Evaluation and Application of a Semantic Network Partition

James Geller, Yehoshua Perl, Michael Halper, Zong Chen, Huanying Gu

Abstract— **Semantic networks are excellent knowledge representation structures. However, large semantic networks are hard to comprehend. To overcome this difficulty, several methods of partitioning have been developed which rely on different mixes of structural and semantic methods. However, little has appeared in the literature concerning the question whether a partition of a semantic network creates subnetworks that agree with human insight. We address this issue by presenting a comparison between the results of an algorithmic partitioning method and a partition created by a group of experts. Subsequently, we show how a network partition can be used to generate various partial views of a semantic network, which facilitate user orientation. Examples from the UMLS Semantic Network are used to demonstrate partial views.**

Keywords— **Semantic Network, Evaluation, Partitioning, Semantic Type, Orientation, Subnetwork, Partial View**

I. Introduction

Semantic networks [1], [2], [3] are excellent repositories for conceptual knowledge. The standard representation of every existing semantic network is a graphical language, although internally in a computer, semantic networks are represented by some kind of symbolic knowledge representation formalism. Well designed, small semantic network diagrams are easy to understand and interpret. However, the same does not hold true for large semantic networks. At the same time, only large semantic networks are of any practical use. Diagrams of large semantic networks are confusing in the best case, and completely "unreadable" due to intersections and overlaps in the worst case. This situation is somewhat reminiscent of the difficulties in understanding programming language code before the popularization of *modular programming.* Due to the nonlinear nature of semantic networks, it has taken much longer to develop structures similar to the modules in programming. Recently, methodologic approaches for generating such modules have become available under names such as *partitioning.* Various methods for partitioning semantic networks have appeared in the literature. Some of them are algorithmic, e.g., [4], [5], [6], while others are semantic, e.g., [7], [8], [9]. However, these approaches typically leave questions open about whether the resulting partitions are meaningful in the eyes of human experts. In this paper, we present a study evaluating the results of the partitioning method of [4].

We are still left with the problematic situation of small diagrams helping comprehension, but large, complex diagrams overwhelming the viewer. It is widely believed that "a picture is worth a thousand words." Diagrammatic representations make use of the high bandwidth of the human vision channel. Moreover, certain operations are hard-wired into the human vision system, such as line detection. However, in a jumble of intersections it is hard to detect and follow a line. Contrary to popular belief, zooming does not help in such a case, because it typically makes the end points of the line disappear from the viewing surface. The only resolution to this contradiction is to carve out *meaningful* views from an overwhelming semantic network diagram. Such views need to contain a few select elements that are chosen *not* by physical distance but by semantic proximity. The partition of [4] allows us to formulate several partial views of a semantic network that conform to this principle, and we will define these views in detail below.

Many practical applications of semantic networks can be found in medical informatics. The one semantic network that probably has the widest distribution in the field is the Semantic Network [10] of the Unified Medical Language System (UMLS) [11], subsequently referred to as SN. The SN serves as a high level abstraction for the Metathesaurus (META), which is the UMLS concept repository. While the methods of [4] are completely general, we chose the UMLS SN for our study, as few other semantic networks are as widely known.

Section 2 describes how we evaluated an algorithmic partition of SN by comparing it with a partitioning of SN created by human experts. Section 3 discusses how to utilize the partition by offering views to orient the user to the UMLS SN. Section 4 contains conclusions.

II. Evaluating an Algorithmic Partitioning

The hypothesis underlying this paper is that although the partitioning technique of [4], yielding what is called the *cohesive partition*, is based primarily on structural aspects, it 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 partition, called in [4] a *semantic-type collection*, is expected to be a unified group of nodes describing some specific subject area. In other words, we assume that if two nodes (called *semantic types*) in the UMLS SN have identical (or even approximately identical) sets of relationships, then they are also close semantically. How can we evaluate whether an algorithmically obtained partition is meaningful to human experts? To address this question, we submitted the algorithmic partition of [4] for review to one of the UMLS contractors. His judgment was—taking into account the constraints expressed by Chen *et al.* [4]—that the partition looks acceptable from a semantic perspective [12].

