MEDINFO 98
B. Cesnik et al. (Eds)
Amsterdam: IOS Press
© 1998 IMIA. All rights reserved

A Concept Model for the Automatic Maintenance of Controlled Medical Vocabularies

Juan Rodriguez, Victor Maojo, Jose Crespo, Isabel Fernandez

Medical Informatics Group. School of Computer Science
Universidad Politecnica de Madrid. Campus de Montegancedo. Boadilla del Monte
28660 Madrid, Spain

Abstract

A controlled medical vocabulary is a fundamental requirement in a range of medical informatics applications. Large vocabularies development and maintenance is labor intensive and costly. Maintainers of medical vocabularies need appropriate tools to do their work correctly. In this paper, we describe our concept model for a controlled medical vocabulary. We present how this model can check vocabulary consistency. We propose a set of tools in a distributed environment, which permits edition, visualization and maintenance of medical terminologies.

Keywords

Semantics; Nomenclature; Vocabulary

Introduction

Medical vocabularies are not static they are constantly evolving. The model by which they are created and maintained is an important factor in their consistency. There is an agreement about the properties of the model used for vocabulary representations [1,2,3,4]. These features are:

- Domain completeness. The model must not restrict terminology size.
- Consistency. There must not be discrepancy among concept definitions, concept positions in the hierarchies, and concept relationships. The model must support the implementation of methods for consistency verification.
- Extensibility. New information can be inserted and useless information deleted without loss of consistency.
- Nonredundancy. We need a mechanism that detects that multiple terms representing the same concept are unique concepts in the vocabulary.
- Synonymy. The model must support multiple terms associated to the same concept.
- Nonvagueness: There must not be partially defined concepts. All concepts in the vocabulary must have a complete meaning.
- Nonambiguity: Concepts must have only one meaning. Thus, the model must resolve ambiguous terms.
- Multiple classification: A concept can be a subclass of several classes; that is, it can be associated to several more generic concepts.
- Explicit relationships: Different inter-concept relationships must be defined and their meaning must be clear.

The Concept Model

Controlled medical terminologies require a deeper representation than the traditional tree structure [5,6,7,8].

We propose a model based on conceptual graphs [9]. The major structure of our model is composed of three hierarchies:

- a concept hierarchy
- a relation hierarchy
- a generalization hierarchy or conceptual graph hierarchy

The concept hierarchy

There is a partial ordering over concepts corresponding to an *is-a* hierarchy. All concepts are nodes in the concept hierarchy, as immediate descendants of at least one other node. One concept, called "Generic Entity" – represented by T – serves as the topmost node. Each concept may have several parents, but a concept may not be its own descendant. Thus, the concept hierarchy is a directed acyclic graph.

Each concept in the model has the following entries:

- identifier: a unique constant number.
- concept type: a unique name that can change during the maintenance of the vocabulary.
- referent: an individual marker or a variable.
- set of definitions: each definition is a conceptual graph, which determines the position of the concept in the hierarchy.
- Schemata set: each schema presents a characteristic of the concept. This characteristic does not determine the meaning of the concept.

Definitions and schemata are represented by conceptual graphs. A conceptual graph is a finite, connected, bipartite graph. The two kinds of nodes of the bipartite graph are concepts and conceptual relations. Every conceptual relation has one or more arcs, each of which must be linked to some concept. A single

