Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

A comparison of two methods for retrieving ICD-9-CM data: The effect of using an ontology-based method for handling terminology changes

Alexander C. Yu *, James J. Cimino

Department of Biomedical Informatics, Columbia University, New York, NY, USA

ARTICLE INFO

ABSTRACT

Objective: Most existing controlled terminologies can be characterized as collections of terms, wherein the terms are arranged in a simple list or organized in a hierarchy. These kinds of terminologies are considered useful for standardizing terms and encoding data and are currently used in many existing information systems. However, they suffer from a number of limitations that make data reuse difficult. Relatively recently, it has been proposed that formal ontological methods can be applied to some of the problems of terminological design. Biomedical ontologies organize concepts (embodiments of knowledge about biomedical reality) whereas terminologies organize terms (what is used to code patient data at a certain point in time, based on the particular terminology version). However, the application of these methods to existing terminologies is not straightforward. The use of these terminologies is firmly entrenched in many systems, and what might seem to be a simple option of replacing these terminologies is not possible. Moreover, these terminologies evolve over time in order to suit the needs of users. Any methodology must therefore take these constraints into consideration, hence the need for formal methods of managing changes. Along these lines, we have developed a formal representation of the concept-term relation, around which we have also developed a methodology for management of terminology changes. The objective of this study was to determine whether our methodology would result in improved retrieval of data.
Design: Comparison of two methods for retrieving data encoded with terms from the International Classification of Diseases (ICD-9-CM), based on their recall when retrieving data for ICD-9-CM terms whose codes had changed but which had retained their original meaning (code change).
Measurements: Recall and interclass correlation coefficient.
Results: Statistically significant differences were detected ($p < 0.05$) with the McNemar test for two terms whose codes had changed. Furthermore, when all the cases are combined in an overall category, our method also performs statistically significantly better ($p < 0.05$).
Conclusion: Our study shows that an ontology-based ICD-9-CM data retrieval method that takes into account the effects of terminology changes performs better on recall than one that does not in the retrieval of data for terms whose codes had changed but which retained their original meaning.

1. Introduction

Most existing controlled terminologies can be characterized as collections of terms that are arranged in a simple list or organized in a hierarchy. These terminologies are useful for standardizing terms and encoding data and are used in many existing information systems. As such, large amounts of data have been recorded using these terminologies. However, they suffer from a number of limitations that make data reuse difficult. Relatively recently, it has been proposed that formal ontological methods can be

applied to some of the problems of terminological design. In the application of formal ontological methods, a crucial distinction to be made is the one between concepts (embodiments of knowledge about biomedical reality) and terms (what is used to code patient data at a certain point in time, based on the particular terminology version). Techniques from computer science have historically been used to represent terminologies in computer-based information systems. *Ontological engineering* is defined as "the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies" [1]. The AI community has developed methods in the form of languages and reasoning algorithms that can be used to create terminologies that are amenable to computer-based automated reasoning. While ontological engineering has always drawn

* Corresponding author. Present address: 4093 Coventry Green Circle, Williamsville, NY 14221, USA.

E-mail address: alex.yu@dbmi.columbia.edu (A.C. Yu).

from different disciplines, more recently there has been an effort to directly incorporate philosophical notions in ontological engineering methods. *Formal ontology* is defined by Smith as "the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality" [2]. As an example of the perceived contributions of formal ontology, its practitioners seek to provide formal meanings of fundamental relations, such as the subsumption (or *is-a*) relation and the *part-of* relation as well as to provide guidelines on the proper use of these relations. Furthermore, it is argued that these methods will lead to the creation of terminologies that are of higher quality than many of the currently existing ones.

In addition, terminologies evolve over time in order to suit the needs of users. There are a number of types of changes that occur in terminologies that can have an effect on the reusability of data that are encoded with these terminologies. For example, when a *major name change* occurs, the term associated with a code changes so much that the meaning is essentially different, even as the code remains the same. Other types of described changes include *simple addition, refinement, deletion due to obsolescence, deletion due to redundancy, minor name change, precoordination, disambiguation, code change,* and *code reuse* [3].

We have developed a formal representation of the relation between concepts and terms, around which we have also developed a methodology for management of terminology changes. We have also implemented our methodology in a terminology maintenance tool. To evaluate our methodology, we compared two methods for retrieving ICD-9-CM data, based on recall when retrieving data for ICD-9-CM terms whose codes had changed but which had retained their original meaning. Our evaluation is focused on *code change*, which occurs when the code for a term changes but the term's meaning remains exactly the same. For example, in the 2005 version of ICD-9-CM, the code for "Meconium aspiration syndrome" is 770.1, but in the 2006 version, its code was changed to 770.10. Recall can be decreased when searching for ICD-9-CM-encoded data because when the code for a term changes, a retrieval method that does not properly manage the code change may miss cases that are encoded with the new code for the term.

2. Background

2.1. Controlled terminologies in biomedicine

Controlled terminologies are important in medicine mainly because of the need to cope with the volume of information involved in patient care. Applications such as direct patient care, clinical research, epidemiology, and billing all rely on information about diagnoses and procedures. In the age of computer-based patient records, controlled terminologies facilitate tasks such as data entry, data retrieval, data aggregation, and data analysis [4–7]. Furthermore, they play an integral role in more sophisticated computer-based applications such as decision-support systems and information retrieval systems, where they provide crucial links among data about the patient, computer-based knowledge sources, and computer-based reasoning programs [8]. Terminologies inevitably have to be updated in order to keep up with users' needs. Ideally, the meaning of recorded data should never change. However, without a systematic methodology for managing changes, the meaning of data may be lost or changed [3].

2.2. Desiderata for controlled terminologies

Given the importance of terminologies, it was only natural that researchers would try to define those properties that are required of a good terminology. Curators of controlled biomedical terminologies need definitions of properties that make a terminology adequate to serve its intended purposes. Various researchers had already been independently articulating these requirements, and in 1998, we published a review paper that brought together many of the common themes [4]. Although the desiderata arguably rest on a reasonably wide consensus within the informatics community, a few stand out as particularly relevant to the problem of terminology maintenance.

