Journal of Biomedical Informatics 42 (2009) 468–489

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

The Neighborhood Auditing Tool: A hybrid interface for auditing the UMLS ☆

C. Paul Morrey [a], James Geller [a,*], Michael Halper [b], Yehoshua Perl [a]

[a] New Jersey Institute of Technology, Newark, NJ 07102, USA
[b] Kean University, Union, NJ 07083, USA

ARTICLE INFO

Article history:
Received 18 June 2008
Available online 5 February 2009

Keywords:
Unified Medical Language System
Auditing of terminologies
Auditing of ontologies
Auditing of the UMLS
Software tool
Auditing tool
User interface
Hybrid diagram/text user interface

ABSTRACT

The UMLS's integration of more than 100 source vocabularies, not necessarily consistent with one another, causes some inconsistencies. The purpose of auditing the UMLS is to detect such inconsistencies and to suggest how to resolve them while observing the requirement of fully representing the content of each source in the UMLS. A software tool, called the Neighborhood Auditing Tool (NAT), that facilitates UMLS auditing is presented. The NAT supports "neighborhood-based" auditing, where, at any given time, an auditor concentrates on a single-focus concept and one of a variety of neighborhoods of its closely related concepts. Typical diagrammatic displays of concept networks have a number of shortcomings, so the NAT utilizes a *hybrid diagram/text interface* that features stylized neighborhood views which retain some of the best features of both the diagrammatic layouts and text windows while avoiding the shortcomings. The NAT allows an auditor to display knowledge from both the Metathesaurus (concept) level and the Semantic Network (semantic type) level. Various additional features of the NAT that support the auditing process are described. The usefulness of the NAT is demonstrated through a group of case studies. Its impact is tested with a study involving a select group of auditors.

1. Introduction

The Unified Medical Language System (UMLS) [4,5,18,19] is an important resource for the medical informatics community. Currently, the UMLS is derived from over 100 source vocabularies, with its Metathesaurus [40,42] housing more than 1.5 million concepts [43]. The Semantic Network (SN) [22–26] provides an abstraction layer consisting of high-level, broad categories called semantic types. One or more semantic types are assigned to each of the Metathesaurus's concepts. This dual knowledge arrangement supports the ongoing integration [20] and auditing activities [9,11,13–16,33].

Auditing is an important part of the terminology design life-cycle [28]. The purpose of auditing a terminology is to detect errors in its stored knowledge. Such knowledge may be factually wrong, e.g., a wrong target concept for a lateral relationship or a misleading definition. Other kinds of errors express contradictory situations, e.g., a cycle of hierarchical relationships or a concept appearing as a child of another concept when according to its definition it should be a parent or a sibling. Furthermore, two concepts may be duplicates or one concept may be ambiguous. Typically, an audit report will identify potential errors and suggest possible ways to resolve them.

The situation with the UMLS is different. The UMLS was not designed as a terminology but as a terminological system intended to integrate many source vocabularies to enable electronic access to their information and interoperation among them. Hence, inconsistencies are expected in the UMLS, and many cannot be resolved due to the commitment to represent each source in its entirety, independent of the content of other sources. Nevertheless, as we will discuss, some kinds of inconsistencies in the UMLS are the responsibility of the UMLS editors and can be resolved. Such inconsistencies should be the focus when auditing the UMLS.

In this paper, we present a novel software tool that facilitates the process of auditing the UMLS. Our design proceeded from the supposition that most inconsistencies are discovered locally, especially in the context of closely related concepts such as parents and children. As such, we based the tool on the *neighborhoods* of a concept that the auditor is focusing on. The tool is called the *Neighborhood Auditing Tool* (*NAT*), and, in fact, presents an auditor with a variety of neighborhood view options at both the concept level and the semantic-type level.

An important aspect of the NAT is the manner in which its neighborhoods are displayed. In the world of terminological and semantic networks, "box and line" diagrams are commonly employed for content presentation [36]. The Semantic Navigator[1] is

☆ This work was partially supported by the NLM under Grant R-01-LM008445-01A2.

* Corresponding author. Fax: +1 9735965777.
E-mail address: geller@njit.edu (J. Geller).

[1] http://mor.nlm.nih.gov/perl/semnav.pl. This tool requires a UMLSKS Password.

1532-0464/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2009.01.006

an example of such an interface for the UMLS. However, their usefulness can break down when a network is very large, as is the case with the UMLS. For example, a proliferation of intersecting relationship-lines can cause major confusion. Moreover, a diagrammatic display of concepts, in general, does not afford good neighborhood views. Adding the UMLS's Semantic Network on top of the concepts makes such visualization all the more difficult. As an auditor navigates from concept to concept within the diagram, it becomes necessary to "visually" construct these neighborhood views over and over again. When presented properly, though, a graphical display can be very powerful, with the interconnections between concepts readily apparent.

With this in mind, the NAT offers a stylized view of a neighborhood, uniform in its layout from concept to concept. It has a *hybrid diagram/text interface* that retains some of the best features of a diagrammatic display while eliminating the clutter of intersecting lines. The NAT's hybrid diagram/text interface uniformly gathers all neighborhoods' similarly related concepts into a single text box. That is, all the children of the focus concept are in a children text box, while the parents appear in a parent text box. This avoids their being scattered about. The auditor is thus alleviated of the burden of discerning concept parentage as he navigates through the Metathesaurus by selecting a new focus concept. The natural diagrammatic rendering of children below parents is preserved with the placement of the children text box below the focus-concept text box, and the latter below the parents text box.

The NAT was designed as a general-purpose auditing tool for the UMLS, with an emphasis on serving the needs of editors and auditors. Two goals of the NAT are the availability of and ease of access to relevant knowledge about concepts without overloading the user. The notion of neighborhoods is used for both structuring the presentation of basic knowledge in an intuitive way and for displaying extended knowledge upon request. Furthermore, easy navigation enables the auditor to further explore the reasons for an inconsistency for any aspect of a concept's properties, and the boundaries of its propagation to other inconsistencies. We hope that these characteristics of the NAT will make it suitable for auditing processes in the UMLS.

The functionality of the NAT in the auditing process is demonstrated through a group of case studies. These examples show that the NAT can help an auditor in finding inconsistencies. To test the efficacy of the NAT, a study measuring its impact on the work of a select group of auditors is carried out. The results of this study are presented.

2. Background

2.1. Structure of the UMLS

The major building blocks of the Metathesaurus (META) are concepts and relationships. The META consists of many different source vocabularies. If available in the source vocabularies, definitions for a concept are provided. The UMLS also identifies the source vocabularies from which each concept is taken. Relationships are divided into hierarchical and lateral. A hierarchical relationship supports the organization of the concepts in a network where family references, such as parent, ancestor, child, and sibling, are defined. Structurally, for most source terminologies, the hierarchy is a Directed Acyclic Graph (DAG), which means that it is impossible to return to the starting point when following the parent arrows in the network. A known exception is the MeSH [27], where the hierarchy deliberately allows for cycles. Furthermore, there exist hierarchical relationship cycles in the META [3,29], due to the cumulative impact of inconsistent hierarchical relationships in various sources. Following the UMLS, we will refer to the hierarchical relationship as *parent-of* (*child-of*).

With regards to lateral relationships, the situation in the META varies. Some are named as in their sources, such as SNOMED [41], NCI Thesaurus [31], and FMA [38]. The lateral relationships that are unnamed in the sources are given general designations. These include "broader" (RB) or "narrower" (RN). Other lateral relationships are marked with a "catch all" relationship designated "other" (or RO, for short).

The UMLS is unique among medical terminological systems in that it has an interconnected two-level structure. The upper level, comprising the Semantic Network (SN) [22–26], consists of 135 semantic types, which are broad biomedical categories. Semantic types are hierarchically connected via IS-A links and part of links. The 135 semantic types together with the IS-A links are organized as two trees. There are 53 kinds of lateral relationships defined for the SN, which may hold between pairs of semantic types. Thus, the SN is structurally similar to the META. When integrating a new source into the META, the editors assign each new concept one or more semantic types [20] to support the integration process. As discussed in Section 2.2, the SN also plays an important role in support of auditing techniques for the UMLS.

2.2. Auditing the UMLS

Auditing large medical terminologies is, in general, a major challenge. Their size and complexity make it unavoidable that errors will occur. We observed that previous methodologies for developing medical terminologies appear incomplete. Auditing by independent teams of experts is not usually considered a part of the terminology life cycle. In [28], we have argued in detail why auditing should be deemed a major activity. This follows the common practice of quality assurance, for example in software engineering [30]. Recent publications show an increase in attention given to auditing of medical terminologies.

Campbell et al. [5] observed it as an advantage that the UMLS avoids imposing any restrictions on the content, structure, and semantics of the source vocabularies. However, with this flexibility comes an increased danger of introducing inconsistencies into the META. In fact, the policy of retaining all knowledge given in each source, even though the knowledge of one source may contradict that in another, effectively guarantees some inconsistencies. Examples of possible inconsistencies [5] include omissions, non-uniform classifications, misclassifications, ambiguities, redundant classifications, and synonyms listed independently as separate concepts. Some of these inconsistencies are bound to be visible to UMLS end users. Since the UMLS provides terminological support, e.g., synonyms, for users of systems such as clinical patient records, health care administrative systems, decision-support systems, etc. [8,12], its inconsistencies may cause problems for users of such systems.

While some inconsistencies in the UMLS cannot be resolved, some knowledge in the UMLS is under the direct control of its editors. The identification of a term from a source with a UMLS concept is done by the UMLS editors in the integration process. The semantic types assigned to concepts during the integration process are artifacts of the UMLS and not from the source vocabularies. The designation of relationships as "parent", "broader", or any other kind is decided by the UMLS editors according to some given rules. Some concepts or their terms and some of their relationships are added by the editors. Those have the source designation MTH (short for Metathesaurus).

When auditing the UMLS, an auditor can concentrate on inconsistencies arising with the knowledge elements under the control of the UMLS editors and thus modifiable by them. An example is

a concept assigned two semantic types that are mutually exclusive, or having one of its semantic types with an IS-A relationship to the other. The latter is a case of a redundant semantic type assignment [33], forbidden by McCray and Nelson [26]. A UMLS auditing report should deal with inconsistencies regarding knowledge elements marked with MTH or ones that can be resolved by adding, deleting, or modifying MTH knowledge elements, as well as other knowledge elements controlled by the UMLS editors.

A UMLS audit report may also involve source-sensitive auditing, where inconsistencies in the UMLS can be resolved by correcting errors in the sources. While UMLS editors cannot modify a source terminology, it is possible for an external auditor to communicate a recommended change to the organization in charge of a source, e.g., IHTSDO for SNOMED or NCI for the NCI Thesaurus. If such a correction is made in a source, it will propagate to the next release of the UMLS. Examples of such cases will be shown in the Section 4.

In our own previous auditing work, e.g., [14,15], auditors were given a textual representation of all the knowledge perceived necessary. Here is an example of such an audit form:

```
CPT:  C0836205 Gut Epithelium
SRC:  CSP, NCI
STY:  T023:T024 Body Part, Organ, or Organ
      Component + Tissue
DEF:  [CSP] one or more layers of epithelial cells,
      supported by the basal lamina, which covers the
      gastrointestinal system. | [NCI] The epithelium
      that lines the intestinal tract.
SYN:  gastrointestinal epithelium | Gut Epithelium
PAR:  gastrointestinal systemSTY: Body System |
      EpitheliumSTY: Tissue
CHD:  Esophageal Glandular CellSTY: Cell | Small
      Intestinal Goblet
      CellSTY: Cell | Parietal Cells, GastricSTY: Cell |
      Chief Cells,
      GastricSTY: Cell | EnterocytesSTY: Cell |
      Intestinal
      MucosaSTY: Tissue | Paneth CellsSTY: Cell |
      Esophageal
      Squamous CellSTY: Cell | Gastric Glandular
      CellSTY: Cell |
      Foveolar CellSTY: Cell
```

Working with this format made us realize its deficiencies and led us to the construction of the NAT. When using a graphical tool, as opposed to a strict text representation, an auditor can easily call up additional information on demand and navigate from one concept to other related concepts.

We have previously distributed a questionnaire about the use and future agenda of the UMLS [8]. It was clear from the results that there is a demand for high-quality auditing. Furthermore, the responding UMLS users saw auditing as a high priority since, on average, they would allocate 35% of a putative UMLS budget to auditing, the highest of all given options by a large margin. The three trailing categories, "designing a derived terminology", "improving interfaces", and "extending coverage", were assigned only 24%, 20%, and 16% of the budget, respectively.

Researchers have developed many different methodologies for auditing the UMLS. Semantic methods have been used to detect classification inconsistencies [9]. Detection of the above mentioned hierarchical relationship cycles was dealt with in [3,29]. Techniques are also given for detecting reverse hierarchical relationships [11], concept redundancy and ambiguity [10], and redundant categorizations [33]. Proposed revisions to the SN have included reclassification of its semantic types [39] and enrichment with additional IS-A links [45,46]. Object-oriented models have been utilized in the service of auditing, e.g., [2,16]. The discovery of missed synonymy in the META has been addressed in [17]. The notion of "metaschema" of the SN [34] has been employed in the process of auditing [15].

2.3. The semantic locality paradigm

The NAT relies heavily on a set of different neighborhoods of the focus concept. A related idea, which appears in the literature, is *semantic locality*. In [32] there is an explanation for the purpose of defining semantic locality: To find how a meaning is named in the source of choice, a user must exploit one of these aspects of semantic locality, entering a term somehow related to the term being sought, and navigating to the preferred term. More succinctly, [42] "...navigation is assisted by semantic locality." The intended use of semantic locality influences the choice of which terminology aspects are included in it. Nelson et al. [32] write that the aspects of semantic locality in the Metathesaurus which can be thus exploited are the terms, the semantic types, the use of that term in a source context, and the co-occurrence of terms in MEDLINE.

3. Methods: the design of a Neighborhood Auditing Tool

3.1. The neighborhood paradigm

When auditing a biomedical concept, it is frequently not sufficient to look at the concept itself to determine whether it is correct or not. Rather, the close neighborhood of the concept has to be investigated, too. Thus, we will define a family of neighborhoods and show how they are useful for auditing.