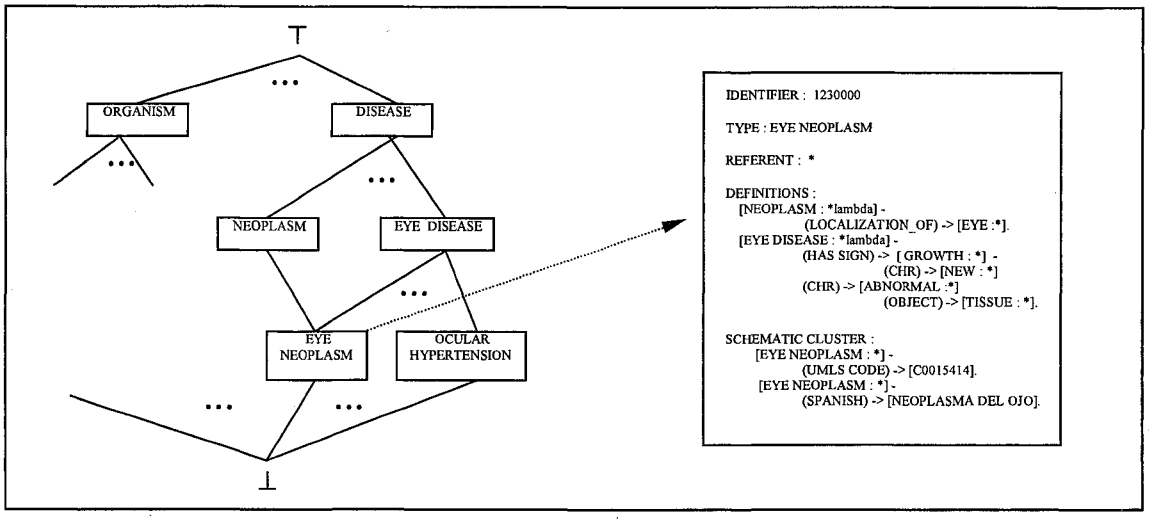

Figure 1 - An example of a Concept Hierarchy.

concept by itself may form a conceptual graph, but every arc of every conceptual relation must be linked to some concept.

Concepts are defined by an Aristotelian approach. Some superordinate concept is named as the *genus*, and a set of features, called the *differentia*, distinguish the new concept from the *genus*. We propose a system in which each concept inherits the distinguishing features of all its superordinates. As an example of concept definition, Figure 1 defines EYE NEOPLASM with *genus* EYE DISEASE and with a *differentia* graph that states a sign which is a new abnormal growth of tissue.

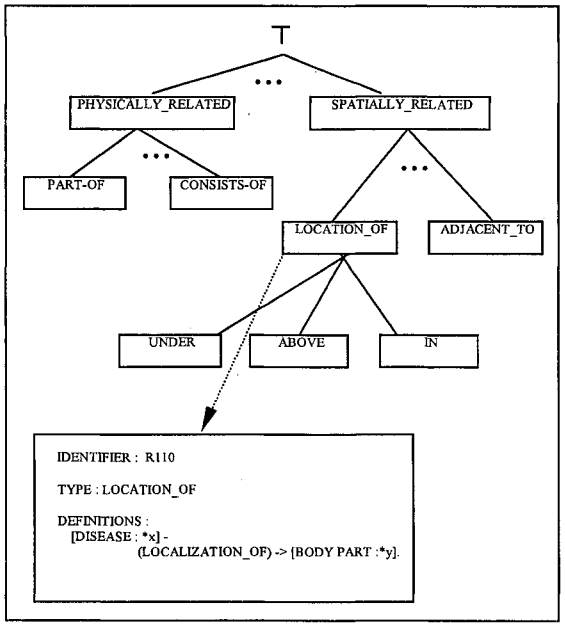

Figure 2 - An example of a Relation Hierarchy

Concept definitions permit automatic classification of concepts. If the definition is not present, the concept should be manually placed into the hierarchy.

Unlike concept definitions, which represent necessary conditions, a schema is not necessary for the meaning of the concept. Each schema is represented by a conceptual graph. Schemata are used for representing, for example, a concept role, a concept translation into several languages, or a concept translation into a coding system. Schemata are also used for representing synonymity or antonymity. This approach permits to separate the meaning of a concept from a particular language or coding system and does not impose a limitation in hierarchy's depth or breadth.

The relation hierarchy

Types classify conceptual relations in the same way that concepts are classified. A hierarchy is also defined over relations. Figure 2 shows a general relation labeled as LOCATION_OF that may have subtypes that specify more details about the location, such as IN, ABOVE, or UNDER.

In this partial ordering of relations, also the "Generic Entity" – represented by T – serves as the topmost node, but concepts have no common supertype with relations. Thus, both hierarchies are separated.

Each relation in the model has the following entries:

- identifier: a unique constant number.
- relation type: a unique name that identifies the relation.
- set of definitions: each definition is a conceptual graph, which determines the range of the concepts, which the relation can link.The generalization hierarchy

There are four formation rules for deriving a conceptual graph w from conceptual graphs *u* and *v*:

- Copy: *w* is an exact copy of *u*.
- Restrict: *w* is obtained replacing any concept *c* in *u* by a subtype.
- Join: if a concept *c* in *u* is identical to a concept *d* in *v*, then let *w* be the graph obtained by deleting *d* and link-

ing to c all arcs of conceptual relations that had been linked to d.

- Simplify: if relations r and s in the graph u are duplicates, then one of them may be deleted from u together with all its arcs.

