Published in final edited form as:
J Biomed Inform. 2009 June ; 42(3): 494–503. doi:10.1016/j.jbi.2009.03.005.

Practical Experience with the Maintenance and Auditing of a Large Medical Ontology

David Baorto, MD, PhD[1], **Li Li, MS**[1], and **James J. Cimino, MD**[2,3]

[1]New York Presbyterian Hospital

[2]Columbia University Department of Biomedical Informatics

[3]National Institutes of Health Clinical Center

Abstract

The Medical Entities Dictionary (MED) has served as a unified terminology at New York Presbyterian Hospital and Columbia University for more than 20 years. It was initially created to allow the clinical data from the disparate information systems (e.g., radiology, pharmacy, and multiple laboratories, etc) to be uniquely codified for storage in a single data repository, and functions as a real time terminology server for clinical applications and decision support tools. Being conceived as a knowledge base, the MED incorporates relationships among local terms, between local terms and external standards, and additional knowledge about terms in a semantic network structure. Over the past two decades, we have sought to develop methods to maintain, audit and improve the content of the MED, such that it remains true to its original design goals. This has resulted in a complex, multi-faceted process, with both manual and automated components. In this paper, we describe this process, with examples of its effectiveness. We believe that our process provides lessons for others who seek to maintain complex, concept-oriented controlled terminologies.

Keywords

terminology; Medical Entities Dictionary; maintenance; semantic network; auditing; vocabulary; ontology

Introduction

When medical centers create central clinical data repositories, they generally find a need for a central controlled terminology by which to code data from disparate sources (such as test results from laboratory systems and medication orders from pharmacy systems). Although the mapping of local terms to a standard terminology might offer advantages, this option has not been practical due in part to the lack of a satisfactory single standard and due in part to the lack of standards adoptions by the local data sources. Instead, developers have found the need to create a unified terminology, consisting of the merger of local terminologies, similar to the approach taken by the National Library of Medicine to unify standard terminologies

Mailing Address: David Baorto, 622 West 168th Street, VC-5, New York, NY 10032, baorto@dbmi.columbia.edu, Fax: 212-305-3302.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

into a Unified Medical Language System (UMLS) [1]. Early examples of this approach include the development of the Directory in the Computer-Stored Ambulatory Record system (COSTAR) [2] and PTXT in the HELP system [3].

Over time, some of these terminologies have evolved into ontologies, as their content has expanded to include biomedical knowledge, application knowledge, and terminologic knowledge. Notable examples include the Vocabulary Server (VOSER) used in 3M's Health Data Dictionary [4] and the Vanderbilt Externalized General Extensions Table (VEGETABLE) [5]. The expansion and maintenance of these terminologies requires significant effort on the part of the developers, with constant vigilance towards continued maintenance of terminology quality [6]. The terminologic knowledge they contain adds to the burden of keeping the content accurate, but also provides some support for the task in the form of knowledge-based terminology maintenance.

The original plan for the clinical information system being constructed at Columbia University and the New York Presbyterian Hospital (NYPH, formerly Presbyterian Hospital) in 1988 required that a single coding system be used to encode data acquired from multiple sources, for storage in a single, coherent data repository.[7] The data sources did not use the same (or often, any) standard terminology, but no single standard terminology existed to which the source terms could be mapped. Rather than attempting to create a comprehensive controlled terminology ourselves, we sought to create a "local UMLS" that brought together the disparate controlled terminologies used by source systems into a single conceptual dictionary of medical entities that could serve as that comprehensive terminology. From the beginning, this Medical Entities Dictionary (MED) was conceived as a terminologic knowledge base that could be used to support its own maintenance and auditing [8]. As such, it has proven to be a fertile substrate for terminologic research by ourselves [9] and others [10], [11]. However, the MED supports a number of important day-to-day patient care, educational, research and administrative operational activities at NYPH and Columbia [12]. Thus, the auditing of its content, like similar efforts at other medical centers, is more than an academic exercise.

The process by which we maintain the MED, including auditing for continual quality monitoring, has evolved over the past two decades as a result of concerted informatics research into the application of good terminology principles, together with intensive analysis of data sources and their terminologies. The purpose of this paper is to describe the requirements that shaped the maintenance process, and to describe that process itself (with special attention to auditing and error detection) that has resulted from those requirements.

Requirements

Terminology model

The MED was designed along the lines of the UMLS: when a term from a terminology was added to the MED, it was to be mapped to an existing concept identifier (MED Code) if an appropriate one already existed in the MED. If not, a new MED Code would be created to accommodate the term. Like the UMLS Concept Unique Identifiers (CUIs), MED Codes could correspond to multiple terms from multiple terminologies.

There were, however, several important differences. First, there was no assumption that different terminologies would necessarily contain terms that were synonymous across the terminologies (that is, terms mapping to the same MED Code). In fact, the opposite was generally considered to be the case. For example, if two laboratory systems included terms for a serum glucose test, these were considered to refer to distinct entities in reality, and therefore were given unique MED Codes. Their similarity was instead captured by making each concept a child of a MED class called “Serum Glucose Test”.[9]

A second departure from the UMLS model was to attempt to include in the MED formal definitional information about each term, to the extent possible and practical, expressed through semantic relationships between MED concepts. For example, each laboratory test concept was to be related to appropriate MED concepts through “Substance Measured” and “Has Specimen” relationships, while each medication concept was to be related to appropriate MED concepts through “Has Drug Form” and “Has Pharmaceutic Component” relationships.

Other ways in which the MED approach differed from the UMLS included the organization of all concepts into a single directed acyclic graph of “Is A” relationships (with “Medical Entity” as the sole top node), the assignment of unique preferred names for each MED Concept that attempted to convey the meaning of the concept (as opposed to the sometimes-telegraphic names from source terminologies), and the introduction of new concept attributes (including the potential for semantic relationships) at single points in the “Is A” hierarchy. As the MED developed, auditing methods were needed to assure adherence to all of these requirements.

Sources

As the clinical information system at Columbia grew to include new data sources, the MED needed to incorporate the relevant terminologies. Initial sources included the laboratory, radiology, pathology and billing systems. Later sources included many other systems in ancillary departments of the medical center. For the most part, systems had their own local terminologies (or set of terminologies) that were maintained in a variety of *ad hoc* ways, in disparate systems and formats. Applications that were developed as part of the clinical information system (such as clinician documentation and laboratory summary reporting) often had their own terminologies as well. As systems and applications were replaced, their successors often came with new terminologies that had to be added to the MED, while retaining the retired terminologies to allow proper interpretation of historical patient data.

National and international standard terminologies were not initially included in the MED, since they were not used by source systems. Over time, however, some adoption of standards began, adding to the terminology requirements of the MED.

Finally, we found that we often needed to add our own terms to the MED to support the knowledge representation requirements. Such knowledge included classification terms (such as the “Serum Glucose Test” class) and terms needed to support definitions (such as “Digoxin”, to allow the proper representation of terms such as “Serum Digoxin Test” and “Digoxin 0.25mg Tablet”).

Publishing the MED

The complex requirements for developing and maintaining MED content precluded the simple approach of including the MED in the clinical information system database and editing it in that environment. Instead, we needed a more flexible, dynamic environment for editing, which led to the added requirement for publishing the MED in a way that made it available to the clinical information system and other systems as well. As this system evolved into a Web-based architecture, the need to distribute the MED to additional environments increased further.

