Designing Metaschemas for the UMLS Enriched Semantic Network

Li Zhang, Yehoshua Perl, Michael Halper[2], James Geller

Computer Science Dept.
NJIT
Newark, NJ 07102
{lxz1853, yehoshua.perl,
james.geller}@njit.edu

[2]Mathematics & Computer Science Dept.
Kean University
Union, NJ 07083
mhalper@kean.edu

Reprint requests & all communications to:

Li Zhang, MS
lxz1853@njit.edu
CS Dept., NJIT
NJIT, University Heights, Newark, 07102
phone: (973) 596-2867
fax: (973) 642-7029

Abstract

The Enriched Semantic Network (ESN) has previously been presented as an enhancement of the Semantic Network (SN) of the UMLS. The ESN's hierarchy is a DAG (Directed Acyclic Graph) structure allowing for multiple parents. The ESN is thus more complex than the SN and can be more difficult to view and comprehend. We have previously introduced the notion of a metaschema for the SN as a compact abstraction to support SN comprehension. We extend the definition of metaschema to make it applicable to a DAG classification hierarchy, such as the one exhibited by the ESN. We specify the requirements for and describe the general process of deriving such a metaschema. We derive two particular metaschemas of the ESN based on a pair of partitions. These two metaschemas and their underlying partitions are compared. Both metaschemas serve as compact representations of the ESN, allowing for convenient viewing of its hierarchy and easier comprehension.

Keywords: UMLS, Enriched Semantic Network (ESN), Partition, Metaschema, Multiple Inheritance, Visualization

1 Introduction

The Unified Medical Language System (UMLS) [1, 2, 3, 4] was designed by the National Library of Medicine (NLM) to overcome problems arising from discrepancies between various medical terminologies. Its concepts reside in a repository called the Metathesaurus (META) [5, 6]. Currently, there are about 871,000 concepts [7]. The Semantic Network (SN) provides an overarching abstraction of the META [8]. The network, consisting of 134 nodes (called semantic types [9]), is organized as a pair of trees rooted at **Event** and **Entity**[1] respectively. The relationships that are the links of the tree structures are IS-A relationships, each of which connects a child semantic type to its parent semantic type.

While the SN is an important abstraction of the META, it is still a difficult mechanism to employ for comprehension due to its large number of semantic types and semantic (i.e., non-IS-A) relationships. Some previous work has been done to help the visualization and navigation of the

[1] A bold font will be used for semantic types.

UMLS knowledge. In [10], a Hypercard browser of Meta-1 (MetaCard) was adapted to enable users to continue the browsing process, extended from the Metathesaurus to a variety of different knowledge sources. In [11], a review about visualization and navigation of knowledge in medical domain was presented. In our previous work [12, 13], we introduced the notion of a metaschema, based on a partition of the SN [14]. A metaschema is a higher-level network that serves as a compact abstraction of the SN. As shown in [12, 13], the notion of the metaschema offers various compact (partial) views which can help users in their orientation to the SN.

In the current version of the SN with its two-tree hierarchy, each semantic type has at most one parent semantic type and can inherit relationships only from this unique parent. Some semantic types are naturally specializations of more than one semantic type. The tree structure does not allow for this kind of multiple parents arrangement. To improve the SN's structure, we previously presented two methodologies to add IS-A links and obtain the Enriched Semantic Network (ESN), a network similar to the SN but permitting multiple parents [15]. The SN's hierarchy is tree-structured, whereas the ESN's is a Directed Acyclic Graph (DAG).

Because the ESN has a more complex hierarchy than the current SN, it is even more critical to develop an ESN metaschema to help in its orientation. In this paper, we will concentrate on extending the notion of metaschema to make it applicable to a DAG hierarchy network and thus to the ESN. We also provide a methodology to derive such a metaschema.

The rest of this paper is organized as follows. Section 2 provides a brief review of the ESN. Section 3 introduces the notion of metaschema for a network having a DAG hierarchy. We first discuss the requirements that a higher-level network must satisfy in order to be a metaschema. We then describe a method by which a metaschema can be derived from a partition of a network like

the SN or the ESN. The separate description is intended to emphasize that for the same network, there may exist several useful metaschemas, corresponding to various partitions of the network. Section 4 presents two metaschemas of the ESN based on two different partitioning techniques which have previously appeared [14, 15]. One metaschema is the "qualities metaschema" ("Q-metaschema" for short) based on the partitioning technique in [15] which is a modification of the partition of the SN in [16]; another is the "cohesive metaschema" ("C-metaschema") based on the technique in [14]. Section 5 contains a comparison and evaluation of the two metaschemas. A general example is presented to demonstrate how a user can employ a metaschema to help in orientation to the ESN. Other applications of the metaschema to auditing for classification errors and to the prevention of redundant classifications in the UMLS are also briefly discussed. Conclusions appear in Section 6.

2 Background

A partition of the SN into 15 groups was previously presented in [16]. Each group in this partition represents a subject area. Six qualities were proposed as desired for such a partition: semantic validity, parsimony, completeness, exclusivity, naturalness, and utility. The semantic validity quality means that each group must be semantically coherent [16]. One way to assess a group's semantic validity is to see if its semantic types together with their IS-A links form a connected subgraph of the SN. This is called the *connectivity property* [15]. Since the SN's IS-A hierarchy consists of two trees, such a connected subgraph must form a tree with a unique root.

Some groups in the partition of [16] do not satisfy the connectivity property. Each such group comprises two or more trees. For example, the *Phenomena* group (Fig. 1) contains two trees; one

of them consists solely of **Laboratory or Test Result** having no IS-A links to any other members.

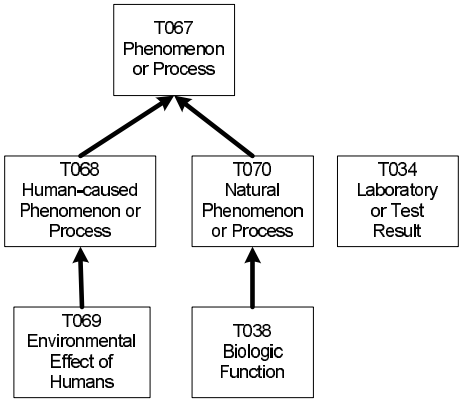

Figure 1: *Phenomena* group

We developed another partitioning technique to derive a cohesive partition of the SN in [12, 13, 14] which requires that all groups in the partition must be connected. Following the cohesive partition in [12, 13, 14], we enforce the *connectivity property* for all groups in the partition of [16] in the design of the ESN in [15]. We presented four transformations to convert each disconnected group into a new connected group, based on reviews of the definitions of all semantic types within a given disconnected group. During the transformations, new potential IS-A links were identified and then, where appropriate, were added.

In another paper,[2] we presented another methodology to identify additional potential IS-A links for the SN. This methodology is based on word-list matching between names and definitions of various semantic types. Using this, we identified four extra IS-A links and added them, too.

Based on our above work, we obtained a new semantic network, referred to as the Enriched Semantic Network (ESN), and an accompanying derived partition of the ESN. The ESN, containing 138 semantic types and 149 IS-A links, has a DAG hierarchy. Eleven semantic types in the ESN have two parents, one has three parents, the two roots semantic types have no parents, and the rest

[2] "An Enriched UMLS Semantic Network with a Multiple Subsumption Hierarchy" submitted for journal publication.

have one parent. The derived partition of the ESN is composed of 19 groups, each of which has a tree hierarchy and thus satisfies the connectivity property. For an excerpt of the ESN hierarchy containing some of the descendants of **Entity**, see Fig. 2. Rectangles represent semantic types and thick arrows represent IS-A links. To emphasize the changes from the original SN, we use dashed thick arrows to denote the added IS-A links and thick dashed rectangles to denote new semantic types (added in the *Anatomy* group only to enable its connectivity [15]). Thin dashed rectangles denote semantic types that originally resided in the **Event** tree of the SN. An ellipsis in a rectangle indicates that some semantic types are not shown due to lack of space.

In the ESN, as in the SN, a pair of semantic types can be linked by 54 kinds of non-hierarchical (semantic) relationships. Each semantic type inherits all the semantic relationships of its parents via IS-A unless such an inheritance is explicitly blocked. Each concept of META is assigned to one or more of the semantic types.

The advantage of the ESN over the SN is that it more accurately captures the existing hierarchical relationships between semantic types. Consider, e.g., **Laboratory or Test Result**. In the SN, it has one parent **Finding** from which it inherits many relationships. (From now on, the term "relationships" refers to semantic relationships, while "IS-A relationships" are IS-As or hierarchical relationships). However, **Laboratory or Test Result**, according to its definition, is also a specialization of **Phenomenon or Process** and should inherit its relationships as well. Thus, in the ESN, **Laboratory or Test Result** also has the relationships inherited from **Phenomenon or Process** (Fig. 3).