2.2.1. Concept orientation

Concept orientation means that the basic unit of the terminology is the meaning of the term, rather than the string of letters used to refer to that meaning. It implies that terms must not be vague and should correspond to at least one meaning (non-vagueness), concepts should correspond to no more than one meaning (non-ambiguity) and lastly, particular meanings should only correspond to single terms (non-redundancy).

2.2.2. Concept permanence

The meaning of a concept should never change and concepts should never be deleted in system implementations. The importance of this requirement becomes clear when one considers the possible dire effects on encoded data and interoperating systems if this requirement is not met. When data that are encoded using the older meaning of a concept are interpreted according to a new meaning for the concept, inferences may no longer be valid. Furthermore, deleting concepts will prevent us from being able to interpret all of the data that were encoded and stored prior to the deletion, including data whose meanings depend on deleted terms.

2.2.3. Graceful evolution

Controlled terminologies evolve over time in order to keep up with the needs of their users. Terminology curators must strive for graceful evolution of the content and structure of the terminology, in order to avoid problems due to ill-conceived changes and to facilitate understanding of the changes [3]. The description of the requirement of *graceful evolution* alludes to the use of a general editorial policy for change management. On the other hand, *concept orientation* and *concept permanence* are clearly defined, specific requirements that are directly related to the research problem because any methodology for organizing and maintaining terminologies must ensure that the terminology fulfills these requirements.

2.3. Temporal databases

Many applications of databases are temporal in nature, such as medical-record keeping, accounting, and airline scheduling [9]. Applications like these rely on temporal databases, which record time-referenced data. Some of the techniques developed in temporal database research are relevant to the problem of terminology maintenance, since many of the issues in terminology maintenance are also temporal in nature. Much like ontologies, databases model a part of reality. In addition, they also collect information about specific facts in that reality, using a variety of structures collectively called *database entities*. In general, temporal data are associated with facts recorded in these entities. Among these aspects is the notion of *valid time* – a span including the past, present, or future – when the fact is true. All facts have a valid time by definition. On the other hand, the *transaction time* of a database fact refers to the time when the fact is current in the database [10]. The transaction time spans from insertion to deletion of the fact in the database. The notion of a *valid time* is particularly relevant to terminology maintenance because relationships between codes, terms, and concepts (or meanings) change over time (as terminologies are updated) and thus have a valid time, i.e. it is only during

the valid time of the relationship that a code, a term, and a concept are related to each other in a particular configuration. The notion of transaction times, on the other hand, would be useful in local installations of terminologies, because information regarding *when* the facts regarding a particular code-term-concept configuration are added to the database can be used in "rollbacks" (i.e., restoring a configuration to a previous, stable state) and conflict resolution (e.g. resolving conflicts between a shared version and a local version).

2.3.1. Temporal data models and temporal entity relationship modeling

A significant amount of research work has gone into extending the conventional data models used in database management systems in order to support the capture of temporal data. Many relational models have been proposed [11], and each of these models include techniques for representing valid time and transaction time. The Bitemporal Conceptual Data Model, for example, uses timestamps for tuples, corresponding to facts, with values that are sets of (*transaction time*, *valid time*) pairs [11]. The presence of a (*tt*, *vt*) pair in a timestamp of a tuple means that the current state of the database at time *tt* records that the fact represented by the tuple is valid at time *vt*. The disadvantage of this approach is that the size of the database increases very quickly as time passes. An alternative model uses a fixed-length format for tuples [12]. T_s and T_e record starting and ending transaction times, while V_s and V_e record starting and ending valid times. The database stays up-to-date without changing things at every clock "tick"; this is made possible by the introduction of the variable *now* that assume the changing current time value.

These database models would allow terminology maintainers to carry out some of the operations that are needed to handle updates to terms. Notably, handling the semantics of changes in the meaning of "not elsewhere classified" is not possible. Furthermore, if one wanted to extend the models to incorporate relationships with concepts (i.e., in addition to codes and terms), then one would need to add another attribute to the model ("concept"). Concepts, however, are usually represented within ontologies, which are much better suited to handling the structures and semantics that are required for concepts and relations. Configuring a database of concept-term relationships to work with an ontology is possible, but it comes at the cost of maintaining separate structures, each with its own set-up and maintenance requirements. Ideally, we should be able to represent relationships among concepts, terms, and codes within the same structure that we use to represent concepts and relationships among concepts.

2.4. Terminology maintenance

We have previously described our experience coping with the annual updates to the ICD-9-CM terminology and incorporating them into the Medical Entities Dictionary (MED) [3]. The MED had mappings to the ICD-9-CM terminology, and every time the ICD-9-CM terminology changed, the maintainers of the MED first had to detect changes by comparing text versions of the two versions of ICD-9-CM using the UNIX *diff* program. Then, they had to analyze and handle the changes so that the mappings would remain valid. In earlier work, we formally described changes in terminologies that included possible reasons (both good and bad) for the changes. Corresponding to these changes were adaptive mechanisms for properly handling the changes in the MED. It was shown that while it was easy enough, with the right tools, to detect surface-level or *syntactic* changes in terminologies, the problem of determining the *semantics* of the change (i.e. how has the meaning of the term changed, if at all) was another problem altogether – one that was more difficult, and ultimately, more

important. For the results of syntactic analyses were deceptively simple, and only by doing a semantic analysis would it be possible to prevent the introduction of problems such as term *redundancy*, term *ambiguity*, and *undetected changes in meanings of terms*. To be specific, each of these problems would potentially result in a lowering in the quality and usefulness of inferences that could be made based on encoded data. To help in the prevention of these problems, he proposed a classification of terminology changes, alongside a methodology to handle these changes according to a concept-oriented paradigm that adhered to desiderata of controlled terminologies.