Originally, the MED was maintained in a PC-based LISP environment, using commercial knowledge representation software. A simple table-based representation was exported that could be incorporated into the database of the clinical information system. When the MED outgrew this environment, we moved to a mainframe-based version of the product but soon the MED outgrew that as well, with a deterioration in performance. We then developed a “temporary” MUMPS-based solution that was used for over ten years as we worked to develop tools more appropriate to a modern, distributed, Unix-based environment. Although these transitions were disruptive to the maintenance processes, the same export mechanism was used by each version, so that the clinical information system continued to function without interruption.

Solutions

Some of the requirements described above were determined at the outset of the MED development.[8] However, many other requirements were established over the ensuing years, sometimes by natural evolution, sometimes by trial and error. With each new requirement came a need to develop maintenance methods that would assure adherence to that requirement. The result has been a collection of techniques. Some are automated, while others are manual; some are general purpose, while others are specific to a particular source terminology; and some are executed at the time of terminology updates (“instant audits”) while others are applied retrospectively.

Structure

Regardless of the representational form (LISP, MUMPS, relational, etc.), the MED is conceptually a frame-based model, with string attributes and semantic relationships, represented by slots. Slots in the MED are sequential numerical attributes that hold values for concepts. Strings are held in string-valued slots, such as LAB-TEST-LONG-NAME and CERNER-FORMULARY-CODE, while semantic relationships are represented with reciprocal pairs of slots, for example, ENTITY-MEASURED and MEASURED-BY-PROCEDURE, that take MED Codes as values.

Slots are introduced at a single, appropriate point (“fathered”) at any level within the hierarchy. For example, slot 61 “DRUG-TRADE-NAME” is fathered at concept 28103 – PHARMACY ITEMS, meaning that only descendants of MED Code 28103 can have slot 61 values. The MED slots, their names and characteristics are modified by a slot editor program that modifies the slot definition file. The slot definition file identifies a number of characteristics of slots including the type (long_string, semantic, synonym, etc), the MED Code where it is fathered, and the slot number for its reciprocal slot (for the semantic slots).

Editing

The MED editing application consists of a browser-based interface to a suite of locally developed Common Gateway Interface (CGI) programs written in C, comprising the MED viewer, MED batch editor, MED manual editor, and MEDchecker. These programs are responsible for providing a user interface, for processing batch edit files, and for calculating inheritance and slot value refinement. They read and modify a series of structured text files that hold the MED content during the editing phase. The text files include the pre-edit cycle MED master file, the modified MED file, the slot definition master file, and various log files. The primary access to MED content by applications occurs by a locally developed shared memory implementation written in C. The MED editing environment resides on a Unix server using the IBM AIX operating system, and the shared memory implementation is disseminated to multiple Unix servers running either AIX or the Sun Solaris operating system.

Editing the underlying MED files occurs via the final endpoint of creating a batch file that is run through the batch editor to modify the underlying MED master files. Lines in the batch file begin with one of several commands, “+” for adding a new MED Code, “ASV” for adding a slot value, “RSV” for removing a slot value, “REPLACE” for replacing a slot value, and “RENAME” for renaming a MED concept. For example, the line “ASV|69467|7|35495” would instruct the batch editor to add the MED concept 35495 (“CPMC Laboratory Test: Amphotericin B”) to slot 7 (“HAS-PARTS” slot) of MED concept 69467 (“CPMC Battery: Fungal Susceptibility”). The line “RSV|61690|211|On Formulary” is the command to remove the value “On Formulary” from slot 211 (“DRUG-IN-CERNER-FORMULARY” slot) of MED Code 61690 (“Cerner Drug: Aluminum Hyd Gel Chew Tab 600 Mg”).

All MED changes, including changes to the hierarchy, can be effected using this command set. We refer to the batch files containing these commands as “asvrsv files”. The MED editing environment also supports single changes in a frame-based graphical interface (Figure 1). The graphical editor is seldom used in practice because making individual changes one-by-one is cumbersome. However, the visually related graphical viewer is used extensively by MED editors and by external users of the terminology to review MED content.

Terminology design considerations

The first step for additions to a terminology is a thoughtful design process. In the MED, each term from an external terminology usually becomes a unique MED Concept, related to similar terms by the hierarchical structure. Knowledge about terms can be represented in a number of ways, either as string slots, semantic relationships to preexisting or new MED terms, or as hierarchical relationships. The model will ideally be chosen to best support downstream users of the MED, and sometimes involves considerable planning.

One example of design choice is the method by which the MED represents information used by data display applications. For example, laboratory display spreadsheets appear in the clinical information system as clinically–related aggregates of test results that can be built to any specifications and are generated in real time from on-the-fly MED queries. The spreadsheets are modeled as individual MED concepts, with each column represented as a semantic relationship (called “HAS-DISPLAY-PARAMETERS”) to a test class whose descendants are the individual local test concepts that are displayed in the column. This allows spreadsheets to be built quickly within the MED terminology.[13]

Most additions to the terminology involve some question about the best structural representation, with the modeling decisions frequently being choices between incorporating knowledge as string attributes, semantic relationships, hierarchical relationships, or a combinatorial approach. The efficiency of service to downstream applications is often a primary concern. Design considerations are usually not made specifically with auditing as the objective. However, the design has implications for auditing as well.

Personnel

The personnel managing the MED content have extensive experience in clinical medicine and informatics. Both are physicians. One has a PhD in pathology, residency training in laboratory medicine, fellowship training in medical informatics, and ten years experience in clinical terminology; the other has masters degrees in computer science and medical informatics.

Terminology Maintenance

The solutions described above set the stage for the establishment and growth of the MED. Specifically, the decision to add a data source to the clinical information system requires a corresponding determination of the controlled terminologies that are needed to represent the data. This determination, in turn, triggers a careful analysis to determine if the terminology already exists in the MED (unusual, unless a standard terminology is involved), if the new terminology closely relates to concepts already in the MED (the usual case when a new system is replacing one previously represented in the MED), or if the new terminology represents an entirely new concept domain for the MED (the usual case when a new type of data source is being added).

The modeling process is followed by a one-time update process in which the new terminology is added *en masse* to the MED as an asvrsv file. Update mechanisms are then established, to apply changes to the MED as source terminologies change. Auditing processes are established to monitor changes and prevent inconsistencies from being introduced or to simply report them when detected.

Local terminology sources

The first source system we addressed was a home-grown clinical laboratory system. We obtained the terminology from that system as a simple listing of names and codes. Updates to the terminology were infrequent, and addressed through ad hoc e-mail messages describing the changes. When the laboratory system was replaced by a commercial system, complete with an entirely new controlled terminology, we realized that we needed to develop more automated methods, especially for coping with more frequent changes.[14]

As we went through a similar experience with a new pharmacy system, we found that obtaining terminology updates from commercial systems was often difficult or impossible. A more viable approach involved obtaining entire copies of current terminologies, and then making our own comparisons to prior copies in order to determine the interim changes.[14] In allusion to the unix diff function, we refer to this as the “diff” approach.

