numbers and drug classifications. Through linking elements in these files, every drug can be associated with one or more drug classes. The total number of such drug classes is 160. Since each class number contains 4 digits, we can decompose these numbers to get a two-level hierarchy out of the drug classification scheme. The classes, whose drug class numbers end with 00, are considered major classifications. For example, the class **ANTIMICROBIALS** has the class number 0300. Therefore, it is a major classification. All major drug classifications will be defined to sit in the first level of the formulary's concept hierarchy. Each is denoted by a node in the network. The number of major classes is 21.

All other classifications, totaling 139, are placed in the second level, called minor classifications. A minor class is defined to be the child of that major class whose class number has the same first two digits. For example, the minor classes **PENICILLINS** and **POLYMIXINS** have the class numbers 0346 and 0349, respectively. Both of them are children of **ANTIMICROBIALS** which has the number 0300. Since each minor class has exactly one major class as its parent, the two-level drug classification hierarchy has a forest structure; i.e., it is a collection of trees. After identifying the drug classification hierarchy, we link all drugs to one or more classes by manipulating the appropriate files provided by the NDC directory. All these drugs become the children of one or more drug class in the two-level hierarchy. Each drug is defined as a node in the resulting CMV. E.g., the drug **AMOXICILLIN 250 MG CAP** is linked to the minor class **PENICILLINS**. Due to this, it is placed in the network as a child of **PENICILLINS**.

After we obtain the drug classification network, we need to impose it on the CHREF source formulary. Since a CMV is assumed to be singly rooted [5], we first need to introduce an absolute root into the network. We call this root **DRUG**, and all major classifications in the network become its children. Once this is done, we can now link the 4177 drug concepts from the CHREF source formulary into the corresponding classifications in the network. Thus, the CHREF-OOHVR network will consist of 20 major classifications, 90 minor classifications, 4177 drug concepts, and a root named **DRUG** (see Figure 1). The difference from the original NDC is due to the lack of some classes in the drug set provided by CHREF.

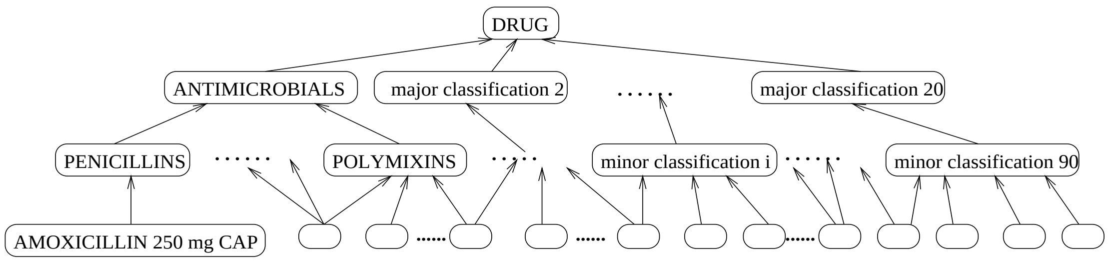

Figure 1: Classification network of CHREF source formulary

CHREF-OOHVR Schema

The CMV network developed in the previous subsection can now be mapped into an equivalent OODB representation which we call the CHREF-OOHVR. The first step in this process is the creation of an OODB schema.

At the outset, the root **DRUG** of the network is mapped into a corresponding root object class of the schema. Every major class in the network is also mapped into its own corresponding object class. Names are preserved in the mapping. That is, the name of the object class in the schema is identical to the name of the corresponding class in the network. All major object classes become subclasses of the root in the schema. E.g., the major object class **ANTIMICROBIALS**, mentioned above, becomes a subclass of the root object class **DRUG**.

Every minor class is also mapped into its own object class. These become subclasses of the corresponding major object classes. E.g., the minor object class **PENICILLINS** is a subclass of **ANTIMICROBIALS**.

Now, we need to deal with all the drug concepts themselves. We let every drug in the network be an instance of an object class which corresponds to its parent node in the network. The result is that all drugs which are linked to a single drug class node in the network become instances of the corresponding object class in the OODB schema. For instance, drug **AMOXICILLIN 250 MG CAP** is an instance of object class **PENICILLINS**.