However, there is a difference between what experts accept as a semantically sound partition and what they will do if assigned the task of partitioning the SN according to their own semantic considerations. To address this issue, the following study was performed. The participants in this study were the five authors of this paper, two additional Ph.D. students from our research group, and one additional professor, all of whom have some background knowledge of the SN. They performed the task as part of a weekly research seminar. The participants did not have a time limit and submitted their work when they were satisfied with it. Each participant received a page of instructions (see Appendix) and two pages with diagrams of the IS-A hierarchy of the SN, i.e., the two trees rooted at **Event** and **Entity**. Fig. 1 shows the Event tree in a format similar to the one shown to the participants. However, the number lists attached to the nodes were not given to the participants. These numbers are experimental results.

The instructions are a simplified version of a human-machine methodology we have used previously to partition an OODB

James Geller, Yehoshua Perl, and Zong Chen are at the CS Department, New Jersey Institute of Technology, Newark, NJ 07102. Michael Halper is at the Department of Mathematics & Computer Science, Kean University, Union, NJ 07083. Huanying Gu is at the Department of Health Informatics, UMDNJ, Newark, NJ 07103. Corresponding Author: James Geller: geller@homer.njit.edu

schema of the MED terminology [7]. The opportunity for the simplification arises out of the fact that the MED schema is a DAG while the SN is a tree. At the same time, they take into consideration the cohesive partitioning rules [4], e.g., the need for singly rooted collections and the prohibition against singleton leaves.

Note that although the instructions seem quite elaborate, they only define structural limitations, such as "no single nodes allowed" or "groups must be connected." These limitations are necessary to make the computation of a valid comparison score between the partition of the subjects and the algorithmically obtained partition possible. On the other hand, our instructions do not limit the semantic decisions of the subjects, who still have the complete freedom to assign semantic types to groups of their choice.

Participants were also given verbal instructions about how to solve an example problem of a subnetwork rooted at **Entity** (see Fig. 2) approximately as follows:

"The figure is scanned top-down by a domain expert to identify semantic-type collections. Since **Entity** is the root of the tree, it should be the root of a semantic-type collection. Scanning down from **Entity**, the semantic type **Physical Object** still belongs to the subject area of **Entity** since both are very general terms. However, the next semantic type down, **Organism**, is significant and starts a new subject area of living creatures which is different from the **Entity** subject area. **Organism** has seven children, each of which is a specific kind of organism different from the general subject area of **Organism**. However, five of them are leaves and therefore do not qualify to begin separate semantic-type collections. Only two non-leaf children of **Organism**, **Plant** and **Animal**, start new semantic-type collections. Scanning further down, the only non-leaf descendants of **Animal** are **Vertebrate** and **Mammal** which are judged to be in the semantic-type collection rooted at **Animal**. Fig. 3, containing four semantic-type collections (each enclosed in a dashed bubble), shows the resulting partition."

Most importantly, the participants did not use any knowledge of the non-IS-A relationships that were used by the structural partitioning method. Therefore, the participants relied exclusively on their understanding of the semantic types, based on the names and positions of nodes in the SN IS-A hierarchy. Except for Z.C. and Y.P. the participants also did not study the details of the cohesive partition. (Before performing the study, Z.C., under the supervision of Y.P., applied the partitioning technique by hand and reviewed the resulting cohesive partition. Y.P. reviewed parts of the cohesive partition).

Evaluating the results showed that the partitions of different participants were quite different. Fig. 1, the Event tree, shows for every node which participants marked it as a root. (Participants are numbered from 1 to 8. The number 0 denotes the cohesive partition [4].) For instance, **Activity** is labeled (3, 6, 7, 8), meaning that it was marked by subjects 3, 6, 7 and 8 as a root. Since 0 is not listed, **Activity** was not chosen as a root by the cohesive partitioning algorithm. The figure shows how subjective semantic decisions are.

Table I demonstrates the high variability of subject responses. The table shows inter-subject agreement. The number in row i and column j indicates how many roots subject i and subject j agree on. For instance, subjects 3 and 5 agree on 21 roots. Table II shows agreement between the cohesive partition and the subjects' partitions. The average inter-subject agreement is 21.22. The average agreement of the subjects with the cohesive partitioning is 21.125.

Although individual participants' responses varied greatly, when accumulating all responses, some choices were made by a majority of subjects. Our approach is to identify a concept as the root of a semantic-type collection if at least T participating subjects chose this concept as a root. We subsequently computed recall (R) and precision (P) of the human subjects relative to the results of [4]. We will refer to T as the cut-off value. We then varied T as an independent variable and computed R and P over all concepts of the hierarchy as dependent variables. We also computed Rijsbergen's F value which combines precision and recall into one number as

$$F = 2 * P * R / (P + R).$$

In Table III, the columns are: Cut-off value T; number of roots marked by at least T subjects; number of roots marked by at least T subjects that were also identified by the cohesive partitioning; recall; precision; and F value.