When using the diff approach, the specifications, fields and file formats were determined at the time the terminologies were first incorporated into the MED. These source systems run regularly scheduled scripts that create extract files reflecting the current state of the source terminology dictionary for all the pre-defined fields. The scripts then send these files, via an automated secure file transfer protocol, to the main terminology server for use by the MED maintenance team.

Some local source systems do not even have the capacity to generate sufficient automated terminology extracts. In these cases, we have made specific arrangements for the system owners to gather the information at regular intervals (usually weekly) and create an interval change file. We presently use this method to obtain terminology updates for one of our clinical laboratory systems, our radiology system, and a clinical documentation system.

Standard terminologies

The MED also includes national and international standard terminologies, either because they are used by some source system or because they provide some added value to the MED – for example, the ability to translate local data into standard coded form. These are obtained at intervals corresponding to official releases. While some releases include specific information about changes, we often must identify the changes ourselves.

Terminology sources for national and international standards are composed of files from the Centers for Medicare and Medicaid Services or the Centers for Disease Control Websites (i.e., the International Classification of Diseases, 9th Edition, Clinical Modifications, or ICD-9-CM), licensed files from American Medical Association (Current Procedural Terminology – Fourth Edition, or CPT4), or Logical Observation Identifier Names and Codes (LOINC) files and the Regenstrief LOINC Mapping Assistant (RELMA) application from the LOINC Website. In each case, we use the “diff” approach (described above) to determine interim additions, deletions, and changes in the terminology.

Because the MED contains large sets of terms for diagnoses and procedures, ICD-9-CM and CPT4 codes are added as attributes of existing concepts. Where needed, new concepts are added to the MED to accommodate these codes [14]. When codes are retired or reassigned to other concepts (that is, the meaning of the code changes) the previous code assignments are moved to slots that hold old terminology-specific codes, along with dates for the changes. These slots add an historical perspective to changes in the standards, but are not strict versioning. This process is described in detail in [14].

Because of the potentially large combinatorial nature of LOINC terms, the MED does not model the entire LOINC database, but incorporates mapping to LOINC only of those codes that are represented by local terms. Therefore, the entire LOINC database is downloaded only as an informational and mapping resource, but not incorporated into the MED in its entirety.

Maintenance processes

The regular updates to terminologies in the MED that require frequent and intensive updating are accomplished using a series of interactive scripts that read the regularly–obtained source system extract files described previously and create the appropriate asvrsv files for the batch editor. The tasks of determining primary parent, creating display names of various lengths, and adding attributes to appropriate slots are generally fully automated aspects of the scripts. The scripts also attempt to create canonical names for new MED concepts, which are fully specified names that distinguish, in a meaningful way, each MED concept from all others. For example, one of the test terms from one of the clinical laboratory systems is called “Copper”; the canonical MED name is “NYH LAB TEST: COPPER, URINE CONCENTRATION”, which identifies both the clinical entity urine copper concentration, and also the campus where the test is performed (that is, NYH or New York Hospital).

The scripts also include steps that require intervention by a terminology content expert. These steps are presented to the expert as lists of suggestions for semantic and hierarchical additions, including placement of terms in classes, guiding creation of new classes, and assignment of values to semantic slots (e.g., substances measured by tests, chemicals contained in formulary items, clinical displays in which newly created classes should appear, etc.). The lists of suggestions are generated by a text based search for semantically relevant MED concepts.

Auditing Source Terminologies

Table 1 lists many of the terminology sources that are presently modeled and maintained in the MED, including local source terminologies, standard terminologies, and application parameters (such as those used to construct laboratory result displays). The process of updating the MED to reflect changes in these terminologies can detect errors and inconsistencies that originate in the source terminologies themselves.

Auditing local source terminologies prior to addition to the MED

For the local terminology source systems, such as the laboratory information system, the pharmacy system, and the radiology system, auditing the source at the time of acquisition allows us to provide feedback so that changes can be made to those systems. Over the years, many errors in source terminologies have been detected at the point of acquisition, including changes in the meaning of existing codes, creation of redundant terms, lexical errors, and even the presence of non-printing characters in the source system master files.

In one case, for one example, we noted that one of our laboratory systems changed the name of a laboratory test from “HIV 1” to “HIV 1/2”, suggesting a change in the substance measured by the test (and therefore the actual meaning of the test code). In another example, we recently detected that one of our local laboratory systems attempted to add new codes for specimen terms that already existed under different codes (e.g., “Bronchial Lavage”). The attempt to add this term to the MED resulted in the removal of the duplicates from the source system. The errors in both of these cases were detected through manual inspection of the “diff” files.

Auditing standard terminologies prior to addition to the MED

We encounter true errors in standard terminologies very infrequently. However, in some cases, changes that occur in a standard terminology lead to incompatibilities with our concept-oriented modeling of the standard terms. As we have previously described,[14] we detect these changes through manual review of “diff” files to identify situations where changes to term names change the meaning of their corresponding codes, while additions or deletions to the terminology may affect the implied meaning of “other” codes. For example, when ICD-9-CM added the code 530.85 “Barrett’s Esophagus” in October, 2003, the meaning of 530.89 “Other Specified Disorders of Esophagus” suddenly excluded “Barrett’s Esophagus”. There is no mechanism to provide feedback to the maintainers of ICD-9-CM; indeed, it is not clear that they even recognize this type of semantic drift as a problem. However, in order to adhere to our concept-oriented approach to terminology representation, we must take somewhat extraordinary steps to accommodate such changes [6, 14, 15].

Auditing source terminologies during addition to the MED

Many of the audits that are implemented for inbound terminologies are inextricably tied to the regular editing and update process of the MED. The attributes of inbound concepts are automatically compared to the attributes of concepts that already exist in the MED. When inconsistencies are noted, they are presented to the MED content manager for manual review.

One type of audit specifically looks for cases where an ancillary department has reused a code that has been used in the past for a concept with a different meaning. The automated scripts will flag these cases because semantic relationships or attributes in the MED will suddenly no longer match. For example, the script that processes the update files from one of our local laboratories flagged an “illegal change in hierarchy” for an existing laboratory concept because the laboratory was attempting to use the same code to represent an orderable procedure for “Methamphetamine and Metabolite” that was previously used to identify a non-orderable result component (“Methamphetamine”) of another procedure.

Auditing source terminologies after addition to the MED

The act of supporting systems downstream of the MED often requires the addition of knowledge and structure from other origins to supplement the source terminology. This modeling process often yields additional opportunities for auditing of the source terminology. This outside knowledge may be needed for functions such as laboratory results displays (as discussed above) and infection control [14] (which would require both coded results “Pasteurella bettyae” and “CDC GroupHB5” to point to the same organism)[16], and links to external knowledge resources (through applications called “infobuttons”)[17]. Some examples of such additional knowledge are included in Table 1B.

The act of seeking knowledge from external sources often provides a default cross-check of the terminology in local sources. In fact, errors are often inferred from lack of concordance between multiple terminology sources. For example, when pharmacy input files contained the new drugs, “Treandra” and “Vimflunine” and we were unable to find these in alternative information sources that we use to classify drugs in the MED, our feedback to the pharmacy corrected the misspellings in their system (to “Treanda” and “Vinflunine”, respectively). In another example, we were able to detect discrepancies between the allergy codes explicitly assigned to drug terms in a pharmacy system with the allergies that the MED hierarchy implied for those terms, resulting in hundreds of corrections to the pharmacy system’s knowledge base[18].