Oliver built on the work on the MED and proposed the CONCORDIA (CONcept and Change Operation Representation for DIAlects) methodology for synchronization of shared controlled medical terminologies with local versions (or "dialects") that could include extensions. The approach was centered on a formal model of medical concepts largely similar to those used in frame-based and semantic net-based knowledge representation systems. This formal representation in turn allowed Oliver to describe detailed and formal operations to handle the types of changes that we had earlier described [13]. For evaluation, she created a small sample test set based on three existing sources of medical knowledge in one subdomain (rickettsial diseases) and demonstrated evolution of the medical terminology in two divergent directions (one local version and one shared version). Then, a proof-of-concept demonstration was performed by synchronizing the local version with the shared version using a synchronization-support tool which incorporated the CONCORDIA model and a model for change logs that could be disseminated among independent sites.

Campbell demonstrated new methods to support an evolutionary approach to controlled medical terminology development. In the system that he created, multiple authors were allowed to independently define terms, and then partially rely on the system to detect and manage conflicts in the definitions. Conflicting differences in definitions were detected using the logically-based definitions of terms, and a description-logic classifier detected conflicting definitions based on semantic equivalence rather than syntactic equivalence. Furthermore, the configuration management methods that he developed relied on "change sets" that contained information on changes that had been made by authors. These change sets were used to support terminology verification and automated migration [14,15].

Noy and Musen developed the PROMPT set of tools that work with the Protégé ontology editor. One of the PROMPT tools handles semi-automated detection and handling of changes in ontologies. The tool produces a *structural diff* (analogous to the result of the "diff" UNIX program) that represents the structural differences between two versions of the same ontology. PROMPT also includes PROMPTDIFF, which is a set of heuristic algorithms that attempt to detect matches between concepts in different versions, as well as a user interface that helps human editors evaluate the results of PROMPT and make their own final decisions [16,17].

SNOMED-CT is currently being developed and promoted by the International Health Standards Development Organization (IHTSDO) [18,19]. SNOMED-CT has file formats for the corresponding notions of concept and relationship (among others). The IHTSDO developers include two fields (*effectiveTime* and *Active*) in each of the SNOMED-CT release file formats. Together, these fields enable the use of a log system to track all changes to the terminology's components. The *Active* field is filled with a boolean value that indicates whether a component (e.g. Concept, Relationship, Description, etc.) is valid within a particular release of SNOMED-CT *effectiveTime* serves as a timestamp for the log entries. Using this mechanism, it is possible to see both current values and historical values of any component at any point in time. It appears that it would be possible to implement most (if not all) of the terminology

changes that were described in [3], but documentation of the specific techniques to handle changes is limited to simple addition and deletion (due to obsolescence).

2.5. Reuse of encoded data in healthcare databases

A large amount of data is stored in health information systems in the form of encoded data. A significant portion of this is currently in the form of administrative data, which are generated as a result of administering healthcare delivery, reimbursing for services, and reporting morbidity and mortality statistics [7,20–24]. There are limitations to the use of this data. The clinical content of administrative data includes only the demographic characteristics and diagnoses of patients and codes for procedures. Furthermore, there is evidence that suggests that administrative data lack important information when compared with concurrently collected clinical data [25]. Still, the data are often reused for other purposes, such as the evaluation of the clinical outcomes and quality of care as well as creating datasets for clinical research [21,26], because they are readily available, relatively inexpensive to acquire, and computer-readable.

Typically, healthcare data will be stored in large data repositories in standard database management systems. Although maintainers of terminologies may release information on the changes that have occurred between versions, insuring that these changes do not affect data-related tasks is an awkward (and usually *ad hoc*) process. Unless the maintainers of the data repository institute procedures that incorporate the changes, similar to what we do with the MED [3], the effects of changes in terminologies may not be taken into account. A sample of studies that utilized encoded data over multiple years (hence, multiple versions of the terminology involved) did not mention methods to handle the possible effects of terminology changes on the performance of the data retrieval [26–30].

3. Methods

3.1. A formal representation of the Concept-Term Relation

We developed the notion of the *ConceptTermRelation*, which is based on a formal representation that captures information about the relation between concepts and terms. This representation is meant to be used in conjunction with a domain ontology constructed according to formal ontological principles. While the domain ontology serves as a representation of the things in a domain, the *ConceptTermRelation* notion is used to represent associations between terms and concepts.

The concept-term relation is represented as a reified *ConceptTermRelation* concept, which has as its attributes:

(1) The *hasCode* attribute that is filled by the terminology's code for the concept.
(2) The *hasTerm* attribute that is filled by the terminology's term (or *one* of its terms) for the concept. We note that it is possible that a concept may have more than one term associated with it. In that case, a separate *ConceptTermRelation* instance is to be created. On an ontological level, this reflects the reality that the ConceptTermRelation instance is for a particular association of concept and term. On a practical level, creating individual ConceptTermRelation instances allows maintainers to modify the other attributes (i.e., *hasStartDatetime*, *hasEndDatetime*, etc.) of the relation between a particular concept and term without affecting the relation between the same concept and another term.
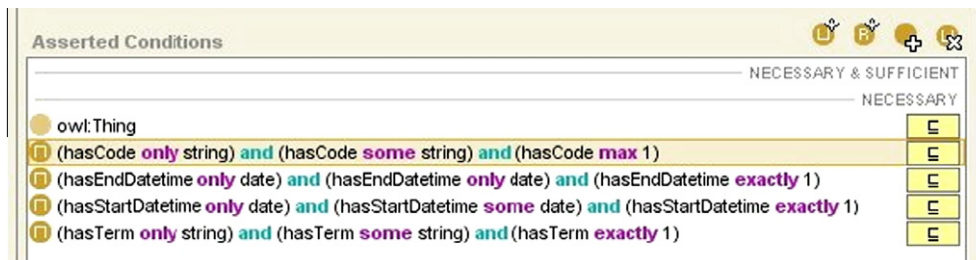