Our general auditing approach is that an auditor is given a concept that is considered suspicious. To assess potential inconsistencies, the auditor needs to view details of the concept. The auditor would start with a "small" environment, such as the close neighborhood of a concept, to avoid mental overload. If this neighborhood does not provide enough information, he can transition to a larger neighborhood.

We introduce and motivate four different kinds of neighborhoods that we have found useful for auditing. They are called *immediate neighborhood*, *extended neighborhood*, *up-extended neighborhood* and *down-extended neighborhood*. For each, there exists a corresponding neighborhood at the SN level consisting of semantic types. The auditor starts with one suspicious concept as the focus concept, but in the process of auditing, might navigate to other concepts. The elements of the neighborhoods include the potential future focus concept in the navigation process. Hence, the neighborhoods support navigation as well.

3.1.1. Knowledge elements

The knowledge about a focus concept consists of two kinds of knowledge elements: textual knowledge elements and contextual knowledge elements. Textual knowledge elements of a concept include its name, its Concept Unique Identifier (CUI), its terms and their Lexical Unique Identifiers (LUIs), its definitions, source terminologies, and the semantic type(s) assigned to it. Note that there may be several definitions from different sources. Contextual knowledge elements are other concepts providing context for the focus concept. Adjacent concepts of the focus concept, e.g., its parents, its children, its siblings and those related to it through lateral relationships, define the major contextual knowledge elements of a concept.

While they are concepts in their own right, the contextual knowledge elements provide knowledge that plays an important

role in expressing the meaning of a concept. Many UMLS concepts are highly technical and specialized and even an MD might not know exactly what their definitions mean. For example, when no definition is provided for a concept—a common situation in the UMLS—the contextual knowledge elements sometimes can suggest one. This follows the approach of Aristotle [1] in basing the definition of a species on the genus and differentiae. The parent provides the genus knowledge, while the differences (differentiae) exist between the siblings. In fact, many definitions that are currently contained in the UMLS are constructed around the expression of an explicit or implicit IS-A relationship. For example, the NCIT definition of *Gut Epithelium*, described in Section 2.2, as "The epithelium that lines the intestinal tract" follows the Aristotelian definition pattern, since *Epithelium* is one of the parents of *Gut Epithelium*.

Knowledge elements, both textual and contextual ones, are essential for auditing a concept. Thus, including contextual knowledge elements in a neighborhood will facilitate this process.

3.1.2. Immediate neighborhood

Thinking of the META as a graph structure, the parents and children are important contextual knowledge elements and should definitely be part of the *immediate neighborhood* of the focus concept, which is defined as follows.

Definition (*Immediate neighborhood*): The immediate neighborhood of a focus concept contains the focus concept plus all the contextual knowledge elements that are connected to the focus concept by a single relationship, either hierarchical or lateral. That is, the immediate neighborhood of a concept contains all concepts at a distance of one, i.e., its parents, children, and concepts that are the targets of lateral relationships emanating from the focus concept.

Fig. 1 shows the immediate neighborhood of the concept *Microsporidia, Unclassified*. (We illustrate neighborhoods that will later be used in a case study.) There is a practical problem when looking at concept neighborhoods. In many cases, there are several parents and many children. Diagrams of these parents and children often show them scattered and intermingled with other unrelated concepts. In such cases, it becomes difficult for an auditor to visually construct the neighborhood for a given focus concept.

To solve this problem, it is necessary to keep the children close together, parents close together, but the parents apart from the children. This is expressed in Fig. 1 by surrounding all children with a box, all lateral targets with a box, and all (one) parents by another box. The significance of these boxes will be discussed in detail below.

3.1.3. Advanced neighborhoods

3.1.3.1. Extended neighborhood. Definition (*Extended neighborhood*): The extended neighborhood of a focus concept contains the focus concept, all contextual knowledge elements of the immediate neighborhood, and all contextual knowledge elements that are separated from the focus concept by a distance of two hierarchical relationships. That is, the extended neighborhood of a focus concept also contains its grandparents, grandchildren, and siblings, which are the other children of the concept's parents.

The example of the extended neighborhood of *Microsporidia, Unclassified* can be seen in Fig. 2.

Viewing the parents, children, and relationship targets of a focus concept may give a good understanding of this concept, but it might still not be enough to see the origin and effect of an inconsistency in the META. Thus, after studying such an immediate neighborhood, an auditor might want to see the extended neighborhood of the focus concept.

Looking initially at the extended neighborhood might be overwhelming for an auditor, but after having digested the immediate neighborhood, it will be easier to look one additional level up and one additional level down. The previously mentioned problem of constructing the environment visually is now even more complicated, as it involves five levels.

Referring again to Fig. 2, we see that this neighborhood contains too many children, relationship targets, and grandchildren to be displayed comfortably. This problem will be addressed later. In preparation, Fig. 2 shows all the concepts of the extended neighborhood in boxes, where all the children are in one box and all the grandchildren are in another box (and similarly for the other boxes). Note that the box to the left of the focus concept box contains the sibling(s) of the focus concept.

3.1.3.2. Up-extended neighborhood. Typically, a concept has more children than parents and many more grandchildren than grandparents. Thus, the extended neighborhood might display too many grandchildren, resulting in information overload, while the grand-

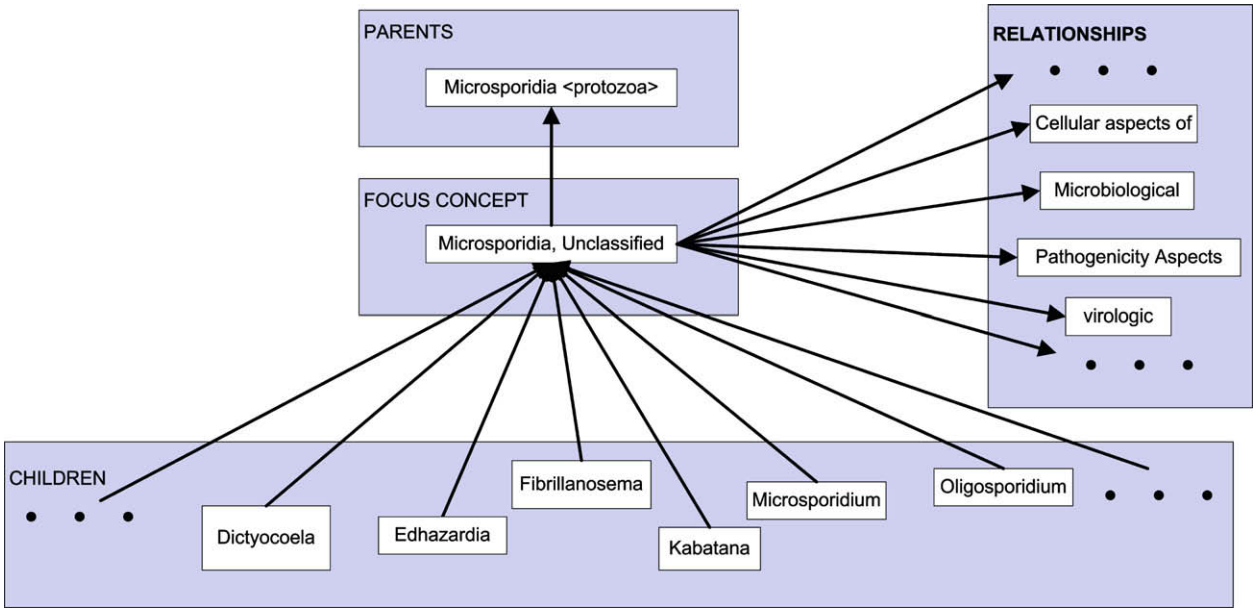

Fig. 1. *Microsporidia, Unclassified* immediate neighborhood, boxed display.

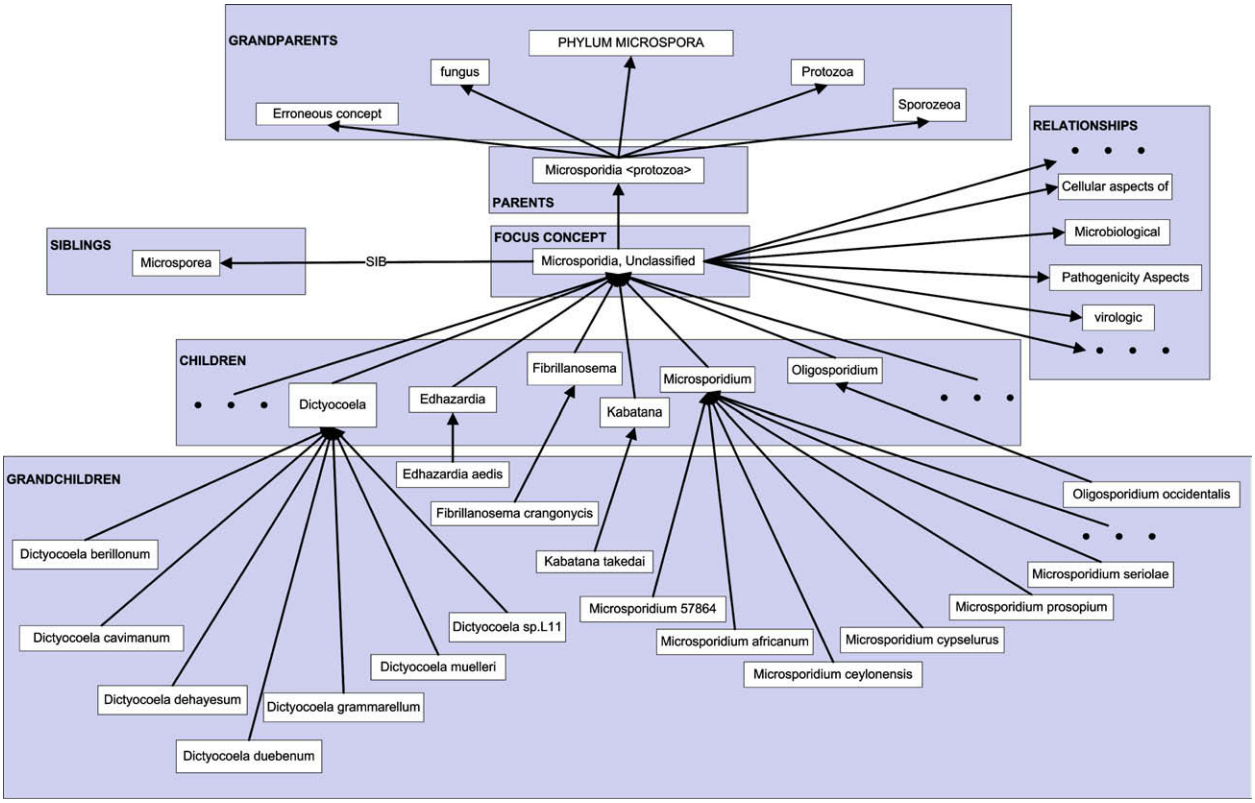

Fig. 2. *Microsporidia, Unclassified* extended neighborhood boxed display.

parents might be of interest. In such a case, an asymmetric neighborhood with grandparents but without grandchildren is called for. We call this kind of neighborhood the *up-extended neighborhood*.

For example, when reviewing *Microsporidia, Unclassified*, insight may be gained by looking at its grandparents. They include both *Fungus* and *Protozoa*, which is a surprising combination. On the other hand, the grandchildren do not provide any additional insights at this step, while their number is overwhelming.

3.1.3.3. Down-extended neighborhood. If auditing a concept uncovers an inconsistency, it is prudent to check its children and grandchildren to see whether the inconsistency has been propagated downwards. For this step, the display of grandparents is not helpful anymore. Thus, there is sometimes a need for a *down-extended neighborhood*, which is a mirror image of the up-extended neighborhood. It contains children and grandchildren, but no grandparents. Due to the potentially large number of grandchildren, there is still a danger of mental overload. Combating this problem will be discussed below in Section 3.2, where we will introduce the hybrid display paradigm.

3.1.4. Semantic-type neighborhoods

As mentioned above, each concept of the META is assigned one or more semantic types (STs). Those types define a broad category for the concept, specifying whether it is, say, a disease (ST **Disease or Syndrome**), a finding (ST **Finding**), or a kind of cancer (ST **Neoplastic Process**). Such high-level knowledge is very important in auditing, since it captures the perception of the UMLS editors about the nature of the specific concept. Inconsistencies many times stem from misconceptions about a concept. Thus, irregularity in the ST assignments of a concept may indicate an ST assignment inconsistency, which may lead the auditor to uncover more inconsistencies in other knowledge elements or other concepts.

One example of an ST irregularity occurs when a concept is assigned two semantic types that are incompatible. Another example is a pair of a concept and its parent, where the STs assigned to these two concepts are inappropriate for a parent–child configuration [7,11].

For each neighborhood of a concept, there is a corresponding neighborhood of STs. That is, a graph exists with the same structure as that of a concept neighborhood, but with nodes representing the STs of the concepts rather than the concepts themselves. Fig. 3, shows the immediate semantic-type neighborhood of *Microsporidia, Unclassified*. We can see some irregularities in this ST configuration. For example, the semantic type of the focus concept is **Invertebrate**, but a semantic type of some of its children, e.g., *Dictyocoela, Fibrillanosema, Myosporidium, Trichotuzetia*, is **Fungus**. However, these two STs, which are children of **Organism**, are exclusive and are not refinements of one another. Thus, they do not constitute a valid ST configuration for a parent-child concept configuration [7,11].

Also, the fact that some of the children have the semantic type **Fungus**, while others have **Invertebrate**, e.g., *Edhazardia, Kabatana, Oligosporidium, Visvesvaria*, is surprising and deserves a review by an auditor. Since the ST information is important for auditing, we looked for a way to integrate it into the NAT interface. Its use will be shown later in Section 4.1.

We note that the difference between the neighborhoods that we have defined and the semantic locality of [32,42] derives from their different intended uses. Semantic locality is term-oriented, as it supports navigation to find a term for a given meaning. Our neighborhoods are concept-oriented, providing access to different neighboring concepts, which can help expose inconsistencies for the focus concept.