We need to develop a metaschema to help in the orientation to the ESN. However, since the ESN is a DAG rather than two trees, the definition of metaschema (as proposed in [12, 13]) is not

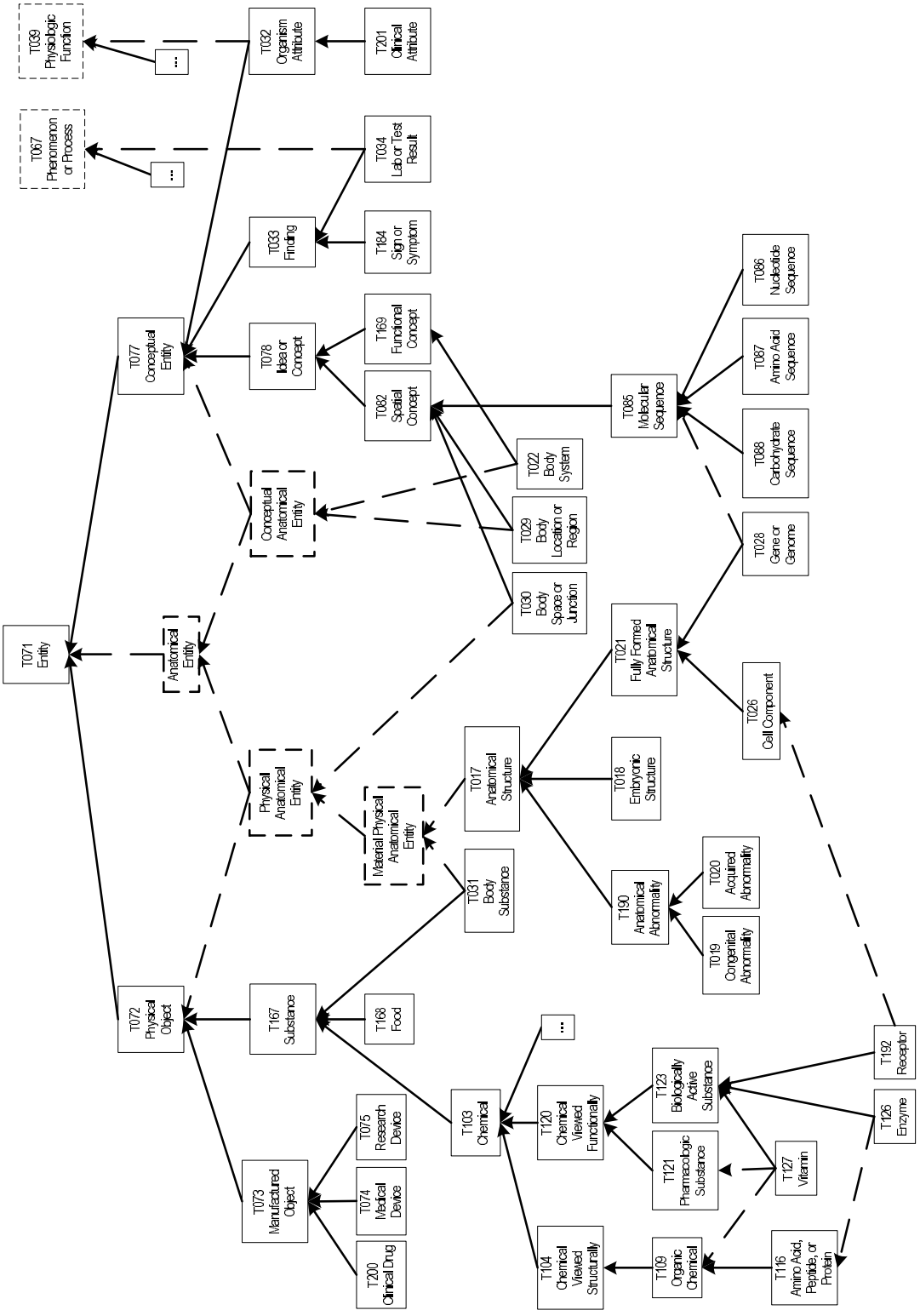

Figure 2: Part of the **Entity** component of the Enriched Semantic Network

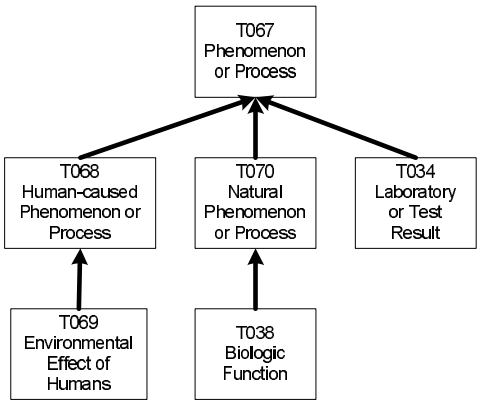

Figure 3: *Phenomenon or Process* group in the ESN

applicable to the ESN. For example, in [12, 13] the hierarchical relationships of the metaschema were derived under the assumption that each semantic type had at most one parent. This is not true for the ESN. In the next section, we will consider the definition of a metaschema for a network with a DAG-structured hierarchy.

3 Methods

The SN is an abstraction of the META which can help users in their orientation to the META. However, since the SN itself is large and complex, we need further help in this orientation task. In [12, 13], we introduced the cohesive metaschema for the SN, based on the cohesive partition [14]. The metaschema is a higher-level abstraction network defined with respect to a given partition of a semantic network.

The notion of metaschema defined in [12, 13] assumes that the underlying semantic network had a multiple-tree hierarchy. It was not designed to handle networks having DAG hierarchies, like the ESN. Therefore, we need to extend the definition of metaschema to be able to derive one for the ESN.

We differentiate between the requirements for and the actual derivation of a metaschema. In

Section 3.1, we characterize the properties of a metaschema for a given semantic network, independent of the way an actual metaschema is derived. For this, we specify the requirements a network should satisfy to qualify as a metaschema of a DAG hierarchy network. The derivation of a metaschema is described in Section 3.2. The separate description is intended to emphasize that for the same semantic network, there may exist several useful metaschemas, corresponding to various partitions of the network.

3.1 Metaschema Requirements

For the requirements of a metaschema, we need some definitions.

Definition (Partition): A *partition* of a set V of elements is a family of subsets $\{V_1, V_2, \ldots, V_k\}$ such that $\bigcup_{i=1}^{k} V_i = V$, and $V_j \cap V_l = \emptyset$ when $j \neq l$. $\Box$

That is, a partition of V is a set of disjoint subsets such that each element of V belongs to exactly one subset.

A partition of the set of semantic types of the SN was presented in [16]. For example, the *Phenomena* group of [16] is {**Phenomenon or Process**, **Human-caused Phenomenon or Process**, **Natural Phenomenon or Process**, **Laboratory or Test Result**, **Environmental Effect of Humans**, **Biologic Function**} (Fig. 1). However, the SN is more than the set of its semantic types; it is a network where the semantic types are connected via hierarchical (IS-A) and non-hierarchical (semantic) relationships. Thus, we need to consider a partition of a graph (network) rather than a set. In particular, we are interested in a partition of the hierarchy of the SN consisting of the semantic types and all the IS-A relationships connecting them. For this, we need the following definition. In all our discussions a graph refers to a directed graph.

Definition (Induced Subgraph): An *induced subgraph* of a graph $G = (V, E)$ induced by a subset of nodes V' ($V' \subseteq V$) is a graph $G' = (V', E')$ where E' contains all the edges of E for which both endpoints are in V'. □

In other words, the V'-induced subgraph of G contains the nodes in V' and all the edges of G connecting them. For example, when G is the hierarchy of the SN, the graph induced by the *Phenomena* group of [16] appears in Fig. 1.

Definition (Partition of a DAG): A *partition* of a DAG $G = (V, E)$ based on a partition $\{V_1, V_2, \ldots, V_k\}$ of V is a collection of subgraphs $\{G_1, G_2, \ldots, G_k\}$ where $G_j = (V_j, E_j)$, $1 \leq j \leq k$, is the subgraph of G induced by V_j. □

Definition (Connected Partition): A partition of a graph is a *connected partition* if each of its subgraphs is a connected graph having a unique root. □

Note that a connected subgraph of a tree must have a unique root, but this is not necessarily true for a DAG. Thus, when dealing with the ESN having a DAG hierarchy, rather than the SN having a tree hierarchy, the requirement for a unique root must be added to the definition.

The partition of the SN hierarchy of [16] is not a connected partition since, for example, the subgraph of the *Phenomena* group is not connected. (See Fig. 1.) On the other hand, the partition of the ESN in [15] is a connected partition. For example, see the subgraph of the *Phenomenon or Process* group in Fig. 3.

Based on the above definitions, we can define the notion of a metaschema for a DAG as follows.

Definition (Metaschema): A *metaschema* of a network G with a DAG hierarchy is a directed network which consists of a set of nodes called meta-semantic types (MSTs) connected via hierar-

chical *meta-child-of* relationships and non-hierarchical meta-relationships satisfying the following two conditions:

1. The set of MSTs represents a connected partition of the given DAG hierarchy.

2. The hierarchy of the metaschema which consists of MSTs and all the *meta-child-of* relationships connecting them is a DAG.

The reason for condition 1 is that an MST standing for a set of semantic types, say S, represents the subgraph of G induced by S. That is, a set of semantic types together with all their hierarchical relationships and semantic relationships. The set of subgraphs of G's hierarchy induced by the set of MSTs in a metaschema make up a connected partition of G. The reason for condition 2 is obvious, in order to qualifying for a hierarchy a network must be a DAG, since a cycle contradicts a hierarchy of its nodes.

3.2 Metaschema Derivation

We will derive the metaschema based on a connected partition. For each group of the partition, we define a meta-semantic type (MST) to represent the group. The MST is named after the unique root of the corresponding group. We will denote by "root of an MST" the ST which is the root of the semantic type group represented by this MST. After defining the MSTs, we need to derive the *meta-child-of* relationships and the meta-relationships for the metaschema.

Let $\{G_1, G_2, \ldots, G_k\}$ be a connected partition of a network G with a DAG hierarchy. Then ST **A** is the unique root of the semantic-type group represented by MST *A*.[3] We called it root of the *A*

[3] An italic font will be used for MSTs.

for short. Since G has a DAG hierarchy, **A** may have several parents $\mathbf{P}_1$, $\mathbf{P}_2$, $\cdots$, $\mathbf{P}_j$. There are two cases.

Case 1: All j parents are associated with a single MST *B*.

Then we define a *meta-child-of* relationship in the metaschema from *A* to *B*. All semantic types associated with *A* are descendants of the root semantic type **A**. Since all **A**'s parents are descendants of the root semantic type **B** of *B*, all semantic types in *A* are descendants of semantic type **B** of *B*. □

Case 2: The j parents $\mathbf{P}_1$, $\mathbf{P}_2$, $\cdots$, $\mathbf{P}_j$ are not associated with one MST.

Suppose these j parents are associated with l MSTs $M_1, M_2, \cdots, M_l$. Then there should be a *meta-child-of* relationship from *A* to each of the l MSTs. Therefore, all semantic types associated with *A* are descendants of each of the roots M_i ($1 \leq i \leq l$) of the l MSTs. □

After we derive the hierarchical *meta-child-of* relationship for the metaschema, we further derive the meta-relationships between two MSTs as follows.