The F value peaks at a cut-off of 6. However, the F values are almost identical for the cut-off values 3, 4, 5, and 6. The F value of about 0.8 indicates similarity between the cohesive partition and a *consensus partition* derived from nodes which were marked by at least T subjects (T = 3,4,5,6).

There is, of course, a trade-off between recall and precision. For example, with an impressive precision of 0.909, at least 6 subjects marked 20 out of the 28 roots of the cohesive partition, corresponding to a recall of 0.714. The recall increases to 0.892 when at least 3 subjects marked 25 out of 28 roots of the cohesive partition, but the precision decreases to 0.714. A middle point between 3 and 6, balancing recall and precision, is

TABLE I
INTER-SUBJECT AGREEMENT MATRIX

	1	2	3	4	5	6	7	8
1		23	19	21	23	20	16	22
2			26	24	22	25	23	18
3				24	21	23	26	25
4					19	24	19	20
5						20	15	19
6							22	22
7								18

TABLE II
ALGORITHM-SUBJECT AGREEMENT

Human	1	2	3	4	5	6	7	8
Algorithm	24	23	19	21	24	20	16	22

TABLE III
RESULTS OF EVALUATION

Cut-Off	Marked	Marked & Struc.	R	P	F val.
8	12	10	0.357	0.833	0.499
7	14	12	0.428	0.857	0.570
6	22	20	0.714	0.909	0.799
5	30	23	0.821	0.766	0.792
4	31	23	0.821	0.741	0.778
3	35	25	0.892	0.714	0.793
2	39	25	0.892	0.641	0.745
1	41	26	0.928	0.634	0.753