Auditing the MED

As described above, the processes by which terms from source terminologies are added to the MED can identify errors and inconsistencies related to the source terms. However, once the terms are added to the MED, they become part of a bigger whole. This larger perspective requires different methods for identifying more global problems, such as inconsistencies between multiple terminologies. These methods depend upon the knowledge used to model terms in the MED – often, knowledge that goes beyond that which is supplied with the source terminologies, such as hierarchical information and semantic relationships.

Automated and semiautomated knowledge-based additions

Much of the detailed modeling that occurs in the MED is either partially automated followed by manual review or completely interactive. Having a controlled process based on expert review or existing content significantly reduces the chance for error. Semiautomated, interactive processes are regularly used to update source terminologies that are modeled in the MED.

For example, once terms are added to the MED, they often require additional classification to make them consistent with previously existing terms. In the example below, an interactive script has generated a list of possible classes in which to place the new test “LYME TOTAL ANTIBODY, SERUM”. Note that the choices are generated from similarity to the name of the test, but with the additional criteria that the listed concepts be in the class “Laboratory Test”, so the disease entity “Lyme Disease” (not an appropriate choice for a test class) will not be displayed (Figure 2). This type of knowledge-based process reduces the likelihood of errors, since the user will not be presented with a choice from an incorrect hierarchy in which to classify the test.

Based on choosing the second option in Figure 2, the script will make the new MED Code a child of MED Code 46736, causing it to inherit all the semantic relationships of its new parent, such as ENTITY-MEASURED: “32338 - Lyme Antibody”, and IS-DISPLAY-PARAMETER-OF: “46679 - Lyme Disease Display”. Figure 3 demonstrates that the classification of the new concept results in the inheritance of the ENTITY-MEASURED relationship. The figure also demonstrates slot refinement for MED Code 48970, whereby the more specific concept “Lyme IgM Antibody” takes precedence over the inherited value “Lyme Antibody”.

Automated addition methods are currently used for the two laboratory systems and one pharmacy system. Figure 4 shows an example of the addition of a pharmacy concept. The formulary file contained a new drug “Etravirine 100 mg Tablets”. Completely automated processes are indicated in blue. They are used to add the concept itself as well as certain attributes and hierarchical relationships (also indicated in blue). Red components in the diagram represent those that are added through semi-automated, interactive steps. These include the preparation classes, links to substance concepts (and the addition of those concepts when necessary), semantic links from preparation classes to substance concepts (which are inherited) and the synonyms. The dotted blue line indicates a hierarchical relationship that was initially built by the automated script, but pruned after the alternate pathway from “Antivirals” to “Cerner Drug: Etravirine Tab 100 mg” was created by the intervening Etravirine tree.

In some respects, a terminology with ontologic aspects such as the MED can be considered as “introspective”, using internal knowledge to support its maintenance [8]. Some of the basic rules built into the MED editing environment, such as the automated inheritance of semantic relationships by all descendants, themselves result in a knowledge-based addition of content. A local laboratory test called “CPMC Laboratory Test: Anti-Cardiolipin IgM Antibody”, when made a child of the test class “Serum Anticardiolipin IgM Antibody Tests”, automatically inherits “Anti-Cardiolipin Antibody” as ENTITY-MEASURED and the specific displays where this tests is to be appear in clinical systems. Although string attributes are not inherited as a rule, they are often filled using a knowledge-based algorithm. An example is the propagation of Infobutton links to slot values of MED Codes based on hierarchical relationships [17].

Knowledge-based approaches also are used to create hierarchical relationships based on string attributes or semantic relationships (often referred to as *automated subsumption*), and can also be used to confirm and audit classification choices. For an example, based on string attributes, all drug preparations in the MED are maintained under multiple hierarchies, including one hierarchy based on the American Hospital Formulary Service (AHFS) classes and a second hierarchy based on Drug Enforcement Agency (DEA) classes. When new formulary items are added to the MED, the codes in the source system file are parsed and used to automatically place drugs in the correct class in each of these hierarchies.

Semantic relationships are also used in the MED to drive the creation of classes and placement within classes. An example is the use of an automated classification algorithm to build laboratory test classes based on hierarchies of semantically related SUBSTANCES-MEASURED values [19] or to infer allergy classes of drugs based on their PHARMACEUTIC-COMPONENT [18]. Using the knowledge that exists in the MED introspectively to build and edit content serves as a powerful auditing mechanism because 1) small areas of existing MED knowledge are frequently presented to users for review and 2) the addition process is controlled and guided by the existing content.

Automated and semiautomated knowledge-based auditing

One of the principal reasons for including terminologic knowledge in the MED has been to exploit it for maintenance and auditing purposes.[8] Certain characteristics of the MED’s design can be represented by explicit rules that can be entirely automated (e.g., no two MED concepts may have the same name, no cycles are allowed in the is-a hierarchy, etc.). In other cases, we can only apply heuristics that suggest areas where problems may exist (e.g., if a term name has changed, perhaps its meaning has changed as well). Although the results of heuristic auditing methods require human judgment, they can at least focus that judgment on areas where the likelihood of errors is relatively high.

Auditing subclassification—One type of knowledge-based auditing uses a sub-classification heuristic to find inconsistencies in the class structure for clinical documents in the MED, and to build a complete hierarchy of document type classes. In the MED, clinical document types from ancillary systems are included in a structured document ontology based on the four LOINC document axes of subject matter (what the note is about, e.g. cardiology), setting (where the note is written, e.g., ICU), role (what caregiver wrote the note, e.g., attending physician), and type of service (what service the note provides, e.g., consult).

In the MED each of these four axes is represented as a hierarchy. Individual clinical documents types from ancillary systems then are given semantic links to the appropriate concept in each axis. For example the document concept “Eclipsys East Campus Document: Neurology Resident Consult Note” would have the following semantic slot values, Document-Has-Subject-Matter -> Neurology, Document-Has-Role -> Resident, and Document-Has-Type-Of-Service -> Consult. In this case there would be no link to ‘setting’ since it is not specified.

The semantic relationships are used to find missing document classes, i.e., combinations of the four axes for which no document class yet exists in the MED. The audit builds the missing class into the hierarchy, and moves the appropriate individual document codes under the new class. In this case the document class “Neurology Resident Consult Note” would then be created, and all the applicable individual document concepts from the various clinical systems would be subsumed by this new class concept.

Detection of redundancy—Pharmacy concepts being added to the MED are audited to find redundancy by comparing slots values for formulary name, generic name, drug form, route of administration, dose strength, and dose units. If all these values are identical for two distinct formulary codes, then they are flagged for manual review as possible redundant formulary concepts.

Automated cross-mapping between terminologies—The MED contains terminologies from major clinical laboratories on each of the two NYPH campuses. Each of these terminologies has many order terms (i.e., ‘batteries” or ‘panels’) as well as the individual test terms associated with each order. For example, both laboratories have similar orders for electrolyte panels, which include individual component tests such as sodium, potassium, etc. Cross-mapping between these terminologies is desirable for several reasons, including the need to have consistent order sets for the physician order entry systems on both campuses and, more recently, the installation of a single, bi-campus laboratory system.

