ACADEMIC
PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Biomedical Informatics 35 (2003) 194–212

Journal of
Biomedical
Informatics

www.elsevier.com/locate/yjbin

The cohesive metaschema: a higher-level abstraction of the UMLS Semantic Network

Yehoshua Perl,[a,*] Zong Chen,[a] Michael Halper,[b] James Geller,[a] Li Zhang,[a] and Yi Peng[a]

[a] Computer Science Department, New Jersey Institute of Technology, Newark, NJ 07102, USA
[b] Department of Mathematics and Computer Science, Kean University, Union, NJ 07083, USA

Received 3 October 2002

Abstract

The Unified Medical Language System (UMLS) joins together a group of established medical terminologies in a unified knowledge representation framework. Two major resources of the UMLS are its Metathesaurus, containing a large number of concepts, and the Semantic Network (SN), containing semantic types and forming an abstraction of the Metathesaurus. However, the SN itself is large and complex and may still be difficult to view and comprehend. Our structural partitioning technique partitions the SN into structurally uniform sets of semantic types based on the distribution of the relationships within the SN. An enhancement of the structural partition results in cohesive, singly rooted sets of semantic types. Each such set is named after its root which represents the common nature of the group. These sets of semantic types are represented by higher-level components called meta-semantic types. A network, called a metaschema, which consists of the meta-semantic types connected by hierarchical and semantic relationships is obtained and provides an abstract view supporting orientation to the SN. The metaschema is utilized to audit the UMLS classifications. We present a set of graphical views of the SN based on the metaschema to help in user orientation to the SN. A study compares the cohesive metaschema to metaschemas derived semantically by UMLS experts.

Keywords: UMLS; Semantic Network; IS-A relationship; Metaschema; Partitioning; Semantic type; Abstraction

1. Introduction

The Unified Medical Language System (UMLS) [1–3] is a large knowledge representation system that combines many medical terminologies. The UMLS can be used to overcome problems caused by discrepancies in different terminologies [4–6]. Two resources of the UMLS are the Metathesaurus (META) [7,8] containing medical concepts and the Semantic Network (SN) containing semantic types. The UMLS's enormous size and complexity (871,584 concepts in the Metathesaurus in the year 2002 edition of the UMLS [9]) can pose serious comprehension and orientation problems for potential users [10].

The UMLS's Semantic Network (SN) [11–13] helps to orient users [14] to the vast knowledge content of META. The SN is composed of a set of 134 semantic types which categorize the concepts of META. Each concept in META is assigned to one or more semantic types in the SN. Overall, the SN's semantic types are arranged in a hierarchy of IS-A relationships. In addition, there are 53 other kinds of (semantic) relationships which connect semantic types.

However, the "small" SN can still be too large and complex for orientation and comprehension purposes. Typically, a convenient way for a user to get oriented to such a knowledge structure is by studying a diagrammatic representation. People often prefer a graphical representation to an equivalent textual form, which may be quite extensive and unruly. (As is said: "A picture is worth a thousand words.") The image allows for human information processing at a very high bandwidth. ([15] refers to "the well-known high bandwidth of the human-vision channel.") Moreover, the image facilitates oper-

[*] Corresponding author.
E-mail addresses: perl@oak.njit.edu (Y. Perl), zong@ylem.njit.edu (Z. Chen), mhalper@kean.edu (M. Halper), geller@oak.njit.edu (J. Geller), lxz1853@njit.edu (L. Zhang), yxp1898@njit.edu (Y. Peng).

1532-0464/02/$ - see front matter © 2002 Elsevier Science (USA). All rights reserved.
doi:10.1016/S1532-0464(02)00528-2

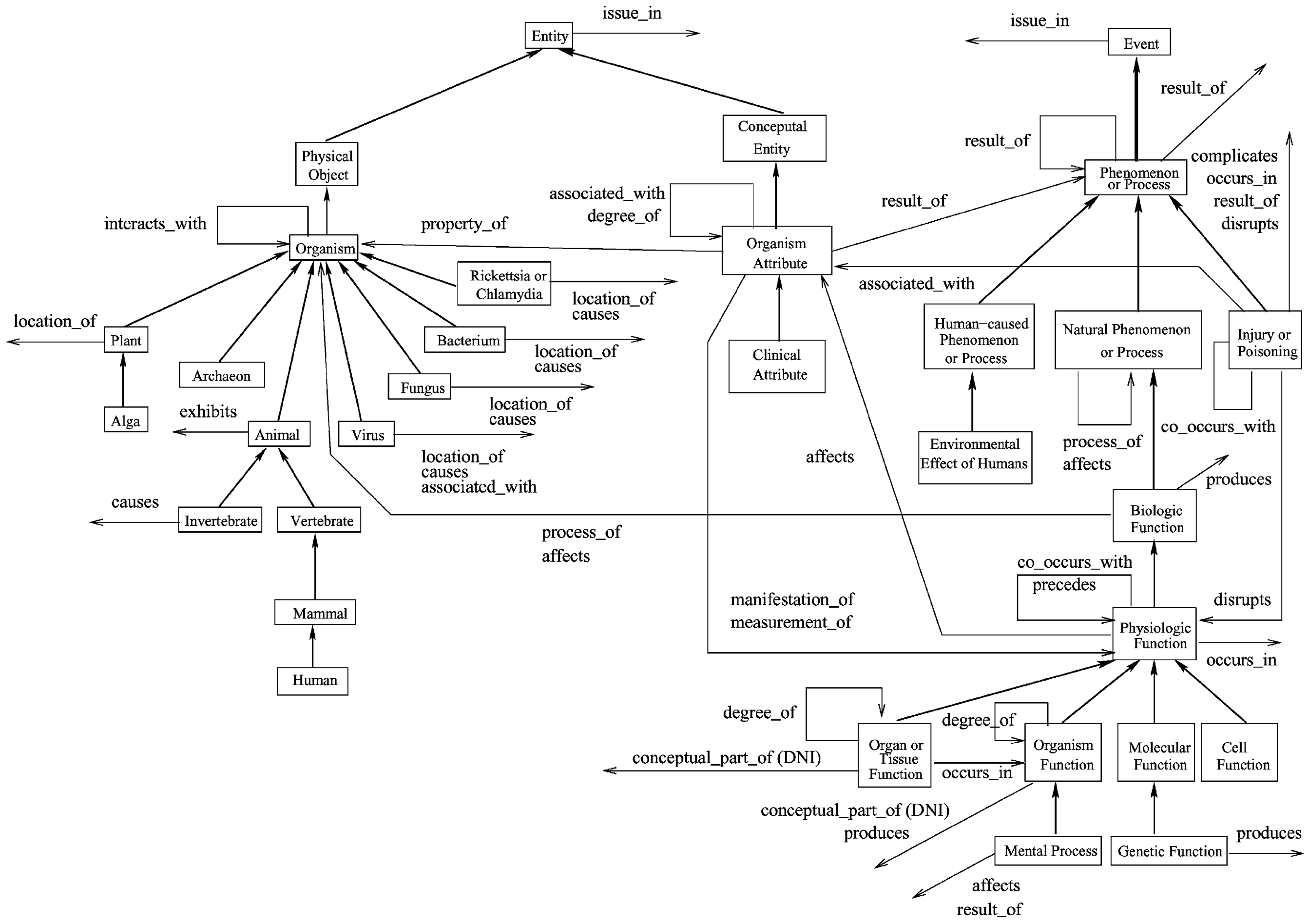


Fig. 1. A portion of the UMLS SN.

ations that are hard-wired into our visual systems, such as focusing on a detail, or detecting and following an edge. However, the knowledge is typically not committed to memory. Thus, any operation that relies on the graphical representation is hampered in its effectiveness if part of the image is obstructed. For example, if a diagram spans several pages, then this built-in functionality cannot be used effectively because only part of the image is available simultaneously. During scrolling, it becomes difficult to follow an edge from one part of the image that is not in the field of vision anymore to another part of the image that is not yet visible. Similarly, it is difficult to focus on a detail in a jumping image. Therefore, only a static graphical representation that captures the whole image on one screen will be of maximum benefit. Ideally, we would like to provide such a graphical view of the knowledge structure.

To give an idea of the complexity of the SN, we show a portion of it in Fig. 1 with 32 semantic types, 30 IS-A relationships, and 51 (semantic) relationships. Note that the figure displays neither the incoming relationships from semantic types out of the scope of the figure nor the inherited relationships of the semantic types. (Note that for a semantic type appearing outside the scope of the figure, the relationship is “dangling.”) For example, the semantic type **Virus**[1] has—in addition to the three outgoing relationships shown—two inherited relationships *issue_in* and *interacts_with*, and nine incoming relationships of four different kinds, *process_of*, *indicates*, *associated_with*, and *location_of*, from semantic types outside the figure. Fig. 1 uses a graphical notation where rectangles represent semantic types, IS-A relationships are represented by bold arrows, and other relationships appear as labeled thin arrows.

In this paper, we concentrate on providing comprehensible access to the SN through simpler and more compact views which fit easily onto a single screen. Such a need is even more urgent in light of a refined UMLS object-oriented database schema of 1296 classes which we created as an extension of the SN [16]. Specifically, we will present, in Section 2, a technique for partitioning

[1] Let us note some typographical conventions used throughout the paper: A semantic type will be written in a bold font. The name of a semantic relationship will be written in italicized lowercase letters.

the SN based on its relationship configuration. (From now on, whenever we use "relationship" we mean a semantic relationship rather than IS-A.) Considering, in Section 3, some modifications in our partitioning technique leads to a revised methodology that partitions the SN into cohesive, semi-structurally uniform collections of semantic types abstracted as *meta-semantic types*. The outcome of our technique, presented in Section 4, is a new, higher-level abstract metaschema which provides a powerful viewing mechanism for the SN. The advantages of the metaschema are demonstrated in Sections 5 and 6. In the first, we describe auditing the classification of META's concepts. In Section 6, we provide views for supporting UMLS users in orientation to the SN. An evaluation study is described in Section 7, and a discussion appears in Section 8. Conclusions are in Section 9. An early short version of this paper appeared in [17].

2. Structural partitioning

Since our partitioning technique is based on the distribution of the relationships among the semantic types of the SN, let us look closely at the structure of the SN's relationships. In the SN, the IS-A hierarchy supports the inheritance of the semantic relationships among semantic types. When two semantic types are linked via IS-A, the child inherits all the relationships defined for the parent. For example, the relationship *process_of* is stated to hold from the semantic type **Biologic Function** to the semantic type **Organism**. Therefore, it also holds from its child semantic type **Physiologic Function** to **Organism**.

By the transitivity of the inheritance, a relationship is introduced at a given semantic type and is inherited by all its descendants (unless the inheritance is interrupted, as we shall see, by the DNI or blocking designation). E.g., all descendants of **Phenomenon or Process** inherit *result_of*, which is introduced at that point. In Fig. 1, we do not draw the inherited relationships in order to avoid clutter, since the information on those relationships can be deduced from the inheritance. However, when a semantic type inherits a relationship from its parent and the target semantic type is refined, we show the inherited relationship explicitly. For example, **Organ or Tissue Function** inherits the relationship *occurs_in*, defined at its parent **Physiologic Function** with the target **Temporal Concept**. However, **Organ or Tissue Function** defines a new target, **Organism Function**, for the *occurs_in* relationship.