Fig. 1. Representation of the *ConceptTermRelation* concept in OWL using the OWL editing environment in Protégé.
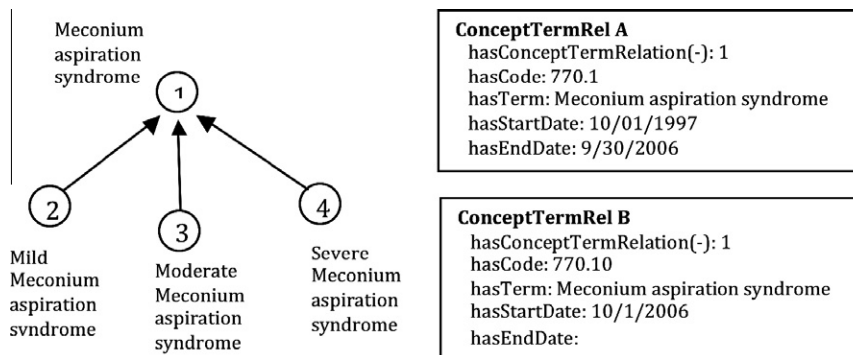

Fig. 2. An example that illustrates the use of *ConceptTermRelation*. On the left is a taxonomic hierarchy. Solid arrows stand for subsumption (is-a) relations. To the right of the graph are the *ConceptTermRelation* instances (ConceptTermRel A and B) for some of the concepts. The has*ConceptTermRelation* relations (shown as inverses) link concepts to *ConceptTermRelation* instances. There are two *ConceptTermRelation* instances shown for concept 1. Although the terms for concepts 2–5 are not included in ICD-9-CM, the ontology includes these concepts, and their instances are coded with "770.1" (before 10/1/2006) and "770.10" (from 10/1/2006 onwards) beciause they inherit the link to *ConceptTermRel*'s A and B from Concept 1.

(3) The *hasStartDatetime* attribute that is filled by the date when the particular association of code and term begin to be used in the terminology.

(4) The *hasEndDatetime* attribute that is filled by the date when the particular code and term case cease to be used or associated with each other in the terminology.

Specific relations between concepts and terms are represented by instantiating the *ConceptTermRelation* concept as particular *ConceptTermRelation* instances. Fig. 1 shows how we defined the *ConceptTermRelation* concept in OWL (Web Ontology Language) [31], and Fig. 2 illustrates the use of the *ConceptTermRelation* methodology to represent and handle inferences about code changes.

We note that in a production system managed by multiple curators, the fields we have included in our system would be incomplete: there would be at least three other fields necessary, i.e., one noting the type of change, another explaining the reason for the change, and yet another identifying who made the change. The system implemented in our study was designed to handle particular problems we encountered in current systems. Other requirements make it necessary for other fields to be included.

3.2. Terminology change management with ConceptTermRelation

The *ConceptTermRelation* model is general enough to handle each of the possible types of changes originally described by us. Each operation corresponds to one of the terminology changes that we originally described [3]. In addition, we describe a new type of change – changes to "Not elsewhere classified". One point to emphasize is that while it is possible to track down the corresponding meaning of a code for a particular point in time, the approach presented here would allow for automatic determination of this correspondence without having to resort to historical change-files. This is possible precisely because knowledge about the concept-term relation is incorporated into the model.

3.2.1. Simple addition
When a new term is added to a terminology, there are two possible cases to consider.

(1) *The concept to which it refers already exists in the ontology*. In this case, two new *ConceptTermRelation* instances are created and values are entered for the attributes of each instance. Two new instances are created in order to be able to handle "Not elsewhere classified"-type terms and changes to these terms (see Section 3.2.10). Note that while the *StartDate* attribute is filled, the *EndDate* attribute remains unfilled, to represent the fact that the concept-term relation continues to hold (until such a date has been specified). Also, *hasConceptTermRelation* relations are asserted between the concept and the two new *ConceptTermRelation* instances.

(2) *The concept to which it refers does not exist in the ontology*. In this case, unless the term is of the "NEC"-type, a new concept has to be created and properly placed in the ontology. Then, as in case 1 above, two new *ConceptTermRelation* instances are created and values are entered for the attributes of each instance. Also, *hasConceptTermRelation* relations are asserted between the concept and the two new *ConceptTermRelation* instances.

In both cases, if two new redundant terms are introduced to the terminology, then two (2) *ConceptTermRelation* instances have to be added for each term, for a total of four (4) *ConceptTermRelation* instances.

3.2.2. Refinement
Refinement occurs when one or more terms are added to provide a greater level of detail than was present in a previously existing term. In terms of change management operations using the *ConceptTermRelation* model, there is no difference between handling simple addition of a term and handling the addition of a refining term.

3.2.3. Precoordination
Precoordination occurs when a new complex term is created by combining two or more pre-existing simpler terms. In an ontology, this may correspond to a concept that is subsumed by or related to (e.g. part-of, has-site) two or more different concepts. Assuming that the *is-a* assertions are valid, this is handled in the following manner:

(1) *The concept for the new term already exists in the ontology*. The ontology itself may have to be updated in order to reflect that the concept should have been subsumed by two parent concepts (i.e. a case of adding "missing" knowledge). Otherwise, this is treated the same as case (1) of simple addition of a new term.

(2) *The concept for the new term doesn't exist in the ontology*. A new concept will have to be created, and any necessary subsumption (*is-a*) relations will have to be specified. Again, from this point forward, this is treated the same as case (1) of simple addition of a new term.

3.2.4. Deletion due to obsolescence
Deletion of a term due to obsolescence is handled by entering a value for the *hasEndDate* attribute of the corresponding *ConceptTermRelation* instance(s); the value will be the date when the term was deleted from the terminology. Note that the corresponding concept in the ontology is never deleted (thus holding to the requirement of concept permanence). Instead, the knowledge of when the term stops being used in the terminology is captured in the *ConceptTermRelation* instance.