3.2. Screen design for hybrid interface

In Fig. 2, there were too many children and grandchildren to generate a good diagrammatic display. The names of concepts written inside of boxes are typically long, which makes it impossible to

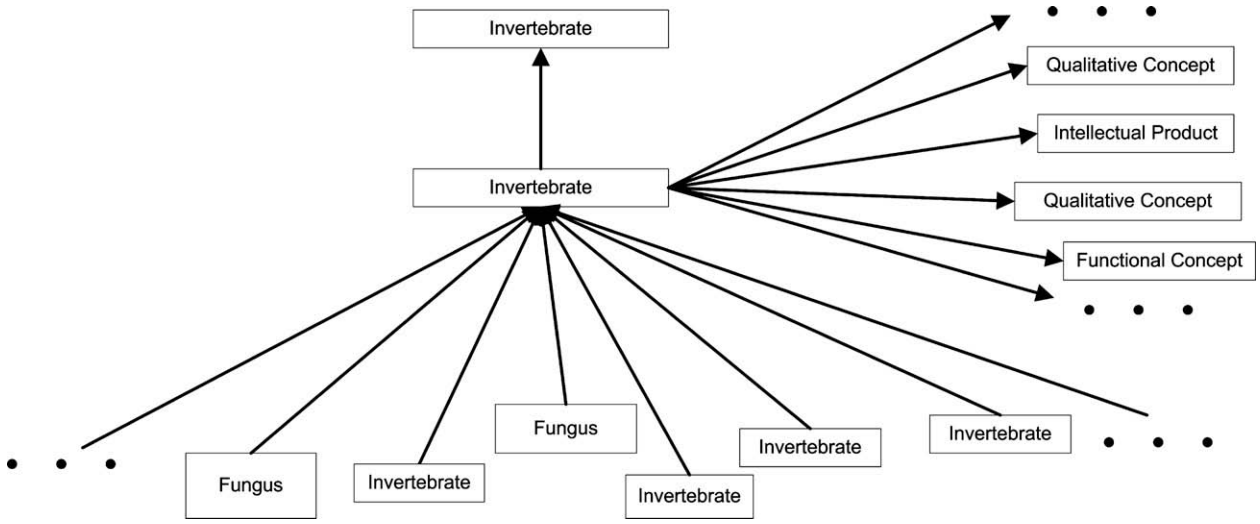

Fig. 3. Immediate semantic type neighborhood of *Microsporidia, Unclassified*.

place all grandchildren next to each other at the same vertical level. For concepts with many children, even the children cannot be displayed at the same vertical level in the diagram. Thus, concepts that are logically at the same level appear physically at different levels to make better use of the available drawing space. When an automatic layout tool is used, which is necessary for all but the most trivially small terminologies, the quality of the diagrams is usually even lower than for a layout that is done by a human.

There is another factor that makes both automated layout and visual comprehension of a concept diagram difficult, namely, the existence of multiple parents. Even in simple cases, this might lead to intersections of the parent/child arrows. When many lines intersect, diagrams become hard to read. Furthermore, intersections between arrows and boxes need to be avoided. If a layout tool indeed does not allow any such intersections, the number of constraints on a layout grows considerably, and closely related concepts, such as siblings, might appear far from each other. This leads to the previously mentioned problem of visually constructing neighborhoods over and over again. Thus, diagrams of concepts have clear disadvantages. On the other hand, pure text representations, even as indented lists, make it hard to follow a path from a concept's grandchildren to its grandparents, and vice versa. The existence of multiple parents makes the latter task especially difficult, because in the indented list only one parent precedes each child.

To solve these problems, we have developed a *hybrid diagram/text display* that provides the "best of both worlds" to the user. This display is dynamically produced by our Neighborhood Auditing Tool (NAT). Let us now see how the previously introduced neighborhoods are materialized with the hybrid display.

3.2.1. Layout for immediate neighborhood

In addition to the contextual elements of the immediate neighborhood, its layout contains some textual elements. The layout principles used for the immediate neighborhood are as follows.

(i) Every display is organized around the focus concept.
(ii) Lines (arrows) are completely eliminated. (A single arrowhead serves as a placeholder for all parent/child relationships between the children and the focus. The same is true for the parent/child relationships between the focus and its parents.)
(iii) Children are easily recognizable by being located below the focus concept. Parents are easily recognizable, by being located above the focus concept. Concepts related through lateral relationships are easily recognizable by being located to the right side of the focus concept.
(iv) Synonyms are displayed in a text box to the left of the focus concept, although they are textual knowledge elements and not concepts. The reason for including synonyms at the contextual level is that although they are textual elements, they are names of a concept. Hence, they refer by name to the focus concept and belong among the concepts.
(v) (a) In cases with few children, the display provides complete information, just as a diagram would. (b) In cases with many children, the display shows a manageable number of them and allows the user to scroll. (The same principle applies to parents and relationship targets.)
(vi) When definitions are available in the UMLS, the NAT allows the user to display them on demand.
(vii) In some cases, e.g., for ambiguous concepts, the deeper meaning of a term may be understood based on its source vocabularies. On demand, the tool displays these.

The last two options are controlled by marking a check box to the right of the parent box. We note that as a result of the elimination of lines (item (ii) above), intersections are eliminated and no sophisticated algorithms for layout are needed.

Fig. 4 shows the immediate neighborhood of the focus concept *Microsporidia, Unclassified* in the NAT. It corresponds to Fig. 1. Children of the focus concept are displayed in the subwindow with the label CHILDREN below the focus concept. Parents are displayed in the PARENT subwindow above the focus concept. Thus, we maintain the natural down-position of the children and the up-position of parents relative to the focus. This is an important cognitive advantage of diagrams and makes understanding them easier than text. The three boxes in Fig. 1 symbolize the corresponding subwindows.

3.2.2. Advanced layouts for various neighborhoods

3.2.2.1. Immediate neighborhood with semantic types. Fig. 5 shows the same neighborhood as Fig. 4; however, this time semantic types are also displayed for each concept. In this figure, the semantic type neighborhood, as shown in Fig. 3, is integrated into the NAT screen. Each concept name is immediately followed by the semantic type. To make it visually easier to distinguish the semantic types from the concepts, all semantic types are displayed in blue. Furthermore, as there may be more than one semantic type

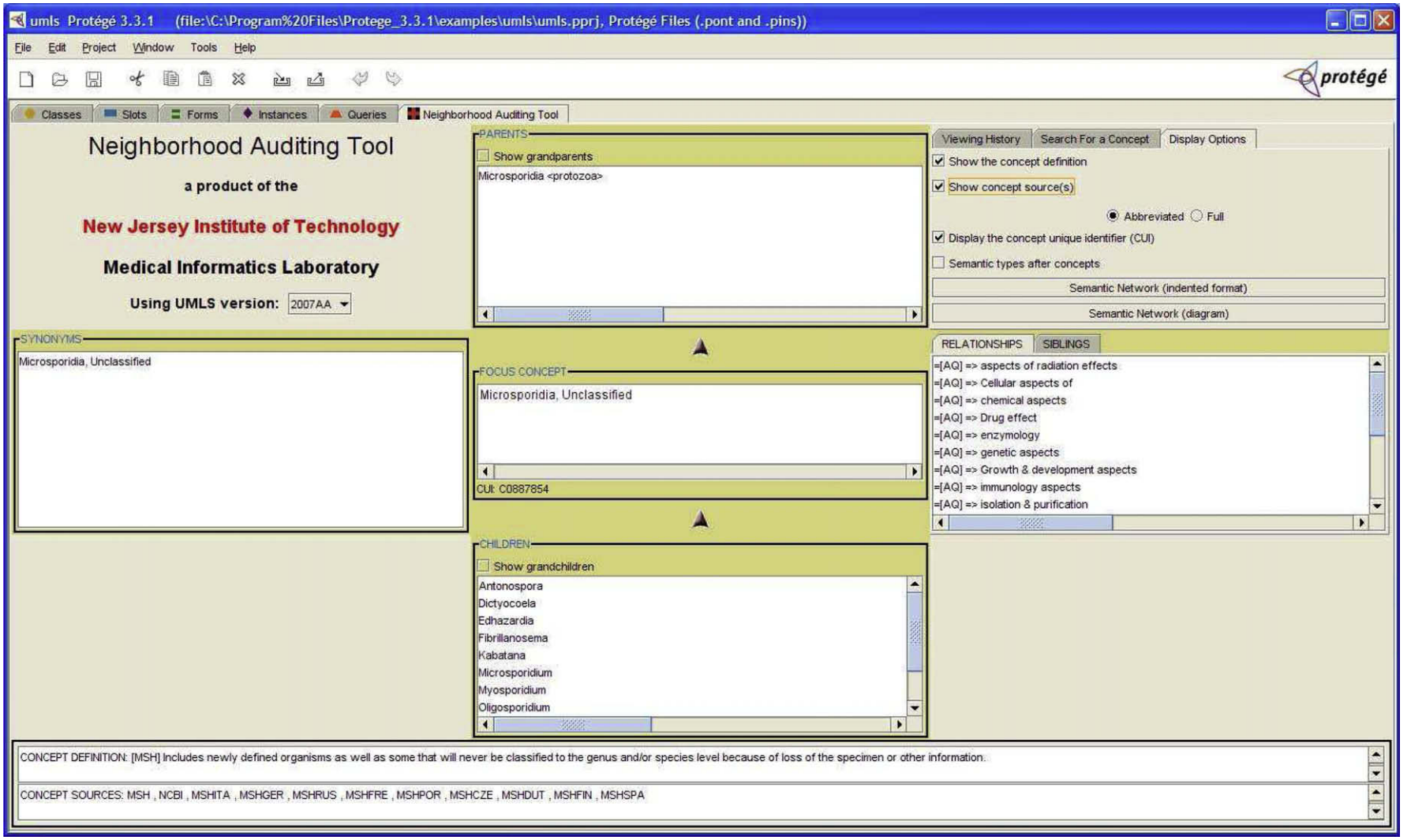

Fig. 4. NAT immediate neighborhood display of *Microsporidia,Unclassified*.

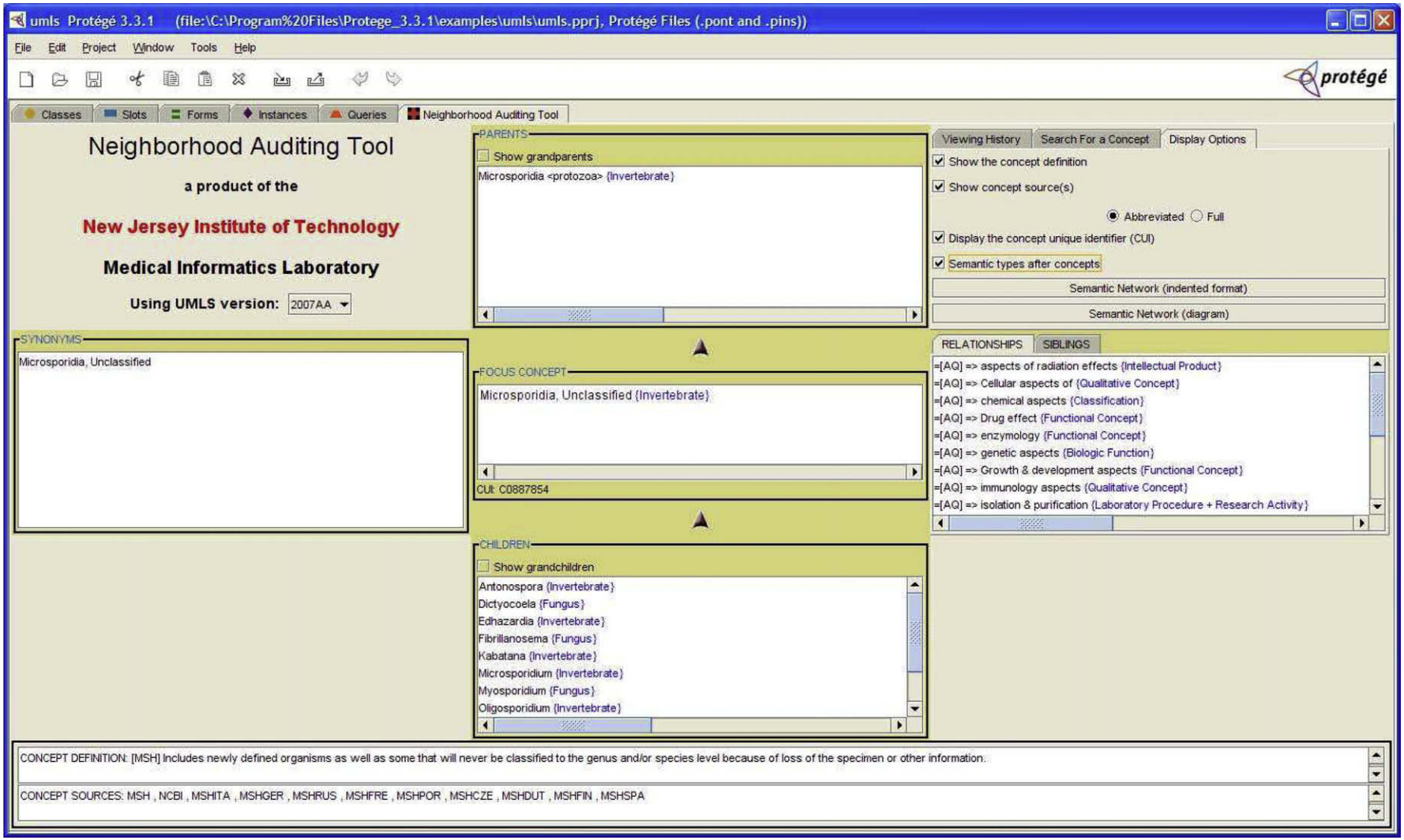

Fig. 5. NAT immediate neighborhood display of *Microsporidia, Unclassified* with semantic types.

for each concept, the semantic types are surrounded by blue curly brackets.