In the MED, the test terms from multiple laboratories are modeled together under a single classification hierarchy. This common classification facilitates the cross mapping; for example, the serum sodium test terms from both campuses share the parent “Serum Sodium Test”, which in turn suggests that they can be cross-mapped. The audit algorithm compares the component results of each order from both campus laboratories. By taking advantage of a common class in the MED, the function finds the best candidates for equivalent or best-matched orders between the two laboratories terminologies.

The MEDchecker—Although the MED embodies design principles that apply to all of its concepts, there are many cases where domain-specific requirements arise, related to the types of terms, their source, or the applications in which they are used. General auditing methods may be inappropriate or insufficient in such cases, requiring domain-specific auditing methods. MED knowledge may still be useful for these situations, but specific, ad hoc programs are often needed to use this knowledge appropriately. We have assembled these ad hoc programs into a single program, called the MEDchecker, that can be used to audit the entire MED, applying domain-specific methods where appropriate.

The MEDchecker is run after the completion of each editing cycle, just before the changes are committed to production systems. The audits performed by MEDchecker probe deeper into structural changes than those performed by the editor, such as the multiple concepts with the same preferred name, introduction of illegal hierarchical cycles and redundant relationships between concepts, presence of slots in concepts that are not descendants of the fathers of those slots, inappropriate slot values, or missing slot values that should otherwise be present through inheritance.

Altogether, there are 31 distinct error messages that can be produced by the MEDchecker. Some are simple housekeeping checks, such as the detection of duplicate preferred names. Others, however (listed in Table 2), examine the slot values (including hierarchical and reciprocal relationships) to identify logical inconsistencies. Error 18, the classic hierarchical cycle, has rarely, if ever, been seen. Error 17 commonly appears, basically representing a redundant hierarchical relationship. Errors 12 and 13, also occasionally seen, are two error types that would prevent slot refinement from occurring. Slot refinement is the property of semantic inheritance by which the most highly specified value prevails (Figure 3). Error 12 indicates that there are 2 hierarchically related values explicitly instantiated, which would violate refinement. Error 13 indicates that a MED concept has an ancestor with a more refined slot value than it has, again violating refinement.

Discussion

The task of maintaining a central controlled terminology for a system that aggregates clinical data from multiple sources is a labor-intensive process. We recently increased our staff from one full time person to two, while other institutions have even larger devotion of personnel to the task. The degree to which computer systems can assist in the creation, addition and updating of the terminology content, and the degree to which computer systems can detect errors and inconsistencies, depends in part on the identification of simple, well-defined repetitive tasks, for which such systems are well-suited. More sophisticated tasks, such as identifying appropriate term classification or creation of appropriate (nonhierarchical) inter-term relationships, typically require a domain expert or knowledge engineer. However, the knowledge content of the terminology can be brought to bear on this problem to be used by algorithms, or perhaps heuristics, to assist those experts or perhaps even work autonomously.

The maintenance and auditing of the Columbia University Medical Entities Dictionary is a “mission critical” task that cannot be tolerant of errors. If a laboratory test term is not in the correct class, it will not appear in a results display spreadsheet; if a medication term is not in an appropriate class, it will fail to be considered by an alerting system. With over 100,000 terms in the MED and over 1,000,000 attributes and values, manual detection of errors is simply not possible. While we can never be sure that the MED is completely correct, we feel confident that our methods are detecting and eliminating a large percentage of potential errors.

When the MED was originally conceived, we thought that a principled, knowledge-based design would allow us to apply state-of-the-art techniques from object oriented technology and artificially intelligent tools to drive the creation, maintenance and auditing of the MED content.[8] However, in practice this approach was impractical for wholesale terminology maintenance. For example, stating that protein is measured by a urine protein seems like a simple statement of truth, but (because insulin is a protein) yields the somewhat farcical inference that one could measure insulin with a urine protein test. In reality, we needed to strike a balance between the application and relaxation of principled techniques.[13]

The actual implementation of our various methods are specific to our institution and setting, but we believe that the description we provide here should be helpful to those who seek to carry out similar tasks at their own institutions, with their own terminologies. The operationalization of the methods should be fairly straightforward; thus, the effort to make our tools “open source” and for others to adapt them to their own environments is likely to be more effort than simply creating tools that employ our methods.

Success of the methods is most dependent on the principled structure of the terminology and the quality of its content. We are fortunate today that many controlled terminologies in health care have been created as, or are evolving towards, high-quality ontologies [20] to which methods such as ours can be (and often already are) applied.

Acknowledgments

Dr. Cimino is supported by intramural research funds from the NIH Clinical Center and the National Library of Medicine.

References

1. Lindberg DA, Humphreys BL, McCray AT. The Unified Medical Language System. Methods Inf Med. 1993 Aug; 32(4):281–91. [PubMed: 8412823]
2. Barnett GO, Winickoff R, Dorsey JL, Morgan MM, Lurie RS. Quality assurance through automated monitoring and concurrent feedback using a computer-based medical information system. Med Care. 1978 Nov; 16(11):962–70. [PubMed: 101723]
3. Pryor TA, Gardner RM, Clayton PD, Warner HR. The HELP system. J Med Syst. 1983 Apr; 7(2): 87–102. [PubMed: 6688267]
4. Rocha RA, Huff SM, Haug PJ, Warner HR. Designing a controlled medical vocabulary server: the VOSER project. Comput Biomed Res. 1994 Dec; 27(6):472–507. [PubMed: 7895474]
5. Stead WW, Miller RA, Musen MA, Hersh WR. Integration and beyond: linking information from disparate sources and into workflow. J Am Med Inform Assoc. 2000 Mar-Apr;7(2):135–45. [PubMed: 10730596]
6. Cimino JJ. Letter to the Editor: An approach to coping with the annual changes in ICD9-CM. Methods of Information in Medicine. 1996; 35(3):220. [PubMed: 8952307]
7. Hripcsak G, Cimino JJ, Johnson SB, Clayton PD. The Columbia-Presbyterian Medical Center decision-support system as a model for implementing the Arden Syntax. Proc Annu Symp Comput Appl Med Care. 1991:248–52. [PubMed: 1807598]
8. Cimino, JJ.; Hripcsak, G.; Johnson, SB.; Clayton, PD. In: Kingsland, LW., editor. Designing an Introspective, multipurpose controlled medical vocabulary; Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care; Washington, D.C.. November, 1989; p. 513-518.
9. Cimino JJ, Clayton PD, Hripcsak G, Johnson SB. Knowledge-based approaches to the maintenance of a large controlled medical terminology. J Am Med Inform Assoc. 1994 Jan-Feb;1(1):35–50. [PubMed: 7719786]
10. Kannry JL, Wright L, Shifman M, Silverstein S, Miller PL. Portability issues for a structured clinical vocabulary: mapping from Yale to the Columbia medical entities dictionary. J Am Med Inform Assoc. 1996 Jan-Feb;3(1):66–78. [PubMed: 8750391]
11. Gu H, Halper M, Geller J, Perl Y. Benefits of an object-oriented database representation for controlled medical terminologies. J Am Med Inform Assoc. 1999 Jul-Aug;6(4):283–303. [PubMed: 10428002]
12. Cimino JJ. From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. J Am Med Inform Assoc. 2000 May-Jun;7(3):288–97. [PubMed: 10833166]
13. Cimino JJ. In defense of the desiderata. Journal of Biomedical Informatics. 2006; 39(3):299–306. [PubMed: 16386470]
14. Cimino JJ. Formal descriptions and adaptive mechanisms for changes in controlled medical vocabularies. Methods Inf Med. 1996 Sep; 35(3):202–10. [PubMed: 8952304]