The UMLS provides two additional modeling features that affect the inheritance of relationships. The first feature, called "blocking," nullifies the definition of an inherited relationship. **Mental Process** and **Plant** are descendants of **Biologic Function** and **Organism**, respectively. By inheritance, **Mental Process** would be *process_of* **Plant**. Since plants are not sentient beings, this relationship is defined as "blocking" between these two semantic types.

The second feature allows a newly introduced relationship to be designated as "defined but not inherited" ("DNI"), which means the relationship is not inherited by any of the children of the semantic type that is introducing it.

The focus of our approach is placed on the relationships because of their overall semantic importance.

Definition (*Structure of semantic type*). The *structure* of a semantic type is the set of its defined relationships, whether they are introduced directly or inherited. It is denoted *Structure(A)*, where *A* is a semantic type.

For example, the semantic type **Entity**, one root of the SN hierarchy, only introduces the relationship *issue_in*; therefore, $Structure(\textbf{Entity}) = \{issue\_in\}$. **Physical Object** inherits **Entity**'s *issue_in* relationship and does not introduce any new relationship of its own. Thus,

$$\begin{aligned} Structure(\textbf{Physical Object}) &= Structure(\textbf{Entity}) \\ &= \{issue\_in\}. \end{aligned}$$

Organism inherits *issue_in* from **Physical Object** and introduces a new relationship *interacts_with* directed to **Organism** itself, thus $Structure(\textbf{Organism}) = \{issue\_in, interacts\_with\}$.

To define our partitioning technique, we need the following:

Definition (*Structurally uniform*). Let *A* and *B* be semantic types. If $Structure(A) = Structure(B)$, then *A* and *B* are called structurally uniform.

Definition (*Semantic-type group*). A *semantic-type group* is a set of all semantic types with the exact same set of relationships.

Hence, in a semantic-type group, each pair of semantic types are structurally uniform. The identical nature of their relationship structure suggests that they bear a close resemblance in meaning. A similar idea is expressed in [18]: "Semantic validity may also be measured by an analysis of the relationships in which the semantic groups participate." It is therefore justified to group them together along that dimension of similarity to form a higher-level conceptual abstraction. All semantic types exhibiting the exact same set of relationships are grouped together. See [19,20] for an example of modeling a schema using structural similarity of concepts for the MED (Medical Entities Dictionary) [21].

Definition (*Root of a semantic-type group*). A semantic type is a root of a semantic-type group if none of its parents belong to the semantic-type group.

Definition (*Partition*). A partition is a collection of disjoint sets of semantic-types such that their union yields all the semantic types of the SN.

Definition (*Structural Partition*). The partition of the semantic types into semantic-type groups is called the structural partition.

Clearly, a semantic type which introduces a new relationship will be a root of its semantic-type group. Most, but not all, semantic-type groups have a unique root. If a semantic-type group has a unique root, then all other semantic types in the group are its descendants.

As stated above, **Entity** introduces the relationship *issue_in* and therefore starts a new semantic-type group. *Structure*(**Physical Object**) = *Structure*(**Entity**). Hence, **Physical Object** belongs to **Entity**'s semantic-type group. **Organism** introduces a new relationship *interacts_with* and thus starts a new semantic-type group. In another example, **Mental Process** introduces two new relationships *affects* and *result_of*. Therefore, **Mental Process** starts a new semantic-type group. Since the relationship *process_of* is defined as "blocking," this relationship will not be included in the structure of **Mental Process**. See Fig. 2 (where semantic-type groups with more than one member are enclosed in bubbles) for the portion of the structural partition pertaining to the subnetwork in Fig. 1. This portion consists of 16 semantic-type groups.

There are, in the structural partition of the semantic types, cases of semantic-type groups with multiple roots. In our discussion, we will concentrate on those appearing in Fig. 2. One example contains the sibling semantic types **Organ or Tissue Function** and **Organism Function**. Both inherit all relationships of their parent **Physiologic Function** and introduce the new relationship *degree_of*. A similar situation occurs with the three semantic types **Bacterium**, **Fungus**, and **Rickettsia or Chlamydia**, which are the children of **Organism**. These three introduce the relationship *causes* to **Pathologic Function**, and the relationship *location_of* to **Biologically Active Substance**. Hence, they form a semantic-type group with three roots.

For the entire SN, there are 71 semantic-type groups. Of these, 47 contain just one semantic type. (We call such groups "singletons.") We note that 45 of 47 singletons are leaves (semantic types without children). Such leaf singletons are expected based on genus/differentia. Eleven groups have two semantic types; five groups have three semantic types; three groups have four semantic types; two groups have five semantic types; one group has six semantic types, and one other has eight. Finally, there is one group with 14.

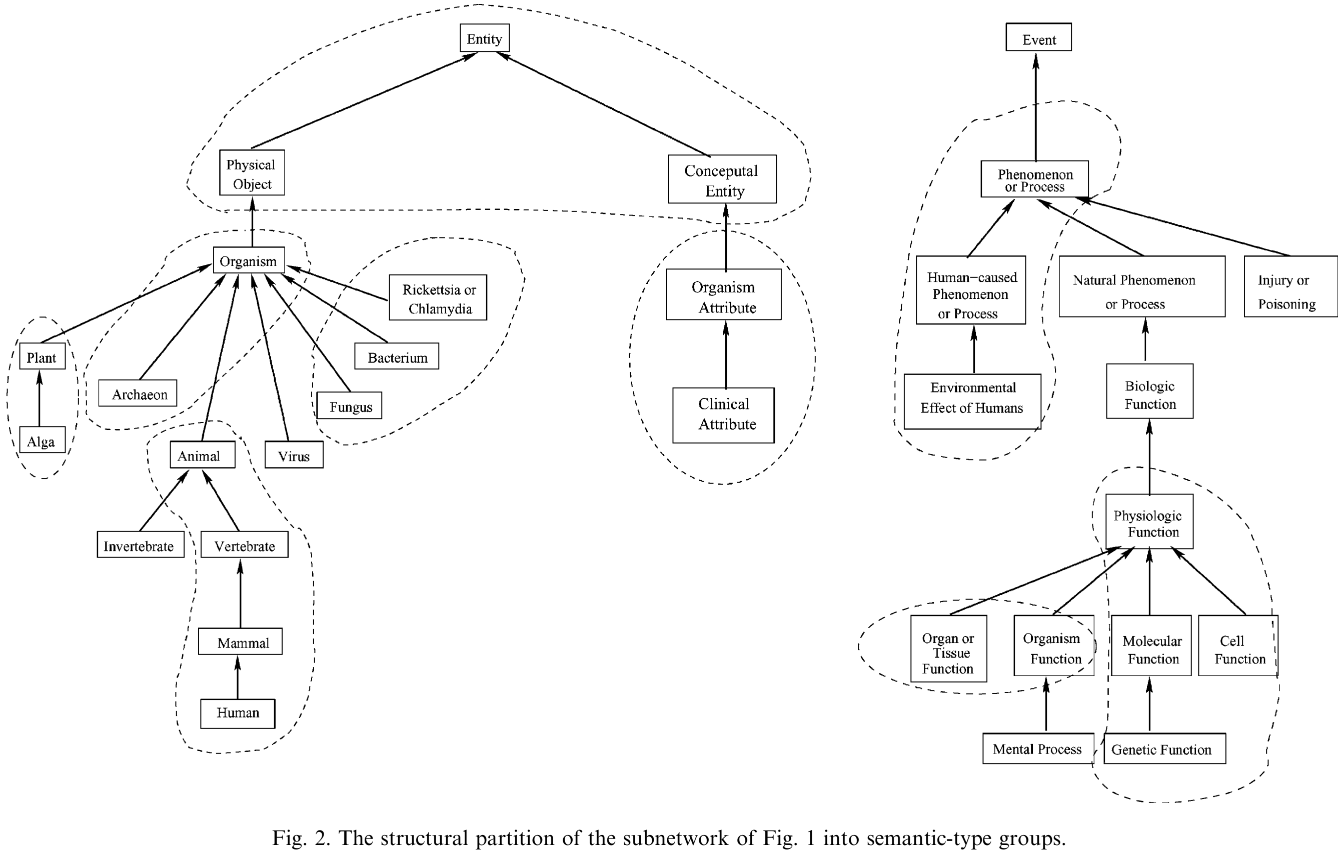


Fig. 2. The structural partition of the subnetwork of Fig. 1 into semantic-type groups.

Note that a semantic type S which introduces a DNI relationship is a root of a group since its parent semantic type's structure does not contain this relationship. On the other hand, this relationship is not contained in the structure of any child semantic type of S because of the lack of inheritance due to the DNI designation. We call such a semantic type a *DNI-root*.

In a structural partition, a DNI-root is either a singleton or, in the case where other semantic types share the same structure, a root in a multi-rooted group. In Fig. 2, we have one such multi-rooted group, containing two DNI-roots, **Organ or Tissue Function** and **Organism Function**, both of which introduce the same *conceptual_part_of* (DNI) relationship. Similarly, a semantic type with a blocked relationship cannot be in its parent's group.

3. Cohesive partitioning

In an effective partition of the SN, necessary later in the definition of a metaschema, a group of semantic types should not just be structurally uniform but also cohesive, i.e., unified in the sense that the semantic types belong together.

Definition (*Cohesive group*). A group is called cohesive if it has a unique root, i.e., one semantic type which all other semantic types in the group are descendants of.

We say that a group of semantic types with a unique root is *singly rooted*. The importance of a singly rooted group is its cohesion derived from the fact that each one of the semantic types in the group is a specialization of the unique root. Hence, by naming a singly rooted semantic-type group after the root, this name properly reflects the overarching nature of the group. The possibility of naming a semantic-type group in such a way is critical for the definition of the metaschema in the next section. As we see in Fig. 2, most of the semantic-type groups have unique roots. This phenomenon shows that structurally uniform groups tend to be cohesive most of the time, but not all the time. This tendency supports the observation from [18] mentioned earlier: "Semantic validity may also be measured by an analysis of the relationships in which the semantic groups participate."

Definition (*Cohesive partition*). A partition into cohesive groups is called a cohesive partition.

Since cohesion is required for the metaschema definition, we will provide, in this section, rules and an algorithm to transform the structural partition into a cohesive partition. For this transformation, we will need to make some trade-offs, meaning some multi-rooted groups will lose their structural uniformity in order to become singly rooted. However, the new sets will still have "semi-structural uniformity" (defined below).

Another problem with the structural partition is its large number of singletons. Let us recall that the metaschema's purpose is to provide an abstract, compact view of the SN that desirably can fit legibly on one computer screen. In other words, the metaschema should manifest some size reduction, i.e., substituting several semantic types by one meta-semantic type. Clearly, singletons do not contribute to this since one semantic type is substituted by one meta-semantic type. As we shall see, a leaf singleton does not contribute to the metaschema's structure, e.g., as a branching point, either. If a singleton contains only a leaf semantic type (i.e., a semantic type without children, like **Virus**), there is no contribution by such a semantic-type group to a size-reduced view of the SN. Thus, we will provide a rule to add leaf singletons to their parent's semantic-type groups to decrease the number of singletons in the partition. Again, this implies creating sets which are not structurally uniform, since those singletons were induced due to structural differences. The rule that we provide will, nevertheless, result in new sets that have semi-structural uniformity.