3.2.2.2. Extended neighborhood. The layout principles used for the extended neighborhood are as follows.

(i) When grandchildren and children are displayed, they are distinguished by indentation. The grandchildren are indented, and thus the information about which grandchildren belong to which child is still maintained. Note that such information would be lost by separate boxes for children and grandchildren. This option is controlled by a check box above the subwindow.
(ii) Similarly, when grandparents and parents are displayed, they are distinguished by indentation. The parents are indented. The information about which parents belong to which grandparent is again maintained. This option is controlled by a check box above the subwindow.
(iii) Siblings of the focus concept are displayed on demand. Logically, they should be placed at the same level as (i.e., laterally relative to) the focus concept. Due to the limited lateral screen space, the display of relationships and their targets is replaced by the sibling display. A user can switch back and forth between these two choices by clicking on the proper tab above the window.

Fig. 6 shows the layout of the extended neighborhood of the focus concept *Microsporidia, Unclassified*, corresponding to Fig. 2. While Fig. 2 is fairly complex and confusing, Fig. 6 shows almost the same information about the focus concept in a simple format. The user will need to scroll in the CHILDREN AND GRANDCHILDREN window; however, even in the diagrammatic display, some children and grandchildren were only hinted at by using an ellipsis. Thus, the comparison of these corresponding figures shows the power of the hybrid display to stay comprehensible while providing a large selection of the desired information.

The auditing tool is referred to as a *hybrid* tool because it uses indented text in two subwindows, but also maintains the relative diagrammatic positions of parents, children, etc. This leads to the plus-sign layout, accentuated by a darker (ocher) background area in Fig. 6.

3.2.2.3. Up-extended and down-extended neighborhoods. We omit a display of the up-extended and down-extended neighborhoods to save space. By marking only the check box "Show grandchildren", the immediate neighborhood would be transformed into the down-extended neighborhood. By marking only the check box "Show grandparents", the immediate neighborhood would be transformed into the up-extended neighborhood. Examples of up-extended and down-extended neighborhood screens are used in describing case studies in Section 4.

3.3. Additional tool features for auditing

The following additional features have been implemented in the NAT.

(i) A display of the complete SN as a diagram; it is displayed on demand.
(ii) A display of the complete SN as an indented list. In some cases, semantic types are far (several screens) apart, and a compact representation as an indented list is easier to understand. This option is also available in the NAT.
(iii) Definitions of semantic types. Understanding the essence of a semantic type is necessary to decide whether it has been assigned correctly to a concept. When the mouse is moved over a semantic type, its definition appears as a tool-tip.

3.4. A study to measure the impact of using the NAT

In [14], we reported on a study analyzing the performance of a group of auditors in an auditing task. Specifically, the task was auditing 70 UMLS concepts, each of which was assigned two semantic types. Furthermore, these ST combinations appeared only for small numbers of concepts in the whole UMLS. In fact, each combination of semantic types in the study appeared for only one to six concepts. According to [15], such concepts have a high likelihood of inconsistencies.

In a recent study, we wanted to investigate the impact of using the NAT on the performance of the auditors. In this study, four auditors dealt with two samples, labeled *A* and *B*, of UMLS concepts. All four auditors have medical training and experience with terminologies. The concepts were selected in the same way as in [14]. Each auditor handled one sample using the tool and one sample using simple text files, as described in Section 2.2 and in [14]. Two auditors processed sample *A* with the NAT and sample *B* using the files. The other two auditors processed the two samples in the opposite manner, i.e., sample *B* with the NAT and sample *A* with the files. The correctness of their reports was measured against a consensus auditing report, obtained by the two more experienced auditors, one from each of the above pairs, after reviewing the scrambled results of all four auditors.

Here, we just concentrate on the comparison of the performance of the four auditors using the NAT versus using the text files. For this purpose, we do not show the performance for the samples *A* and *B*. We just compare the results with and without the NAT for each of the four auditors. The study was designed with the two pairs of auditors switching the way they handled the auditing, to enable such a comparison, independent of the samples. We measure the recall (what fraction of the inconsistencies of the consensus report was found) and the precision (what fraction of the inconsistencies reported by an auditor actually appears in the consensus). The *F*-measure [37] is also calculated.

4. Results

In this section, we start off with an extensive case study that demonstrates the usefulness of the NAT in auditing. This is followed by two additional smaller case studies that further demonstrate how the NAT's various features help in the discovery of inconsistencies. Details about the implementation of the NAT are also described. Finally, the results of the impact study are presented.

The NAT is a general-purpose auditing tool that can support various auditing techniques. Whatever concept is chosen or submitted for review, the NAT provides choices for accessing the proper related knowledge needed for auditing it. To demonstrate this characteristic of the NAT, we will present three case studies showing how the NAT was used to support various auditing techniques in our research. The first, extensive case study in Section 4.1 is based on our research [13–16] showing that small groups of concepts (in this case, one concept) of a unique combination of multiple semantic type assignments have a high likelihood of inconsistencies. As is demonstrated, the discovery of one concept with incorrect multiple semantic type assignments helps to expose some inconsistencies in the META, as well as many other outdated semantic type assignments, for concepts assigned only one semantic type. The second case study concentrates on a cycle of child-of

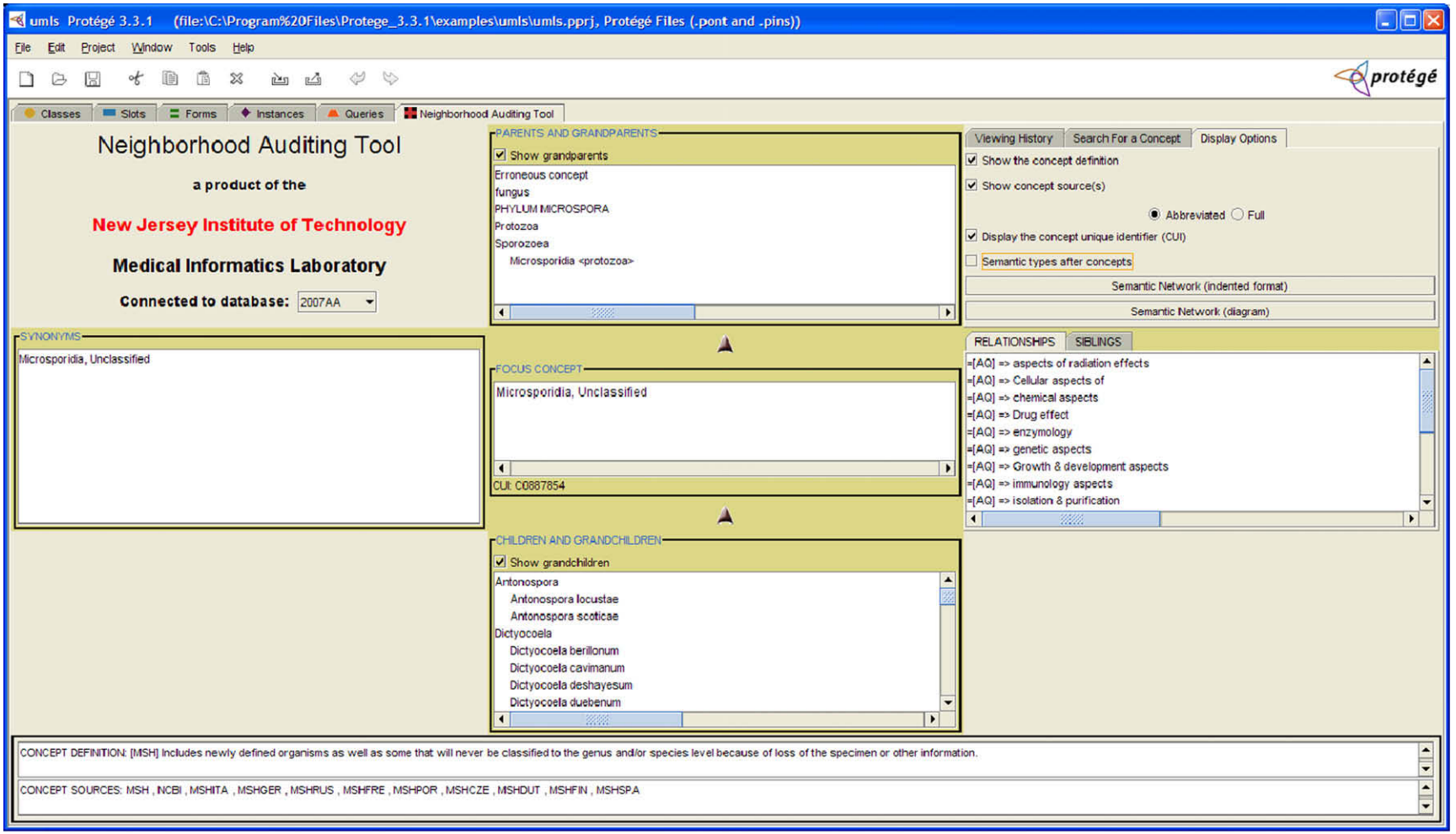

Fig. 6. NAT extended neighborhood display of *Microsporidia, Unclassified*.

relationships with three concepts. This example follows Bodenreider's research on cycles of hierarchical relationships in the META [2,29]. Two inverse hierarchical relationships from the same source terminology, DSM-IV, are discovered, one of which is an obvious error in the DSM-IV. The third example shows the analysis of a concept with a suspicious semantic type assignment, following our recent research on structural group auditing of a UMLS semantic type's entire group of assigned concepts (i.e., its extent) [7].

As we mentioned, the interplay between the two layers of the UMLS, the META and the SN, is utilized heavily in auditing the UMLS. We note that although the discovery of inconsistencies sometimes uses the semantic type assignments of concepts (which are also sometimes modified), more inconsistencies, not involving semantic types, may be located in the META. Thus, the SN supports the diagnostic process in those cases.

4.1. An extensive auditing case study

As a concrete case study, we present a review of an auditing session involving the concept *Antonospora Locustae* as it appeared in the UMLS 2007AA release. This concept was proposed to us as suspicious by our auditing research programs [14,15]. It has two semantic types, **Invertebrate** and **Fungus**. Looking at the animal kingdom, as partially embodied in the SN, these two semantic types should be mutually exclusive. Thus, our immediate judgment was that this combined semantic type assignment indicated a problem. However, we needed to collect more information about this concept to find the proper unique semantic type assignment for it.

We navigated to its parent *Antonospora* and looked at the immediate neighborhood (with semantic types), as defined in Section 3.2.2 and shown in Fig. 7. The result was striking. *Antonospora*, its parent *Microsporidia, Unclassified*, and one of its children, *Antonospora Scoticae*, were all classified as **Invertebrate**. Only the other child, *Antonospora Locustae*, had the suspicious double semantic type assignment. This example shows that the immediate neighborhood allowed us to literally surround an incorrect semantic type assignment. One thing that we noticed in this auditing process was that neither *Antonospora Locustae* nor *Antonospora* has a definition in the META. Thus, we advanced one level up, to *Microsporidia, Unclassified* (Fig. 5), in the hope of finding a definition. Interestingly enough, a definition exists, but it focuses on the "unclassified" part of the concept, providing a generic definition for any "unclassified organism" concept, not for microsporidia:

> [MSH] Includes newly defined organisms as well as some that will never be classified to the genus and/or species level because of loss of the specimen or other information.

We judged the MeSH definition for this concept to be improper. Reporting this fact to the editors of MeSH could result in an update of the MeSH, which would eventually propagate to a new release of the UMLS. Next we wanted to see the neighborhood of *Microsporidia, Unclassified*. The children of *Microsporidia, Unclassified* provided an interesting picture. Some of them had the semantic type **Invertebrate**, while others had **Fungus**. None had both—which should not happen for exclusive semantic types. This configuration of siblings of *Antonospora* assigned two exclusive types was strange and probably hinted at an inconsistency. But which one was the proper semantic type?

Because we did not get clarification about the nature of the concepts retrieved up to this point, we switched to the extended neighborhood of *Microsporidia, Unclassified* (Fig. 8) with the hope that the extra knowledge elements would bring such clarification. We saw that the grandparents included both the concepts *Fungus* (ST **Fungus**) and *Protozoa* (ST **Invertebrate**). This explained, on an intuitive level, where the assignment of the two semantic types came from. Yet, it still did not clarify which one was proper. Moving up one level (see Fig. 9) to a down-extended neighborhood of *Microsporidia* ⟨*Protozoa*⟩, we finally found a definition which appeared valid.

> [CSP2006] minute intracellular parasites with spores of unicellular origin; they have been treated as protozoa; in the most recent phylogenetic analyses microsporidia branch among the fungi. | [MSH2007-2007-05-01] A phylum of fungi comprising minute intracellular PARASITES with FUNGAL SPORES of unicellular origin. It has two classes: Rudimicrosporea and MICROSPOREA.

We noticed that this concept had been reclassified recently in phylogenetic research, and is now considered a **Fungus**. Thus, the semantic type **Invertebrate** appeared to be a "leftover" from old knowledge that was not updated according to new definitions. The assigned semantic type should be **Fungus**. The down-extended neighborhood of *Microsporidia*, ⟨*Protozoa*⟩ (Fig. 9) showed all of the children and grandchildren which should have been assigned **Fungus**. On the other hand, the grandparent *Protozoa* was properly assigned **Invertebrate**. Furthermore, *Protozoa*, a single-cell organism, is properly a second parent of *Microsporidia*, ⟨*Protozoa*⟩, due to the latter's "spores of unicellular origin".

In an effort to verify the needed widespread modification to all descendants of *Microsporidia* ⟨*Protozoa*⟩, we found among its children *Apansporoblastina*, which has a semantic type **Invertebrate** (Fig. 9). We looked at its definition, which contains the designation "fungus".

> [MSH] Apansporoblastina: A suborder of FUNGI in the phylum MICROSPORIDIA, commonly lacking a pansporoblastic membrane. The sporoblast is usually dinucleate.

We next looked at *Encephalitozoon*, another child assigned **Invertebrate**. It also has a definition based on "fungus".

> [MSH] A genus of FUNGI originally considered a member of the class SPOROZOEA but now recognized as part of the class MICROSPOREA.

Thus, we found more support for our suspicion that all children and grandchildren of *Microsporidia* ⟨*Protozoa*⟩ should be assigned **Fungus**. Our conclusion is that all descendants of *Microsporidia* ⟨*Protozoa*⟩ assigned **Invertebrate** should be considered for assignment of **Fungus** instead.