One problem that arises with this approach is dealing with those drugs that fall under more than one drug classification in the network. For example, drug **MEPROBAMATE 400 MG TAB** is under both minor classifications **ANTIANXIETY** and **SKELETAL MUSCLE HYPERACTIVITY**. This is a problem because **MEPROBAMATE 400 MG TAB** cannot be made an instance of both object classes **ANTIANXIETY** and **SKELETAL MUSCLE HYPERACTIVITY** at the same time (a fundamental restriction imposed by OODBs). To solve this problem, we create a new type of object class, called an intersection class, which is the subclass of two or more existing object classes. E.g., we create the intersection class **ANTIANXIETY/SKELETAL MUSCLE HYPERACTIVITY** as a subclass of **ANTIANXIETY** and **SKELETAL MUSCLE HYPERACTIVITY**. The drugs linked to more than one class node in the network will become instances of such corresponding intersection classes. The drug **MEPROBAMATE 400 MG TAB** becomes an instance of the intersection class **ANTIANXIETY/SKELETAL MUSCLE HYPERACTIVITY** in the OODB.

At this point, all drug classification nodes in the network have been mapped into the CHREF-OOHVR schema as corresponding object classes. All drug nodes become the instances of those object classes. Overall, the CHREF-OOHVR schema contains 20 major object classes, 90 minor object classes, and 130 intersection classes (see Figure 2). All 4177 drugs provided by the CHREF source formulary become instances of the corresponding classes in the database. Since all drugs in the network have the same properties (namely, a set of attributes which can contain primitive data like integers or strings), we define the object class **DRUG** to have these properties. All other object classes in the schema inherit these properties via subclass relationships directly or indirectly from **DRUG**. In this way, all concepts in the CHREF-OOHVR exhibit the required set of attributes.

The resulting CHREF-OOHVR schema allows for the visualization of the 4177 drug concepts as 241 object classes. (In fact, this schema can accommodate more drug concepts if they are added into the source formulary.) This abstract view has provided valuable information to users and has served as an aid in comprehending and navigating through the CMV.

CREATION OF THE CHREF-OOHVR

In order to create the CHREF-OOHVR, we have utilized a program called the *OOHVR Generator* that we have built [5,10]. This program takes as its input a description of the source CMV (in this case, the source formulary transformed into a CMV network structure) and converts it automatically into the form of an OOHVR. Having such a program, of course, alleviates the burden of

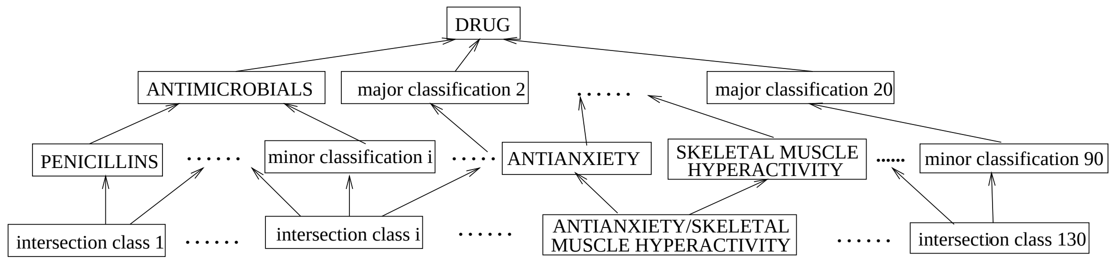

Figure 2: CHREF-OOHVR schema

creating the OODB by hand, a task that can be extremely tedious in the case of a large CMV.

The actual input to the program is assumed to reside in two specially formatted text files. If these files are not available for the source CMV, then a custom Preprocessor component is prepended to the OOHVR Generator so that the proper input files can be produced. Such a Preprocessor was, in fact, created for the task of handling the conversion of the formulary into the CHREF-OOHVR. This Preprocessor automated the task of combining the information supplied to us by CHREF and the NDC semantic hierarchy as described in the previous section. The original formulary came to us in some "flat" files. Using these and NDC's Drug Class File, the Preprocessor produced the two new text files that captured the formulary in its CMV configuration.