The cohesive partition that will emerge from applying our rules to the structural partition will be composed of *semantic-type collections*. Each semantic-type collection is a singly rooted set of semantic types in the SN. Each such collection is named after its root.[2] Some semantic-type collections are also semantic-type groups. Others have semi-structural uniformity.

Rule 1. Each semantic-type group with a non-leaf unique root becomes a semantic-type collection and is named after its root.

Rule 2. If a leaf semantic type L is a singleton in the structural partition, then L is added to its parent's semantic-type collection.

Applying Rule 2 helps to merge many singleton semantic-type groups into larger semantic-type collections. For instance, the singleton containing the leaf **Invertebrate** (Fig. 2) is combined with the semantic-type collection *Animal* to produce a new semantic-type collection with five members (Fig. 3).

For the cases of multi-rooted semantic-type groups, we need to introduce an additional rule.

[2] The name of a semantic-type collection will be written in an italicized font, with uppercase letters appearing at the beginning of significant words. The same convention will be used for the name of a "meta-semantic type," defined in the next section.

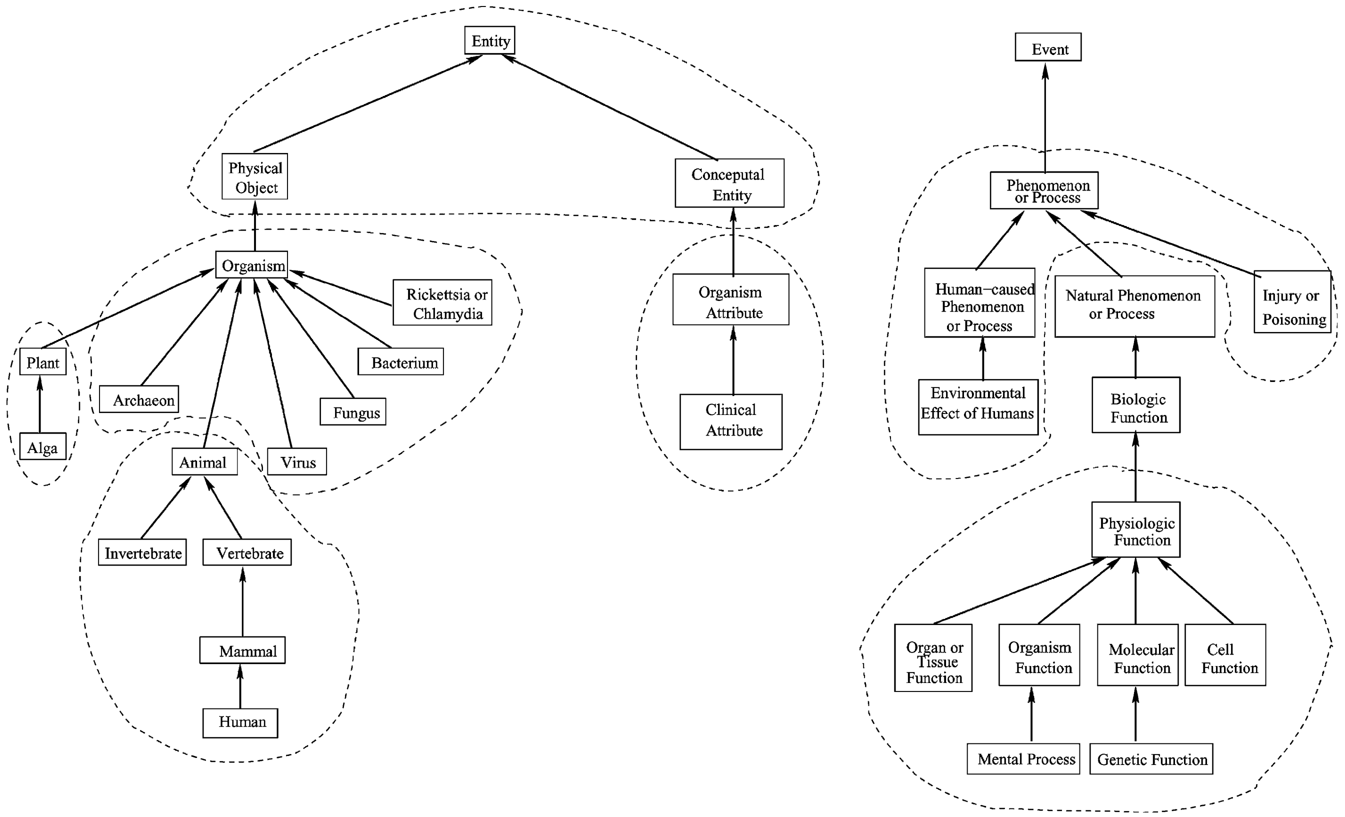


Fig. 3. Cohesive partition of Fig. 1 into semantic-type collections.

Rule 3. Let the semantic types $A_1, A_2, \ldots, A_n$ ($n \geqslant 2$) be roots of the same semantic-type group G of the structural partition. If there exists a lowest common ancestor A of $A_1, A_2, \ldots, A_n$ in the IS-A hierarchy, then add all the semantic types of G to the semantic-type collection of A. If this common ancestor is one of these roots, say A_i, then this is a case where another of these roots say A_j is a descendant of A_i. This can only happen if on the IS-A path from A_j to A_i there exists a semantic type B which belongs to another group. Add the group which contains the semantic type B to the collection of the group G.

Note that the reason that such an intermediate semantic type B does not belong to the same group containing $A_1, A_2, \ldots, A_n$ is that it is a DNI-root. That is, it introduces a DNI relationship and thus B has more relationships than A_i, but its descendant A_j does not inherit this DNI relationship and thus has the same structure as A_i. Also if no lowest common ancestor exists, then Rule 3 does not have any effect.

Consider the multi-rooted semantic-type group containing the semantic-type leaves **Bacterium**, **Fungus**, and **Rickettsia or Chlamydia**. According to Rule 3, it is added to the collection rooted at **Organism**, their common parent, which already was assigned **Virus** (formerly a singleton) according to Rule 2.

To demonstrate the case where one root is a descendant of another root in a multi-rooted group, consider the following example, with these three groups: the first group containing **Entity**, **Physical Object**, **Conceptual Entity**, and **Classification**, the second group containing **Intellectual Product**, and the third group containing **Regulation or Law** (see Fig. 4). The semantic-type **Intellectual Product** is a DNI-root with two chil-

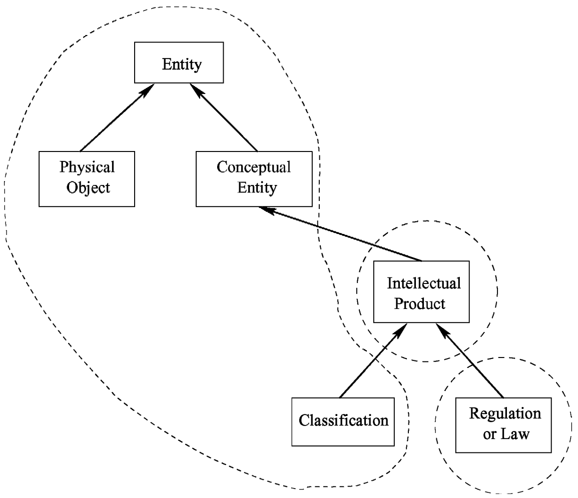


Fig. 4. An extract of the structural partition of the SN.

Table 1
Semantic-type collection list

Collection	*Size*	*Semantic Types in Collection*	*# Concepts*
Anatomical Abnormality	3	Anatomical Abnormality; Congenital Abnormality; Acquired Abnormality	10,790
Anatomical Structure	2	Anatomical Structure; Embryonic Structure	1001
Animal	9	Animal; Invertebrate; Vertebrate; Amphibian; Bird; Fish; Reptile; Mammal; Human	11,634
Behavior	3	Behavior; Social Behavior; Individual Behavior	1192
Biologic Function	1	Biologic Function	179
Biologically Active Substance	7	Biologically Active Substance; Receptor; Vitamin; Enzyme; Neuroreactive Substance or Biogenic Amine; Hormone; Immunologic Factor	60,633
Chemical	16	Chemical; Chemical Viewed Functionally; Hazardous or Poisonous Substance; Inorganic Chemical; Biomedical or Dental Material; Element, Ion or Isotope; Carbohydrate; Indicator, Reagent or Diagnostic Aid; Chemical Viewed Structurally; Organic Chemical; Organophosphorus Compound; Steroid; Eicosanoid; Amino Acid, Peptide, or Protein; Lipid; Nucleic Acid, Nucleoside, or Nucleotide	327,486
Entity	8	Entity; Physical Object; Conceptual Entity; Group Attribute; Language; Intellectual Product; Classification; Regulation or Law	7107
Event	4	Event; Activity; Daily or Recreation Activity; Machine Activity	809
Finding	3	Finding; Lab or Test Result; Sign or Symptom	47,841
Fully Formed Anatomical Structure	6	Fully Formed Anatomical Structure; Cell; Cell Component; Tissue; Gene or Genome; Body Part, Organ, or Organ Component	42,221
Group	6	Group; Professional or Occupational Group; Population Group; Family Group; Age Group; Patient or Disabled Group	6440
Health Care Activity	4	Health Care Activity; Laboratory Procedure; Diagnostic Procedure; Therapeutic or Preventive Procedure	91,290
Idea or Concept	14	Idea or Concept; Functional Concept; Body System; Temporal Concept; Qualitative Concept; Quantitative Concepts; Spatial Concept; Geographic Area; Body Location or Region; Molecular Sequence; Carbohydrate Sequence; Amino Acid Sequence; Body Space or Junction; Nucleotide Sequence	25,661
Manufactured Object	4	Manufactured Object; Medical Device; Research Device; Clinical Drug	95,758
Natural Phenomenon or Process	1	Natural Phenomenon or Process	583
Occupation or Discipline	2	Occupation or Discipline; Biomedical Occupation or Discipline	968
Occupational Activity	3	Occupational Activity; Educational Activity Governmental or Regulatory Activity	3063
Organism	6	Organism; Archaeon; Virus; Bacterium; Fungus; Rickettsia or Chlamydia	9119
Organism Attribute	2	Organism Attribute; Clinical Attribute	26,464
Organization	4	Organization; Health Care Related Organization; Professional Society; Self-help or Relief Organization	2193
Pathologic Function	6	Pathologic Function; Experimental Model of Disease; Cell or Molecular Dysfunction; Disease or Syndrome; Mental or Behavioral Dysfunction	63,969
Pharmacologic Substance	2	Pharmacologic Substance; Antibiotic	116,171
Phenomenon or Process	4	Phenomenon or Process; Injury or Poisoning; Human-caused Phenomenon or Process; Environmental Effect of Humans	31,892
Physiologic Function	7	Physiologic Function; Organ or Tissue Function; Organism Function; Mental Process; Molecular	5078

Table 1 (*continued*)

Collection	*Size*	*Semantic Types in Collection*	*# Concepts*
		Function; Genetic Function; Cell Function	
Plant	2	Plant; Alga	3481
Research Activity	2	Research Activity; Molecular Biology Research Technique	948
Substance	3	Substance; Body Substance; Food	5487

dren. **Classification** belongs to the **Entity** group since it does not inherit the *conceptual_part_of* (DNI) relationship from **Intellectual Product**. **Regulation or Law** is a singleton leaf since it introduces the relationship *affects*.