3.2.5. Deletion due to discovered redundancy
Assuming that the terms were linked to the same concept in the first place (i.e. the ontology adhered to the principle of *concept orientation*), deletion of a term due to discovered redundancy is handled in the following manner.

(1) If one of the redundant terms is kept in the terminology and the other(s) is(are) dropped, then we handle the deletion of the term(s) the same way as deletion due to obsolescence.

(2) If all terms are dropped and a new term is introduced to replace the redundant terms, then we handle the deletion of all dropped terms the same way as deletion due to obsolescence. In addition, we treat the addition of the new term as a case of simple addition of a new term (with pre-existing concept, case (1)).

If the redundant terms are linked to different concepts, and the concepts are found to be equivalent in meaning, this implies that the ontology did not adhere to *concept orientation*. The maintainer(s) of the ontology would have to retire one or more the redundant concepts and any associated term(s). Methods for retiring concepts is beyond the scope of our methodology, but has been discussed in the terminology literature [3], and is implemented in some terminologies, notably in SNOMED-CT [32].

3.2.6. Major name change and minor name change
In terminologies that are used as coding systems, the term associated with a code can change. A *major name change* occurs when a

term sufficiently differs from its previous version so that the term actually refers to a different concept. For example, in 1994, the ICD-9-CM term "Flaccid Hemiplegia" (342.0) was changed to "Flaccid Hemiplegia and Hemiparesis" (342.0). A major name change can be seen as a composite change involving the deletion of the old term and an addition of a new one. In the example, "Flaccid Hemiplegia" (342.0) is deleted and the new term "Flaccid Hemiplegia and Hemiparesis"(342.0) is added; it just so happens that they map to the same code (in non-overlapping time periods). Handling a major name change involves two steps:

(1) Treating the deletion of the old term the same way as a deletion due to obsolescence.
(2) Treating the addition of the new term the same way as simple addition of a new term.

When a term changes but the concept to which it refers remains exactly the same, then this is a case of *minor name change*. Changes may be made to correct spelling errors or to keep up-to-date with naming conventions; they can also be made to make the meaning of a term more explicit. There are two things to consider when this type of change occurs. First, users might search for the concept using either the new term or the old term, so we have to explicitly represent that both terms are (or were) used (during different periods of time) to refer to the same concept. Second, encoded data that are stored after the minor name change will probably be retrieved using the new term, and data that are recorded before the change will be retrieved using the old term. Therefore, whenever a minor name change occurs in the terminology:

(1) The relevant *ConceptTermRelation* instance(s) are changed by entering value(s) for the *hasEndDate* attribute. Therefore, we are able to retain the old term as well as knowledge of its relations to the concept and the code. We did not allow for this kind of functionality in earlier work [3]. Although in Oliver's methodology it is possible to retain the old term and its relations to the concept and the code, this kind of knowledge is not represented in the ontology itself. One would have to extract the information from a log of changes in order to accomplish the same task, and there is no straightforward way to keep track without resorting to reification [13].
(2) New *ConceptTermRelation* instance(s) are created with the same values for *hasCode*. The *hasStartDate* attribute will have as its value the new date, and *hasTerm* will have the new term. A *hasConceptTermRelation* relates the concept to the new *ConceptTermRelation* instance(s).

3.2.7. Disambiguation

Sometimes, it becomes clear that an existing term could have had more than one possible interpretation or meaning. Two possible cases can be identified:

(1) *The existing term was actually used to refer to several meanings.* If there are *n* possible meanings, we handle this case as 1 minor name change (change of the ambiguous term to an unambiguous term that refers to the same concept) plus *n-1* simple additions of new terms (one for each possible meaning).
(2) *The existing term was always used to mean just one of the possible meanings.* This case is treated as a minor name change.

3.2.8. Code reuse

Code reuse occurs when the term associated with a code changes dramatically [3]. Code reuse can be perceived as being the same as major name change, except that the semantic distance between the old concept being referred to and the new concept is

greater. Therefore, the mechanisms for handling this type of change are the same as for handling major name changes.

3.2.9. Code change

Code change occurs in a terminological coding system when a term (and its corresponding concept) remains the same but the code that is associated with the term changes. In order to handle this type of change, the following operations are performed:

(1) Enter values for the *hasEndDate* attribute of the relevant *ConceptTermRelation* instance(s).
(2) Create new *ConceptTermRelation* instance(s) with the new code value and a new *hasStartDate* value; the *hasEndDate* attribute remains unfilled and the value for *hasTerm* is identical to one in the original *ConceptTermRelation* instance(s). The concept is then linked to the new *ConceptTermRelation* instance by a *hasConceptTermRelation* relation.

3.2.10. Change in meaning of "Not elsewhere classified"-type term

A special type of change may occur when terms are added, deleted, or undergo a major name change in subdomains where a "Not elsewhere classified"-type term is used. The meaning of the "Not elsewhere classified"-type term changes because its meaning depends on the meaning of the other terms.