Aside from the Preprocessor, the OOHVR Generator contains three major components: (1) Schema Extractor, (2) Program Generator, and (3) Concept Loader. While the details of each are unimportant in this context, let us just note that in conjunction these components automatically build the requisite OODB schema, create an OODB that exhibits such a schema, and then populate the OODB to produce the final product: the OOHVR itself.

At present, the OOHVR Generator employs the ONTOS OODB management system, a commercially available product, as its implementation vehicle. Therefore, the output, i.e., the OOHVR, is an ONTOS OODB.

Let us emphasize that the OOHVR Generator is a general-purpose tool; it not only works for the CHREF-supplied formulary, but also for any CMV that can be modeled in a network structure [1,5]. Only the Preprocessor needs to be changed to handle different source formats.

Currently, the CHREF OOHVR is up and running on top of ONTOS. One can access it via sundry ONTOS utilities like DBDesigner or using an *ad hoc* query language like OQL [11]. In addition, we have built a Web-based interface [7] which we discuss in the next section.

A WEB INTERFACE

The main screen of the CHREF-OOHVR's Web-based interface is shown in Figure 3. This interface was built as a JAVA applet. In the middle and on the left-hand side, there are six windows that display aspects of the drug classification. These are labeled: Current, Children, Parents, Ancestors, Descendants, and Siblings. The Current window displays a selected drug class from the CHREF-OOHVR schema. The other five windows display, respectively, the selected class's subclasses, superclasses, ancestor classes, descendant classes, and sibling classes in the schema.

To the right of the Current window is the Terms window which shows a list of the drugs which are instances of the selected class. One of the items in the Terms window can be selected for the viewing of its property values, which appear in the Attributes window immediately above.

In Figure 3, we see that the Current class is **PENICILLINS ANTIBACTERIALS MISCELLANEOUS** and it has two drugs as its instances. These are **TIMENTIN** and **TICARCILLIN CLAVULATE**, the latter being currently selected for property viewing. Looking at the Attributes window, we see all of its properties' names along with their values. For example, **TICARCILLIN CLAVULATE**'s NDC (National Drug Code) is 00029657140.

CONCLUSIONS

The representation of a Controlled Medical Vocabulary as an Object-Object Database—a representation which we call an OOHVR—is very beneficial for a number of reasons. Among these is the increased comprehension promoted by the OOVHR's two dimensions of schema level and concept level. In this paper, we have described the process of converting an integrated hospital formulary into an OOHVR. Actually, our source formulary, provided by CHREF, was not in the form of a CMV. In particular, it lacked a hierarchical structure, one of the centerpieces of all CMVs. Utilizing the NDC directory, accessible via the Web, we first expanded the formulary into a hierarchical network structure. Thereafter, we were able to apply our previously developed methodol-