According to Rule 2, the singleton **Regulation or Law** will join its parent **Intellectual Product**. However, the *Entity* group will still be multi-rooted. But by then applying Rule 3, the entire group containing **Intellectual Product** and **Regulation or Law** will be added to the *Entity* group resulting in a singly rooted group. Note that the same group would result from applying Rule 3 first and then Rule 2.

To achieve the transformation from the structural partition to the cohesive partition, we apply the following transformation algorithm consisting of three steps:

1. Apply Rule 1 for all semantic-type groups.
2. Apply Rule 2 for all leaf singleton groups.
3. Apply Rule 3 for all multi-rooted groups.

The semantic-type collections yielded by the application of the three rules to the structural partition form a new partition of the SN. While the semantic-type collections are not necessarily structurally uniform, they are characterized by approximated structural uniformity, which we formally define in the following and call "semi-structural uniformity."

Definition (*Structure of semantic-type collection*). The *structure* of a semantic-type collection is the set of all relationships of its root excluding any DNI relationships. It is denoted *Structure*(A), where A is a semantic-type collection.[3]

The reason DNI relationships of a root of a collection are not considered part of the structure of a collection is that they are not inherited by the rest of the semantic types, thus, they are only properties of the root, not of the collection.

As examples:[4]

$$\mathit{Structure}(\mathit{Animal}) = \{\mathit{exhibits},\ \mathit{issue\_in},\ \mathit{interacts\_with}\}$$

$$\mathit{Structure}(\{\mathit{Bacterium},\ \mathit{Fungus},\ \mathit{Rickettsia\ or\ Chlamydia}\}) = \{\mathit{causes},\ \mathit{exhibits},\ \mathit{issue\_in},\ \mathit{interacts\_with}\}$$

$$\mathit{Structure}(\mathit{Invertebrate}) = \{\mathit{causes},\ \mathit{exhibits},\ \mathit{issue\_in},\ \mathit{interacts\_with}\}$$

Note that *Structure*(*Invertebrate*) $\supset$ *Structure*(*Animal*), where "$\supset$" denotes "properly contains," i.e., contains but is not equal to. Also *Structure*({*Bacterium*, *Fungus*, *Rickettsia or Chlamydia*}) $\supset$ *Structure*(*Animal*).

For an example of a group with a DNI-root, consider the semantic-type collection *Intellectual Product*. *Structure*(*Intellectual Product*) = {*issue_in*}, since *conceptual _part_of* is a DNI relationship and is excluded from the structure in this situation. *Structure*(*Regulation or Law*) = {*issue_in*, *affects*}. Hence, *Structure*(*Regulation or Law*) $\supset$ *Structure*(*Intellectual Product*).

Definition (*Semi-structurally uniform*). A singly rooted set C of semantic types with root R is *semi-structurally uniform* if:

1. For all $T \in C$, *Structure*(T) $\supseteq$ *Structure*(R).
2. C does not contain three semantic types x, y, and z such that x is a non-leaf descendant of y which is a descendant of z and *Structure*(x) $\supset$ *Structure*(y) $\supset$ *Structure*(z).

Condition 2 of the definition is intended to prevent the addition of two subsequent layers with incremental structures to a structural group. Only one layer with incremental structure is allowed.

Note that the characteristic of semi-structural uniformity subsumes the characteristic of structural uniformity (see Section 2). That is, every structurally uniform set is semi-structurally uniform, but the reverse does not necessarily hold.

The application of Rule 2 yields semantic-type collections that are semi-structurally uniform. When a singleton leaf semantic type L is added to a semantic-type collection C whose root is R, then *Structure*(L) $\supset$ *Structure*(R). The reason is that L introduces at least one new relationship, in addition to the relationships it inherits from R. This is true whether or not R has a relationship designated DNI. Thus, in such a case, although the semantic types in the semantic-type collection

[3] Here, we are using *Structure* for semantic-type collections, whereas above it was used for semantic types. By looking at the parameters, the reader can distinguish them.

[4] In the second example, we are using the "{ }" notation to denote a semantic-type collection, i.e., the collection's semantic types are enumerated in the braces.

do not always have uniform structure, the collection is semi-structurally uniform.

The child **Invertebrate** of **Animal** was originally a singleton. On the one hand, it inherits all relationships of **Animal**. On the other hand, it is structurally different from its parent. **Invertebrate** introduces a new relationship *causes* (directed to **Pathologic Function**). As we noted before, *Structure*(**Invertebrate**) $\supset$ *Structure* (**Animal**). Hence, the semantic-type collection obtained by adding **Invertebrate** to the semantic-type group rooted at **Animal** is a semi-structurally uniform collection.

The application of Rule 3 also yields semantic-type collections that are semi-structurally uniform. Hence, all semantic-type collections of the cohesive partition are semi-structurally uniform.

For example, the semantic-type group rooted at **Organism** has two semantic types, **Organism** and **Archaeon**, sharing the structure of the group. The structure of the multi-rooted group containing **Bacterium**, **Fungus**, and **Rickettsia or Chlamydia** has the extra relationship *location_of*, and the structure of the singleton semantic-type **Virus** has two extra relationships, *location_of* and *associated_with*. According to our definition, the collection, after applying Rule 3 to add the multi-rooted group following the applications of Rule 2 to add the singleton, is semi-structurally uniform in spite of the variety of the structures of the semantic types.

As another example, consider the multi-rooted semantic-type group whose roots are **Organ or Tissue Function** and **Organism Function**. The lowest common ancestor of these two roots is **Physiologic Function**. Hence, both roots **Organ or Tissue Function** and **Organism Function** join the *Physiologic Function* collection according to Rule 3. Note that the singleton **Mental Process** was added previously according to Rule 2 to its parent **Organism Function** which is a DNI-root, before **Organism Function** was added to the *Physiologic Function* collection (Fig. 3). Now, all the descendants of **Physiologic Function** belong to its semantic-type collection.

It is possible to show semi-structural uniformity in such cases as well. **Organ or Tissue Function** and **Organism Function** and its child **Mental Process** have all the relationships of **Physiologic Function**, and an extra relationship *degree_of*. Moreover, **Organ or Tissue Function** and **Organism Function** both introduce the DNI relationship *conceptual_part_of*. Hence, *Structure* (**Organ or Tissue Function**) = *Structure*(**Organism Function**) $\supset$ *Structure*(**Mental Process**) $\supset$ *Structure*(**Physiologic Function**). Note that this DNI relationship belongs to the structure of two semantic types, **Organ or Tissue Function** and **Organism Function**, which are not roots of a collection. Thus, the *Physiologic Function* collection is semi-structurally uniform. Note that Condition 2 in the definition of semi-structurally uniform is satisfied as **Mental Process** IS-A **Organism Function** and not the reverse.

Note that if we would add both **Natural Phenomenon or Process** and its child **Biologic Function** to the *Phenomenon or Process* group, we will violate Condition 2 in the definition of semi-structurally uniform. This is because *Structure*(**Phenomenon or Process**) = $\{$*issue_in*, *result_of*$\}$. The semantic-type **Natural Phenomenon or Process** has relationships *process_of* and *affects* in addition to the relationships inherited from **Phenomenon or Process**. Furthermore, its child **Biologic Function** has one more relationship *produces*. However, our rules do not allow the addition of these two semantic types to the *Phenomenon or Process* group because they are not leaves.

Fig. 3 shows the cohesive partition of the subnetwork of Fig. 1. There are eleven semantic-type collections in the figure. In the entire SN, there are 28 semantic-type collections. Two of them are the just mentioned singletons. (As we see, the algorithm does not eliminate non-leaf singletons which may play the role of branching points in the hierarchy.) Six semantic-type collections have two semantic types; five semantic-type collections have three, and five have four; four semantic-type collections have six semantic types; two have seven. Finally, there are four large semantic-type collections containing, respectively, eight, nine, fourteen, and sixteen semantic types. Table 1 shows the semantic types and lists the number of concepts of META in each collection of the cohesive partition. Note that the total number of concepts in the table is larger than the number of concepts in META since concepts may be assigned to multiple semantic types which may be in different collections.

We note that in the transformation from the structural partition to the cohesive partition, a set of semantic types are typically becoming less uniform in their semantics by adding singletons and transforming multi-rooted groups into singly rooted groups. For example, in the structural partition, the *Idea or Concept* group has 7 semantic types. In the transformation to a cohesive group, 7 more leaf singletons are added to the group. Some like **Qualitative Concept** fit the group semantically in spite of their extra relationships. However, some like **Geographic Area** are quite different from the rest of the semantic types in the group. However, this is a price we need to pay as a tradeoff for our decision to disallow singleton leaves in the partition. Semantically, we can have a singleton collection *Geographic Area*. However, due to the lack of contribution of such a collection to size reduction of the metaschema, we prefer to avoid such a collection by merging it into its parent collection.

4. Metaschema

With the cohesive partition of the SN now established, we are ready to define the notion of *metaschema*, a network that provides a compact abstract view of the SN. Before defining metaschema, we need the following.

Definition (*Meta-semantic Type*). A *meta-semantic type* is an abstract entity which represents one semantic-type collection.

In other words, a meta-semantic type is the abstraction of a single semantic-type collection, which in turn functions as the extent of the meta-semantic type. We use the expression "root of a meta-semantic type" to mean the root of its corresponding semantic-type collection. Furthermore, a meta-semantic type is labeled with the name of its root.

Definition (*Metaschema*). A *metaschema* is a network whose nodes are meta-semantic types and whose links are of two types: *meta-child-of* relationships and *meta-relationships*.

In the following, we define the notions of *meta-child-of* relationship and *meta-relationship*, both of whose occurrences in the metaschema are induced by underlying relationships in the UMLS SN.

Definition (*Meta-child-of*). Let a and b_r be semantic types in the semantic-type collections of meta-semantic types A and B, respectively. Furthermore, let b_r be the root of B and let b_r IS-A a. Then, in the metaschema, there exists a *meta-child-of* relationship directed from B to A.

The *meta-child-of* connecting a meta-semantic type B to a meta-semantic type A forms an abstraction that denotes the fact that all semantic types in B are subtypes (specializations) of some element of A (and indeed A's root) via the transitivity of IS-A. We can see that the IS-A hierarchy of the underlying SN induces an entire hierarchy of *meta-child-of* relationships within the metaschema. This metaschema hierarchy, consisting of meta-semantic types and the *meta-child-of* relationships connecting them, is an important skeletal view of the metaschema, aiding both in orientation to the metaschema itself and the IS-A hierarchy of the SN.