15. Tuttle MS, Nelson SJ. A poor precedent. Methods Inf Med. 1996 Sep; 35(3):211–7. [PubMed: 8952305]

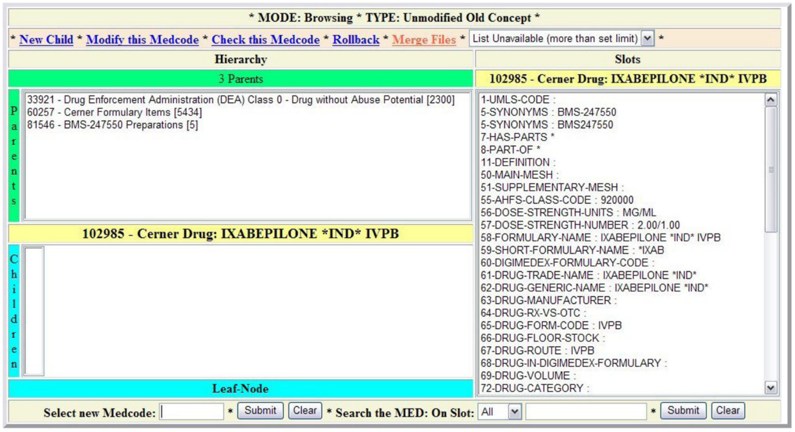
16. Haas JP, Mendonça EA, Ross B, Friedman C, Larson E. Use of computerized surveillance to detect nosocomial pneumonia in neonatal intensive care unit patients. Am J Infect Control. 2005 Oct; 33(8):439–43. [PubMed: 16216656]

17. Cimino JJ, Elhanan G, Zeng Q. Supporting infobuttons with terminological knowledge. Proc AMIA Annu Fall Symp. 1997:528–32. [PubMed: 9357682]

18. Cimino JJ, Johnson SB, Hripcsak G, Hill CL, Clayton PD. Managing vocabulary for a centralized clinical system. Medinfo. 1995; 8 Pt 1:117–20. [PubMed: 8591133]

19. Cimino, JJ.; Hricsak, G.; Johnson, SB.; Friedman, C.; Fink, DJ. In: Clayton, PD., editor. Clayton. Representation of Clinical Laboratory Terminology in the Unified Medical Language System; Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care; Washington, D.C.. November, 1991; p. 199-203.

20. Cimino JJ, Zhu X. The practical impact of ontologies on biomedical informatics. IMIA Yearbook of Medical Informtics. 2006:124–35.



* MODE: Browsing * TYPE: Unmodified Old Concept *
* New Child * Modify this Medcode * Check this Medcode * Rollback * Merge Files * List Unavailable (more than set limit) *
Hierarchy
Slots
3 Parents
Parents
Children
Leaf-Node
Select new Medcode:
Submit
Clear
* Search the MED: On Slot: All
Submit
Clear

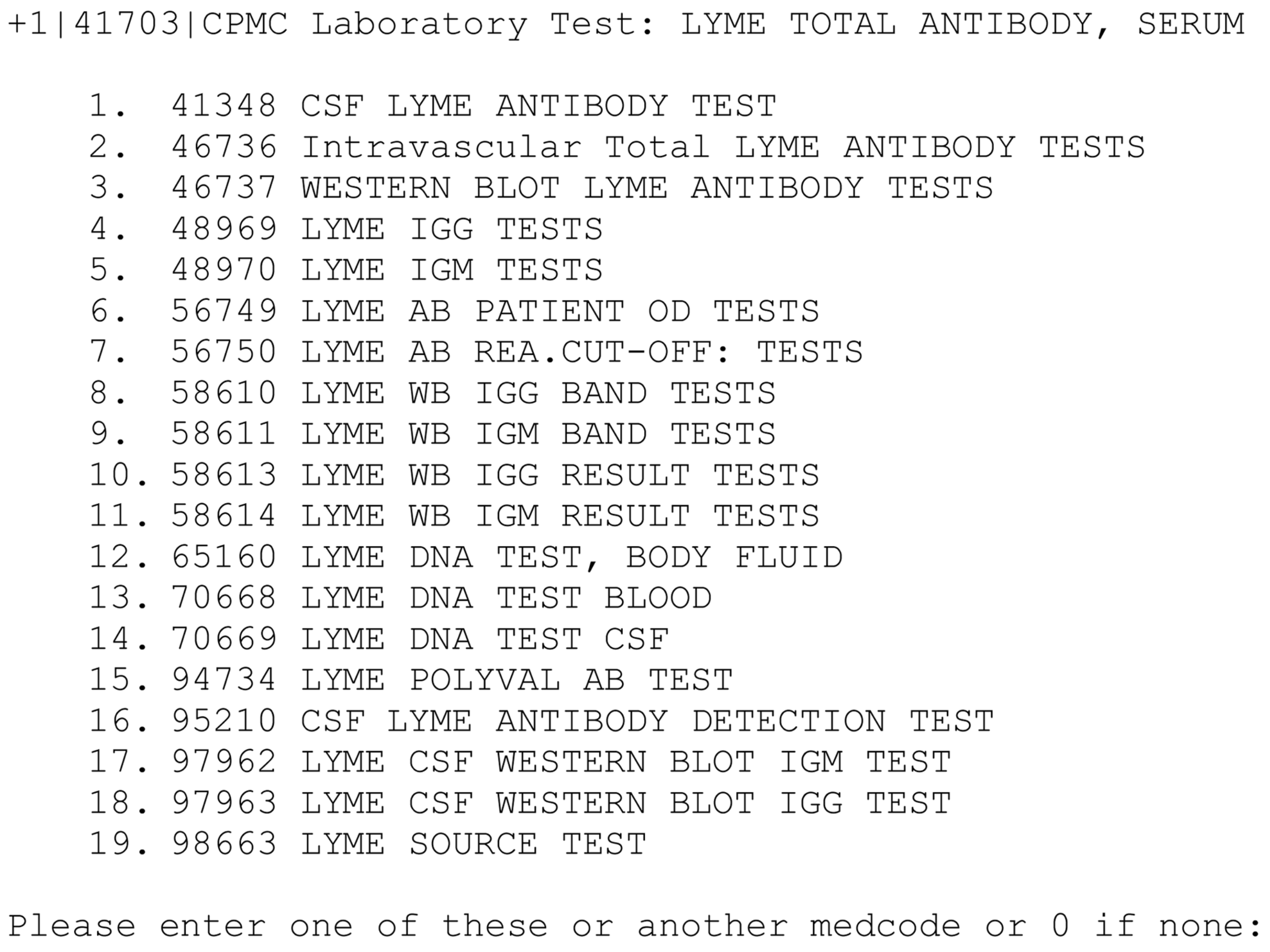
102985 - Cerner Drug: IXABEPILONE *IND* IVPB