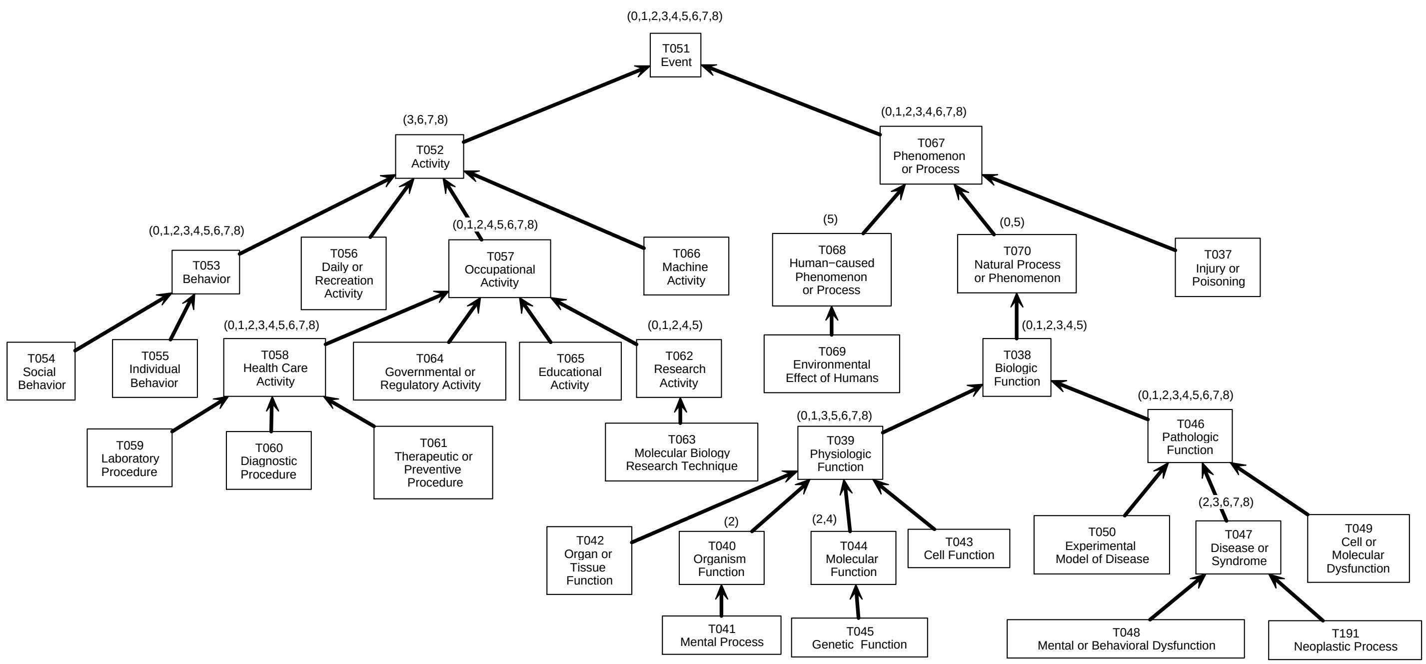

Fig. 1. Event Tree Hierarchy with Results

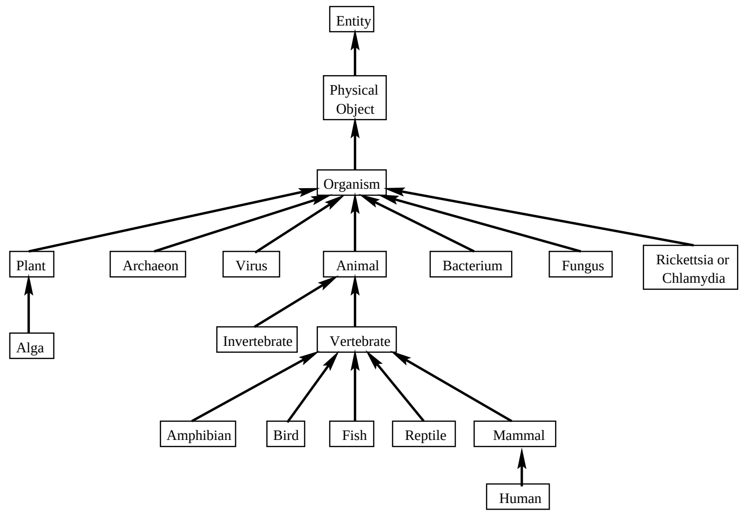

Fig. 2. A Portion of Entity Hierarchy

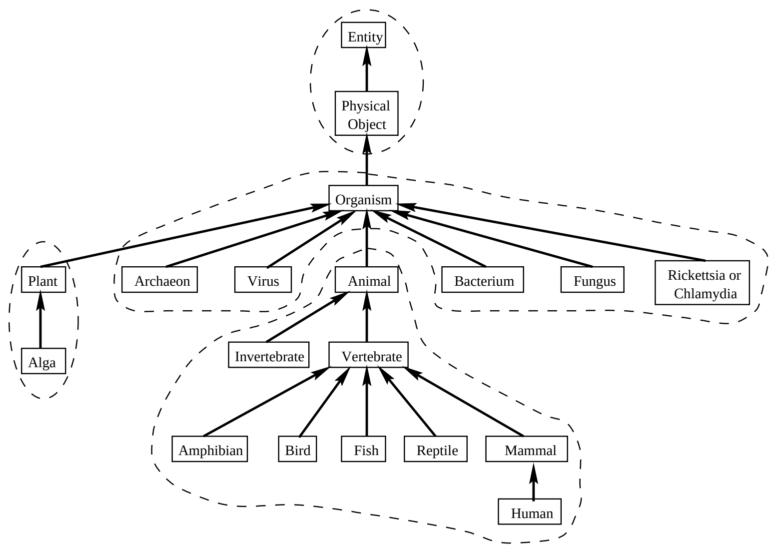

Fig. 3. Semantic Partition of a Portion of Entity Hierarchy

obtained for the cut-off value of 5. At least 5 subjects marked 23 of the 28 roots of the cohesive partition, for a precision of 0.766. Thus, our evaluation shows the usefulness of the cohesive partition and the high degree of agreement with the partition obtained by our subjects. This supports the claim that the cohesive partition is an effective semantic partition of the SN.

III. Using the Partition for Orientation to the SN

Above, we compared the need for modularity in SW development with the need for partitioning in semantic networks. The need for modularity of program code arose primarily out of difficulties in maintaining a growing base of installed software. Similarly, it should be easier to maintain semantic networks if they have been partitioned into groups of nodes.

In this section, we will show how a meaningful partition of a semantic network can be used to generate partial views of it, which are easier to understand than the whole network. We stress again that partial views are in no way specific to the UMLS Semantic Network. However, as all examples in [4] were given based on the SN, we continue with it.