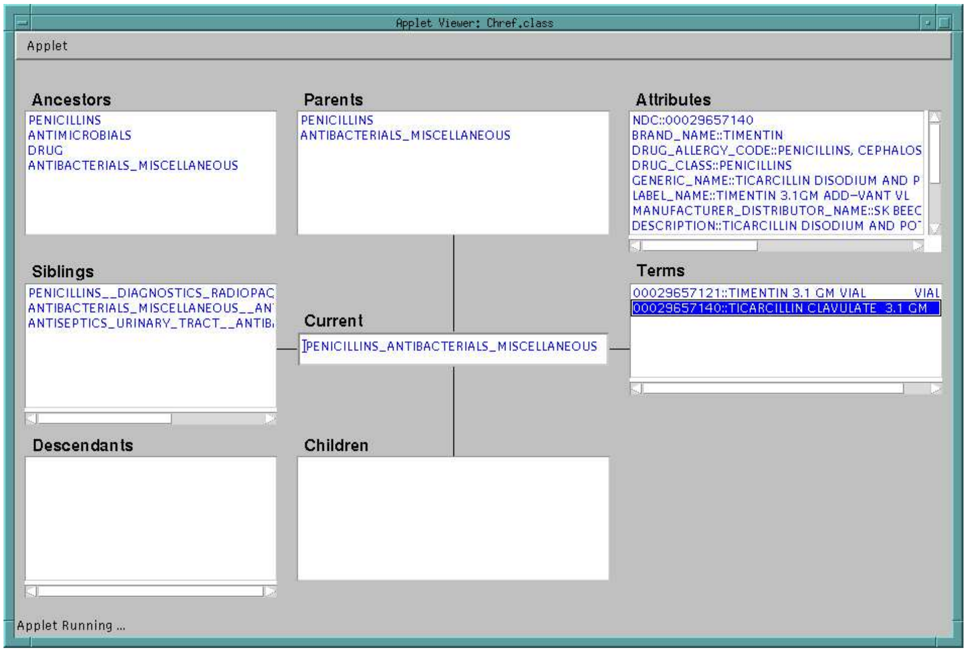

Figure 3: The CHREF-OOHVR Web Interface

ogy to construct an OODB schema and produce the CHREF-OOHVR with the formulary network as its content. The application of our methodology was carried out automatically by a program called the OOHVR Generator. Currently, the CHREF-OOHVR is running on top of ONTOS and is available on the Web through an interface we described.

Acknowledgments

This research was done under a cooperative agreement between the NIST Advanced Technology Program (under the HIIT contract #70NANB5H1011) and HOST, Inc. consortium.

References

1. J. J. Cimino, G. Hripcsak, S. B. Johnson, and P. D. Clayton. Designing an introspective, multipurpose, controlled medical vocabulary. In *Proc. Thirteenth Annual Symposium on Computer Applications in Medical Care*, pp. 513–517, Washington, DC, Nov. 1989.
2. B. H. Forman, J. J. Cimino, S. B. Johnson, S. Sengupta, R. Sideli, and P. Clayton. Applying a controlled medical terminology to a distributed, production clinical information system. In *Proc. '95 AMIA Annual Fall Symposium*, pages 421–425, 1995.
3. H. Gu, J. Cimino, M. Halper, J. Geller, and Y. Perl. Utilizing OODB schema modeling for vocabulary management. In J. Cimino, editor, *Proc. 1996 AMIA Annual Fall Symposium*, pp. 274–278, Washington, DC, Oct. 1996.
4. H. Gu, Y. Perl, J. Geller, M. Halper, J. Cimino, and M. Singh. Partitioning a vocabulary's IS-A hierarchy into trees. In D. R. Masys, editor, *Proc. 1997 AMIA Annual Fall Symposium*, pages 630–634, Nashville, TN, October 1997.
5. L. Liu, M. Halper, H. Gu, J. Geller, and Y. Perl. Modeling a vocabulary in an object-oriented database. In K. Barker and M. T. Özsu, editors, *CIKM-96, Proc. 5th Int'l Conference on Information and Knowledge Management*, pages 179–188, Rockville, MD, November 1996.
6. J. J. Cimino, P. D. Clayton, G. Hripcsak, and S. B. Johnson. Knowledge-based approaches to the maintenance of a large controlled medical terminology. *JAMIA*, 1(1):35–50, 1994.
7. M. Halper, R. Galnares, J. Geller, and Y. Perl. An analogical, Web-based interface to an OODB medical vocabulary. In preparation, 1998.
8. OOHVR Query Interface Project. URL: http://object.njit.edu:2000/~newoohvr-/FBI/QIP.html.
9. The National Drug Code Directory. URL: http://www.fda.gov/cder/ndc/index.htm.
10. L. Liu, M. Halper, J. Geller, and Y. Perl. Controlled vocabularies in OODBs: Modeling issues and implementation. Submitted for journal publication, 1998.
11. R. G. G. Cattell and D. K. Barry, editors. *The Object Database Standard: ODMG 2.0*. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1997.