Let **A** be the root of the MST *A*, and let $\mathbf{B}_i$ be a semantic type in the MST *B*. If in the original network there exists a semantic relationship `rel` connecting **A** to $\mathbf{B}_i$, then in the metaschema there exists a link labeled "`rel`"[4] connecting *A* to *B*. Such a link is called a meta-relationship.

Note that semantic type $\mathbf{B}_i$ does not need to be the root of *B*, but the source semantic type of the `rel` relationship must be the root **A** of *A*. Sometimes in the original network, there is a semantic relationship $\texttt{rel}_1$ from ST **C** to ST **D** where **C** is not a root of an MST. Then, we do not say there exists a meta-relationship $\texttt{rel}_1$ from the MST associated with **C** to the MST associated with

[4] A verbatim font will be used for semantic relationships and meta-relationships.

D. The reason for this asymmetry in the requirements for the source and target semantic types of meta-relationships is as follows. For a meta-relationship `rel` to be defined from MST *A* to MST *B*, we want to have a situation that for each semantic type $\mathbf{A}_j$ of MST *A*, there should be some semantic type $\mathbf{B}_i$ of MST *B* such that $\mathbf{A}_j$ `rel` $\mathbf{B}_i$. For this we require that `rel` should be defined at the root semantic type **A** of MST *A*, so `rel` is inherited by all semantic types of MST *A*, which are all descendants of the root semantic type **A**. Such a requirement is not needed for the target semantic type $\mathbf{B}_i$ of the relationship, since not every semantic type in MST *B* has to be a target of such a relationship. It is enough that there exists some semantic type in MST *B* which is a target of `rel` for each source semantic type $\mathbf{A}_j$ of MST *A*.

To reflect the relationship inheritance of the original network, we can define the inheritance of meta-relationships along the hierarchical *meta-child-of* links in the metaschema. Suppose there exist three MSTs *A*, *B*, and *C*, where a *meta-child-of* link connects *B* to *A*. If there is a meta-relationship `rel` from *A* to *C*, then *B* also has a meta-relationship `rel` to *C*, and to all MSTs that have *meta-child-of* links or a chain of *meta-child-of* links to *C*.

The relationships of the metaschema should reflect the relationships in the SN. For example, if *A* is *meta-child-of B*, then every semantic type in *A* should be a descendent of some semantic type in *B*. Similarly, if there is a meta-relationship `rel` from *A* to *B*, then there should be a relationship `rel` defined for every semantic type in *A* to some semantic type in *B*.

In the Section 4, we will apply the metaschema derivation described to the ESN network with its DAG hierarchy.

4 Results: Two Metaschemas

For a given semantic network, any connected partition leads to a metaschema. Each such metaschema will be named after its partition. In this section, we present two possible metaschemas for the ESN, both derived using the method given in the previous section.

4.1 Qualities Metaschema of the ESN

Definition (Qualities Partition): A partition of a set is called a *qualities partition* if it possesses the six qualities (principles) listed in [16]: semantic validity, parsimony, completeness, exclusivity, naturalness, and utility. □

We use "Q-partition" as an abbreviation for "qualities partition" throughout the remainder of the paper.

The partition of the SN in [16] is a Q-partition but not a connected partition. Thus, it cannot be used to derive a metaschema for the SN. However, the partition of the ESN obtained in [15] is a connected Q-partition. Thus, we can derive a metaschema based on the connected Q-partition of the ESN. We refer to the resulting metaschema as the qualities metaschema (Q-metaschema for short).

The hierarchy found in each group in the Q-partition [15] is a tree with a unique root. For each group, we define an MST whose name is the root of the group. Therefore, we get a metaschema of 19 MSTs (see Table 1).

Now, we will derive the hierarchical *meta-child-of* relationships for the Q-metaschema relating to the above Q-partition. For example, the root of MST *Phenomenon or Process* is the semantic

MST	# of STs contained in	# of outgoing relationships
Anatomical Abnormality	3	11
Anatomical Entity	15	1
Chemical	25	2
Clinical Drug	1	0
Conceptual Entity	12	0
Entity	4	1
Event	7	1
Finding	2	8
Geographic Area	1	2
Group	6	7
Manufactured Object	3	2
Molecular Sequence	5	0
Occupation or Discipline	2	1
Occupational Activity	9	3
Organism	17	1
Organization	4	3
Pathologic Function	7	14
Phenomenon or Process	6	2
Physiologic Function	9	4
Total: 19 MSTs	138	63

Table 1: MSTs, STs, and meta-relationships in the Q-metaschema

type **Phenomenon or Process** which is a child of **Event**. **Event** is associated with *Event*; hence, there is a *meta-child-of* from *Phenomenon or Process* to *Event* in the Q-metaschema. The root of *Pathologic Function*, the semantic type **Pathologic Function**, is a child of **Biologic Function** which resides in *Phenomenon or Process*. Thus, there exists a *meta-child-of* from *Pathologic Function* to *Phenomenon or Process*.

By applying this *meta-child-of* derivation process to all 19 MSTs, we get the entire Q-metaschema hierarchy consisting of 17 *meta-child-of* links. Fig. 4 shows this hierarchy. Each node contains the name of the MST and "The # of constituent STs". It is interesting to note that no root of a group in the Q-partition actually has more than one parent. Multiple parents occur only for non-root seman-

tic types in the Q-partition. Hence, the hierarchy of the Q-metaschema has a two-tree structure, as did the original SN.

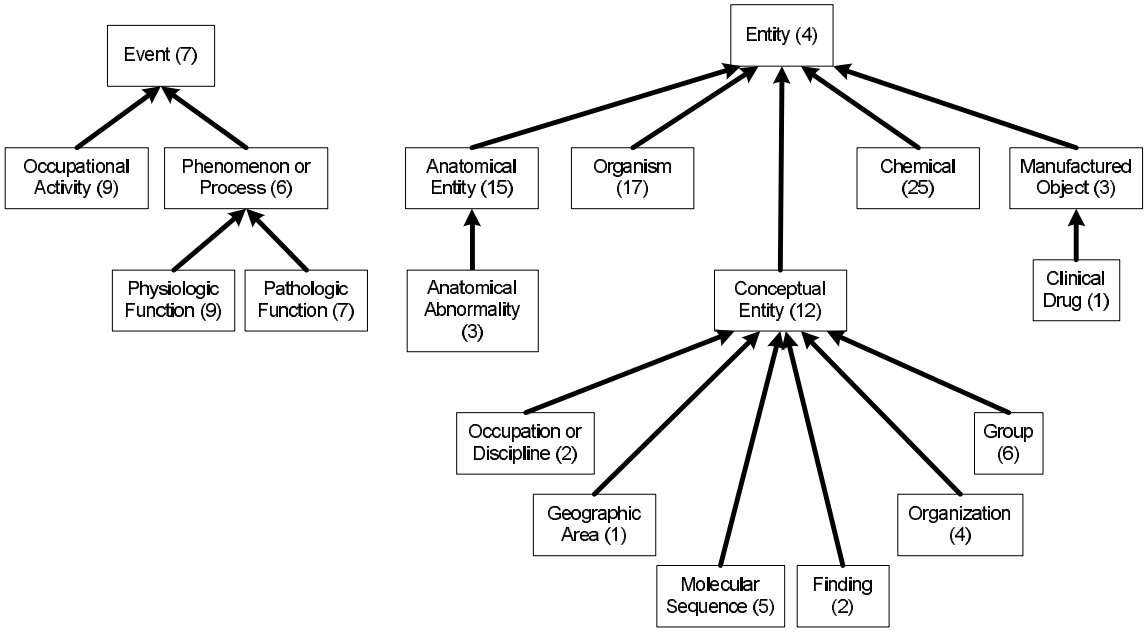

Figure 4: The Q-metaschema hierarchy of the ESN based on the Q-partition

Besides the *meta-child-of* relationships, we need to derive the meta-relationships. For example, **Pathologic Function** introduces the `manifestation_of` relationship to **Physiologic Function**. Since **Pathologic Function** is the root of *Pathologic Function*, and **Physiologic Function** is in *Physiologic Function*, there is a meta-relationship named `manifestation_of` from *Pathologic Function* to *Physiologic Function*. There is a relationship `occurs_in` from **Pathologic Function** to **Group**. Thus, in the metaschema, there is also an `occurs_in` meta-relationship from *Pathologic Function* to *Group*. **Pathologic Function** also defines `co-occurs_with`, `complicates`, `manifestation_of`, and `occurs_in` relationships to **Injury or Poisoning**, which is in *Finding*. Thus, there are also meta-relationships `co-occurs_with`, `complicates`, `manifestation_of`, and `occurs_in` from *Pathologic Function* to *Finding*.

In total, there are 63 meta-relationships belonging to 22 kinds of relationships. Fig. 5 shows