The professionals who maintain the META of the UMLS, 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 to META. 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 an orientation. However, SN itself is too large and complex to be laid out on a computer screen. The SN's partition, which provides compact partial views of SN, can help us in this regard.

Our purpose is to provide various views which enable the user to study each element of SN, i.e., each semantic type and each relationship (IS-A relationship or semantic relationship) within a network small enough to be conveniently displayed on a computer screen. Those views show semantic types and their relationships within the vicinity of the relevant neighboring semantic types. Proper use of such views will enable a user to gain comprehension of the SN or parts of the SN of interest.

We will first list the various kinds of views, followed by examples. Later, we will describe a scenario of a user employing a sequence of such views to achieve a satisfactory degree of ori-

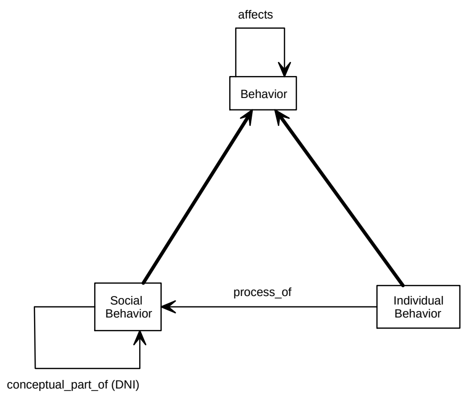

Fig. 4. *Behavior*-Collection Subnetwork

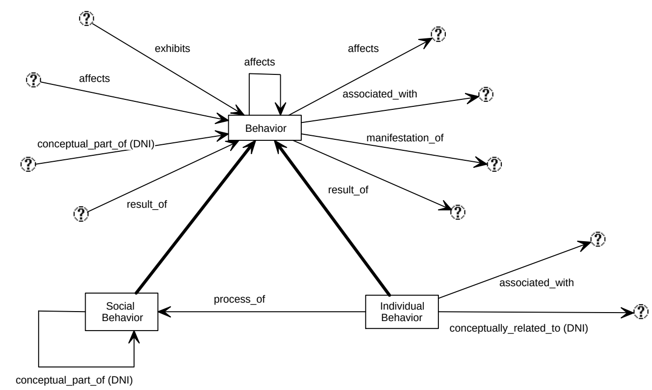

Fig. 5. *Behavior*-Collection Environment

entation. We will need the following definitions.

Definition (Induced Subnetwork): Let $G = (V, E)$ be a network where V is the set of nodes and E is the set of edges. Let $V' \subset V$ be a subset of nodes. The induced subnetwork of V' is $G' = (V', E')$ where $E' = E \cap (V' \times V')$, that is, E' contains edges of E where both nodes of such an edge are in V'. ■

Definition (C-Collection Subnetwork): Let C be a semantic-type collection of the cohesive partition of the SN. The C-collection subnetwork is the induced subnetwork of C. ■

The collection subnetwork contains the edges which are internal to the collection. Fig. 4 shows the *Behavior*-collection subnetwork. It contains three semantic types, two IS-A relationships, and three semantic relationships.

The collection subnetwork shows the internal connections of 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 those semantic types. For considering the external relationships of the collection, we need the following.

Definition (C-Collection Environment): Let C be a semantic-type collection of the cohesive partition of the SN. The C-collection environment is a network containing the C-collection subnetwork and all the (external) relationships of the SN for which only one semantic type is in C. ■

The other semantic type of each such relationship is not included in the environment. Every semantic type that is outside of the environment is labeled with "?." Fig. 5 shows the *Behavior*-collection environment containing three semantic types of the collection and three internal relationships. There are 22 external relationships incident on the *Behavior*-collection subnetwork's nodes, thirteen of which (five kinds) are exiting the collection, and nine of which (four kinds) are entering the collection. These 22 relationships belong to seven kinds of relationships (two kinds of relationships are both entering and exiting) which are displayed in Fig. 5. We do not display all the 22 occurrences of the relationships since the semantic types at the other ends of the relationships, which can distinguish the occurrences of the relationships, are outside the figure. A figure containing 22 external relationships along with their "other end" semantic types will be too large to appear readably on a screen and will not be helpful for orientation. Note that the relationship *issue_in*, inherited by **Behavior** and its children from **Event**, is not shown. Furthermore, in order to prevent clutter, the exiting relationships of **Behavior** inherited by both children are not displayed in Fig. 5. They can be deduced, using the rules of inheritance. The one exception is the relationship *associated_with*, exiting **Individual Behavior**, since its target is refined.