In order to represent "Not elsewhere classified"-type terms, we make further modifications to *ConceptTermRelation*. These modifications are introduced in order to be able to accurately map "Not elsewhere classified"-type terms to the concepts to which they refer without requiring the introduction of artificial "NEC" concepts. The modification is relatively small: we introduced an *inheritable* attribute to the *ConceptTermRelation* concept, which can be filled with one of two possible Boolean values ("true" or "false"). When a *ConceptTermRelation* instance has an *inheritable* value of true, then the descendants of the concept to which the *ConceptTermRelation* instance is linked (via the *hasConceptTermRelation* relation) can inherit the link to that *ConceptTermRelation* instance. "Not elsewhere classified"-type concepts are not directly represented as a concept in the ontology. Instead, it is through the use of the *inheritable* attribute in *ConceptTermRelation* that one is able to handle changes in meaning (semantic drift) of "Not elsewhere classified"-type terms. In those subdomains where a "Not elsewhere classified"-type term is used (e.g. "Viral pneumonia due to other virus, not elsewhere classified"), some concept *X* that corresponds to the root term ("Viral pneumonia" or "Viral pneumonia, not otherwise specified") will have at least one *ConceptTermRelation* instance that has the value "true" for its *inheritable* attribute and whose *hasTerm* attribute will be the "Not elsewhere classified"-type term (e.g. "Viral pneumonia, not elsewhere classified"). After inspecting the start and end datetimes of the *ConceptTermRelation* instances of the child concepts of concept *X*, we can determine which of the child concepts of *X* will inherit a link to inheritable *ConceptTermRelation* instance(s) at a particular point in time. For the point of time we are interested in (e.g. "March 5, 2006"), if the time period of an inheritable "NEC" *ConceptTermRelation* instance is valid (e.g. "January 1, 2006 – "), then any concepts that do not have any *ConceptTermRelation* instances that are valid for that point in time will inherit the link to the inheritable *ConceptTermRelation* instance. Furthermore, those concepts that are children of concept *X* that do not have *any ConceptTermRelation* instances linked to them (i.e. those concepts without a corresponding term in *any* version of the terminology) will automatically inherit the links to the inheritable *ConceptTermRelation* instance. Thus, any changes to the meaning of the "Not elsewhere classified"-type term are automatically handled because the "inheritable" *ConceptTermRelation* instance serves as a virtual pointer to those concepts that are *not* referred to by a term in a particular

version of the terminology. For example (see Fig. 3), let's say a new term is added (simple addition) in the new version (January 1, 2007) of a terminology and the term's corresponding concept is a child of concept *X*. The addition of the new *ConceptTermRelation* instances (start date January 1, 2007) that are related to the child concept tells us that from January 1st 2007 onwards, the link to the inheritable *ConceptTermRelation* instance will *not* be inherited by that child concept. Therefore, no change is needed: the handling of a change in the meaning of an "NEC"-type term is implicit in the representation mechanism. Figs. 3 and 4 show how our method might improve data retrieval when "NEC"-type terms are involved.

3.3. Implementation and evaluation

We implemented our methodology in a terminology mainte-nance tool as an extension to the Protégé ontology editor. We used the tool to interface a domain ontology derived from SNOMED CT (2005) and parts of ICD-9-CM (1997 version). We also used the tool to handle successive changes to ICD-9-CM (1998–2006).

Then, as part of our evaluation of the methodology, we com-pared two information retrieval methods based on their perfor-mance on retrieving ICD-9-CM-encoded data after the occurrence of a code change. Method 1 was based on a static view of the ICD-9-CM terminology (i.e. without regard for changes in concepts,
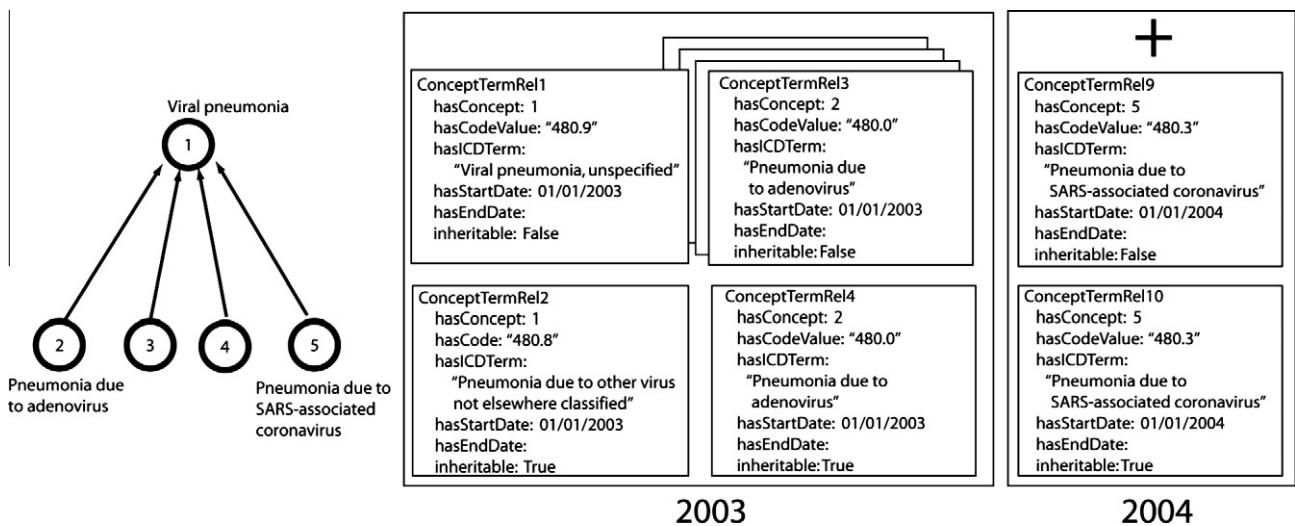

Fig. 3. On the left we show the taxonomical hierarchy for the "Viral pneumonia" domain. Solid arrows stand for subsumption relations. To the right of the graph are the *ConceptTermRelation* instances for some of the concepts. In the 2003 version, we show two *ConceptTermRelation* instances for concept 1 ("Viral pneumonia") and two *ConceptTermRelation* instances for concept 2 ("Pneumonia due to adenovirus"). In 2003, although it isn't a coded term in ICD-9-CM 2003, the ontology already has the concept for "Pneumonia due to SARS-associated coronavirus". Hence, it "inherits" the code "480.8" from its "parent" concept (concept 1). The 2004 version of ICD-9-CM adds the entry for "Pneumonia due to SARS-associated coronavirus", so we create two new instances of *ConceptTermRelation* for concept 5. Since concept 5 now has its own *ConceptTermRelation* instances, the code "480.3" overrides the inheritable code "480.8" from the parent concept. Not shown are the *ConceptTermRelation* instances for concepts 3 and 4.