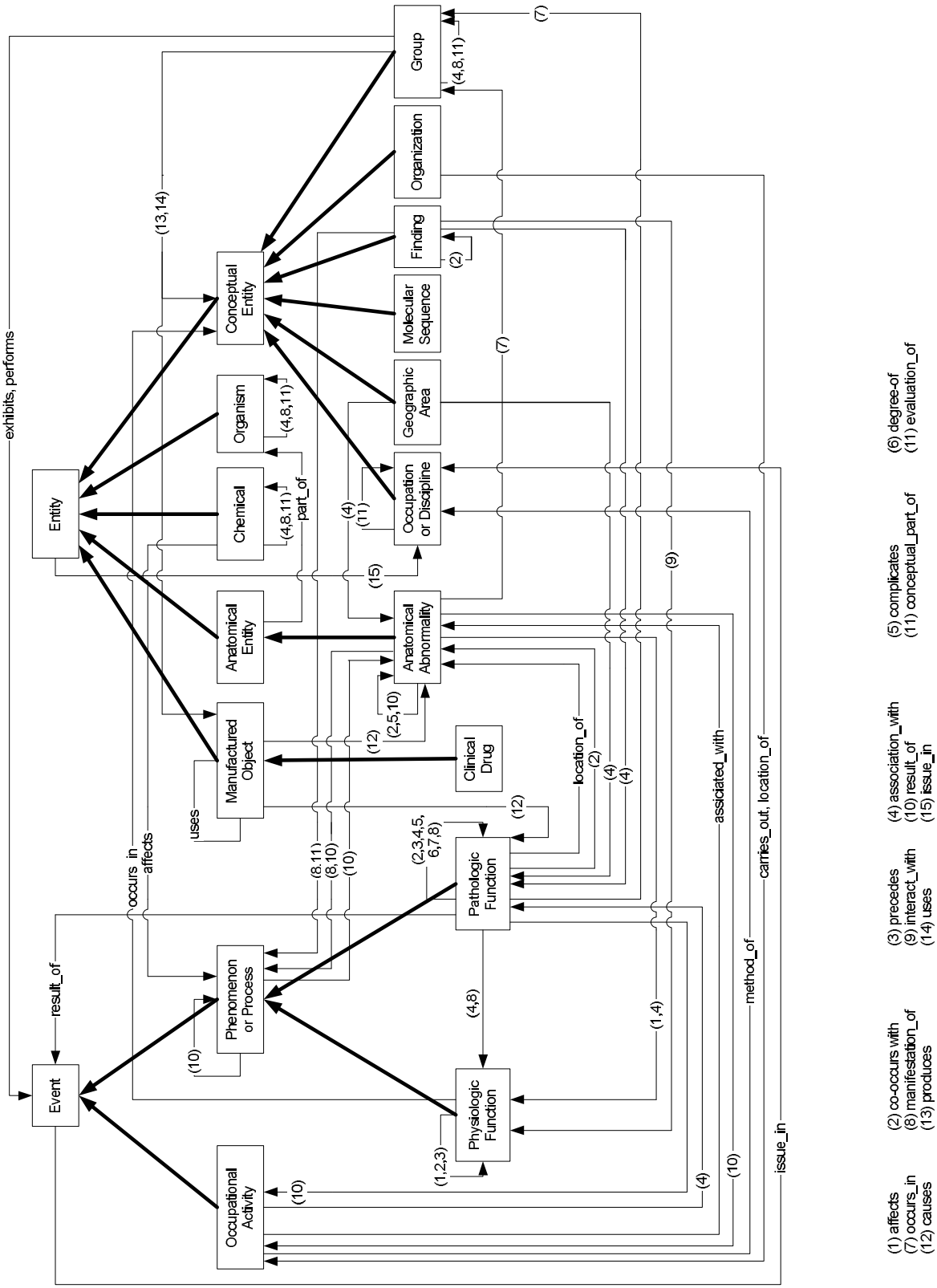

Figure 5: Q-metaschema of the ESN based on the Q-partition of [15]

the whole Q-metaschema, including its 19 MSTs, 17 *meta-child-of* relationships, and 63 meta-relationships. Note that only the meta-relationships introduced at an MST are shown in the figures; the inherited meta-relationships are not shown to avoid clutter. The existence of these additional relationships is easily derived from the figure. A thick arrow denotes a *meta-child-of* relationship, while a labeled thin arrow denotes a meta-relationship. This metaschema, which is displayed on only one page, serves as a compact abstraction of the ESN and can help with user orientation.

4.2 Cohesive Metaschema of the ESN

The technique for deriving a metaschema for the SN described in [12, 13] first defined the "structure" of a semantic type as the set of its defined relationships, either introduced directly or inherited. Semantic types with the same structure were grouped as one semantic-type group. Thus, a structural partition of the SN was obtained. However, that partition was not connected. By applying the three rules defined in [13], a cohesive partition was obtained, consisting of cohesive (singly rooted) semantic-type collections. An MST was then defined to represent each cohesive semantic-type collection. It should be noted that elements at the structural partition were called groups to distinguish from elements at the cohesive partition that were called collections. Based on the cohesive partition, the cohesive metaschema of the SN was derived in [12, 13].

We will now derive a second metaschema of the ESN, referred to as the cohesive metaschema, based on an application of the methodology of [12, 13]. First, we need to obtain a structural partition of the ESN. Note that the structural partition of the ESN will differ from the structural partition of the SN due to the multiple parent configuration and the new distribution of inherited relationships. We will then apply three rules to derive a cohesive partition from the structural partition. Finally, we use the method of Section 3.2 to obtain the cohesive metaschema of the

# of STs in group	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Number of groups	50	10	5	4	3	0	0	1	0	0	0	0	0	1

Table 2: Size Distribution of semantic-type groups

ESN. We use "C-metaschema"and "C-partition" as abbreviations for the cohesive metaschema and the cohesive partition of the ESN, respectively.

4.2.1 Cohesive Partition of the ESN

Since the structural partition depends on the relationships defined at semantic types, it is important to note the relationships of the four new semantic types of the ESN. Following the precedent set by the Digital Anatomist Foundational Model [17], the new **Anatomical Entity** semantic type in the ESN is defined as "a biologic entity which forms the whole or part of or is an attribute of the structural organization of a biological organism." Thus, **Anatomical Entity** introduces the `part_of` relationship directed at **Organism**[5] instead of having its descendant **Anatomical Structure** introduce it, as in the current SN. Thus, in the ESN, **Anatomical Structure** inherits `part_of` from **Anatomical Entity**; it still introduce the `location_of` relationship. The introduction of these relationships is relevant to the structural partition of the ESN, as each of these two semantic types is a root of a semantic-type group.

For the ESN, we get a structural partition consisting of 74 semantic-type groups. We find that most of these contain only one semantic type. Such groups are called singletons. See Table 2 for the distribution of the numbers of groups according to their sizes.

To obtain the C-partition of the ESN, we need to apply the following three rules [13] to the 74 semantic-type groups.

[5]Cornelius Rosse, private communication, 2002.

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. □

Rule 3: Let the semantic types $A_1, A_2, \ldots, A_n$ $(n \geq 2)$ 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. □

However, in applying Rule 2, we found that there are 8 leaf singletons that have multiple parents. Note that some leaves with multiple parents are not singletons as they share the same structure (relationship set) and thus the same group with one of the parents. For example, **Vitamin** has three parents, but it has the same structure as its parent **Biologically Active Substance** and is thus in the same group of that semantic type.

Each of the 8 leaf singletons has a different relationships set from all its parents. Besides this, its parents exhibit different structures and thus are not in the same semantic type group. Rule 2 stated that a leaf singleton should be merged into its parent's semantic type collection. In such a case of multiple parents, we need to determine to which semantic-type collection each singleton should be added since each semantic type must belong to exactly one semantic-type collection in the C-partition. For this, we need to differentiate between different kinds of parents of such a singleton. Among the parents, we need to identify only one parent to be considered the "primary parent" of the singleton; other parents will be considered "secondary parents". The singleton will then be merged to the group of its primary parent. Of course, if the singleton has only one parent, then this parent is considered the primary parent. The process of identifying the primary parent is

discussed in the following subsection.

4.2.2 Identifying the Primary Parent among Multiple Parents

The process of differentiating multiple IS-A links from a singleton to all its parents is guided by the analysis of the names and definitions of the singleton semantic type and its parents. We provide the following guidelines, which are modifications of our guidelines in [18, 19].

We distinguish in a definition of a singleton semantic type among three kinds: the descriptive kind, the functional kind, and the characterizing kind. The descriptive kind captures the essence or nature of the semantic type. The functional kind captures the functionality or usage of the semantic type. The characterizing kind does not describe the essence of the knowledge or its function, but characterizes what kind of knowledge is represented. A definition sometimes has both a descriptive part and a functional part.

For each singleton semantic type having multiple parents, we will find, from all its parent semantic types, which parents are descriptive, which parents are functional and which parents are characterizing. Typically, all parents contribute to the definition of the child; a descriptive parent highlights the essence or nature of the child semantic type; a functional parent highlights the function or usage of the child semantic type; a characterizing parent classifies the kind of knowledge rather than concentrating on the knowledge itself.

Case 1: *Some of the parents are descriptive and the others are functional*.

First check the descriptive part and the functional part of the singleton's name or definition, and determine which part is the primary part.

If the primary part is the descriptive part and there is only one descriptive parent, then choose

this descriptive parent as the primary parent; otherwise, choose the primary parent from among the group of descriptive parents using Case 2.

If the primary part is the functional part and there is only one functional parent, choose this functional parent as the primary parent; otherwise, choose the primary parent from among the group of functional parents using Case 3. □

Case 2: *All parents are descriptive* (*or the primary part of the singleton's name or definition is descriptive*). Among these descriptive parents, distinguish the primary parent by linguistic analysis of the name or definition of the singleton. If the name of one parent is used as a noun and the names of the other parents are used as adjectives in the singleton's name or definition, then the noun defines the primary parent.

If the names of all parents are used as nouns in the name or definition of the singleton, then the last noun is considered the primary noun. The corresponding parent is chosen as the primary parent. If the names of all parents are used as adjectives in the name or definition of the singleton, then the adjective closest to the noun in the name or definition is considered the primary adjective. The corresponding parent is chosen as the primary parent. □

Case 3: *All parents are functional* (*or the primary part of the singleton's name or definition is functional*).

Again, use the linguistic analysis described in Case 2 to identify the primary (functional) parent. □

Case 4: *Some parents are characterizing*.

Examples of such parents are: **Physical Object**, **Functional Concept**, **Spatial Concept**,

and **Conceptual Entity**.

The only case where such a parent semantic type will be the primary parent of a child semantic type is when the child is also considered characterizing. In all other circumstances, we will pick another parent as primary using the other three cases after removing the characterizing parents from consideration. □

In each case, the singleton is merged into the collection of its primary parent in the partition. To capture this situation of a singleton with more than one parent, Rule 2 defined in [12, 13] must be restated as:

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

Let us demonstrate how we identify the primary parent for the 8 singletons with multiple parents, following the method described above. For example, let us consider **Enzyme**, a singleton in the structural partition of the ESN having two parents. One is the old parent **Biologically Active Substance**; the other one is the new parent **Amino Acid, Peptide, or Protein**. **Enzyme** is defined as "a complex chemical, usually a protein, that is produced by living cells and which catalyzes specific biochemical reactions." The descriptive part in the definition is "a complex chemical, usually a protein," while the functional part is "that is produced by living cells and which catalyzes specific biochemical reactions." We need to review the two parents' definitions. **Biologically Active Substance** is defined as "a generally endogenous substance produced or required by an organism, of primary interest because of its role in the biologic functioning of the organism that produces it." This definition emphasizes the role (or usage) of the substance, in our case **Enzyme**, in an organism. Hence, **Biologically Active Substance** is a functional parent. **Amino Acid, Peptide, or**

Protein is defined as “amino acids and chains of amino acids connected by peptide linkages.” This describes the chemical composition of **Enzyme**. (Enzyme is a kind of protein.) Therefore, **Amino Acid, Peptide, or Protein** is a descriptive parent. Since one parent is functional and the other one is descriptive, we need to check both the descriptive part and the functional part of **Enzyme**’s definition and determine which of them is the primary part. We find that what makes enzyme different from other proteins lies in its function (usage), which is catalyzing specific biochemical reactions of an organism. Thus, we think that the functional part of **Enzyme** is the primary part of its definition. So, the functional parent **Biologically Active Substance** is chosen as the primary parent, and **Enzyme** will be merged into the *Biologically Active Substance* group.