We now need the following two definitions.

Definition (Adjacent Collection): Two semantic-type collections C and D (of the cohesive partition of the SN) are adjacent if there exists a semantic type c in C and a semantic type d in D such that there is a relationship (either IS-A or semantic) connecting c to d or vice versa in the SN. ■

Definition ((C, D)-Adjacency Subnetwork): Let C and D be two adjacent semantic-type collections. The (C, D)-adjacency subnetwork contains the C-collection subnetwork, the D-collection subnetwork, and all relationships of the SN with one semantic type in C and one semantic type in D. ■

Fig. 6 shows the (*Behavior*, *Pathologic Function*)-adjacency subnetwork. It shows, in addition to the two collection subnetworks, the interaction between their semantic types. There is one edge from **Behavior** to **Mental or Behavior Dysfunction** and one edge in the opposite direction. There is also one edge from **Pathologic Function** to **Behavior** and one edge from **Individual Behavior** to **Pathologic Function**.

We will now present a scenario showing how a user can employ a sequence of various such views to achieve orientation to the SN. Note that each of these views can conveniently be displayed on a computer screen.

There are 28 semantic-type collections in the cohesive partition of the SN. Each is named after its unique root semantic type. Thus, a user can identify (according to search interest) a desired semantic-type collection which is called the *focus semantic-type collection*. As an example, let the focus semantic-type collection be the *Behavior* collection. Next, the user can view the semantic-type collection subnetwork of the focus semantic-type collection. Fig. 4 gives an example of the *Behavior*-collection subnetwork where a user sees its three semantic types and the relationships connecting them. However, the user cannot get a full understanding of these semantic types without reviewing all their external relationships. The focus semantic-type collection environment captures all the external relationships of the semantic types of the focus semantic-type collection. Fig. 5 shows the *Behavior*-collection environment. There are many relationships defined for the environment which are not shown in Fig. 5. If all the relationships of this environment were included in the figure, it would not fit on a screen anymore. Furthermore, the collection environment view does not include the semantic types on the other side of the external relationships, which are needed for full study of the semantic

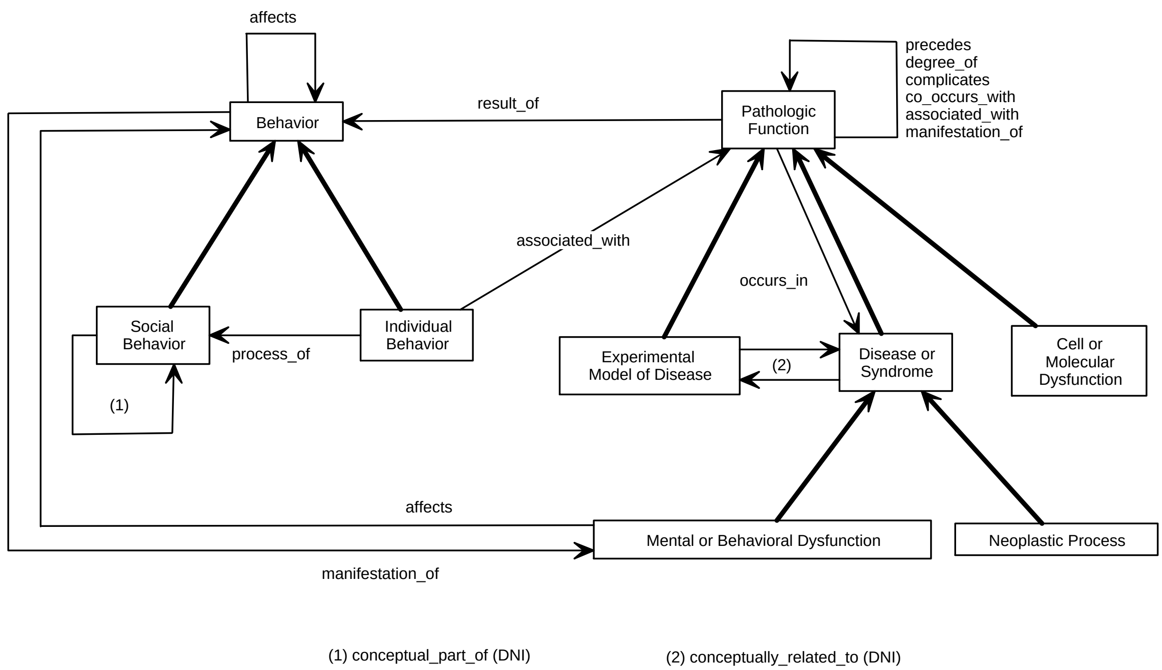

Fig. 6. (*Behavior*, *Pathologic Function*)-Adjacency Subnetwork

types' structure. (Note that the user will not be able to use the C-collection environment view as it is typically too large. It is described here only to demonstrate the need to view all these external relationships.)