Data encoded in 2003	Data encoded in 2004
Case 1: SARS-associated pneumonia (encoded 480.8)	Case 6: SARS-associated pneumonia (encoded 480.3)
Case 2: Hantavirus pneumonia (encoded 480.8)	Case 7: SARS-associated pneumonia (encoded 480.3)
Case 3: SARS-associated pneumonia (encoded 480.8)	Case 8: Hantavirus pneumonia (encoded 480.8)
Case 4: SARS-associated pneumonia (encoded 480.8)	
Case 5: Hantavirus pneumonia (encoded 480.8)	

Fig. 4. If the terminology does not take into account the change in the meaning of the "NEC"-type term "Pneumonia due to other virus not elsewhere classified" (code 480.4), there are unwanted effects that can occur that affect data retrieval. To illustrate this, we extend the example in Fig. 3. In this example, several cases of SARS-associated pneumonia and Hantavirus pneumonia are encoded in 2003 and 2004. Neither of these diseases have their own term and code in ICD-9-CM in 2003, but in 2004, a code and term are added for SARS-associated pneumonia. (1) When retrieving cases of SARS-associated pneumonias, a method that ignores the fact that 480.8 in 2003 will include SARS-associated pneumonias will not retrieve those cases (decreasing recall). In the example above, Cases 1, 3, and 4 will not be retrieved by methods that do not recognize that the meaning of "480.8"("NEC") includes these cases encoded in 2003 when 480.8 had a broader definition. Our method incorporates these facts and so is able to retrieve these cases. (2) Invalid comparisons might be made between data encoded as 480.8 in different years. This can happen because data encoded as 480.8 in 2003 will include SARS-associated pneumonias while data encoded as 480.8 in 2004 will not. A naïve assumption that 480.8 in 2003 and 2004 are comparable will not take this into account. Our method can track these differences, and the precision of comparisons across years can be improved by more precise definitions of "NEC" according to the terminology version.

terms, and codes); this is the approach typically used in retrospective studies [26,28,29]. Method 2 was based on an approach for managing ICD-9-CM that took into account the effects of terminological changes using an ontological view. In the case of a code change, Method 2 utilizes the information about the code change, so that searches against an ICD-9-CM-encoded database will retrieve cases with the new code as well as the old code. Since Method 1 does not utilize this information, only cases encoded with the old code will be retrieved.

ICD-9-CM-encoded clinical data were obtained from the Columbia University Medical Center (CUMC) clinical data repository (January 1, 1997–May 1, 2006) with the approval of the Columbia University Institutional Review Board. All data were formatted and deidentified using Perl scripts and stored on a secure database running on MySQL. We analyzed all the changes that occurred in ICD-9-CM and categorized them according to the terminology change types [3]. Terminology changes that occurred across the 1997–2006 versions of the ICD-9-CM were listed and sorted into bins of change types.

We obtained a random sample of ten code changes from the population of terminology changes in the bin for code changes. For each code change, we generated a corpus of cases using a combination of keyword- and pattern matching-based searches to screen for cases that had the discharge diagnosis from the CUMC clinical data repository. A case was defined as a unique hospital inpatient course that occurred over a period of time beginning at the admission date and ending at the discharge date. Therefore, the same patient could be associated with more than one case. Discharge dates had to be within the time period encompassing the consecutive 24 months (i.e., before October) preceding the code change, and the consecutive 15 months afterward. The date range was based on results of test runs of the case-retrieval program we used to screen for cases.

Cases were allocated equally among five judges. For each of the cases retrieved by the screening process, the judges reviewed the diagnosis section (also called the "impression" or "assessment" section) of the discharge summary associated with each case and made a judgment that served as the reference standard for that case. Judges were asked to determine whether any one of the ten diagnoses was made by the primary physician in each case: After reading the diagnosis section of the discharge summary, the judges determined whether the diagnosis was documented by the primary physician in the case. Possible responses included "Yes", "Maybe", "Cannot tell", and "No". A *positive case* was defined as a case where the diagnosis was given by the primary physician. A *negative case* was defined as a case where the diagnosis was not given by the primary physician. The raters' responses were dichotomized into the two categories in three different ways. This was done because it was found that there was some variation in how judges interpreted the "maybe" and "cannot tell" categories. In the first dichotomization, "Yes" responses were binned into the *positive case* category, and "Maybe", "Cannot tell", and "No"

responses were binned into the *negative cases* category. In the second dichotomization, "Yes" and "Maybe" responses were binned into the *positive case* category, and "Cannot tell" and "No" responses were binned into the *negative case* category. Finally, in the third dichotomization, "Yes", "Maybe", and "Cannot tell" responses were binned into the *positive case* category, and only "No" responses were binned into the *negative case* category.

For each code change, we carried out parallel SQL queries using Methods 1 and 2 against the corpus of cases generated for recall measurement. Recall was computed as the proportion of all cases in the corpus that were classified as a positive case by the human experts and also retrieved by the method, based on the actual codes in the patient records. The recall performance of Methods 1 and 2 on each code change were compared using the McNemar Test, which is appropriate when the data consist of paired observations of nominal data [33].

Finally, in order to estimate the reliability of the measurement process, the judges were asked to give their responses to each case in a separate set of 96 cases. There were five (5) judges and three (3) categories ("Yes","Cannot tell OR Maybe yes", and "No"). Inter-rater reliability was measured using an intraclass correlation coefficient (two-way mixed, single measures, absolute agreement – corresponding to Shrout and Fleiss's ICC(3,1) [34]) using the SPSS statistical software program [35].

4. Results

Table 1 shows the ten selected code changes that we examined in our study, with entries describing the changes between years, the terms associated with the codes that were changed, and the actual codes that were listed in ICD-9-CM. A total of 675 cases were retrieved by the screening process and presented to the judges.