Above (in Fig. 9), we used the down-extended neighborhood of *Microsporidia* ⟨*Protozoa*⟩ to display all its parents, since in the extended neighborhood screen not all parents are visible simultaneously due to the large number of grandparents and parents. Concerning the usefulness of the down-extended neighborhood, one can also use it to quickly tell whether an improper semantic type assignment has been propagated downward one level or more. If there are inconsistencies at the grandchildren level, the auditor would need to navigate downward to see if the problem has spread even farther. On the other hand, the inconsistency propagation might be limited to the children of the focus concept, in which case the down-extended neighborhood is sufficient. In our case study example, an auditor would use the down-extended neighborhood (of Fig. 9) to quickly scan the semantic types of all the children and many grandchildren of *Microsporidia, Unclassified*, utilizing the scroll bar.

Our suspicion is that when the UMLS sources CSP and MSH were changed such that their definitions reflected the change of scientific knowledge about *Microsporidia* ⟨*Protozoa*⟩ and all its descendants, the information was later updated in the next release of the UMLS. However, since the ST assignments do not appear in

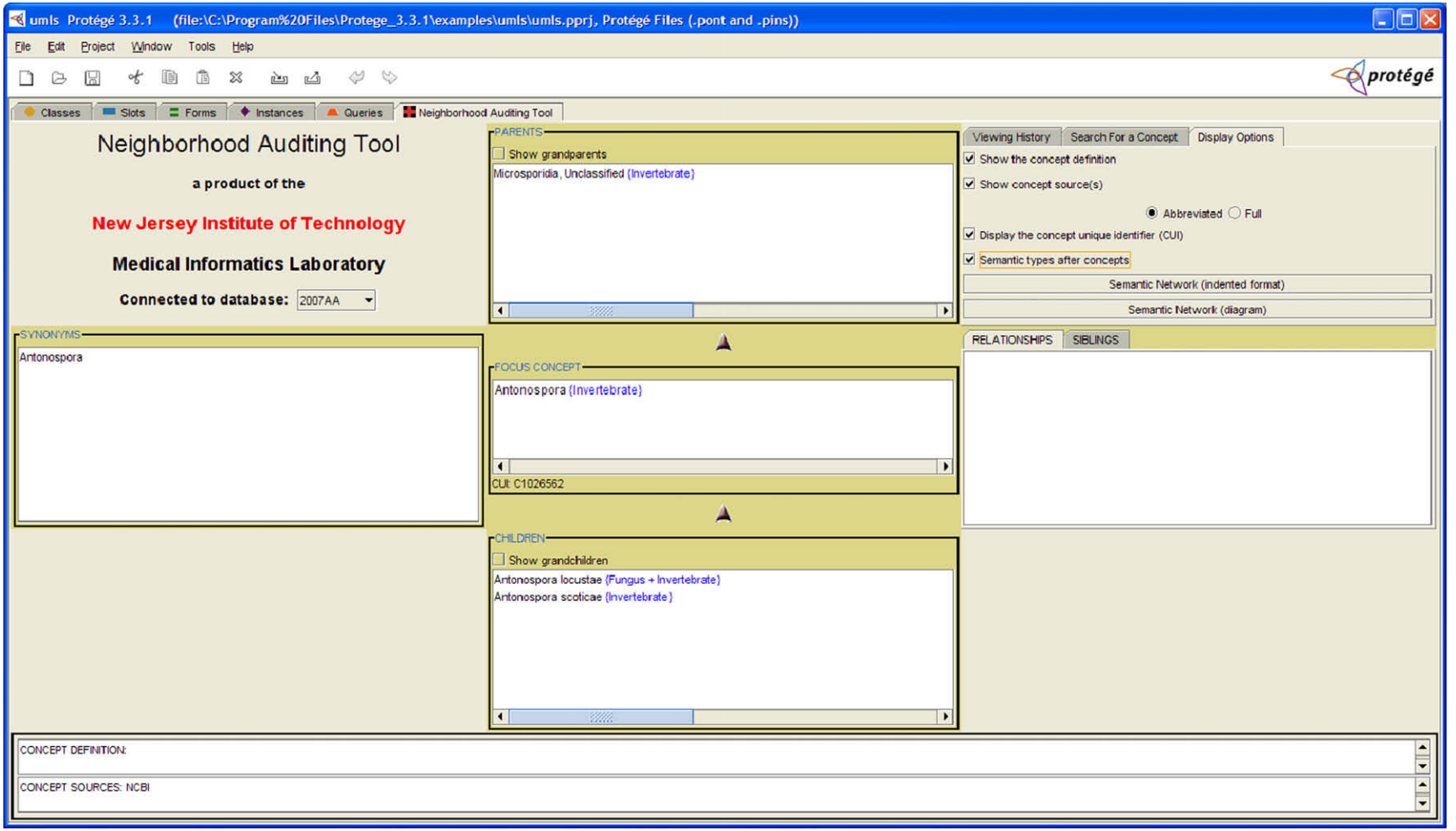

Fig. 7. NAT display of immediate neighborhood of *Antonospora* with semantic types.

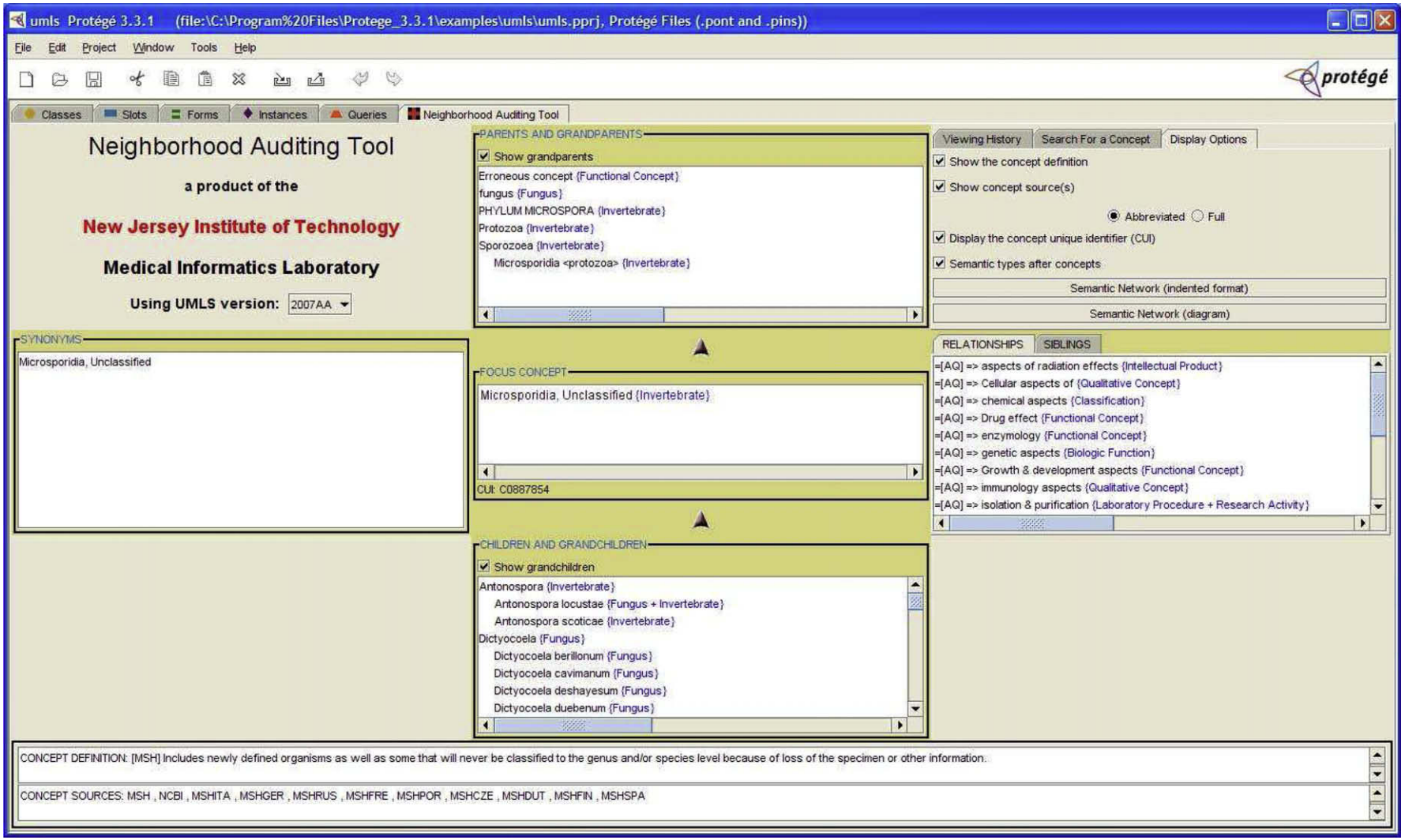

Fig. 8. NAT extended neighborhood display of *Microsporidia, Unclassified* with semantic types.

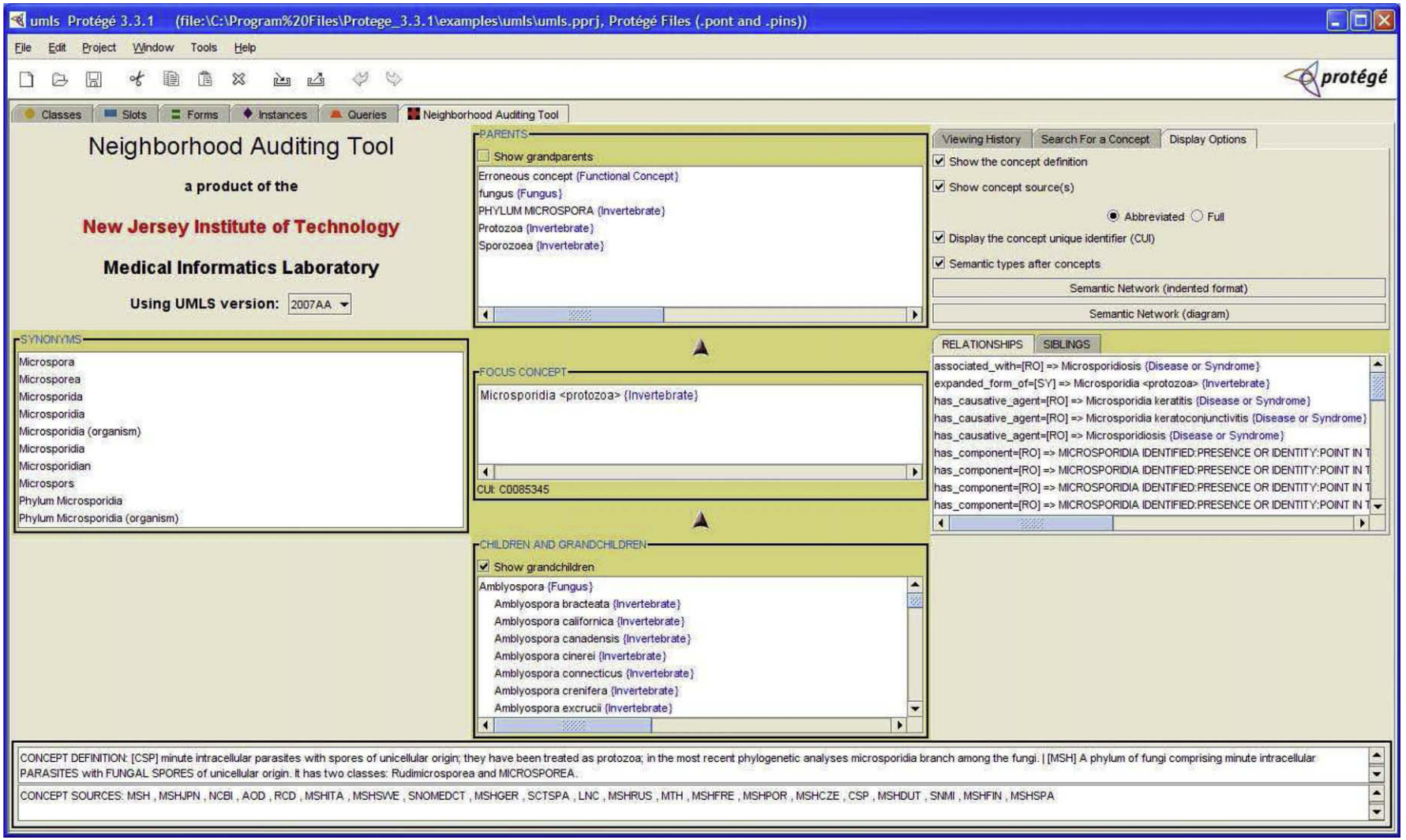

Fig. 9. NAT down-extended neighborhood display of *Microsporidia* ⟨*Protozoa*⟩ with semantic types.