As an example, **Organism**, the root of the meta-semantic type *Organism*, is a child of **Physical Object** which is in the meta-semantic type *Entity*. Therefore, a *meta-child-of* relationship is defined from the meta-semantic type *Organism* to the meta-semantic type *Entity*. Fig. 5 shows the complete metaschema hierarchy, consisting of 28 meta-semantic types and 26 *meta-child-of* relationships. According to its definition, this hierarchy consists of two trees rooted at the *Entity* and *Event* meta-semantic types, similar to the situation in the IS-A hierarchy of the SN. However these two trees are more compact than the corresponding trees of the SN.

Definition (*Meta-relationship*). Let a_r and b be semantic types in the semantic-type collections of meta-semantic types A and B, respectively. Moreover, let a_r be the root of A and let there exist a semantic relationship *rel* connecting a_r to b. Then, in the metaschema, there exists a link labeled "*rel*" connecting A to B. Such a link is called a *meta-relationship*.

It will be noted that the semantic type b in the definition need not be the root of its meta-semantic type. Only the source of the relationship *rel* (i.e., a_r) need be a root in order for a new meta-relationship *rel* to be induced in the metaschema.

Recall that in the SN a child (or descendant) inherits all the relationships defined by its parent (or ancestor), unless this is disrupted by DNI or blocking. Therefore, via inheritance, each semantic type in meta-semantic type A (in the definition)—all of which must be descendants of a_r—has the relationship *rel* to semantic type b,

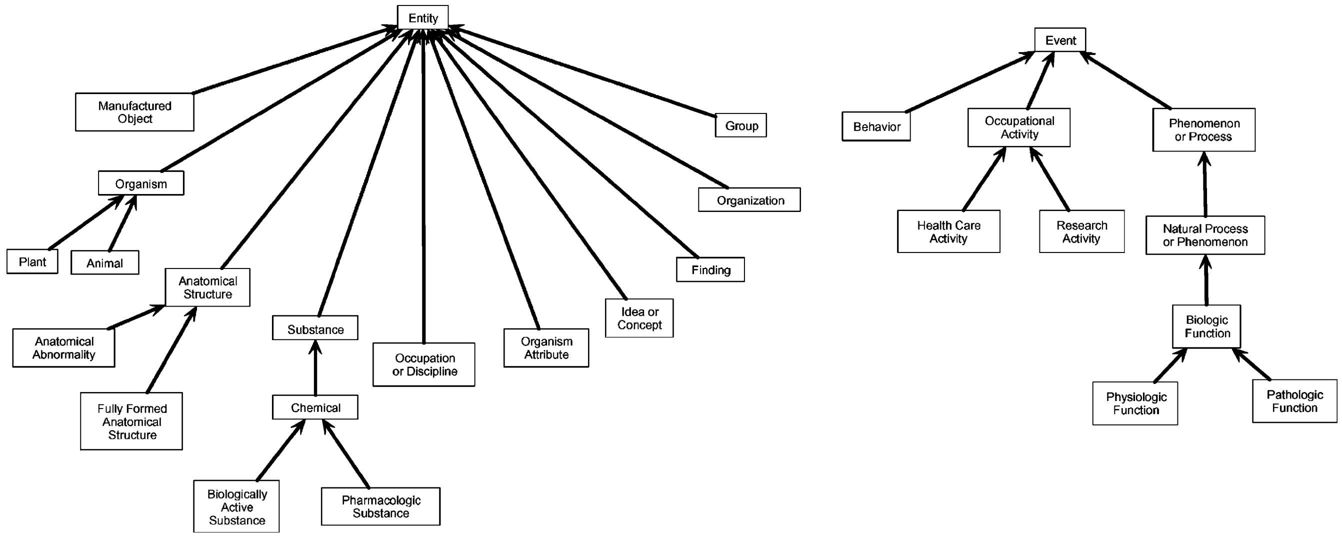


Fig. 5. Metaschema hierarchy for the SN.

or to a semantic type *c* which is a descendant of *b* in the meta-semantic type *B* or in the meta-semantic types which are descendents of *B* in the hierarchy of the metaschema (see discussion of inheritance in the metaschema at the end of this section). The meta-relationship *rel* forms an abstraction capturing this situation.

As examples, there are two relationships, *affects* and *process_of*, defined from **Biologic Function**, which is the root of the meta-semantic type *Biologic Function*, to **Organism**, which is in the meta-semantic type *Organism*. Therefore, two meta-relationships, *affects* and *process_of*, are defined from the meta-semantic type

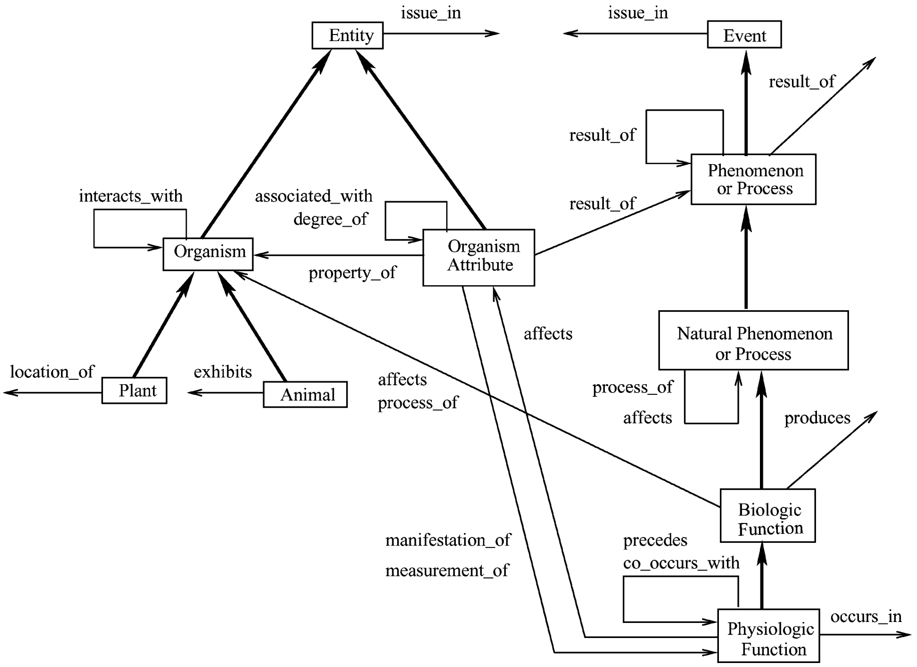


Fig. 6. Metaschema for Fig. 1 subnetwork.

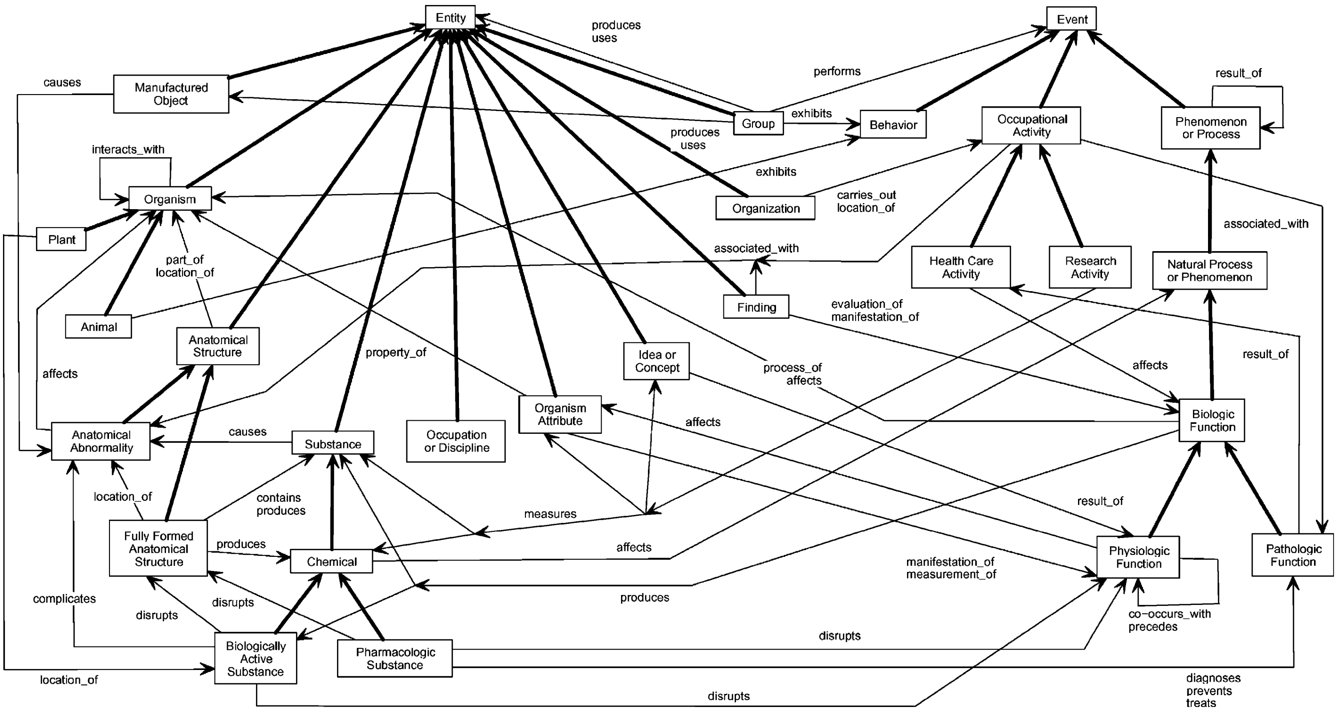


Fig. 7. Metaschema of the SN (some meta-relationships omitted).

Biologic Function to the meta-semantic type *Organism* in the metaschema.

Fig. 6 shows a portion of the metaschema corresponding to the subnetwork of Fig. 1. In the overall metaschema, there are 28 meta-semantic types connected by 26 *meta-child-of* relationships and 139 meta-relationships. Fig. 7 shows the *meta-child-of* hierarchy and some of the 139 meta-relationships of the metaschema. While the metaschema hierarchy fits easily onto a computer screen or printed page (see Fig. 5), the number of meta-relationships in the metaschema causes some difficulty. In Fig. 7, we display only a subset of those relationships. In Section 6, we will discuss views that conveniently display all of them, although not simultaneously.

To reflect the relationship inheritance occurring in the SN, we define inheritance of meta-relationships along the *meta-child-of* links in the metaschema. Specifically, let A, B, and C be meta-semantic types with a *meta-child-of* relationship connecting B to A. If A has a meta-relationship *rel* to C, then B also has a meta-relationship *rel* to C, or to a meta-semantic type which is either a *meta-child-of* C or has a chain of *meta-child-of* relationships to C.

We need to stress that the metaschema is not intended to replace the role of the SN as an abstract level overarching the META. The purpose of the metaschema is to provide an additional compact layer of abstraction. This abstract layer is small enough to be visualized graphically on a computer screen in a comprehensible way, which is not true of the SN. On the other hand, it serves to orient the user so the user can identify which part of the SN is currently of interest. Such parts can be studied through partial views of the SN defined with the orientation gained by the metaschema, as described in Section 6. To summarize, the metaschema does not replace the SN, but helps to simplify the access and orientation into it through an extra upper level layer.

5. Using the Metaschema to audit the UMLS Classification

Due to the way the UMLS was created, it is unavoidable that some errors and inconsistencies exist regarding some of its concepts and their classification into semantic types. This is due to the integration of many terminological sources, which are not necessarily consistent. Another reason is that the classification was done by many experts of various backgrounds and viewpoints. It is a challenge facing NLM to audit the UMLS and expose and correct existing errors. However, due to the huge size of META, such a comprehensive audit can be an overwhelming task.