33921 - Drug Enforcement Administration (DEA) Class 0 - Drug without Abuse Potential [2300]
60257 - Cerner Formulary Items [5434]
81546 - BMS-247550 Preparations [5]

```
1-UMLS-CODE :
5-SYNONYMS : BMS-247550
5-SYNONYMS : BMS247550
7-HAS-PARTS *
8-PART-OF *
11-DEFINITION :
50-MAIN-MESH :
51-SUPPLEMENTARY-MESH :
55-AHFS-CLASS-CODE : 920000
56-DOSE-STRENGTH-UNITS : MG/ML
57-DOSE-STRENGTH-NUMBER : 2.00/1.00
58-FORMULARY-NAME : IXABEPILONE *IND* IVPB
59-SHORT-FORMULARY-NAME : *IXAB
60-DIGIMEDEX-FORMULARY-CODE :
61-DRUG-TRADE-NAME : IXABEPILONE *IND*
62-DRUG-GENERIC-NAME : IXABEPILONE *IND*
63-DRUG-MANUFACTURER :
64-DRUG-RX-VS-OTC :
65-DRUG-FORM-CODE : IVPB
66-DRUG-FLOOR-STOCK :
67-DRUG-ROUTE : IVPB
68-DRUG-IN-DIGIMEDEX-FORMULARY :
69-DRUG-VOLUME :
72-DRUG-CATEGORY :
```

Figure 1.

Screen Shot of Web-Based MED Editor.



```
+1|41703|CPMC Laboratory Test: LYME TOTAL ANTIBODY, SERUM

    1.  41348 CSF LYME ANTIBODY TEST
    2.  46736 Intravascular Total LYME ANTIBODY TESTS
    3.  46737 WESTERN BLOT LYME ANTIBODY TESTS
    4.  48969 LYME IGG TESTS
    5.  48970 LYME IGM TESTS
    6.  56749 LYME AB PATIENT OD TESTS
    7.  56750 LYME AB REA.CUT-OFF: TESTS
    8.  58610 LYME WB IGG BAND TESTS
    9.  58611 LYME WB IGM BAND TESTS
    10. 58613 LYME WB IGG RESULT TESTS
    11. 58614 LYME WB IGM RESULT TESTS
    12. 65160 LYME DNA TEST, BODY FLUID
    13. 70668 LYME DNA TEST BLOOD
    14. 70669 LYME DNA TEST CSF
    15. 94734 LYME POLYVAL AB TEST
    16. 95210 CSF LYME ANTIBODY DETECTION TEST
    17. 97962 LYME CSF WESTERN BLOT IGM TEST
    18. 97963 LYME CSF WESTERN BLOT IGG TEST
    19. 98663 LYME SOURCE TEST

Please enter one of these or another medcode or 0 if none:
```

Figure 2.

Sample of interactive script output requesting user input for placement of a new Lyme antibody test into the appropriate class.

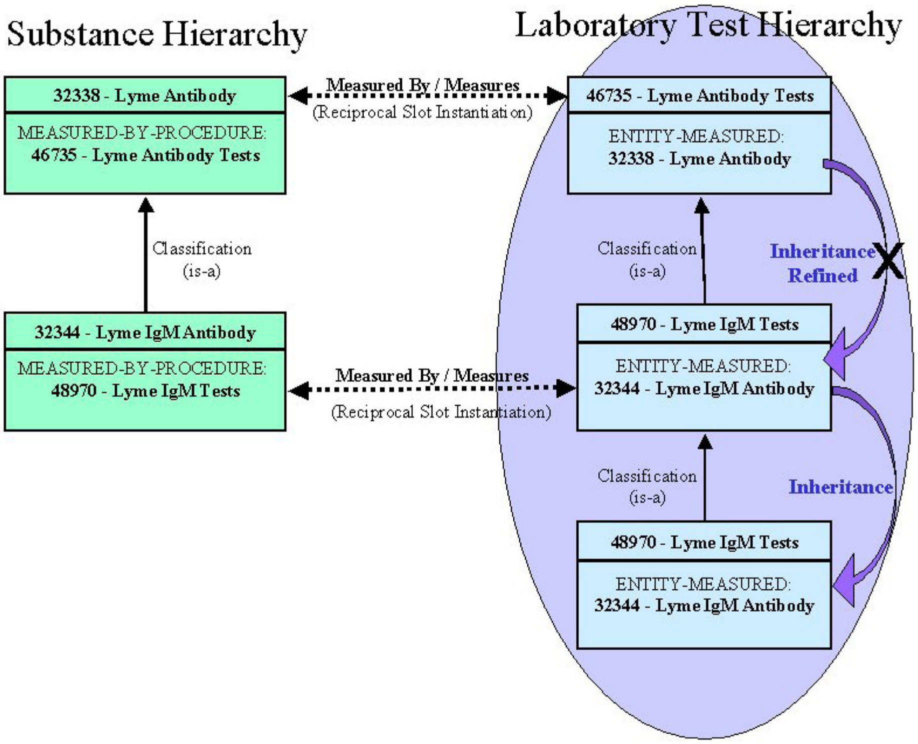
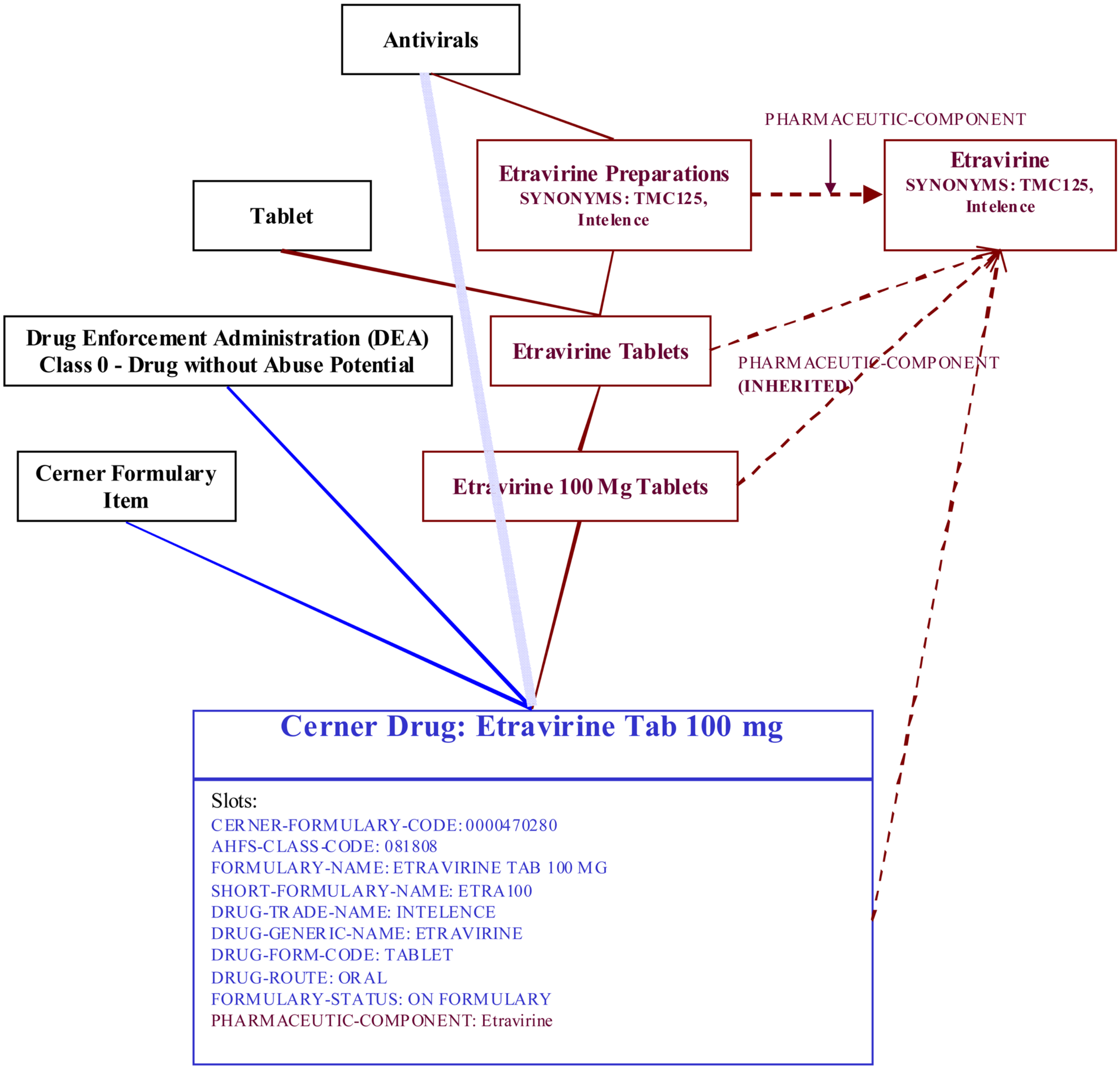