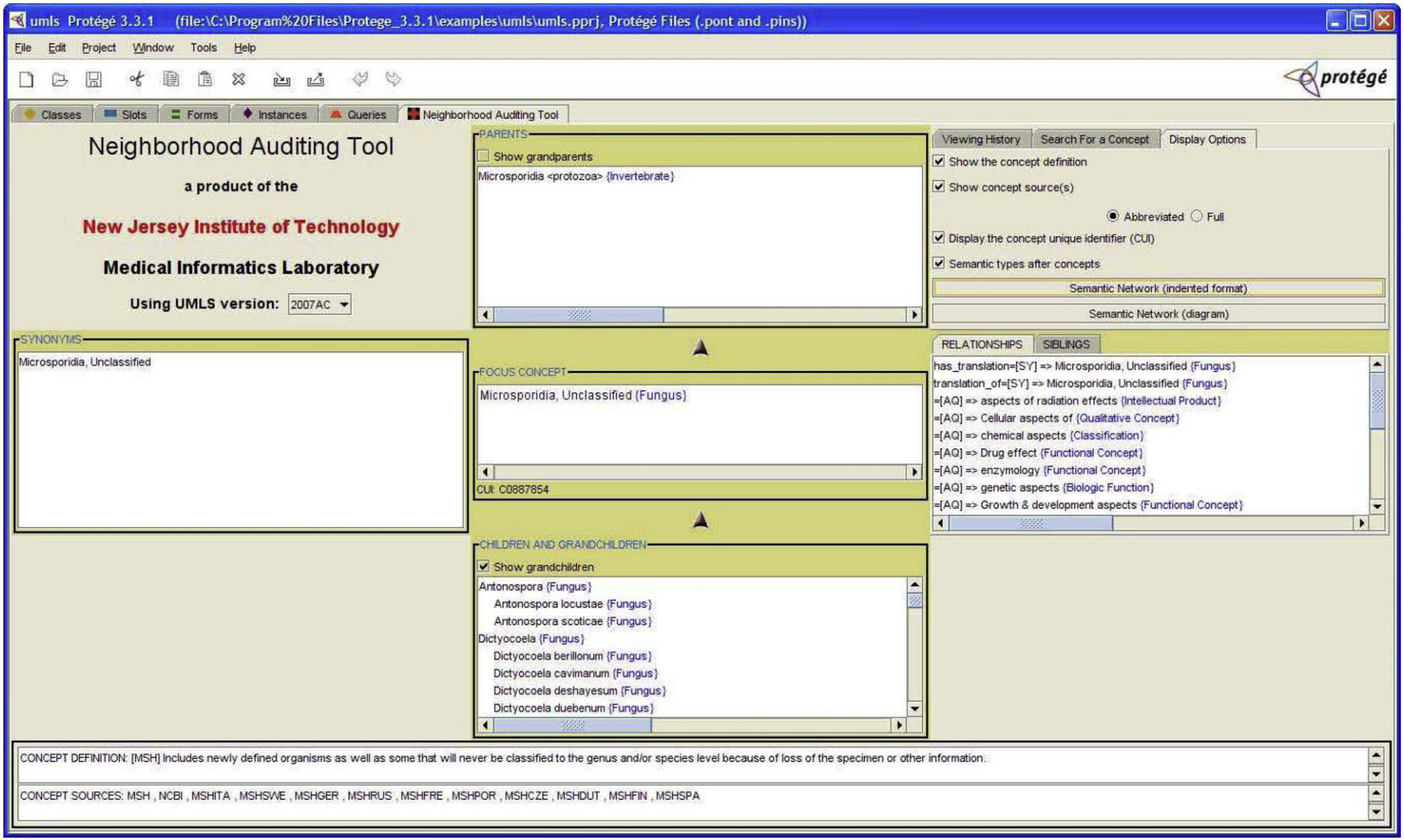

Fig. 10. NAT (2007AC) down-extended neighborhood display of *Microsporidia, Unclassified* with semantic types.

the sources, they were not automatically updated. Hence, only for some of the descendants were the ST assignments updated, leaving the descendants of *Microsporidia* ⟨*Protozoa*⟩ in an inconsistent state of semantic type assignments.

As mentioned, our previous research [13–16] has shown that when very few concepts (or just one) are assigned a specific combination of semantic types, then there is a higher likelihood of inconsistencies. Thus, our research targets such concepts for auditing. Our case study started with such a concept, *Antonospora Locustae*. As a matter of fact, we originally audited this concept with a text file, described in Section 2.2, and our domain experts determined it should be assigned **Fungus**. But without the NAT tool, the auditors did not explore further propagation of the inconsistency. This suggested correction was reported to the NLM. Indeed, in the 2007AC release (see the version number in the upper left corner of Fig. 10), we can see that *Antonospora Locustae* is assigned **Fungus**. Furthermore, the UMLS editors must have reviewed the related concepts, as we see that the sibling *Antonospora, Scoticae*, its parent *Antonospora*, and its grandparent *Microsporidia, Unclassified* were all assigned **Fungus** instead of **Invertebrate**.

In Fig. 10, we can also see that this correction was stopped at *Microsporidia, Unclassified*, but did not spread to children of *Microsporidia, Unclassified* assigned **Invertebrate** before, like *Edhazardia* and its child or to the parent *Microsporidia* ⟨*Protozoa*⟩ and its descendants. Our auditors, when working with the tool, have found that the **Fungus** assignment should propagate to all descendants of *Microsporidia* ⟨*Protozoa*⟩. This case study suggests that easy navigation, extended neighborhoods, and the display of semantic types for all the concepts in a neighborhood encourage auditors to explore the limits of upward and downward inconsistency propagation.

Our experience in previous work has shown us that inconsistencies in semantic type assignments sometimes indicate the existence of other inconsistencies. This phenomenon was also seen in this case study.

We already remarked on the improper definition (from MeSH) of *Microsporidia, Unclassified*. Looking for similar concepts with names that include "Unclassified", we found that there are proper definitions in MeSH for *Fungi, Unclassified* and *Viruses, Unclassified*. Hence the definition of *Microsporidia, Unclassified* should be corrected in a similar manner. By the policy of the NLM, the UMLS has to reflect the contents of the sources, even when erroneous. However, one can report the errors to the organization in charge of the specific source—the NLM in the case of the MeSH. Corrections of the errors in the MeSH will be propagated to the UMLS when the new release of the MeSH is included in the UMLS.

Let us now demonstrate two other inconsistencies exposed while working on this case study. *Microsporidia* ⟨*Protozoa*⟩ has a synonym *Microsporea* (in MeSH) and a child *Microsporea* (in NCBI). The UMLS has to reflect the contents of both sources, but obviously only one can be true. By contacting both organizations responsible for these sources, hopefully, a resolution can be achieved, which would propagate to a later release of the UMLS.

Another inconsistency occurs with regards to the two children of *Microsporidia* ⟨*Protozoa*⟩: *Apansporoblastina* and *Encephalitozoon*. Both are assigned **Invertebrate**, while their definitions say they are fungi, as reported earlier. Upon further review, we see that the second is the child of the first in the MeSH, but they are both children of *Microsporidia* ⟨*Protozoa*⟩ and *Apansporoblastina* in the NCBI and *Encephalitozoon* in the SNOMED-CT. In this case, the contents of the three sources combined exhibits an inconsistency.

In these three examples, we saw that the NAT enabled us to unearth three different kinds of inconsistencies: an incorrect definition, an inconsistency between a child and a synonym of the same name, and an inconsistency in the hierarchical relationships. Although none of those problems can be solved in the UMLS itself, since they are derived from sources, they nevertheless demonstrate the capabilities of the NAT in exposing such problems.

4.2. More case studies

The following examples demonstrate cases where the NAT has helped the auditor by presenting the relevant information in an appropriate format. The various kinds of neighborhoods and additional accompanying material are shown in light of their support for the auditing process.

The examples are taken from our research on designing techniques for automatically exposing concepts with high likelihoods of inconsistencies. Such concepts should be reviewed by domain expert auditors, who can assess the representation.

Example 1: Bodenreider et al. [2,29] found many cycles consisting of parent/child and broader/narrower links in the META. We are interested in cycles consisting only of parent/child relationships, i.e., no broader/narrower links. The child-of relationships are modeled as hierarchical relationships in their source terminologies [2]. Finding cycles where a concept is its own parent (and therefore its own child) is relatively easy. We looked for cycles involving three different concepts, connected by child-of links. Locating such cycles involved a database query. The NAT allowed us to visualize such a cycle and the knowledge relevant to it, and to suggest a way of correcting the inconsistent modeling it represents.

One such cycle that was discovered and analyzed consists of the three concepts *Mood Disorders, Bipolar Disorder,* and *Affective Disorders, Psychotic*. Looking at *Bipolar Disorder* as a focus concept in the NAT (Fig. 11), one can see that *Mood Disorders* is its child. At the same time, *Affective Disorders, Psychotic* is its parent and *Mood Disorders* is its proper grandparent, completing the cycle. Fig. 11 shows this configuration in the NAT. Note that Fig. 11 shows an up-extended neighborhood to capture the four levels needed to illustrate the cycle of three concepts. Fig. 12(a) shows a diagram of the child-of relationships between the pairs of these three concepts and the source terminologies of some of them.

According to the MeSH, these three concepts are defined as follows. (The definitions from other sources are similar.)

Bipolar Disorder: A major affective disorder marked by severe mood swings (manic or major depressive episodes) and a tendency to remission and recurrence.

Affective Disorders, Psychotic: Disorders in which the essential feature is a severe disturbance in mood (depression, anxiety, elation, and excitement) accompanied by psychotic symptoms such as delusions, hallucinations, gross impairment in reality testing, etc.

Mood Disorders: Those disorders that have a disturbance in mood as their predominant feature.

Studying these definitions, we found that *Mood Disorders* is more general than *Bipolar Disorder*, which is a specific mood disorder. Similarly, *Mood Disorders* is also more general than *Affective Disorders, Psychotic*, as *Affective Disorder* is a synonym of mood disorder. However, no child-of relationship should exist between *Bipolar Disorder* and *Affective Disorders, Psychotic* because bipolar disorder is not necessarily psychotic, i.e., accompanied by delusions, etc., while not every psychotic disorder is bipolar. Thus, the child-of from *Mood Disorders* to *Bipolar Disorder*, which comes only from the DSM-IV, also needs to be broken, because the opposite child-of from *Bipolar Disorder* to *Mood Disorders*, in DSM-IV and many other sources, fits the definitions. Note that breaking the cycle made of child-of relationships among these three concepts has to be done in a "source sensitive" manner, due to the policy of the UMLS of accurately representing the content of each source vocabulary, even when the integration of their knowledge leads to contradictions.

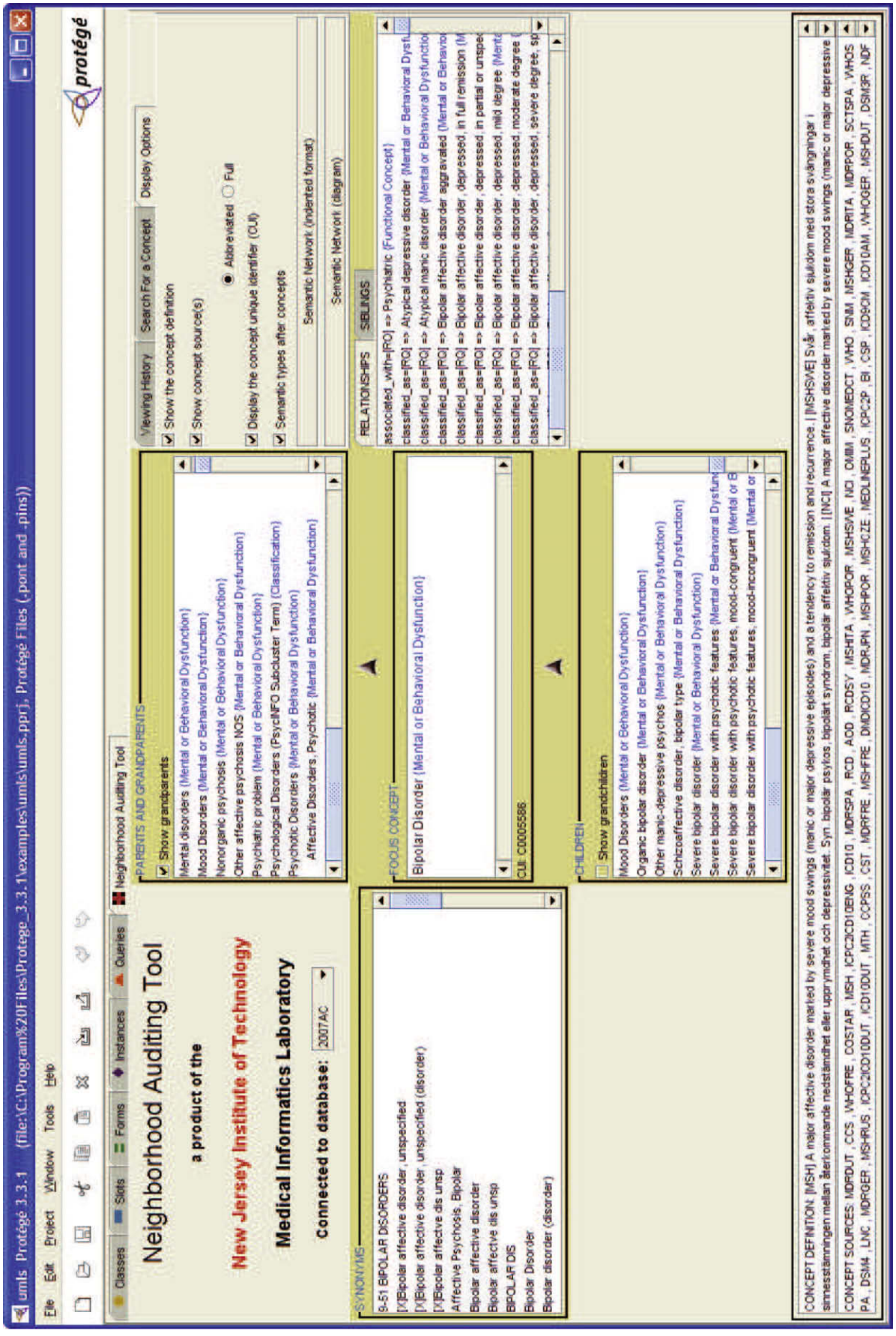

Fig. 11. Cycle of concepts in the NAT.