As another singleton example, **Gene or Genome**, has two parents in the ESN. One is the old parent **Fully Formed Anatomical Structure**; the other one is **Molecular Sequence**. First, we review the definition of **Gene or Genome**, which is defined as “a specific sequence, or in the case of the genome the complete sequence, of nucleotides along a molecule of DNA or RNA (in the case of some viruses) which represent the functional units of heredity.” In the definition, the descriptive part is “a specific sequence, or in the case of the genome the complete sequence, of nucleotides along a molecule of DNA or RNA (in the case of some viruses).” The functional part is “which represent the functional units of heredity.” Next we review the definitions of its two parents. **Fully Formed Anatomical Structure** is defined as “an anatomical structure that exists only before the organism is fully formed; in mammals, for example, a structure that exists only prior to the birth of the organism. This structure may be normal or abnormal.” This definition is descriptive as no function is discussed. **Molecular Sequence** is defined as “a broad type for grouping the collected sequences of amino acids, carbohydrates, and nucleotide sequences.” This

definition is also descriptive since it does not discuss the function or usage of **Gene or Genome**. Since both parents are descriptive, we have to use linguistic analysis to distinguish the primary parent from the secondary one. We found that in the definition of **Gene or Genome**, the primary noun is "sequence"; therefore, **Molecular Sequence** is the primary parent, and **Gene or Genome** will be merged into the *Molecular Sequence* group.

Some leaf singleton semantic types with two parents have one parent which is a characterizing parent, while the semantic type is not of the characterizing kind. Both **Body Location or Region** and **Body Space or Junction** have the characterizing **Spatial Concept** as a parent. **Body System** has the characterizing **Functional Concept** as parent. All these parents are considered to be secondary while the primary parent semantic types are **Physical Anatomical Entity**, and **Conceptual Anatomical Entity** respectively (where by linguistic analysis "anatomical" is the primary adjective being closer to the noun in the name of the semantic type). Note that although these two primary parents have a characterizing part in their names, namely Physical and Conceptual, these two parts are considered secondary in the names of the parents.

By using the above guidelines, we choose, for each leaf singleton having multiple parents, the primary parent (see Table 3). Those singletons will be merged to the groups of their primary parents according to the revised Rule $2'$. When applying the three rules to the 74 semantic-type groups, we obtained 29 collections of semantic types, called cohesive semantic-type collections (see Table 4). The "*# of STs*" column is the number of semantic types in each semantic-type collection. The "*# of rel.*" column in Table 4 is the number of semantic relationships introduced by the root of each semantic-type collection in the ESN. These relationships will imply the meta-relationships when we derive the ESN's cohesive metaschema. The 29 collections together form a partition, called

Singleton	Primary parent ST	Secondary parent ST
Body Location or Region	Conceptual Anatomical Entity	Spatial Concept
Body Space or Junction	Physical Anatomical Entity	Spatial Concept
Body System	Conceptual Anatomical Entity	Functional Concept
Body Substance	Material Physical Anatomical Entity	Substance
Enzyme	Biologically Active Substance	Amino Acid, Peptide, or Protein
Gene or Genome	Molecular Sequence	Fully Formed Anatomical Structure
Laboratory or Test Result	Phenomenon or Process	Finding
Receptor	Biologically Active Substance	Cell Component

Table 3: Primary/Secondary parents for singletons having multiple parents

cohesive partition ("C-partition" for short).

We wish to stress here that the IS-A link from the singleton to the secondary parent is still part of the ESN. It is just labelled so we can determine uniquely the groups of the partition on which the metaschema is based. Interestingly in most cases, the secondary parent was the original parent in the SN, while the connection to the primary parent is a newly added IS-A link.

4.2.3 Derivation of the Cohesive Metaschema

The derivation of the cohesive metaschema (C-metaschema) for the ESN is based on the above C-partition. For each cohesive semantic-type collection, we define an MST to represent it. It is named after the root of the collection. The *meta-child-of* relationships and meta-relationships are derived as described in Section 3.2. The C-metaschema contains 29 MSTs, 28 *meta-child-of* relationships and 124 meta-relationships belonging to 31 kinds of relationships. Figure 6 shows the

Semantic-Type Collection	# of STs	# of rel.	Semantic-Type Collection	# of STs	# of rel.
Anatomical Abnormality	3	11	Anatomical Entity	8	1
Anatomical Structure	2	1	Animal	9	1
Behavior	3	10	Biologic Function	1	4
Biologically Active Substance	7	7	Chemical	16	2
Entity	8	1	Event	4	1
Finding	2	8	Fully Formed Anatomical Structure	5	7
Group	6	7	Health Care Activity	4	1
Idea or Concept	12	2	Manufactured Object	4	2
Natural Phenomenon or Process	1	2	Occupation or Discipline	2	1
Occupational Activity	3	3	Organism	6	1
Organism Attribute	2	6	Organization	4	3
Pathologic Function	7	14	Pharmacologic Substance	2	11
Phenomenon or Process	4	2	Physiologic Function	7	4
Plant	2	1	Research Activity	2	7
Substance	2	3			
Total 29 Groups				138	124

Table 4: Semantic-type collection of the ESN C-metaschema

cohesive metaschema hierarchy of the ESN with 29 MSTs. Note that this metaschema has a DAG hierarchy, as elaborated in the Section 5. The number in each rectangle denotes the number of semantic types in the MST. Interestingly, the choice of the primary parents for the singleton leaves does not have influence on the metaschema itself, since no leaf is an MST in the metaschema, but in the underlying partition, reflected in the number of semantic types for some groups. Figure 7 shows the C-metaschema including all *meta-child-of* relationships and most meta-relationships. Unfortunately, there is insufficient space to draw all the meta-relationships.