If a conceptual graph w is derivable from a conceptual graph u using the formation rules, then u is called a generalization of w. Generalization defines a partial ordering of conceptual graphs called the generalization hierarchy. The graph [T] is a generalization of all other conceptual graphs; thus [T] is the topmost node in the hierarchy. The hierarchy consists only of concept definitions and schemata. Its purpose is to keep the consistency of the vocabulary, as well as be the core to handle the lexical information and to answer queries about implicit information.

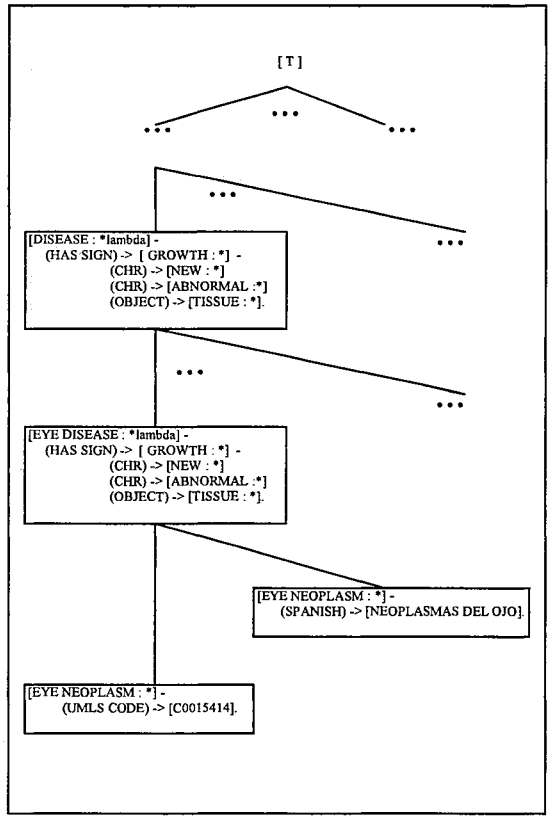

Figure 2 - An example of a Generalization hierarchy

Vocabulary Maintenance

The proposed model fulfills the properties exposed at the beginning of this paper.

- Domain completeness. Since the position of a concept in the hierarchy is not determined by the coding system, there are no limitations in hierarchy depth or breadth.
- Consistency. The consistency test in the proposed model is based on:

  1. Check validity of the graphs that define concepts. It must be tested that concepts in a graph agree with the range specified in the definition of the relations in that graph.

  2. Check that there are no cycles in the three hierarchies (concepts, relations and conceptual graphs). The presence of a cycle is equal to the existence of a contradiction.

  3. Check that concepts of a relation definition are subtypes of concepts that appear in the definition of its predecessor relations in the relation hierarchy.
- Extensibility. Every time information is added it is possible to check its validity; that is, to check that no new information entered contradicts information already present. When information is deleted, it is possible to reorganize the three hierarchies preserving the consistency of the model.
- Nonredundancy: Using formation rules, concept contraction, and concept expansion, it can be detected whether the definitions associated with different concepts are equivalent. Thus, we can check if the information in the model is redundant or not.
- Synonymy: Using conceptual schemata we could represent concept synonyms and we could also use different languages.
- Nonvagueness, nonambiguity: All concepts should have an associated definition and have a position in the hierarchy, which determines completely and without ambiguity their meaning.
- Multiple classification: A concept can be placed at several locations in the hierarchy.
- Explicit relationships: All the inter-concept relationships have a definition and a position in the relation hierarchy, which determines exactly their meaning.

With these properties, this model helps the maintenance operations in a controlled vocabulary. Some of these operations are:

- Add a concept.

  It is enough to facilitate the conceptual graph, which describes the concept definition. The maintainer will consult relations and concepts already defined, and add more relations as needed. Once the maintainer gives the graph, its validity will be checked automatically based on relation definitions and looking for an equivalent graph in the generalization hierarchy. If the graph is not valid, it is possible to determine where the inconsistency is located. If it is valid, it will be placed automatically in the generalization hierarchy and then, the concept defined by the graph will be placed in the concept hierarchy.
- Modify a concept.

  If a concept modification adds/deletes any synonym, code or language dependent term, the maintainer will create a schema for that concept or will modify an already existing schema. This information is not inherent

to the concept meaning so it has no influence in the information consistency. In this case, we only need to check that there are no duplications.