To overcome this information overload, we will divide the task of reviewing all the external relationships of the focus collection into a sequence of small tasks using views which fit on a computer screen.

The external relationships of the focus semantic-type collection are divided into disjoint sets according to the collections containing the semantic types on the other side of the relationships. For example, the 22 external relationships of the *Behavior* collection are divided into 10 different sets, according to the semantic-type collections containing the other ends of the relationships. For instance, consider the external edges between the *Behavior* collection and the *Pathologic Function* collection. To review such a limited set of external relationships, the user utilizes the (C, D)-adjacency subnetwork for the pair of C and D collections. For example, Fig. 6 shows the (*Behavior*, *Pathologic Function*)-adjacency subnetwork. It shows four of the 22 external relationships of the *Behavior* collection and can be conveniently displayed on a screen. One can review all the external relationships of the *Behavior* collection by reviewing 10 such adjacency subnetworks, each of which is small enough for display and study. Of course, if one were interested in only some of these external relationships, fewer such adjacency subnetwork views would be necessary.

It is clearly much easier to get an understanding of each of the semantic types of a collection and the interactions among them separately from figures such as Fig. 4 and Fig. 6, than to get such knowledge from a complete diagram of the Semantic Network. In the network, these aspects are hidden in the overall structure of a large diagram which does not even show the inherited relationships. 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 small enough to be displayed on a computer screen and is easier for a user 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.

IV. Conclusions

In this paper, we have addressed a common criticism of semantic network partitioning methods, namely that their results may not agree with the intuition of human experts about how to partition a semantic network into meaningful groups of concepts. We evaluated the partitioning method of [4] by comparing its results with the results of experts who partitioned the SN into meaningful logical units, based on their understanding of the domain. We found, on average, good agreement, with a peak F value of 0.799.

We then constructed several views of the UMLS Semantic Network, based on the cohesive partition [4], which make it easy to perform an organized study of the SN or of well chosen parts of it. These views are perfectly general. We have demonstrated them with the SN of the UMLS because it is widely known. As the SN itself is an abstraction of the UMLS Metathesaurus, our views will make it easier to study, understand, and use the Metathesaurus.

Acknowledgment:
Li Zhang and Yi Peng have helped with the analysis of the study presented in this paper. Dr. James Cimino has commented on an earlier draft of this paper.

References

[1] M. R. Quillian, "Semantic memory," in *Semantic Information Processing*, M. L. Minsky, Ed., pp. 227–270. The MIT Press, Cambridge, MA, 1968.

[2] F. Lehmann, Ed., *Semantic Networks in Artificial Intelligence*, Pergamon Press, Tarrytown, NY, 1992.

[3] W. A. Woods, "What's in a link: Foundations for semantic networks," in *Representation and Understanding*, D. G. Bobrow and A. M. Collins, Eds., pp. 35–82. Academic Press, New York, NY, 1975.

[4] Z. Chen, Y. Perl, M. Halper, and J. Geller, "Partitioning the UMLS Semantic Network," *IEEE Transactions on Information Technology in Biomedicine*, in this issue.

[5] L. Liu, M. Halper, J. Geller, and Y. Perl, "Controlled vocabularies in OODBs: Modeling issues and implementation," *Distributed and Parallel Databases*, vol. 7, no. 1, pp. 37–65, Jan. 1999.

[6] H. Gu, Y. Perl, J. Geller, M. Halper, L. Liu, and J. J. Cimino, "Representing the UMLS as an OODB: Modeling issues and advantages," *JAMIA*, vol. 7, no. 1, pp. 66–80, Jan./Feb. 2000.

[7] H. Gu, Y. Perl, M. Halper, J. Geller, F. Kuo, and J. J. Cimino, "Partitioning an object-oriented terminology schema," *Methods of Information in Medicine*, vol. 40, no. 3, pp. 204–212, 2001.