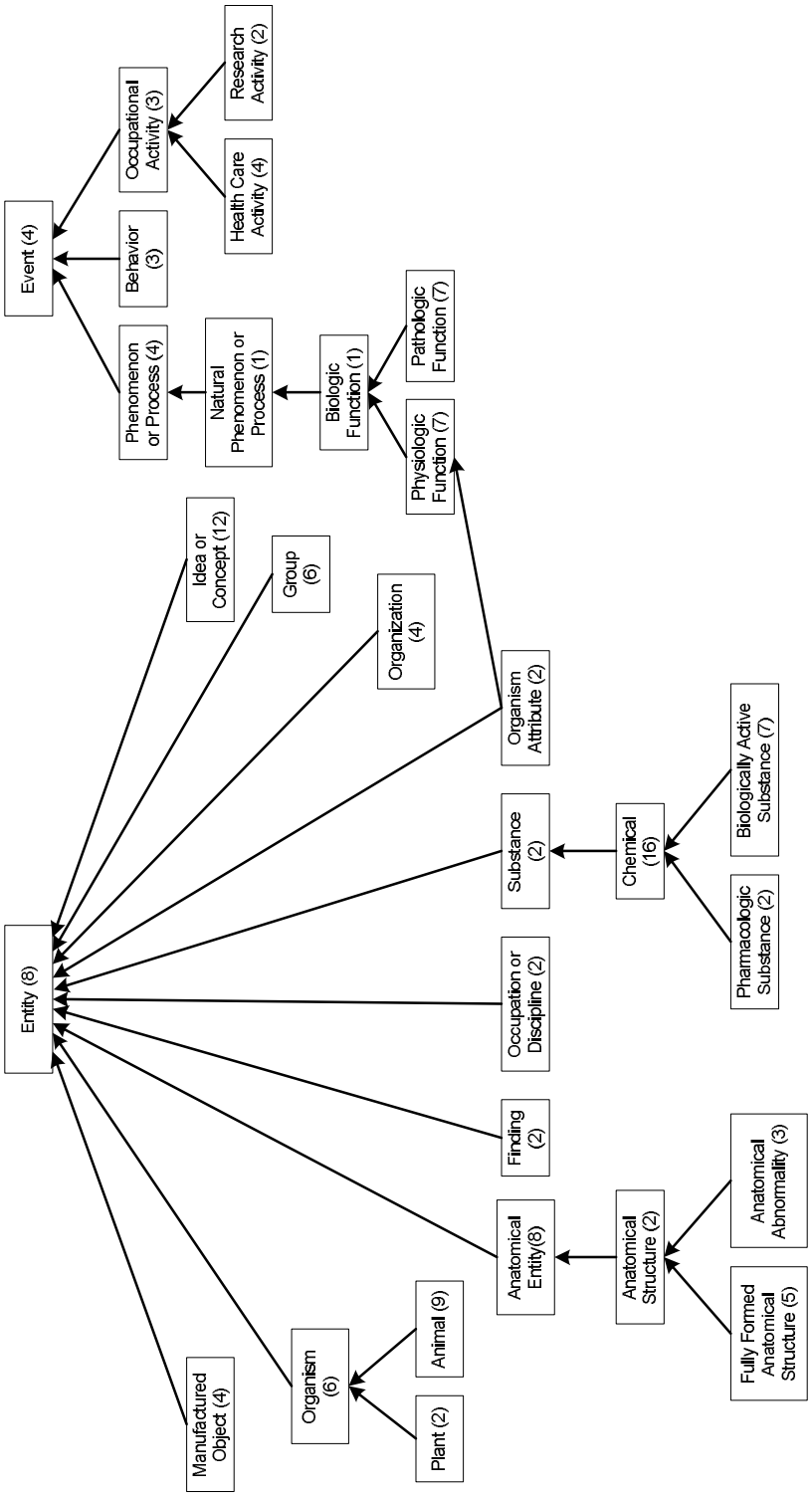

Figure 6: The C-metaschema hierarchy of the ESN

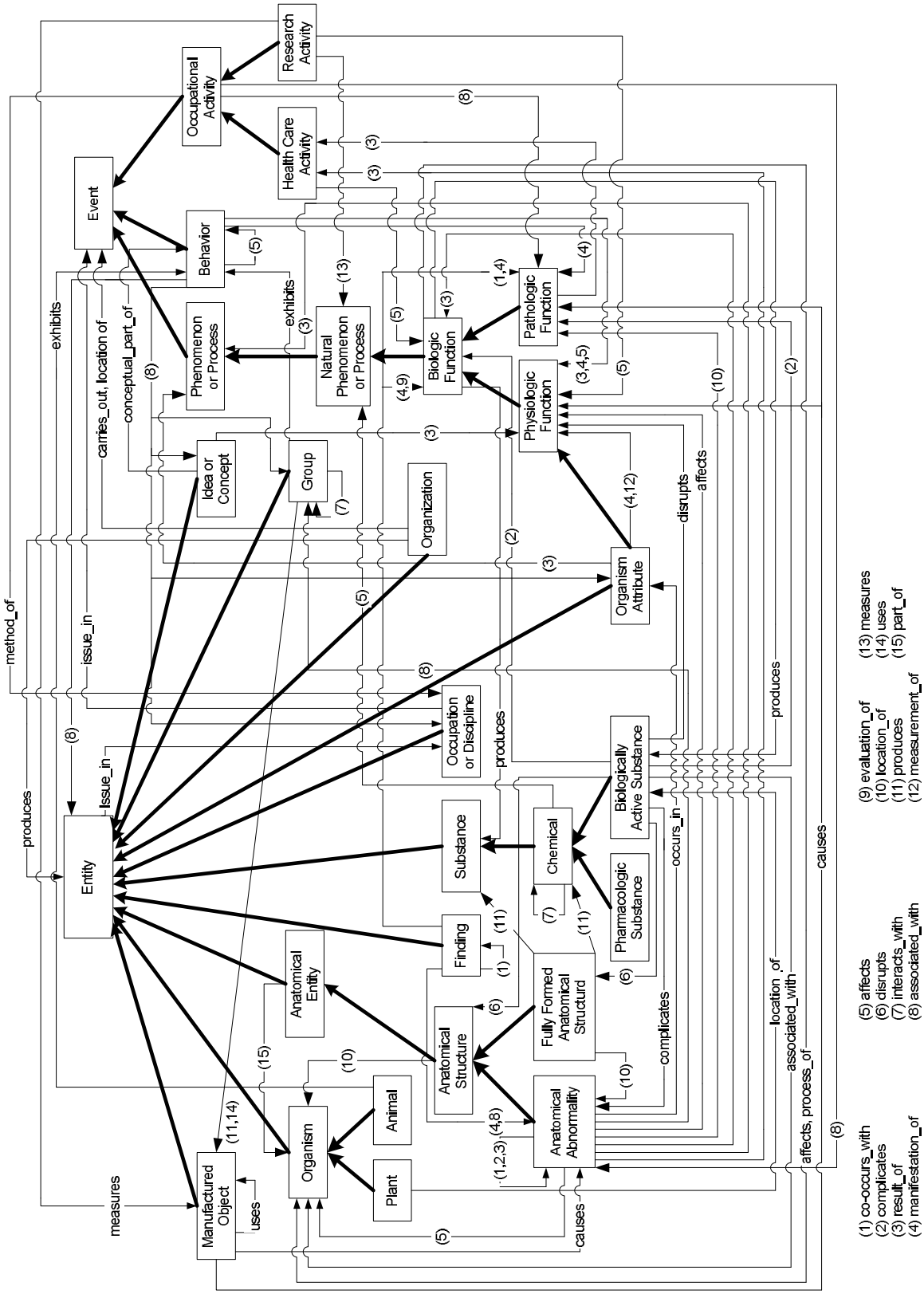

Figure 7: The C-metaschema of the ESN with most meta-relationships

5 Discussion

In [20, 21, 22] we developed techniques to design an upper-level schema for the MED terminology. Similar techniques used can be applied to other medical terminologies such as the SNOMED-CT to abstract its huge concept hierarchy into a schema of classes of groups of similar structural concepts. This role of this schema for the given terminology is similar to the role of the Semantic Network for the META of the UMLS. Then our technique in this paper can be applied to derive a simplified metaschema to serve as a higher-level abstract compact view of the schema and indirectly of the concept hierarchy. The metaschema can be used as the first view for the users to simplify and help their orientation.

5.1 Comparison of Two Metaschemas

Based on the Q-partition of the ESN in [15], we obtain the Q-metaschema of the ESN (Fig. 4). Modifying the method in [12, 13], we get the C-partition and the C-metaschema (Fig. 6) of the ESN. Each of the metaschemas provides an abstract view of the ESN.

The Q-metaschema contains 19 MSTs, while the C-metaschema contains 29 MSTs. There are some common MSTs between the two metaschemas. Among the 19 MSTs in the Q-metaschema, 6 also appear in the C-metaschema, representing the same semantic-type collections in both the Q-partition and the C-partition. That means both metaschemas agree that these 6 MSTs are quite important in the abstraction of the ESN, providing the metaschema with the representation of natural units of semantic types. These 6 MSTs are: *Anatomical Abnormality*, *Finding*, *Group*, *Occupation or Discipline*, *Organization*, and *Pathologic Function* (see Table 5). Together they cover 24 semantic types (i.e., 17.4% of the ESN).

MST	Semantic-type collection
Anatomical Abnormality	Anatomical Abnormality; Acquired Abnormality; Congenital Abnormality
Finding	Finding; Sign or Symptom
Group	Group; Family Group; Age Group; Population Group; Professional or Occupational Group; Patient or Disabled Group
Occupation or Discipline	Occupation or Discipline; Biomedical Occupation or Discipline
Organization	Organization; Professional Society; Health Care Related Organization; Self-help or Relief Organization
Pathologic Function	Pathologic Function; Experimental Model of Disease; Disease or Syndrome; Injury or Poisoning; Neoplastic Process; Mental or Behavioral Dysfunction; Cell or Molecular Dysfunction

Table 5: Identical MSTs in Q-metaschema and C-metaschema

There are some obvious differences between the two metaschemas and their underlying partitions. The Q-metaschema contains two trees, while the C-metaschema is a DAG. In the Q-partition, semantic type **Organism Attribute** and its child **Clinical Attribute** are part of the *Physiologic Function* group. However, in the C-partition, these two semantic types form a separate semantic-type collection due to structural differences; hence, there is an MST named *Organism Attribute* in the C-metaschema. This MST has two parents in the C-metaschema: one is *Entity*, the other is *Physiologic Function*. These two *meta-child-of* relationships make the C-metaschema a DAG.

In the Q-metaschema, the MSTs *Clinical Drug* and *Geographic Area* each represent a semantic-type collection that contains only a leaf singleton semantic type. In the C-metaschema, there is no such case because Rule $2'$ explicitly merges each leaf singleton into its parent's group. On the other hand, the C-metaschema contains two MSTs, *Natural Phenomenon or Process* and *Biologic*

Function, that each represent a semantic-type collection consisting of only one internal (non-leaf) semantic type. This is because a semantic type like **Natural Phenomenon or Process** has a different structure (relationship set) from its parent and its child, and it is not merged into its parent's group since it is internal node of the DAG.

There are also some other differences between the two metaschemas and their underlying partitions. Some semantic-type collections in the Q-partition are split into several separate semantic-type collections in the C-partition, which results in several separated MSTs in the C-metaschema. These MSTs in the C-metaschema are more refined than the corresponding MSTs in the Q-metaschema (Table 6). The number of STs in each MST appears in parenthesis.

MST in Q-metaschema	MST in C-metaschema
Anatomical Entity	Split into three MSTs: Anatomical Entity; Anatomical Structure; and Fully Formed Anatomical Structure
Chemical	Split into three MSTs: Chemical; Pharmacologic Substance; and Biologically Active Substance
Event	Split into two MSTs: Event; Behavior
Occupational Activity	Split into three MSTs: Occupational Activity; Health Care Activity; and Research Activity
Organism	Split into three MSTs: Organism; Plant; Animal
Phenomenon or Process	Split into three MSTs: Phenomenon or Process; Natural Phenomenon or Process; and Biologic Function
Physiologic Function	Split into two MSTs: Physiologic Function; Organism Attribute

Table 6: Refined MSTs in the C-metaschema

For example, the *Chemical* group in the Q-partition is split into three semantic-type collections in the C-partition. One is *Pharmacologic Substance*, which contains **Pharmacologic Substance** and its child. One is *Biologically Active Substance* containing **Biologically Active Substance** and its children. The third is *Chemical*, which contains **Chemical** and all its descendants, except those in the *Pharmacologic Substance* and *Biologically Active Substance* semantic-type collections. This

is because **Pharmacologic Substance** introduces the five relationships $complicates$, $diagnoses$, $disrupts$, $prevents$, and $treats$, and **Biologically Active Substance** introduces $associated\_with$, $complicates$, and $disrupts$. Since these relationships are not defined at **Chemical**, **Pharmacologic Substance** and **Biologically Active Substance** start new MSTs.

Another example is the MST *Anatomical Entity* in the Q-metaschema. This MST represents a group of 15 semantic types. This group is split into three semantic-type collections in the C-partition. One collection contains **Anatomical Entity** and its 7 descendants which are not in the other two collections; the second collection contains **Anatomical Structure** and **Embryonic Structure**; the third contains **Fully Formed Anatomical Structure** and its children. This is because **Anatomical Structure** introduces a new relationship `location_of` that is not defined for its ancestors, and **Fully Formed Anatomical Structure** defines two new relationships, $contains$ and $produces$. Therefore, **Anatomical Structure** and **Fully Formed Anatomical Structure** both begin new MSTs in the C-metaschema.