Figure 3.

Classification causes semantic inheritance with refinement. For MED Code 48970, the “ENTITY-MEASURED” inherited slot value, “32338”, is refined to the explicitly instantiated value, “32344”, since 32344 is more specialized than 32338 in the substance hierarchy.


Antivirals
PHARMACEUTIC-COMPONENT
Etravirine Preparations
SYNONYMS: TMC125, Intelence
Etravirine
SYNONYMS: TMC125, Intelence
Tablet
Drug Enforcement Administration (DEA) Class 0 - Drug without Abuse Potential
Etravirine Tablets
PHARMACEUTIC-COMPONENT (INHERITED)
Cerner Formulary Item
Etravirine 100 Mg Tablets
Cerner Drug: Etravirine Tab 100 mg

Slots:
CERNER-FORMULARY-CODE: 0000470280
AHFS-CLASS-CODE: 081808
FORMULARY-NAME: ETRAVIRINE TAB 100 MG
SHORT-FORMULARY-NAME: ETRA100
DRUG-TRADE-NAME: INTELENCE
DRUG-GENERIC-NAME: ETRAVIRINE
DRUG-FORM-CODE: TABLET
DRUG-ROUTE: ORAL
FORMULARY-STATUS: ON FORMULARY
PHARMACEUTIC-COMPONENT: Etravirine

Figure 4.

Example of the addition of a pharmacy concept to the MED. Components added by fully automated processes are indicated in blue, interactive processes in red. The solid lines are hierarchical relationships, the dashed lines are semantic relationships to other MED concepts as indicated, and the clouded blue line is an automated hierarchical relationship that was broken by the interactive creation of intervening classes in red.

Table 1

Sources, concept counts, and concepts attribute values in the Medical Entities Dictionary.

	Concept Count (approx)	**Attribute Value Count (approx)**
A) Local Sources:		
Laboratory Systems from both major clinical laboratories at NYPH.	27000	400000
Radiology System.	1200	12000
Pharmacy System.	10000	250000
Display Information (Display categories, formatting, Requests for laboratory summaries)	200	6000
Local Clinical Document and Reports	3000	28000
Local Document Template Forms, Sections and Fields, Attributes	6000	60000
B) External Knowledge To Supplement and Classify Local Terminology Entities:		
Laboratory, Pharmacy and Radiology Classes	10000	115000
Chemical Substance, Names, Synonyms	2500	52000
Pharmacy Substance Links		18000
Laboratory Substance Links		9000
Pathogenic Microorganisms diagnosed by specific procedures.		3000
Ideal Clinical Categories for Reviewing Results.		32000
Knowledge Sources for Infobuttons.		30000
NCBI Taxonomy Information	2000	8000
American Hospital Formulary Service (AHFS(TM)) Classes	700	8000
Local Physicians		
C) Standards:		
ICD9 Terms and Codes (Active and Retired)	18000	155000
CPT Terms and Codes	8700	60000
LOINC Laboratory Terms and Codes.		7000
LOINC Document Ontology	1000	3500

Table 2

Examples of the error codes that can be produced the automated MEDchecker program.

Error Code	Error Text	Description
6	Either Value Out Of Range Or Non-All-Digit	A semantic-valued slot value is not a valid MEDcode.
9	Slot Not Defined For This Part Of The Hierarchy BUT In String	Slots in the MED are instantiated at discrete points in the hierarchy, referred to as the “fathers” of the slots; this error indicates that a MED concept has a value for a slot but is not a descendant of the slot’s father
10	Value xxx Is Out-Of-Range For This Slot	Semantic slots are created in reciprocal pairs; the allowed values for a semantic slot are MEDcodes for concepts that are descendants of the father of the reciprocal slot; being the corollary of error 9, this indicates that a semantic slot is filled with a MEDCode for a concept value that is not a descendant of the reciprocal slot’s father – that is, it is not allowed to have the reciprocal slot
11	Missing Reciprocal In Slot xxx Of yyy	When two concepts are related by a pair of reciprocal slots, the MEDCode for each concept is a value in the other concept’s reciprocal slot; this error indicates that the reciprocal value is missing for one of the related MED concepts.
12	Ancestor (xxx) - Descendant (yyy) Relationship Between Explicit Values	In the MED, semantic slot values are inherited and exhibit refinement, meaning the more specified value prevails over a less specified value that is inherited. This error indicates that a slot value is explicitly stated as present, even though an ancestor concept has the same value; this is a problem because it cannot be removed if refinement is desired
13	Ancestor (xxx) - Descendant (yyy) Relationship Between Explicit/ Ancestor Value And Displayed Inherited/Descendant Value	Another error affecting refinement; this error indicates a hierarchical “flip-flop” in values has occurred in a semantic slot, with a more specific value in the ancestor’s slot and a less specific value in the descendant’s slot
15	Multiple Values (xxx) In A Single-Value Slot	The MED can define slots as multi-value or single value; this error indicates that there was an attempt to violate this rule.
16	Duplicate Value (xxx) In Slot	This error indicates an attempt to redundantly instantiate 2 identical values in the same slot for the same MEDcode.
17	Ancestor (xxx) - Descendant (yyy) Relationship Between Parents	This notification indicates that a hierarchical “shortcut” has been created – that is, a direct is-a relationship exists between two concepts that are also related indirectly through a chain of is-a relationships; it is not always an error.
18	Med Code xxx: Child (yyy) Is Also An Ancestor	This error indicates the presence of a classic hierarchical cycle, violating the definition of a directed acyclic graph.
S7	Slot xxx: Introduction Point_Out_Of_Med Code_Range	Indicates attempt to introduce slot (that is, define the slot’s father) at non-existent copncept
S13	Slot xxx: Duplicate Name With Slot xxx	Indicates an attempt to create a slot with the same name as an existing slot.