[8] H. Gu, Y. Perl, J. Geller, M. Halper, and M. Singh, "A methodology for partitioning a vocabulary hierarchy into trees," *Artificial Intelligence in Medicine*, vol. 15, no. 1, pp. 77–98, 1999.

[9] A. T. McCray, A. Burgun, and O. Bodenreider, "Aggregating UMLS semantic types for reducing conceptual complexity," To appear in: *Proceedings of Medinfo 2001*.

[10] A. T. McCray and W. T. Hole, "The scope and structure of the first version of the UMLS semantic network," in *Proceedings of the Fourteenth Annual SCAMC*, 1990, pp. 126–130.

[11] B. L. Humphreys, D. A. B. Lindberg, H. M. Schoolman, and G. O. Barnett, "The Unified Medical Language System: An informatics research collaboration," *JAMIA*, vol. 5, no. 1, pp. 1–11, 1998.

[12] J. J. Cimino, "Personal communication," 2000.

Appendix

Instructions to experimental subjects.

You are given two sheets. Each sheet contains a tree diagram. Each tree diagram is a part of a semantic network called the UMLS. Each node in a tree stands for a medical class. Each arrow connects a specific class to a more general class. A leaf node is a node without children.

The purpose of this experiment is to partition each tree of classes into subtrees, such that the classes of each subtree form a logical group describing one subject matter. A single leaf node does not constitute a group. Your task is as follows.

1) Start at the root node of the tree.

2) Scan through the tree downwards.

3) Whenever you judge that a non-leaf class is important AND is quite different from its parent class, mark it with a star. This class will be called a "new root."

– A marked class starts a group below it.

– The name of a marked class is also the name of the group below it.

– Each unmarked class belongs to the group of its closest root class above.

James Geller received an Electrical Engineering Diploma from the Technical University Vienna, Austria (1979), an M.S. degree (1984) and a Ph.D. degree (1988) in Computer Science from SUNY at Buffalo. James Geller is professor in the Computer Science Department at NJIT where he is also Director of the OODB & AI Laboratory and Vice Chair of the M.S. and Ph.D. programs in Biomedical Informatics. Dr. Geller has published numerous journal and conference papers in knowledge representation, parallel artificial intelligence, medical informatics, and object-oriented databases. His current research interests concentrate on object-oriented modeling of medical vocabularies, and on Web mining.

Yehoshua Perl received his Ph.D. degree in CS in 1975 from the Weizmann Institute of Science, Israel. Since 1985, he has been in the Computer and Information Science Department at NJIT, where he was appointed professor in 1987. Dr. Perl received the Harlan Perlis Research Award of NJIT in 1996. Dr. Perl is the author of more than 80 papers in international journals and conferences. His publications are in object-oriented databases, design and analysis of algorithms, data structures, design of networks, sorting networks, graph theory, and data compression. His current research deals with modeling medical vocabularies using object-oriented databases.

Michael Halper received the CS B.S. degree (with honors) from NJIT (1985), the CS M.S. degree from Fairleigh Dickinson University (1987) and the CS Ph.D. degree from NJIT (1993). He was the recipient of a Garden State Graduate Fellowship from the State of New Jersey. Dr. Halper is an associate professor of computer science at Kean University. His research interests include conceptual and object-oriented modeling, part-whole modeling, extensible data models, object-oriented database systems, and medical informatics. Dr. Halper has published numerous papers in international journals and conferences. He is a member of The Honor Society of Phi Kappa Phi.

Zong Chen received his B.S. degree in Electrical Engineering at the East China Institute of Technology, Nanjing, China (1992), and his M.S. degree in Electrical Engineering at the Southeast University, Nanjing, China (1997). He is currently working on his Ph.D. degree in Computer Science at NJIT. He has published several papers in journals and conferences on topics in Medical Vocabularies.

Huanying Gu received her B.S. degree in Computer Science from Huazhong University of Science and Technology in 1984 and her M.S. degree in Computer Science from Nanjing University of Science and Technology in 1989, both in China. She received a Ph.D in Computer Science from New Jersey Institute of Technology in 1999. Huanying Gu has been with Health Informatics Department in University of Medicine and Dentistry of New Jersey as an assistant professor since 1999. Her research interests include object-oriented modeling, object-oriented databases, knowledge representation, controlled medical terminologies, and biomedical informatics.