On the other hand, the semantic-type collection *Manufactured Object* in the C-partition, containing **Manufactured Object**, **Medical Device**, **Research Device**, and **Clinical Drug**, is split into two groups in the Q-partition. One group is *Clinical Drug*, containing only **Clinical Drug**; the other group is *Manufactured Object*, consisting of the remainder of the three semantic types. This is because in the C-partition, the leaf singleton **Clinical Drug** is merged into the group of its parent semantic type **Manufactured Object**, while in the Q-partition, there is no rule to avoid leaf singletons.

From the above comparison, we can see that the C-metaschema generally provides a more refined abstract view of the ESN than the Q-metaschema. The collections that are similar in the

two metaschemas, up to the refinement level, cover 92 semantic types (i.e., 66.7% of the ESN).

There are 22 semantic types of the ESN that are assigned to MSTs differently in the two metaschemas. The MSTs involved are *Entity*, *Conceptual Entity*, *Molecular Sequence*, and *Geographic Area* in the Q-metaschema and *Entity*, *Idea or Concept*, and *Substance* in the C-metaschema. There are also cases where MSTs with the same name in the two metaschemas represent different semantic-type collections in the underlying partitions. For example, *Entity* appears in both metaschemas, but it represents different semantic-type collections in each. Let us note that the major differences in the two metaschemas involve only 15.9% of the ESN.

An interesting measure for the two metaschemas is how many semantic relationships of the ESN are not reflected by the meta-relationships of the metaschema. There are 571 defined semantic relationships in the ESN, but when we take into account the inherited semantic relationships, the number is 6977. For the Q-metaschema, there are 699 out of the 6977 semantic relationships (about 10%) that are not reflected. For the C-metaschema, there are only 285 out of the 6977 semantic relationships (about 4%) that are not reflected. Hence, the C-metaschema is better at capturing the relationship structure of the ESN. The reason for this is not just the larger number of MSTs; it is also due to the fact that the initial design of the collections is based on the grouping of all semantic types with the same set of relationships. This organization minimizes the cases of relationships introduced at a non-root ST of a collection. Furthermore, all 285 semantic relationships that are not reflected by the C-metaschema are defined at leaves and are not inherited. This is not the case for the Q-metaschema.

Although the Q-metaschema captures less semantic relationships than the C-metaschema, it contains less MSTs. Therefore, its network is more compact and simpler than that of the C-

metaschema. Hence, the whole Q-metaschema with all its meta-relationships can be displayed on one page. To summarize, both metaschemas have their advantages and disadvantages and each can serve as a compact abstraction of the ESN.

Note that conceptually there is loss of knowledge in a metaschema view versus the complete ESN diagram. The loss occurs both in the nodes and in the links. In the nodes, only the roots of the collections are appearing and represent the rest of the semantic types. In the links, we present only the meta-relationships standing for the semantic relationships defined at the roots of the semantic type collections. Hence, we miss semantic relationships whose sources are non-root semantic types. Furthermore, for the meta-relationships the knowledge of the target ST for each relationship is not reflected is the metaschema. Such knowledge loss is unavoidable whenever we try to capture a large network in a compact abstract view.

However, we note that there is no permanent loss of any knowledge as the metaschema is just the first view a user will employ when orienting herself to the ESN. The user will still have access to all the ESN's elements. In Section 5.2, we show how various partial graphical views, based on the metaschema, provide complete knowledge of small, comprehensible portions of the ESN. In particular, the fact that Figure 7 of the C-metaschema cannot show all the 124 meta-relationships defined for it is not so critical, as the missing meta-relationships and the semantic relationships represented by them will be displayed in the various partial views.

5.2 Applications of a Metaschema

In this section, we briefly describe three applications of a metaschema. (These applications were described in detail in [13].)

The first application uses the metaschema notion for auditing the classification of concepts in the UMLS, where concepts of the META are assigned to one or more semantic types of the ESN. Auditing the META concept classification is a persistent, and perhaps overwhelming, task for UMLS professionals. There is a need to design auditing techniques for the UMLS which will minimize the effort and maximize the probability of finding errors.

Previously published papers have exploited UMLS knowledge to help the auditing of the META. For example, in [23], Cimino used semantic methods to uncover the UMLS classification errors. Gu et al. [24] and Bodenreider [25], respectively, described techniques to support the maintenance of the META by constructing object-oriented models of the UMLS. Hole demonstrated a new method to find missed synonymy in the META [26].

Metaschemas, too, can be used to help uncover classification errors in the META. In a metaschema, 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. However, it is more likely that a concept will be erroneously assigned to several semantic types residing in different meta-semantic types than to 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 audit 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 other concepts, and the effort to review them is limited since their number is not very large. For more details and examples, see [27].

One example is the concept SERIAL ANALYSIS OF GENE EXPRESSION that was assigned to **Plant** and **Research Activity** simultaneously. In the C-metaschema, these two semantic types belong to MSTs *Plant* and *Research Activity*, respectively. The MST *Plant* consists of semantic types residing in the **Entity** part of the ESN, while the *Research Activity* contains semantic types residing in the **Event** part. They are quite different in nature. Hence, the classification of a concept assigned to these two MSTs is suspicious. As a matter of fact, from the name of the concept, we see that the assignment of the concept to **Plant** is erroneous and should be removed. A typical user for this application is an NLM employee who is an auditor of the UMLS concept classifications. He (or she) can utilize the metaschema to help in detecting classification errors.

The second application is using various kinds of graphical views, based on the metaschema, to enhance user orientation to the ESN. These views include:

1. A collection subnetwork which is a subgraph of the ESN induced by a semantic-type collection (See Figure 8).

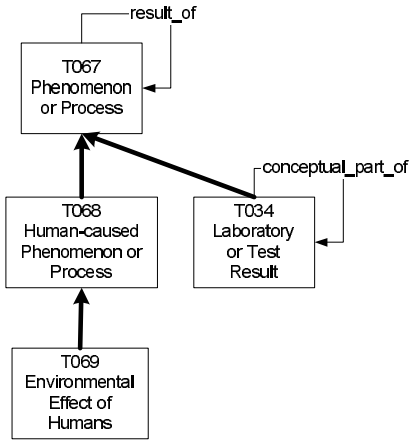

Figure 8: *Phenomenon or Process* collection subnetwork

2. The focus MST submetaschema which contains an MST in which the user is interested (a focus MST) and all its neighboring MSTs (See Figure 9).

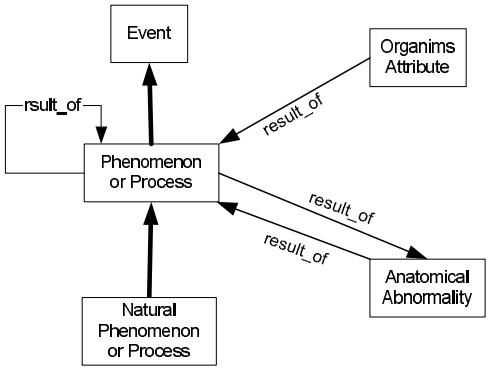

Figure 9: Focus *Phenomenon or Process* submetaschema

3. The bi-collection subnetwork which is the subgraph of the ESN induced by two neighboring collections (i.e., the corresponding MSTs are neighbors). (See Figure 10.)

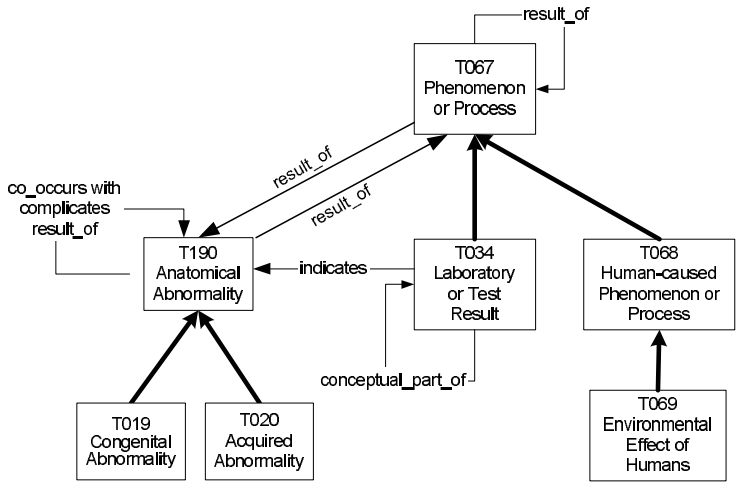

Figure 10: Bi-collection subnetwork of *Phenomenon or Process* and *Anatomical Abnormality*

Let us describe a scenario of a user employing these graphical views to gain an orientation. The user starts by viewing the metaschema hierarchy (Figure 6) to identify which MST is closest to her interest. Suppose it is *Phenomenon or Process*. Then the viewer looks at the *Phenomenon or Process* collection subnetwork (Figure 8), and she can see all the semantic types in the collection and all relationships connecting them. Once the user gains this knowledge, she might want to see the interaction between semantic types of this collection and other external semantic types. But the

number of relationships between semantic types of this collection and other semantic types may be overwhelming. Thus, the user can first view an abstraction of this interaction by viewing the *Phenomenon or Process* focus MST submetaschema where the relationships to and from the various neighboring MSTs of *Phenomenon or Process* are shown (Figure 9). If, for example, the user identifies an interest in the interaction between *Phenomenon or Process* and *Anatomical Abnormality*, she can choose to view the *Phenomenon or Process*/*Anatomical Abnormality* bi-collection subnetwork (Figure 10). The subnetwork contains all the interactions in the ESN between the semantic types of these two collections. Note that this view may show relationships from non-root semantic types of a collection which were not reflected in the metaschema, e.g., the `indicates` relationship from **Laboratory or Test Result** to **Anatomical Abnormality**. That is, the loss of relationship knowledge in the metaschema is not a permanent loss, and the "lost relationships" appear in the bi-collection views. If the user wants to learn about all the external relationships of the *Phenomenon or Process* collection, then she can view a sequence of bi-collection subnetworks, one for each pair of neighboring MSTs in the focus MST submetaschema. In this way, the overwhelming task of reviewing all the relationship interactions of one collection is divided into a sequence of manageable tasks, supporting user comprehension efforts. A potential user for this application is a medical informatics student or professional which is not familiar with the SN of the UMLS and is trying to achieve an orientation into the SN.