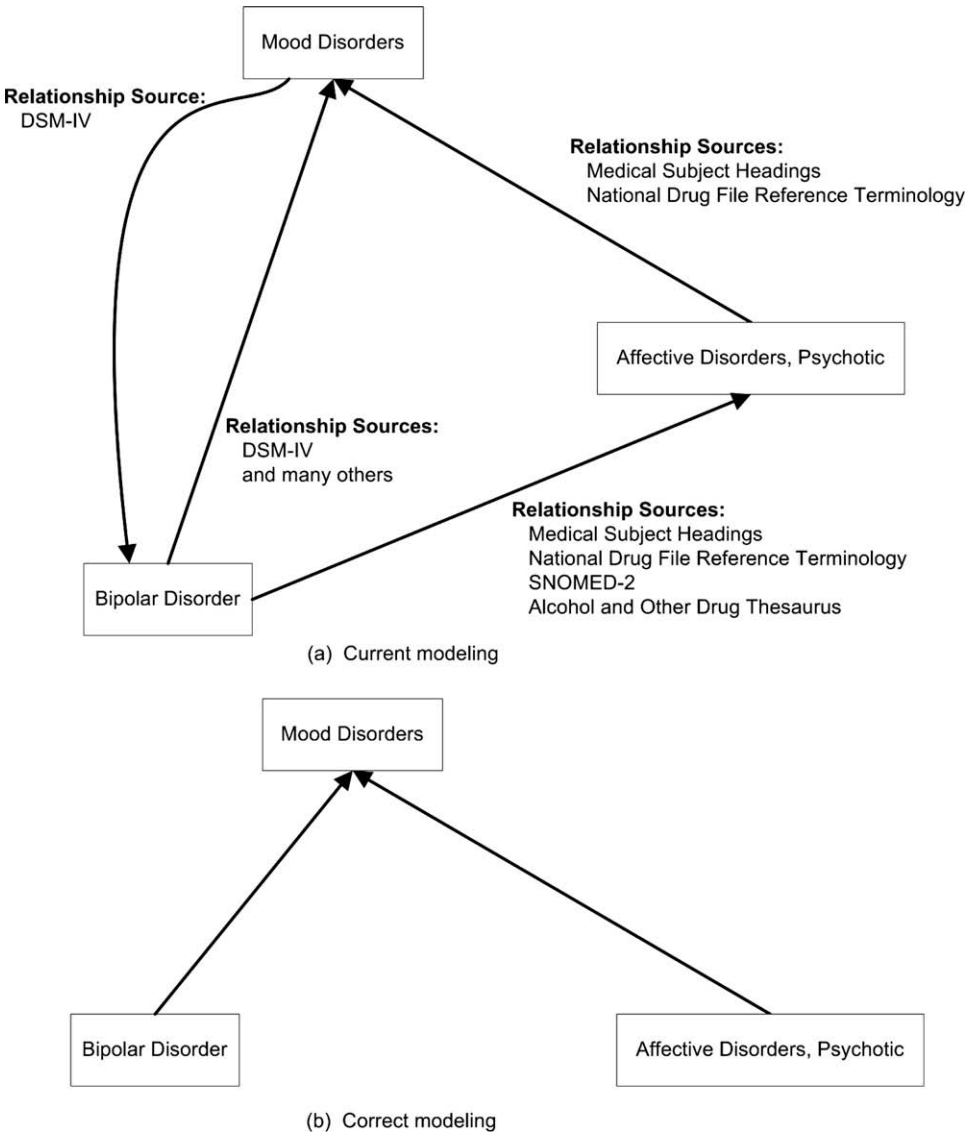

Fig. 12. Cycle of concepts in META.

Interestingly, this instance of a cycle of two opposite child-of relationships in DSM-IV is an error in DSM-IV itself rather than an inconsistency between various source vocabularies. Reporting this error to DSM-IV will eventually propagate to a correction of this inconsistency in the UMLS and a rectification of this cycle.

Furthermore, the child-of from *Bipolar Disorder* to *Affective Disorders, Psychotic* is improperly modeled. The MeSH definitions of the two concepts do not support this relationship. However, this relationship comes from three sources, AOD, MeSH, and NDF. We note that removing the relationship from the UMLS is less critical, because, although it is improper, it does not cause a cycle. The actual removal of this relationship from the UMLS requires changes in the sources by their respective organizations. Fig. 12(b) shows the proper child-of relationships among these three concepts.

Example 2: In [7], the concept *Genetically Engineered Mouse*, assigned the semantic type **Experimental Model of Disease (EMD)**, is considered to have a "suspicious" semantic type assignment in the 2006AC release of the UMLS, since the semantic type of its parent concept *Organism Modification* is **Research Activity (RA)** (see Fig. 13). Specifically, it is considered suspicious because there is neither an IS-A relationship nor an IS-A path in the SN from **EMD** to **RA**. Instead, the semantic type of *Genetically Engineered Mouse* should be **Mammal**. Furthermore, its parent should be *Animals, Laboratory*, assigned **Mammal** (as appears in the 2007AC release), rather than *Organism Modification*. Moreover, of the children of *Genetically Engineered Mouse*, namely, *Knock-in Mouse, Knock-out,* and *Retrovirus Research Technique*, only the first should be its child. Also, the semantic type assignment of *Knock-in Mouse* should be only **Mammal**, not both **Mammal** and **EMD**.

At the same time, *Knock-in* and *Knock-out* should both be children of *Organism Modification*, all of which should be assigned **RA**. In the 2007AC version, *Knock-in* is modeled this way. *Knock-out* is still assigned **EMD**. Listing the semantic types for the parents and children of a focus concept in the NAT helps to expose these inconsistencies in child-of relationships. This example demonstrates how considering one concept with a suspicious semantic type assignment propagates to modifications of parent relationships for several concepts.

4.3. Implementation

The NAT front-end was initially developed as a plug-in for Protégé [35] and later converted into a stand-alone Web-based application based on the Java Network Launching Protocol (JNLP). This

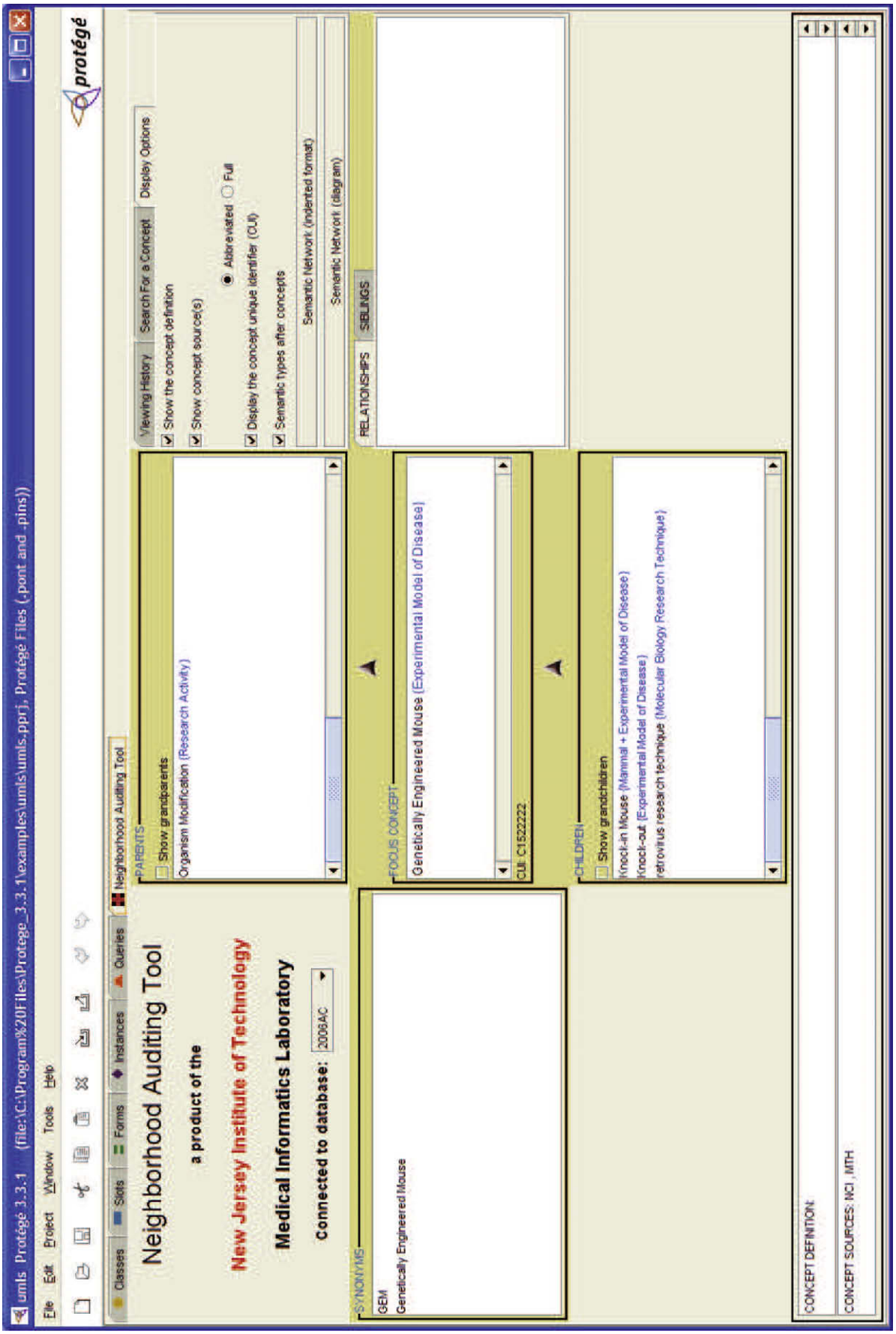

Fig. 13. Suspicious semantic type assignment.

means that the NAT is started from within a Web browser, but is executed directly on a client machine. This implementation makes full use of the screen and is not affected by a number of compatibility problems with different versions of commercial browsers.

Data from the UMLS releases 2006AC up to 2008AA is stored in an Oracle back-end database. When looking for a concept, the user does not need to know the exact spelling of its preferred term, because we implemented a partial match facility. This facility makes full use of the Lexical Variant Generation (LVG) libraries created and provided by the NLM [21]. Thus, if a user types in the words "broken arm", the term returned by the NAT search is *Fracture of upper limb NOS*, which has the same meaning as "broken arm", but contains neither "broken" nor "arm".

We achieved a functional NAT system early in the development process, but it was too slow for practical use. Navigation from one concept to another sometimes took as long as 90 s. Three principal improvements to the system decreased these worst case times for navigation from 90 s to about 2 s. First, concept data is now stored locally in a "cache" instead of being repeatedly retrieved from the database. Thus, whenever the same concept appears before and after a navigation step, the time of a second database retrieval is saved. Second, an optimized Java database query mechanism was used instead of the "textbook" query mechanism. The third improvement involved the elimination of repeated, unnecessary, automatic resize operations of large text areas.

4.4. Evaluation study with auditors

In Table 1, we show the results of the study we conducted to measure the performance of four auditors working with and without the NAT. The performance was measured relative to a consensus reached by the two more experienced auditors after they reviewed the scrambled results of all four auditors (including their own).

We see that the number of inconsistencies found with the NAT was higher than with the simple text files. Furthermore, it was higher for each one of the four auditors. The average number of inconsistencies was 56 versus 44. Similarly, the average recall was 0.65 versus 0.57. Again, the recall with the NAT was higher for each of the four auditors, although for the last one it is almost equal. No significant difference was observed for the precision. The *F*-measure, symmetrically accumulating the contribution of both recall and precision, is better for the NAT, but less significantly so than for the recall.

5. Discussion

5.1. Design considerations

The NAT was designed to facilitate the work of a domain-expert auditor while working on the UMLS. The determination of which knowledge elements of a concept an auditor may wish to see is a challenge. The reason is that in auditing it is not always predictable which extra knowledge elements will be desired after a review of the knowledge in the immediate neighborhood of a focus concept. One extreme is to show an auditor "all" possible knowledge elements in an effort to cover all that he may ever need. But then we face the other challenge of avoiding mental overload. Trying to fill the screen with too much knowledge will almost surely backfire. In such a case, an auditor may feel overwhelmed and his effectiveness will be reduced. Hence, the designers of such a tool need to search for a balance between displaying all desired knowledge and limiting the amount of knowledge to that which is predicted to be most relevant.

Our research group has, over the years, accumulated experience with auditing the UMLS [13–16,33] as well as other terminologies. Utilizing this experience, we formulated the file-based auditing approach discussed earlier. For example, those files included the parents and children of a concept, with their semantic types, but not the grandparents, grandchildren, or concepts related through lateral relationships. If an auditor wished to view more details, he could utilize the UMLSKS system. But, not surprising, in most cases the auditors tended to make decisions based only on the data from the files provided to them.

With the NAT interface, we can enable the auditor to start with the limited knowledge elements of an immediate neighborhood, but extend the screen or navigate to view more elements on demand. With this flexibility, the NAT offers a balance between the wish to capture all potentially desired knowledge and the need to avoid overwhelming the auditor. The use of the scrollable text windows provides a similar balance regarding the use of neighborhoods with an overwhelming number of related concepts, e.g., children or grandchildren. Other balancing options of the auditor are whether he wants to list the semantic types of all concepts, and whether he wants to display the concept definition(s) and concept sources of the focus concept. These balances were achieved after watching the feedback of our domain-expert auditors using earlier prototypes with various options. Hopefully, these characteristics of the NAT will make it suitable for use in the editing process of the UMLS maintenance personnel.

5.2. Evaluating the impact of the NAT on the performance of auditors

Our interpretation of the results of our impact study is that the NAT with its rich offering of knowledge and the possibility of easy navigation helps the auditor to discover more inconsistencies. However, the precision of their inconsistencies with the NAT is not better; in fact, the average precision is even a bit lower, although the difference is not significant. It seems that the precision depends more on the capabilities of the auditors than on the accessibility of knowledge. In other words, the precision seems to depend on the decision process of the auditor, rather than on the accessible knowledge. More experiments are needed to assess the impact of using the NAT on the performance of the auditing. It is also interesting to note that the four auditors in the study expressed preferences of working with the NAT versus the files or the UMLSKS interface.

Table 1
Performance comparison of auditors with and without the NAT.

Auditor	Inconsistencies		Recall		Precision		*F*	
	Tool	No tool	Tool	No tool	Tool	No tool	Tool	No tool
1	57	45	0.97	0.82	0.53	0.51	0.86	0.63
2	22	20	0.43	0.35	0.55	0.55	0.48	0.43
3	39	34	0.64	0.58	0.46	0.53	0.54	0.55
4	56	44	0.55	0.54	0.30	0.34	0.39	0.42
Average	44	36	0.65	0.57	0.46	0.48	0.57	0.51

5.3. Comparison with UMLSKS tools

Our preliminary experience with the NAT has been that it supports auditing more effectively than the UMLS Knowledge Source Server [44] because it brings the most needed information together at one place, without overwhelming the auditor. Before building this tool, much of our auditing was done by (i) using prepared text files containing the results of database searches in an Oracle database version of the UMLS and (ii) accessing a previous version of the UMLSKS when an auditor wished to navigate through the UMLS, and the prepared text file was not sufficient. The comparison with the UMLSKS tools for the purpose of auditing is not entirely fair, as they were not designed for auditing but rather as general-purpose browsing tools for the multiple uses of the UMLS [8,12]. But since they were the only tools available to us for auditing before the NAT, we are presenting this comparison.