Thus, there is a need to design auditing techniques for the UMLS which will minimize the effort and maximize the probability of finding errors. In this section, we will describe such a technique based on the metaschema of the UMLS defined in the previous section.

Concepts of META are assigned to one or more of the semantic types of the SN. In this paper, we have grouped closely related semantic types into semantic-type collections and abstracted these into meta-semantic types. Since a concept may be assigned to several semantic types, it may also be associated with several meta-semantic types (a formal definition of such an association is given below). However, it is more likely that a concept will be assigned erroneously to several semantic types which reside in different meta-semantic types than to be assigned erroneously with several semantic types of the same meta-semantic type. The reason is that, in general, two semantic types of the same meta-semantic type belong to the same domain. On the other hand, if two semantic types are in two different meta-semantic types, they belong to two different domains. This observation leads to the idea of an auditing effort that concentrates on concepts which are associated with several meta-semantic types. The idea is that such concepts are more likely to be in error than general concepts. Of course, there are many concepts which are correctly assigned to several semantic types of the same meta-semantic type or of different meta-semantic types.

5.1. Meta-intersection

Our auditing approach requires a few definitions.

Definition (*Intersection*). An intersection of two or more semantic types is the set of all concepts assigned to all of the respective semantic types.

Such an intersection is named after its constituent semantic types. For example, the intersection **Plant** ∩ **Disease or Syndrome** is a set which contains one concept *toxicodendron*, which is the only one that is assigned to both **Plant** and **Disease or Syndrome**.

Definition (*Concept associated with meta-Semantic type*). A concept is said to be associated with a meta-semantic type if it is assigned to one or more of the semantic types in the meta-semantic type.

Definition (*Meta-intersection*). A meta-intersection of two or more meta-semantic types is the set of all concepts associated with all of the respective meta-semantic types.

Such a meta-intersection is named according to its meta-semantic types. For example, the meta-intersection *Plant* ∩ *Pathologic Function* is a set which contains one concept *toxicodendron*, mentioned above.

In the first step of our auditing process, we generate all meta-intersections of the metaschema. When this list is reviewed, it is seen that most of the meta-intersections are small sets of one or two concepts. On the other hand, there are a few meta-intersections which are very large. Specifically, 332 meta-intersections contain only one concept, 113 meta-intersections contain two concepts, 17 meta-intersections contain eight concepts, and the largest meta-intersection contains 70,436 concepts.

5.2. Review of meta-intersections

As mentioned above, an effective auditing process should expose many errors with limited efforts. With this in mind, we review only meta-intersections that contain very few concepts. The likelihood of a mistake for such a meta-intersection is higher than in the case of a very large meta-intersection. The reason is that if a combination of semantic types makes sense semantically, then there would probably be quite a few—or at least several—concepts associated with it. The case where such a combination of semantic types is associated with only one or two concepts may indicate an erroneous classification where no concepts at all should be classified in such a manner. For example, the largest intersection is **Organic Chemical** $\cap$ **Pharmacologic Substance** which contains 70,436 concepts. It is not a case of an erroneous classification since every one of these drugs is classified both as **Organic Chemical** due to its chemical compounds, and as **Pharmacologic Substance** according to its function as a drug.

On the other hand, there are 332 meta-intersections containing one concept. Those are worth a careful review since some of them may exist due to a mistake. For each such small meta-intersection, we need to review the actual corresponding intersection of semantic types to explore if this intersection reflects an error in classification. As a matter of fact, in [22] and [16] concepts of more than one semantic type were reviewed to uncover errors in the UMLS. The new idea in this paper is to avoid scanning all such concepts, and to concentrate instead on the more suspicious of them—those which are associated with more than one meta-semantic type.

In Table 2, we list some erroneously classified concepts, their assigned semantic types, and their associated meta-semantic types. For example, *toxicodendron* (poison ivy) was suspicious as a member of the meta-intersection *Plant* $\cap$ *Pathologic Function*. It does not seem right that a plant can be a pathologic function. Looking at the corresponding semantic types, we see it is in **Plant** $\cap$ **Disease or Syndrome**. This is a case of a homonymy. The same concept is used for the plant and the disease it causes. Two concepts, *toxicodendron*$\langle 1\rangle$ and *toxicodendron*$\langle 2\rangle$, need to be created instead to resolve the ambiguity.

For similar reasons, another suspicious meta-intersection is *Animal* $\cap$ *Pathologic Function*. As a matter of fact, the concept *lice infestations* is in the intersection **Invertebrate** $\cap$ **Disease or Syndrome**. In this case, it is an outright classification error. The concept *lice* is **Invertebrate**, and *lice infestation* is a **Disease or Syndrome**.

A similar case of a suspicious meta-intersection is *Animal* $\cap$ *Organization*. As a matter of fact, *seeing eye dogs* is the only concept classified in the intersection **Mammal** $\cap$ **Self-help or Relief Organization**. Obviously, it should not be classified as a **Self-help or Relief Organization**.

For the case of the concept *serial analysis of gene expression*, we do not have an inkling as to why it was classified as a **Plant**. Similarly, for the case of *surgical procedures*, *colposcopic*, there is no reason why it was classified as **Human**.

The situation with *stramonium* is different. It is classified correctly as **Plant** $\cap$ **Hazardous or Poisonous Substance**. However, this case indicates a non-uniform classification. This is not the only case of a plant which is a poisonous substance; however, it has been the only one defined that way. Either all such plants should be associated with both semantic types or none should be.

There are cases of meta-intersections which on the surface look suspicious but do make sense when considering the semantic types' intersection. For a few such examples, see Table 3.

Any intersection of many meta-semantic types or many semantic types is also suspicious. For example, there are two concepts "*benzoic acid*, 4-*amino-ethyl ester mixt. with aluminum hydroxide*, *magnesium hydroxide and Me siloxanes*" and "*benzoic acid*, 4-*amino-ethyl ester mixt. with aluminum hydroxide* ($Al(OH)_3$), *magnesium hydroxide* ($Mg(OH)_2$) *and Me siloxanes*," both of which are classified as five different semantic types, **Chemical Viewed Structurally**, **Organic Chemical**, **Carbohydrate**,

Table 2
A few misclassified concepts

Concept	Semantic types	Meta-semantic types
toxicodendron	**Plant; Disease or Syndrome**	*Plant; Pathologic Function*
serial analysis of gene expression	**Plant; Research Activity**	*Plant; Research Activity*
stramonium	**Plant; Hazardous or Poisonous Substance**	*Plant; Chemical*
lice infestations	**Invertebrate; Disease or Syndrome**	*Animal; Pathologic Function*
seeing eye dogs	**Mammal; Self-help or Relief Organization**	*Animal; Organization*
surgical procedures, colposcopic	**Human; Diagnostic Procedure**	*Animal; Health Care Activity*

Table 3
A few properly classified concepts

Concept	Semantic types	Meta-semantic types
hallucinogenic mushrooms	**Fungus; Hazardous or Poisonous Substance**	*Organism; Chemical*
homo sapiens	**Human; Population Group**	*Animal; Group*
partner in relationship	**Human; Family Group**	*Animal; Group*
bone marrow of iliac crest	**Anatomical Structure; Body Substance**	*anatomical Structure; Substance*

Pharmacologic Substance, and **Inorganic Chemical**. First, we observe that since the concepts are **Organic Chemical**, they cannot be **Inorganic Chemical**. Furthermore, as specified by McCray and Nelson [14] discussing the assignment of concepts to semantic types: "In all cases the most specific semantic type available in the hierarchy is assigned to a term." In the SN, **Carbohydrate** IS-A **Organic Chemical** and in turn **Organic Chemical** IS-A **Chemical Viewed Structurally**. Hence, these concepts should only be classified as **Carbohydrate** out of these three semantic types. As a result, these two concepts are associated only with the intersection **Carbohydrate** ∩ **Pharmacologic Substance**. This example also demonstrates the need to check intersections of pairs of exclusive semantic types, such as **Organic Chemical** and **Inorganic Chemical** in an audit since they should be empty.

As we see, the metaschema is very helpful in auditing the UMLS. All the above examples were taken from a small sample of intersections.

6. Using the Metaschema for Orientation to the SN

The professionals who maintain META, performing operations such as adding a new concept, splitting a concept which is found to have two different meanings (homonym), changing the semantic type classification of a concept, etc., need to be well oriented with its content. Achieving such an orientation is difficult due to META's size and complexity. The abstract view of META provided by the SN can help towards reaching such a goal. However, SN itself is too large and complex to be laid out on one computer screen. The metaschema, which provides an abstract, compact view of SN, can help us in this regard.

Let us now describe how the metaschema view will help maintenance personnel in achieving an orientation to the SN. Using a diagrammatic display of the metaschema hierarchy (Fig. 5) which fits onto a single screen, a user can easily identify (according to a search interest) a desired meta-semantic type which we call the focus meta-semantic type. As an example, let the focus meta-semantic type be *Pathologic Function*. Next, the user can view on the computer screen the diagram of the SN subnetwork induced by the semantic-type collection represented by the focus meta-semantic type. We call such a diagram a *collection subnetwork*.

Fig. 8 shows the *Pathologic Function* collection subnetwork. It contains six semantic types, five IS-A relationships, and nine semantic relationships. The collection subnetwork shows only the internal connections within the collection. However, this is not sufficient for studying the full significance of the semantic types of the collection since it does not include the external relationships of the semantic types of the collection. For considering the external relationships of the collection, we need the following definition.

The *collection environment* is a network containing the semantic types of the collection, the (internal) relationships of the collection subnetwork, and all the outgoing (external) relationships of the SN where only one semantic type is in the collection. (The other semantic type of each external relationship is not included in the environment, leaving the relationship pointing to a "?" in the diagram.)