For the third application the user is an NLM employee classifying concepts of the UMLS who can use the graphical views, provided by a metaschema framework, to help detect and avoid redundant classifications within an MST. A classification of a concept to a semantic type while it has a simultaneous assignment to a descendant of the semantic type is called a redundant classification

and is forbidden in the UMLS [28]. We demonstrate this with regards to classifications involving chemicals and will use the *Chemical* collection subnetwork view (Fig 11).

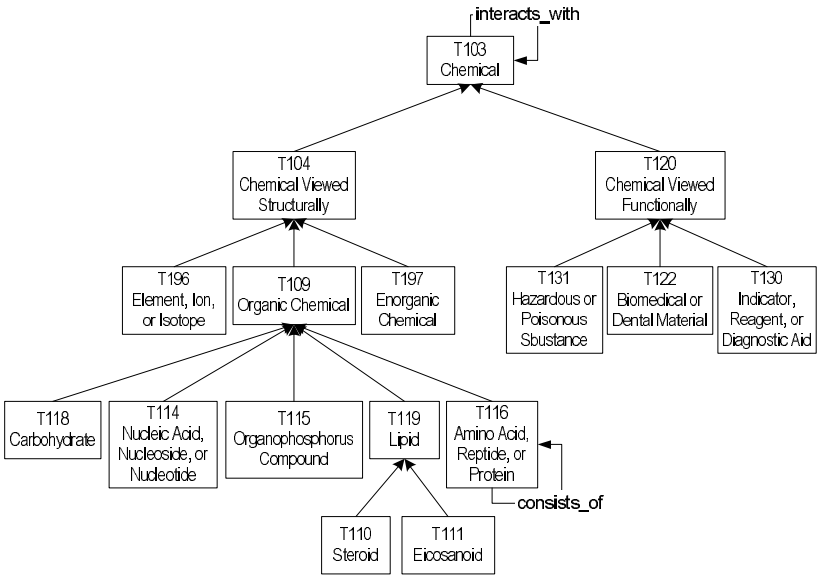

Figure 11: *Chemical* collection subnetwork

As an example, consider the concept CONCENTRIN assigned to semantic types **Steroid**, **Lipid**, and **Organic Chemical**. From the *Chemical* collection subnetwork in Fig 11, we can see that **Organic Chemical** is the parent of **Lipid**, which in turn is the parent of **Steroid**. Therefore, the assignment of concept CONCENTRIN to **Organic Chemical** and **Lipid** is redundant since it can be inferred from the assignment to **Steroid**.

In another example, there are two concepts, FLUOR PROTECTOR and AELITEFIL, assigned to three semantic types within the same subnetwork, **Chemical Viewed Structurally**, **Organic Chemical** and **Inorganic Chemical**, where **Chemical Viewed Structurally** is the parent of the other two semantic types. Hence, the assignment of the two concepts to **Chemical Viewed Structurally** is redundant. Furthermore, a concept cannot be both an organic chemical and an inorganic chemical simultaneously. As a matter of fact, the two concepts are organic chemicals.

The following statistics demonstrate that such users might need help of graphical views in determining concept classification. In [29], while reviewing all intersections of semantic types in the SN of the 1998 version of the UMLS, we discovered that 8,622 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 8,622 concepts. It was found that a portion (38.3%) of the redundant classifications were properly removed. However, a large number of them (57%) were still present. A third portion (4.7%) of the redundant classifications were 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. The graphical views provided by a metaschema framework might help such users in concept classification, especially in avoiding creating new redundant classifications while removing existing redundant classifications.

6 Conclusion

The UMLS's Semantic Network (SN) provides an abstract view for its Metathesaurus and helps with its comprehension. However, the SN itself can be hard to comprehend since it is complex and large. At the same time, the SN does not allow for multiple parents and multiple inheritance. The ESN with its DAG structure [15], enabling multiple parents, is more accurate but also more complex than the SN. In this paper, we presented the requirements for and derivation of metaschemas that support the comprehension of the ESN. We obtained a "qualities metaschema" (Q-metaschema) based on the qualities partition (Q-partition) and the "cohesive metaschema" (C-metaschema) based on the cohesive partition (C-partition). We compared the two metaschemas

and their underlying partitions. The Q-metaschema is a more compact metaschema, and the C-metaschema is more refined. Each metaschema can be used as compact abstract layer of the ESN to help in its comprehension. Potential applications of metaschemas were described.

7 Acknowledgment

This research was supported by contract #N01-1-3543 from the National Library of Medicine (NLM).

References

[1] Campbell KE, Oliver DE, and Shortliffe EH. The Unified Medical Language System: Toward a collaborative approach for solving terminologic problems. *JAMIA*, 5(1):12–16, 1998.

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

[3] Lindberg DAB, Humphreys BL, and McCray AT. The Unified Medical Language System. *Methods of Information in Medicine*, 32:281–291, 1993.

[4] Humphreys BL and Lindberg DAB. Building the Unified Medical Language System. In *Proc. Thirteenth Annual Symposium on Computer Applications in Medical Care*, pages 475–480, Washington DC, November 1989.

[5] Schuyler PL, Hole WT, Tuttle MS, and Sherertz DD. The UMLS Metathesaurus: Representing different views of biomedical concepts. *Bull Med Libr Assoc*, 81(2):217–222, 1993.

[6] Tuttle MS, Sherertz DD, Olson NE, Erlbaum MS, Sperzel WD, and Fuller LF et al. Using META-1, the first version of the UMLS Metathesaurus. In *Proc. Fourteenth Annual SCAMC*, pages 131–135, 1990.

[7] U. S. Dept. of Health and Human Services, National Institutes of Health, National Library of Medicine. Unified Medical Language System (UMLS). 2002.

[8] 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*, pages 45–55, Mekler, Westport, CT, 1993.

[9] McCray AT and Hole WT. The scope and structure of the first version of the UMLS Semantic Network. In *Proc. Fourteenth Annual SCAMC*, pages 126–130, Los Alamitos, CA, November 1990.

[10] Nelson SJ, Sheretz DD, Tuttle MS, and Erlbaum MS. Using metacard: A hypercard browser for biomedical knowledge sources. In *Proc. Annu Symp Comput Appl Med Care*, pages 151–154, 1990.

[11] Tuttle MS, Cole WG, Sheretz DD, and Nelson SJ. Navigating to knowledge. *Methods Inf Med*, 34(1-2):214–231, 1995.

[12] Halper M, Chen Z, Geller J, and Perl Y. A metaschema of the UMLS based on a partition of its Semantic Network. In *Proc. 2001 AMIA Annual Symposium*, pages 234–238, Washington DC, November 2001.

[13] Perl Y, Chen Z, Halper M, Geller J, Zhang L, and Peng Y. The cohesive metaschema: a higher-level abstraction of the UMLS Semantic Network. *Journal of Biomedical Informatics*, 35(3):194 – 212, 2003.

[14] Chen Z, Perl Y, Halper M, Geller J, and Gu H. Partitioning the UMLS Semantic Network. *IEEE Trans. Information Technology in Biomedicine*, 6(2):102–108, June 2002.

[15] Zhang L, Perl Y, Geller J, Halper M, and Cimino JJ. Enriching the structure of the UMLS Semantic Network. In *Proc. of the 2002 AMIA Annual Symposium*, pages 939–943, San Antonio, TX, November 2002.

[16] McCray AT, Burgun A, and Bodenreider O. Aggregating UMLS semantic types for reducing conceptual complexity. In *Proc. Medinfo 2001*, pages 171–175, London, UK, September 2001.

[17] Michael J, Mejino J, and Rosse C. The role of definitions in biomedical concept representation. In Bakken S, editor, *Proc. 2001 AMIA Annual Symposium*, pages 463–467, Washington DC, November 2001.

[18] Gu H, Perl Y, Geller J, Halper M, and Singh M. A methodology for partitioning a vocabulary hierarchy into trees. *Artificial Intelligence in Medicine*, 15(1):77–98, January 1999.

[19] Gu H, Perl Y, Geller J, Halper M, and Singh M. Partitioning an object-oriented terminology schema. *Methods of Information in Medicine*, 40(3):204–212, July 2001.

[20] Gu H, Halper M, Geller J, and Perl Y. Benefits of an object-oriented database representation for controlled medical terminologies. *JAMIA*, 6(4):283–303, July/August 1999.

[21] Liu L, Halper M, Geller J, and Perl Y. Controlled vocabularies in OODBs: Modeling issues and implementation. *Distributed and Parallel Databases*, 7(1):37–65, 1999.

[22] Liu L, Halper M, Geller J, and Perl Y. Using OODB modeling to partition a vocabulary in structurally and semantically uniform concept groups. *IEEE TKDE*, 14(4):850–866, 2002.

[23] Cimino JJ. Auditing the unified medical language system with semantic methods. *JAMIA*, 5:41–51, 1998.

[24] Gu H, Perl Y, Geller J, Halper M, Liu LM, and Cimino JJ. Representing the UMLS as an object-oriented database: modeling issues and advantages. *JAMIA*, 7(1):66–80, Jan-Feb 2000.

[25] Bodenreider O. An object-oriented model for representing semantic locality in the UMLS. *Proc. Medinfo. 2001*, 10(1):161–165, 2001.

[26] Hole WT and Srinivasin S. Discovering missed synonymy in a large concept-oriented metathesaurus. *JAMIA*, 7:354–358, 2000.

[27] Gu H, Min H, Peng Y, Zhang L, and Perl Y. Using the metaschema to audit UMLS classification errors. In *Proc. 2002 AMIA Annual Symposium*, pages 310–314, San Antonio, TX, November 2002.

[28] McCray AT and Nelson SJ. The representation of meaning in the UMLS. *Methods of Information in Medicine*, 34:193–201, 1995.

[29] Gu H, Perl Y, Geller J, Halper M, Liu L, and Cimino JJ. Representing the UMLS as an OODB: Modeling issues and advantages. *JAMIA*, 7(1):66–80, Jan/Feb 2000.