The UMLSKS, both Version 6.0 and the older version, relies exclusively on indented text. The older version is deficient because of the limited control the user has over the amount of information that is displayed. The organization of the data through the filter of the source vocabularies causes much repetition, which might be overwhelming to the auditor. In Version 6.0, which is only a beta version, the repetition is hidden, since the list of children for each terminology is not shown until the user clicks on the appropriate plus-signs. For the task of auditing, it would be more effective to display the list of children without reference to their sources. If an auditor wants to know the sources for a specific child, the list should be displayed on demand. Such a feature is listed in our future work for the NAT.

The Semantic Navigator is philosophically closer to the NAT in that it mixes textual (not indented) and graphical displays. However, the Semantic Navigator only partially maintains the diagrammatic layout of information.

The Semantic Navigator still seems to display too much knowledge, although it does so to a lesser degree. For example, the diagram may require scrolling. Some of the frames, e.g., Siblings, should be displayable on request only. On the other side, some knowledge is only presented in response to clicking, e.g., information on a specific relationship. The color coding of relationships expresses their sources, but it does not scale to the large number of sources and their combinations. The Semantic Navigator also does not eliminate the visual confusion caused by multiple intersections of links.

In contrast, the NAT offers a more uniform presentation of the various kinds of knowledge elements. This stylized and systematic presentation reduces the complexity for the user. Furthermore, on demand, all concepts can be viewed in the NAT with their semantic types, in contrast to the situation with the Semantic Navigator and the UMLSKS.

5.4. Future work

5.4.1. NAT features for future releases

The current version of the NAT has achieved the goal of making it easy to audit the highly complex and extremely large UMLS while avoiding and controlling information overload. In the future, we intend to add the following features, among others, which will further serve to avoid displaying too much knowledge unless it is demanded.

- A feature needs to be added to allow a user to audit the data from one single (or a selected group of) source vocabularies of his choice, e.g., only SNOMED-CT concepts and relationships.
- A user should be able to limit the terms displayed by default to English and by choice to any other UMLS-supported natural language, as auditing is rarely done in a multi-lingual context.
- The current NAT does not allow a user to see from which source(s) a relationship (parent, broader, etc.) has been imported. One possible way of achieving this effect is to make relationships clickable, so that their sources become visible. A second way is to provide a query window. When the user enters the two endpoints of a relationship and submits the query, the relationships between the two endpoints and their sources would be displayed.
- The NAT lists all the sources that a term comes from. This list should be pruned to contain only English sources.

5.4.2. Algorithms for finding input concepts for auditing with the NAT

Algorithms for identifying concepts with high likelihood of inconsistencies are beneficial for effective utilization of the limited auditing resources available for terminologies in general and for the UMLS in particular. Directing auditors to concentrate their efforts on such inconsistent concepts will increase the positive impact of their work. A number of researchers, including ourselves, have devised auditing methodologies combining such algorithms with manual audits of the concepts returned by them. The output of these algorithms can serve as input to auditing with the NAT, as demonstrated by the case studies presented. Another source of input for the NAT may be reports of users who encounter problematic or inconsistent concepts and report them to the UMLS team.

In the future we plan to integrate such algorithms as part of the NAT tool, so that the whole auditing process is handled by one integrated software system. Examples of such algorithms include identifying semantic type intersections of extents of small sizes [13–16], redundant semantic type assignments [33], circular hierarchical relationships [3,29], and pairs of concepts for which the hierarchical relationships in the META are inconsistent with the hierarchical relationships between their assigned semantic types.

In recent work [7,6] we have introduced the principle of group auditing. Rather than considering concepts one by one, groups of concepts which are purportedly uniform in their semantics are reviewed together, enabling the auditor to recognize those concepts that obviously do not belong in a group. Those concepts are likely to exhibit inconsistencies. To better utilize auditing resources, algorithms are used to identify suspicious concepts [7]. For better support of the work of the auditor, these groups of concepts are further partitioned into smaller, cohesive, singly-rooted subgroups of a more refined unified semantics [6]. As shown in [6], concepts with inconsistencies in their semantic type assignments are more likely than other concepts to lack certain parent relationships.

In current research, we are concentrating on concepts missing a semantic type assignment, by identifying the envelope of a group of concepts with uniform semantics, consisting of all the parents and children of the concepts of this group. We are facing the challenge of supporting group auditing by the NAT, by providing an interface for effectively reviewing concept groups from which an auditor can choose focus concepts to concentrate on.

6. Conclusions

We have introduced the Neighborhood Auditing Tool (NAT) and described the process of neighborhood-based auditing supported by it. We presented several useful kinds of neighborhoods, distinguished by different sizes and locations, that can be displayed by the NAT in a stylized manner.

NAT is a hybrid diagram/text interface that captures "the best of both worlds" in the sense of combining the advantages of diagram-based and text-based interfaces. The hybrid diagram/text interface allows for immediate visual readouts as in a diagram, without exposing the viewer to the intersecting line chaos that is often present. It does so by utilizing predefined layouts, consisting of dif-

ferent text boxes, for the various neighborhoods. Thus, the NAT brings together all the information needed in most auditing situations at one place without causing cognitive overload for the auditor. Example inconsistencies found with the use of the NAT were presented. An impact study involving a select group of auditors demonstrated the NAT's efficacy.

Acknowledgments

The NAT was developed with the help of Pratik Shah, Saurabh Singhi, Sirish Motati Reddy, Sandeep Pasuparthy, Ramya Gokanakonda, Suraj Pal Singh, Kartik Gopal, Yakup Kav, Kandarp Shah, Aditi Dekhane, Anisa Vishnani, Saurabh Patel, Rajesh Gupta and Sandeep Ramachandran.

References

[1] Aristotle. Metaphysics. Available at: http://www.greektexts.com/library/Aristotle/Metaphysics/eng/277.html [accessed December 2007].
[2] Bodenreider O. An object-oriented model for representing semantic locality in the UMLS. In: Patel VL, Rogers R, Haux R, editors, Proc MEDINFO 2001, London, UK; 200. p. 161–5.
[3] Bodenreider O. Circular hierarchical relationships in the UMLS: etiology, diagnosis, treatment, complications and prevention. In: Bakken S, editor, Proc. AMIA Symp., Washington, DC; 2001. p. 57–61.
[4] Bodenreider O. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res 2004;1(32):D267–70.
[5] Campbell KE, Oliver DE, Shortliffe EH. The Unified Medical Language System: toward a collaborative approach for solving terminologic problems. J Am Med Inform Assoc 1998;5(1):12–6.
[6] Chen Y, Gu H, Perl Y, Geller J. Structural group-based auditing of missing hierarchical relationships in UMLS. J Biomed Inform; 2008. Available online at: http://www.sciencedirect.com/science.
[7] Chen Y, Gu H, Perl Y, Geller J, Halper M. Structural group auditing of a UMLS semantic type's extent. J Biomed Inform; 2008. Available online at: http://www.sciencedirect.com/science.
[8] Chen Y, Perl Y, Geller J, Cimino JJ. Analysis of a study of the users, uses, and future agenda of the UMLS. J Am Med Inform Assoc 2007;14(2):221–31.
[9] Cimino JJ. Auditing the Unified Medical Language System with semantic methods. J Am Med Inform Assoc 1998;5(1):41–51.
[10] Cimino JJ. Battling Scylla and Charybdis: the search for redundancy and ambiguity in the 2001 UMLS Metathesaurus. In: Bakken S, editor, Proc AMIA Symp., Washington, DC; 2001. p. 120–4.
[11] Cimino JJ, Min H, Perl Y. Consistency across the hierarchies of the UMLS Semantic Network and Metathesaurus. J Biomed Inform 2003;36(6):450–61.
[12] Fung KW, Hole WT, Srinivasan S. Who is using the UMLS and how—insights from the UMLS user annual reports. In: Hersh W, editor, Proc. AMIA Symp., Washington, DC; 2006. p. 274–278.
[13] Geller J, Gu H, Perl Y, Halper M. Semantic refinement and error correction in large terminological knowledge bases. Data Knowledge Eng 2003;45(1):1–32.
[14] Gu H, Hripcsak G, Chen Y, Morrey CP, Elhanan G, Cimino JJ, Geller J, Perl Y. Evaluation of a UMLS auditing process of semantic type assignments. In: Teich JM, Suermondt J, Hripcsak G, editors, Proc. AMIA Symp., Chicago, IL, November 2007. p. 294–8.
[15] Gu H, Perl Y, Elhanan G, Min H, Zhang L, Peng Y. Auditing concept categorizations in the UMLS. Artif Intell Med 2004;31(1):29–44.
[16] Gu H, Perl Y, Geller J, Halper M, Liu L, Cimino JJ. Representing the UMLS as an object-oriented database: modeling issues and advantages. J Am Med Inform Assoc 2000;7(1):66–80.
[17] Hole WT, Srinivasin S. Discovering missed synonymy in a large concept-oriented Metathesaurus. J Am Med Inform Assoc 2000;7:354–8.
[18] Humphreys BL, Lindberg DAB, Schoolman HM, Barnett GO. The Unified Medical Language System: an informatics research collaboration. J Am Med Inform Assoc 1998;5(1):1–11.
[19] Lindberg DAB, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inform Med 1993;32:281–91.
[20] Lomax J, McCray AT. Mapping the gene ontology into the Unified Medical Language System. Comp Funct Genomics 2004;5(5):354–61.
[21] Lexical variant generation libraries. Available at: http://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/current/docs/userDoc/index.html [accessed Jan. 10, 2008].
[22] McCray AT. UMLS Semantic Network. In: Proc. 13th Annu. Symp. Comput. Appl. Med. Care, Washington, DC; 1989. p. 503–7.
[23] 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, Mekler, Westport, CT; 1993. p. 45–55.
[24] McCray AT. An upper-level ontology for the biomedical domain. Comp Funct Genom 2003;4:80–4.
[25] McCray AT, Hole WT. The scope and structure of the first version of the UMLS Semantic Network. In: Proc. 14th Annu. Symp. Comput. Appl. Med. Care, Los Alamitos, CA, November 1990. p. 126–30.
[26] McCray AT, Nelson SJ. The representation of meaning in the UMLS. Methods Inform Med 1995;34:193–201.
[27] Fact sheet Medical Subject Headings (MeSH). Available at: http://www.nlm.nih.gov/pubs/factsheets/mesh.html [accessed December 31, 2008].
[28] Min H, Perl Y, Chen Y, Halper M, Geller J, Wang Y. Auditing as part of the terminology design life cycle. J Am Med Inform Assoc 2006;13(6):676–90.
[29] Mougin F, Bodenreider O. Approaches to eliminating cycles in the UMLS Metathesaurus: naive vs. formal. In: Proc. AMIA Symp., Washington, DC, October 2005. p. 550–4.
[30] NASA software quality assurance. Available at: http://satc.gsfc.nasa.gov/assure/assurepage.html [accessed December 2007].
[31] National Cancer Institute browser. Available at: http://nciterms.nci.nih.gov/NCIBrowser/ [accessed December 2007].
[32] Nelson SJ, Tuttle MS, Cole WG, Sherertz DD, Sperzel WD, Erlbaum MS, Fuller LF, Olson NE, Fuller LF. From meaning to term: semantic locality in the UMLS Metathesaurus. In: Proc. Annu. Symp. Comput. Appl. Med. Care, Washington, DC; 1991. p. 209–13.
[33] Peng Y, Halper M, Perl Y, Geller J. Auditing the UMLS for redundant classifications. In: Kohane IS, editor, Proc. AMIA Symp., San Antonio, TX; 2002. p. 612–16.
[34] Perl Y, Chen Z, Halper M, Geller J, Zhang L, Peng Y. The cohesive metaschema: a higher-level abstraction of the UMLS Semantic Network. J Biomed Inform 2003;35(3):194–212.
[35] Protégé homepage. Available at: http://protege.stanford.edu/ [accessed January 31, 2007].
[36] Protege plugins library: visualization. Available at: http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePluginsLibraryByTopic#nid3QH% [accessed January 31, 2007].
[37] Rijsbergen CJV. Information retrieval. London: Butterworth; 1979.
[38] Rosse C, Mejino JL. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. J Biomed Inform 2003;36(6):478–500.
[39] Schulze-Kremer S, Smith B, Kumar A. Revising the UMLS Semantic Network. In: Fieschi M, Coiera E, Li YCJ, editors, Proc MEDINFO 04, San Francisco, CA; 2004. p. 1700.
[40] Schuyler PL, Hole WT, Tuttle MS, Sherertz DD. The UMLS metathesaurus: representing different views of biomedical concepts. Bull Med Libr Assoc 1993;81(2):217–22.
[41] International health terminology standards development organization—SNOMED CT. Available at: http://www.ihtsdo.org/our-standards/ [accessed December 2007].
[42] Tuttle MS, Sherertz DD, Olson NE, Erlbaum MS, Sperzel WD, Fuller LF, et al. using META-1, the first version of the UMLS Metathesaurus. In: Proc 14th Annu. Symp. Comput. Appl. Med. Care, Washington, DC; 1990. p. 131–5.
[43] UMLS documentation: 2007AA documentation. Available at: http://www.nlm.nih.gov/research/umls/umlsdoc-preface.html [accessed January 31, 2007].
[44] UMLS Knowledge Source Server (UMLSKS). Available at: http://umlsks.nlm.nih.gov/kss/ [accessed January 31, 2007].
[45] Zhang L, Halper M, Perl Y, Geller J, Cimino JJ. Relationship structures and semantic type assignments of the UMLS Enriched Semantic Network. J Am Med Inform Assoc 2005;2(6):657–66.
[46] Zhang L, Perl Y, Halper M, Geller J, Cimino JJ. An enriched Unified Medical Language System semantic network with a multiple subsumption hierarchy. J Am Med Inform Assoc 2004;11(3):195–206.