Fig. 9 shows the *Pathologic Function* collection environment containing six semantic types of the collection. It contains 19 external relationships incident on the *Pathologic Function* collection subnetwork's semantic types, nine of which are exiting the collection and are shown in Fig. 9, and ten of which are entering the collection and are not shown in the figure to avoid overwhelming clutter. In addition, there are eight

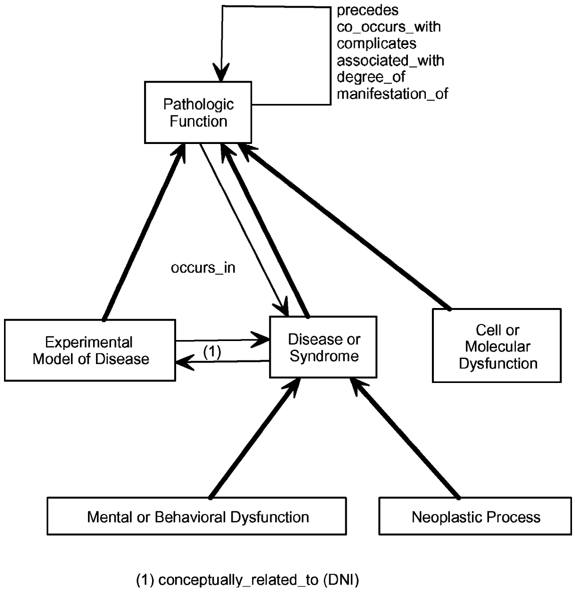
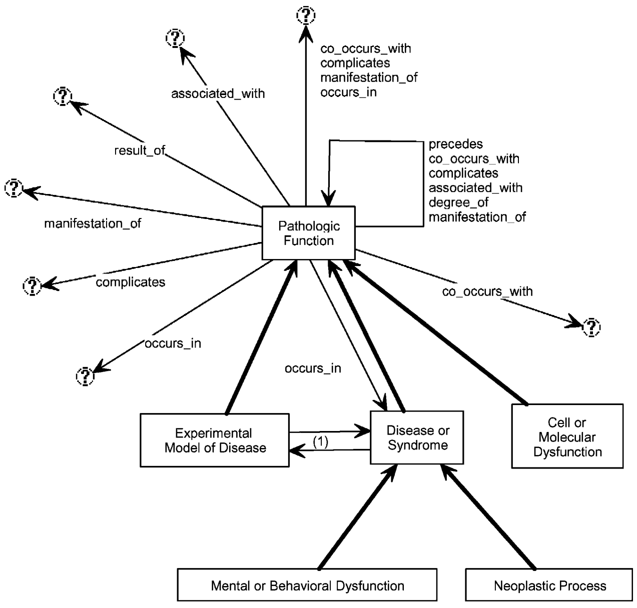


Fig. 8. *Pathologic Function* collection subnetwork.



(1) conceptually_related_to (DNI)

Fig. 9. *Pathologic Function* collection environment.

relationships inherited from the ancestors of **Pathologic Function** defined for each semantic type of the collection which are not displayed in the figure. Furthermore, nine internal relationships are defined in the *Pathologic Function* collection subnetwork.

To study the interactions between semantic types in the focus collection (corresponding to the focus meta-semantic type) and other semantic types, we will first view a subnetwork of the metaschema showing the focus meta-semantic type and all other neighboring meta-semantic types related to it via either a *meta-child-of* or a meta-relationship. Such a subnetwork is called a focus sub-metaschema. See Fig. 10 for the *Pathologic Function* focus sub-metaschema with nine neighboring meta-semantic types, including the *Pathologic Function* meta-semantic type itself. (The entering meta-relationships are omitted from the figure.)

To study the relationships between the semantic types of the collection subnetwork of interest and other semantic types, we concentrate each time on one pair of meta-semantic types, the focus meta-semantic type and a neighboring meta-semantic type. For example, we will study the interaction between the *Pathologic Function* collection subnetwork and other semantic-type collections by reviewing eight pairs of meta-semantic types in Fig. 10. Each time, we will pick one neighboring meta-semantic type which is connected to the *Pathologic Function* meta-semantic type and study the interactions between the two corresponding semantic-type collections.

For example, we can study the relationships between the *Pathologic Function* focus collection and the *Phenomenon and Process* collection. There are four relationships from the *Pathologic Function* collection to the *Phenomenon and Process* collection, and one relationship from the *Phenomenon and Process* collection to the *Pathologic Function* collection. In addition, there are two internal relationships in the *Phenomenon and Process* collection subnetwork and nine internal relationships in the *Pathologic Function* collection subnetwork (see Fig. 11).

As another example, let us look at the *Pathologic Function* collection and the *Physiologic Function* collection and their relationships. There is one relationship from the *Pathologic Function* collection to the *Physiologic Function* collection, and there are five internal relationships in the *Physiologic Function* collection subnetwork, and nine internal relationships in the *Pathologic Function* collection subnetwork.

It is clearly much easier to get an understanding of each of these semantic types and the interactions among them separately from Fig. 10 and several figures like

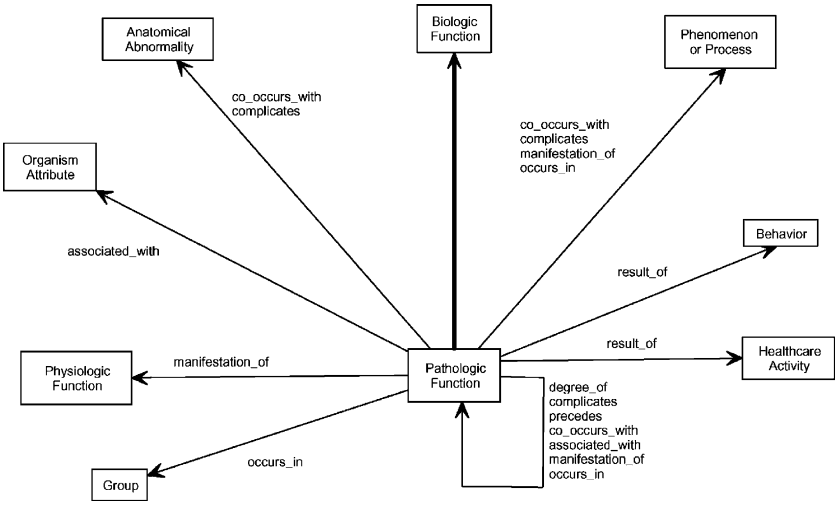


Fig. 10. The *Pathologic Function* focus sub-metaschema.

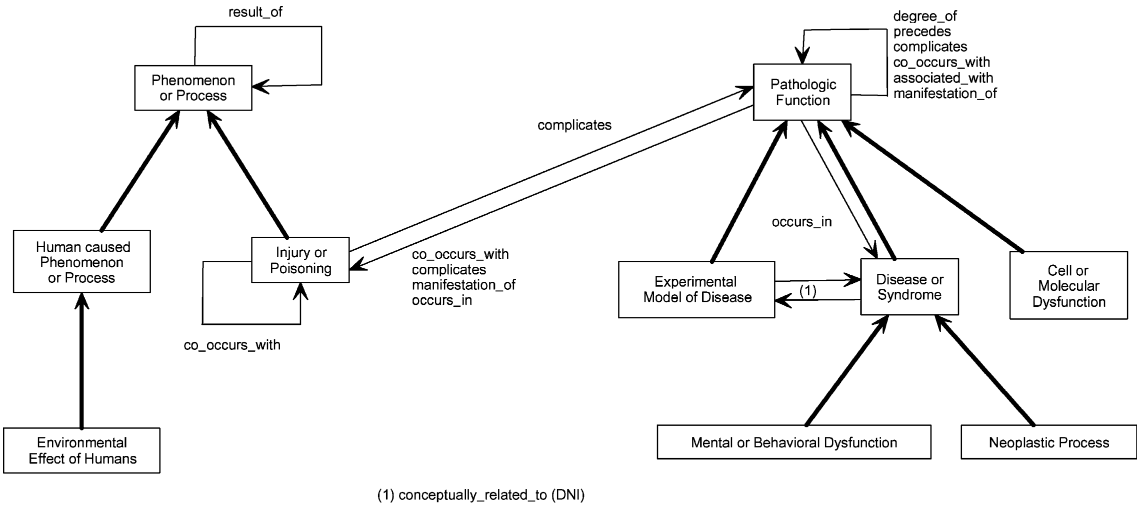


Fig. 11. Interaction between the *Pathologic Function* and the *Phenomenon or Process* semantic-type collections.

Fig. 11, each covering a pair of collections, than to get such knowledge from Fig. 1 where these interactions are hidden in the overall structure of a large network. Concentrating only on the connections between the semantic types of two semantic-type collections at a time, the user can cope with a small network and a limited number of relationships. Such a network is typically small enough to be displayed on one computer screen and is easier to comprehend. By dividing the orientation task of the whole SN into subtasks of comprehending many small networks, the difficulty of the task is meaningfully reduced.

One of the advantages resulting from the various partial views described in this section is related to detection of redundant classification.

Definition (*Redundant Classification*). Let concept *c* be a member of the intersection of two semantic types *B* and *A* such that *B* is a descendant of *A* in the SN. Then the classification of *c* to *A* is called a redundant classification since it can be inferred from the classification of *c* to *B*.

The redundant classification is a violation of an important rule promoted by the SN's designers. Specifically, it was stated that a concept should be explicitly assigned to the lowest (most specialized) possible semantic type in the SN's IS-A hierarchy [14]. For example, a concept should not be assigned to both a child semantic type and its parent.

As an example, the two concepts *Tryplosan* and *Triton* belong to the intersection of **Chemical Viewed Functionally** and **Pharmacologic Substance**. **Chemical Viewed Functionally** is the parent of **Pharmacologic Substance**. Thus, the two concepts have redundant classifications to **Chemical Viewed Functionally**, and these classifications should be removed.

In [16], while reviewing all intersections of semantic types in the SN of the 1998 version of the UMLS, we discovered that 8622 concepts had redundant classifications. This group of redundant classifications was reported to the NLM so they could be omitted in subsequent releases. Recently, a follow-up audit was performed on the 2001 UMLS to determine the status of these 8622 concepts. It was found that a portion (38%) of the redundant classifications was properly removed. However, a large number of them (57%) were still present. A third portion (5%) of the redundant classifications was partially treated. For instance, an existing redundant classification was removed, and a new assignment to another semantic type was added instead, only to create a new redundancy. Additionally, there were cases of multiple assignments causing multiple redundant classifications, and only one of those assignments was deleted. The above audit shows that redundant classification has been a persistent problem in the UMLS.

An obvious question is: how come the experts classifying META's concepts are not aware that they are creating redundant classifications? One explanation is that the various classifications were made by different experts with different opinions. But it is still not clear why an expert, while making a classification, would not realize that his newly created classification should not co-exist with an already defined classification, due to redundancy. Having tools which provide comprehensible views of the SN would enable experts to see the existing or emerging redundancies. The lack of such visual tools for experts is even more striking in the case where redundant classifications were reported to the NLM; a change was made in the classification by an expert, but the removal of one case of redundancy caused a new case to appear. If such an expert had tools providing graphically comprehensible views of the SN, he would realize that the newly introduced classification should not co-exist with the current one. Then the expert would have resolved the redundancy properly.

7. Evaluation study

The hypothesis underlying our research is that although our partitioning technique, leading to the metaschema, is based primarily on structural aspects, the metaschema still captures semantic considerations. That is, even though the cohesive partition is the result of an algorithmic process, it still yields meaningful and useful (to a human) "graphical modules." From a content point of view, each element of the metaschema, called a *meta-semantic-type*, is expected to represent a unified group of semantic types, describing some specific subject area. In other words, we assume that if two *semantic types* in the UMLS Semantic Network have identical (or even approximately identical) sets of relationships, then they are also close semantically. How can we evaluate whether an algorithmically obtained metaschema is meaningful to human experts?

To address this question, the following study was performed. We selected seven experts with reputations in UMLS research and sent them two pages with diagrams of the IS-A hierarchy of the SN, i.e., the two trees rooted at **Event** and **Entity**.

Each participant received a page of instructions as follows:

1. Start marking by star, the root node of the tree and continue to scan the semantic types downwards.
2. While scanning, mark by star, semantic types, which you judge as IMPORTANT AND QUITE DIFFERENT from their parent semantic types.
3. There is one exception: Don't mark semantic types which have no children. Thus, you only need to consider the 45 semantic types with children.
4. The star markings of each participant will be used to define a Metaschema where each semantic type marked by a subject names a meta-semantic-type. This metaschema will be compared with the results of other respondents and with our algorithmically derived Metaschema.