If the concept definition is modified, the new graph validity will be checked and if the graph is valid, its position in the generalization hierarchy will be analyzed automatically. Likewise, the location of the concept in the concept hierarchy will be checked and if there are changes in the position, every successor node of the concept will be analyzed.

- Remove a concept.

  A concept deletion will remove the concept definition and all of its associated schemata. In this case, the generalization hierarchy and the concept hierarchy will be reorganized. The maintainer must confirm the deletion in order to modify the definitions of the concepts that were subtypes of the removed concept, as well as the definitions of the relations where the erased concept appeared.

- Add a relation.

  To add a relation into the model, the maintainer should give the associated definition and put the relation in the relation hierarchy. After this, definition validity can be checked.

Implementation

Since the three hierarchies in the model have inheritance as an important property, we have chosen an object-oriented representation. We are also developing software using CORBA [10], adopting a component-based, distributed approach.

The CORBA types, which have been defined, are:

- concept
- relation
- conceptual_graph. This object has the following methods : copy, restriction, maximal join, projection, simplification, contraction, and expansion
- POSET_structure. The methods associated to this object are: exist (look for an element in the hierarchy), insert (add a new node in the hierarchy), remove (delete an element from the hierarchy), predecessors (immediate predecessors of a node), successors (immediate successors of a node).
- concept_hierarchy
- relation_hierarchy
- graph_hierarchy.

These three hierarchies are POSET structures (Partially Ordered Sets) and they inherit all the methods associated to a POSET_structure.

Conclusion and future work

Investigation on representation methods of medical concepts continues active. Mechanisms for vocabulary maintenance are crucial for the success of controlled vocabularies.

We are currently implementing the proposed model. Our aim is to create a vocabulary server. In a first stage we are developing a set of tools which permit vocabulary creation and maintenance. We plan to develop tools to permit users to navigate and to consult lexical and semantic information stored in the vocabulary server.

Acknowledgments

This research has been funded by the Fondo de Investigacion Sanitaria (FIS), Ministry of Health, Spain (FIS 95/1952).

Computing resources were provided by Hewlett-Packard Initiative HISE 96 (*Healthcare Information Systems Engineering*).

References

[1] Cimino, J.J.: Data Storage and Knowledge Representation for Clinical Workstations. Int J Biomed Comput, 1994; 34:29-44.

[2] Cimino, J.J., Clayton PD, Hripcsak G, Johnson SB. Knowledge-based Approaches to the Maintenance of a Large Controlled Medical Terminology. J Am Med Informatics Assoc. 1994;1:35-50

[3] Rector AL, Nowlan WA, Glowinski A. Goals for Concept Representation in the GALEN project. In Safran C, ed. Proc of SCAMC 93. New York: McGraw-Hill, 1993: 414-418.

[4] Friedman, C.;Huff; S.M.;Hersh, W.R.;Pattison, E.;Cimino, J.J.Ó The Canon Groups Effort: Working Toward a Merged ModelÓ. J. Am Med Informatics Assoc. 1995; 2: 4-18

[5] Cimino JJ. Saying what you mean and meaning what you say: coupling biomedical terminology and knowledge. Acad Med 1993; 68 (4): 257-60.

[6] Nowlan WA, Rector AL, Kay S, Horan B, Wilson A. A patient care workstation based on user centered design and a formal theory of medical terminology: PEN&PAD and the SMK formalism. In: Clayton PD, de. Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care. New York: McGraw-Hill, 1992:855-7.

[7] Rector, A.L." Medical-concept Models and Medical Records: An Approach Based on GALEN a PEN&PAD". J Am Med Informatics Assoc. 1995; 2:19-35

[8] McCray, A."The UMLS semantic network". 13th annual symposium on computer applications in Medical Care. Washington, DC:IEEE Computer Society Press,1989:503-507

[9] Sowa, JF. Conceptual Structures: Information Processing in Mind and Machine. Reading, MA: Addison-Wesley Publishing Company, 1984.

[10] Siegel, J. "CORBA Fundamentals and Programming". Willely & Sons. NewYork 1996

Address for correspondence

Juan Rodriguez Pedrosa
Grupo de Informatica Medica.
Departamento de Inteligencia Artificial
Facultad de Informatica de la UPM
Campus de Montegancedo. Boadilla del Monte. 28660 Madrid. Spain
Tfno: +34 1 336 68 97
e-mail: jrodriguez@infomed.dia.fi.upm.es