Note that although the instructions seem quite elaborate, they only define structural limitations, such as "don't mark semantic types which have no children." These limitations are necessary to make the participant results compatible with the same constraints followed by the algorithmically derived metaschema and enable the computation of a valid comparison score between the metaschema of the subjects and the algorithmically obtained metaschema. On the other hand, our instructions do not limit the semantic decisions of the subjects, who still have the complete freedom to mark semantic types of their choice. Most importantly, the participants were not provided any information of the non-IS-A relationships that were used by the structural partitioning method in order not to bias the participants to follow a structural approach. Therefore, the participants relied exclusively on their understanding of the semantic types, based on their names and positions in the SN IS-A hierarchy.

Evaluating the results showed that the metaschemas of different participants were quite different. The second row of Table 4 shows the number of meta-semantic types of each participant. The third row shows the level of agreement (joint meta-semantic types) between the cohesive metaschema and the subjects' metaschema. The number of meta-semantic types of the participants varies widely between 21 and 35, and the average is about 29. The average agreement of the subjects with the cohesive metaschema is 20.14, with the high of 24 and the low of 15. To gain a perspective into the meaning of this average agreement, we will look at Table 5 which demonstrates the high variability of subject responses. The table shows inter-subject agreement. The number in row *i* and column *j* indicates how many meta-semantic types subject *i* and subject *j* agree on. For instance, subjects 2 and 5 agree on 27 meta-semantic types. The average inter-subject agreement is 20.12. For comparison, a random process conducted shows an intersection agreement of 17.9. Hence, although the responses of the participants vary strongly, they reflect the different opinions of experts, and are significant. We see that the average agreement with our metaschema (the cohesive metaschema) is almost equal to the average agreement among the participants. If the results of the cohesive metaschema were added to Table 5, they would not be distinguishable from the results of other participants.

Although the participants' responses varied greatly, when accumulating all responses, some choices were repeated by many subjects. Our approach is to identify a semantic type as a meta-semantic-type if at least *N* participating subjects chose this semantic type. We

Table 4
Algorithm–subject agreement

Human	1	2	3	4	5	6	7	Average
# of meta-semantic types	21	33	21	35	34	35	25	29.14
Agreement to algorithm	15	24	17	23	23	21	18	20.14

Table 5
Intersubject agreement matrix

	1	2	3	4	5	6	7
1		19	15	16	15	19	12
2			18	28	27	27	20
3				16	16	17	14
4					28	26	23
5						27	20
6							19

subsequently computed recall (R) and precision (P) of the human subjects relative to the cohesive metaschema. We will refer to N as the cut-off value. We then varied N as an independent variable and computed R and P over all semantic types of the hierarchy as dependent variables. We also computed Rijsbergen's F measure which combines precision and recall into one number as: $F = 2 * P * R/(P + R)$.

In Table 6, the columns are: Cut-off value N; number of semantic types marked by at least N subjects; number of semantic types marked by at least N subjects that were also identified by the cohesive metaschema; recall; precision; and F value.

The F value peaks at a cut-off of 3. The F value of about 0.8 with the recall of 0.96 indicates similarity between the cohesive metaschema and a *consensus metaschema* derived from the semantic types which were marked by at least $N = 3$ subjects. At least 3 subjects marked 27 of the 28 semantic types of the cohesive metaschema, with a precision of 0.659. Thus, our evaluation shows the usefulness of the cohesive metaschema and the high degree of agreement with the metaschemas obtained by our subjects. This supports the claim that the cohesive metaschema is semantically meaningful to human experts.

8. Discussion

For another attempt to partition the SN, see [18], where six principles are listed as a requirement for such a partition. The first of these principles is semantic validity: "the group must be semantically coherent." The partition of the SN of [18] consists of 15 groups. However, not all these groups constitute a connected subgraph of the SN which is one way to assess the semantic validity principle. Due to this, the partition of [18] does not lend itself to the definition of a metaschema which is based on connected subgraphs.

Let us contrast the approach of [18] with ours. In [18], they basically use a semantic approach for partitioning. The groups are externally induced by identifying important subjects. Then the semantic types are selected to participate in the proper groups. Both connectivity and the uniformity of structure (the set of relationships) of the semantic types are used to guide this selection, but are not perceived as required properties of the partition. Rather, they are perceived as preferred properties which are considered together with other issues.

On the other hand, our approach is structural in nature. First the structural partition is found, where each group contains exactly all semantic types with the same set of relationships. Then the partition is modified to the cohesive partition to ensure connectivity and avoid isolated leaf groups. Furthermore, the groups in the cohesive partition are induced internally where each group is named after the unique root of its tree structure.

Both approaches share the motivation that the structural properties of a uniform set of relationships and connectivity are good indications for the semantic validity/cohesiveness of the group. The difference lies in the strictness of adhering to these criteria. While in [18], these criteria are preferred, in this paper, they are enforced. Obviously, each of the emerging partitions has its advantages and disadvantages. Our approach is more objective while the semantic approach of [18] depends more on the human designer and his/her perceptions; different designers will partition differently. One striking difference is in the number of groups, 15 versus 28, indicating that the cohesive partition is a finer grained partition.

An alternative approach for structural partition is grouping together all semantic types with the same relationships pointing to the same semantic type. Hence, if a semantic type inherits a relationship from an ancestor semantic type, but it is directed to another target semantic type, then this source semantic type will be in a different group than that ancestor semantic type is.

Table 6
Results of evaluation

Cutoff (N)	Marked	Marked and cohesive	$R = C/28$	$P = C/B$	$F = 2 * P * R/(P + R)$
7	9	9	0.321	1.000	0.486
6	13	12	0.429	0.923	0.585
5	20	15	0.536	0.750	0.625
4	32	22	0.786	0.688	0.733
3	41	27	0.964	0.659	0.783
2	45	28	1.000	0.622	0.767
1	45	28	1.000	0.622	0.767

Another alternative approach is grouping together all semantic types with the same outgoing and incoming relationships. Both are possible approaches, which will lead to a more refined larger metaschema. We did not use these approaches because the current metaschema is already a little too large, as evident from Fig. 7 and the fact it cannot contain all its relationships.

Note that our technique can be applied to other large semantic networks. For example, in [19,20], a schema is presented for the MED [21] terminology of New York Presbyterian Medical Center. A partition of this schema is provided in [23]. However, applying our method to define a metaschema based on this partition requires extension of the definition from a tree-structured schema to a DAG schema.

9. Conclusions

The Unified Medical Language System (UMLS) integrates many medical terminologies and coding systems. It plays a major role in overcoming terminological differences in the design of computerized healthcare information systems. However, the size and complexity of the UMLS make it difficult to comprehend. The Semantic Network (SN) provides an abstract view for the Metathesaurus of the UMLS and helps with its comprehension. However, SN itself is hard to comprehend since it is too large and complex for display on a single computer screen. In this paper, we presented a partitioning algorithm to produce a metaschema that supports the comprehension of the SN. We utilized the metaschema for auditing the classification of the UMLS. A method showing how to use the metaschema for graphical orientation to the SN was given, while an evaluation study confirmed that the metaschema is semantically relevant.

Acknowledgments

We thank the anonymous reviewers for many helpful suggestions incorporated in the paper.

References

[1] Humphreys BL, Lindberg DAB. Building the Unified Medical Language System. In: Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care, Washington, DC. 1989. p. 475–80.

[2] Humphreys BL, Lindberg DAB, Schoolman HM, Barnett GO. The Unified Medical Language System: an informatics research collaboration. JAMIA 1998;5(1):1–11.

[3] Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language system. Methods Inf Med 1993;32:281–91.

[4] Campbell KE, Oliver DE, Shortliffe EH. The Unified Medical Language System: toward a collaborative approach for solving terminologic problems. JAMIA 1998;5(1):12–6.

[5] Cimino JJ. Review paper: coding systems in health care. Methods Inf Med 1996;35:273–84.

[6] Humphreys BL, Lindberg DAB. The Unified Medical Language System project: a distributed experiment in improving access to biomedical information. Methods Inf Med 1992;7(2):1496–500.

[7] Schuyler PL, Hole WT, Tuttle MS, Sherertz DD. The UMLS Metathesaurus: representing different views of biomedical concepts. Bull Med Libr Assoc 1993;81(2):217–22.

[8] Tuttle MS, Sherertz DD, Olson NE, Erlbaum MS, Sperzel WD, Fuller LF, Nelson SJ. Using META-1 the first version of the UMLS Metathesaurus. In: Proceedings of the Fourteenth Annual SCAMC. 1990. p. 131–5.

[9] US Dept. of Health and Human Services, NIH, National Library of Medicine. Unified Medical Language System (UMLS), 2002.

[10] Wickens CD, Gordon SE, Liu Y. An introduction to human factors engineering. New York: Longman; 1998.

[11] McCray AT. UMLS Semantic Network. In: Proceedings of the Thirteenth Annual SCAMC. 1989. p. 503–7.

[12] McCray AT. Representing biomedical knowledge in the UMLS Semantic Network. In: Broering NC, editor. High-Performance Medical Libraries: Advances in Information Management for the Virtual Era. Westport, CT: Meckler; 1993. p. 45–55.

[13] McCray AT, Hole WT. The scope and structure of the first version of the UMLS Semantic Network. In: Proceedings of the Fourteenth Annual SCAMC. 1990. p. 126–36.

[14] McCray AT, Nelson SJ. The representation of meaning in the UMLS. Methods Inf Med 1995;34:193–201.

[15] Catarci T, Costabile M, Levialdi S, Batini C. Visual query systems for databases: a survey. J Visual Lang Comput 1997;8:215–60.

[16] Gu H, Perl Y, Geller J, Halper M, Liu L, Cimino JJ. Representing the UMLS as an OODB: modeling issues and advantages. JAMIA 2000;7(1):66–80.

[17] Halper M, Chen Z, Geller J, Perl Y. A metaschema of the UMLS based on a partition of its Semantic Network. In: Bakken S, editor. Proceedings of the 2001 AMIA Annual Symposium, Washington, DC. 2001. p. 234–8.

[18] McCray AT, Burgun A, Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. In: Proceedings of Medinfo 2001, London, UK. 2001. p. 171–5.

[19] Gu H, Halper M, Geller J, Perl Y. Benefits of an OODB representation for controlled medical terminologies. JAMIA 1999;6(4):283–303.

[20] Liu L, Halper M, Geller J, Perl Y. Controlled vocabularies in OODBs: modeling issues and implementation. Distrib Parallel Dat 1999;7(1):37–65.

[21] Cimino JJ, Clayton PD, Hripcsak G, Johnson S. Knowledge-based approaches to the maintenance of a large controlled medical terminology. JAMIA 1994;1(1):35–50.

[22] Cimino JJ. Auditing the Unified Medical Language System with semantic methods. JAMIA 1998;5(1):41–51.

[23] Gu H, Perl Y, Halper M, Geller J, Kuo F, Cimino JJ. Partitioning an object-oriented terminology schema. Methods Inf Med 2001;40(3):204–12.