Table 2 shows the results of measuring the recall performance of Methods 1 and 2 on the code changes. The results show that Method 2 performed significantly better ($p < 0.05$) than Method 1 for two of the ICD-9-CM terms whose codes had changed (Code changes 3 and 9), regardless of how judges' responses were dichotomized in the reference standard. For a third code change (Code change 8), Method 2 performed better than Method 1 using dichotomization 3. Finally, when all the cases were combined in an "overall" category, Method 2 also performed statistically significantly better ($p < 0.05$) than Method 1.

With regards to the judges, two of the judges had finished 3 years of specialty training in Internal Medicine, 1 judge had finished 1 year of specialty training in Internal Medicine, and 1 judge had finished medical school. The calculated interclass correlation coefficient was sufficiently large at 0.599 (95% CI 0.503, 0.689), so the results of the inter-rater reliability study show that reliability was sufficient for the judges' responses to be used as a reference standard.

Table 1
ICD-9-CM Code changes that were used in the recall study.

Code change	Year	Term	Previous code	New code
1	2006	Meconium aspiration syndrome	770.1	770.10
2	2001	Ulcer of lower limbs	707.1	707.10
3	2005	Decubitus ulcer	707.0	707.00
4	2005	Dysplasia of cervix	622.1	622.10
5	2005	Endometrial hyperplasia	621.3	621.30
6	2005	Prolapse of vaginal wall without mention of uterine prolapse	618.0	618.00
7	2006	Urinary obstruction	599.6	599.60
8	2005	Hyperparathyroidism	252.0	252.00
9	2001	Hyperplasia of prostate	600	600.0
10	1998	Staphylococcal septicemia	038.1	038.10

Table 2
Recall performance of Method 1 and Method 2 on ten (10) code changes (reported in %; n is the number of positive cases found by the judges). Double asterisks indicate a significant difference detected with the McNemar Test ($p < 0.05$). Method 2 performed significantly better for Code changes 3 and 9, regardless of how judges' responses were dichotomized for the reference standard. Method 2 also performed significantly better for Code change 8 under the third dichotomization.

Code change	Dichotomization 1 (Positive = Y Negative = N,M,C)[*]		Dichotomization 2 (Positive = Y,M Negative = N,C)[*]		Dichotomization 3 (Positive = Y,M,C Negative = N)[*]	
	Retrieval Method 1	Retrieval Method 2	Retrieval Method 1	Retrieval Method 2	Retrieval Method 1	Retrieval Method 2
1	54.55% ($n = 11$)	54.55% ($n = 11$)	46.15% ($n = 13$)	46.15% ($n = 13$)	42.86% ($n = 14$)	42.86% ($n = 14$)
2	42.86% ($n = 14$)	64.29% ($n = 14$)	42.86% ($n = 14$)	64.29% ($n = 14$)	42.86% ($n = 14$)	64.29% ($n = 14$)
3	49.34%[**] ($n = 152$)	53.29%[**] ($n = 152$)	49.67%[**] ($n = 153$)	53.59%[**] ($n = 153$)	50.00%[**] ($n = 154$)	53.90%[**] ($n = 154$)
4	37.50% ($n = 24$)	37.50% ($n = 24$)	37.50% ($n = 24$)	37.50% ($n = 24$)	38.46% ($n = 26$)	38.46% ($n = 26$)
5	35.50% ($n = 16$)	56.25% ($n = 16$)	33.33% ($n = 18$)	50.00% ($n = 18$)	33.33% ($n = 18$)	50.00% ($n = 18$)
6	20.00% ($n = 15$)	20.00% ($n = 15$)	20.00% ($n = 15$)	20.00% ($n = 15$)	18.75% ($n = 16$)	18.75% ($n = 16$)
7	14.29% ($n = 14$)	21.43% ($n = 14$)	13.33% ($n = 15$)	20.00% ($n = 15$)	13.33% ($n = 15$)	20.00% ($n = 15$)
8	22.00% ($n = 50$)	32.00% ($n = 50$)	21.57% ($n = 50$)	31.37% ($n = 50$)	21.82%[**] ($n = 55$)	32.72%[**] ($n = 55$)
9	26.32%[**] ($n = 266$)	39.85%[**] ($n = 266$)	26.02%[**] ($n = 269$)	39.41%[**] ($n = 269$)	25.64%[**] ($n = 273$)	39.19%[**] ($n = 273$)
10	18.52% ($n = 54$)	18.52% ($n = 54$)	19.18% ($n = 73$)	20.55% ($n = 73$)	18.18% ($n = 77$)	19.48% ($n = 77$)
Overall	32.14%[**] ($n = 616$)	40.91%[**] ($n = 616$)	31.53%[**] ($n = 647$)	40.03%[**] ($n = 647$)	31.16%[**] ($n = 661$)	39.67%[**] ($n = 661$)

[*] Y = Yes, N = No, M = Maybe, C = Cannot tell.
[**] Statistically significant difference ($p < 0.05$).

5. Discussion

Our method builds upon previous work on using a formal analytical approach to detecting and managing terminology changes. Furthermore, we have adopted a formal representation of terminology changes that is compatible with the widely-adopted Web Ontology Language (OWL) representation for ontologies. One advantage of an ontology-based approach is that it allows us to handle changes in the terminology in a way that propagates these changes down a class hierarchy correctly (inheritance). This in turn allows for an easier time managing changes in terminologies and incorporating them in a way that does not alter or lose the meaning of existing data. The method also takes into account problems with managing other types of terminology changes that are not solved by simply querying for the class. For example, in major name changes, the meaning of a term associated with the code changes but the code remains the same. Our method allows for the representation and handling of these kinds of changes. SNOMED-CT, as mentioned above, includes a log-style change mechanism that employs the use of the *effectiveTime* and *active* fields. These fields are used to handle changes to the ontology itself, and are used in the core ontology files (i.e. Concept, Description, Relationship, Reference Set). There is some overlap with the methods in our model that pertain to changes in the ontology (as opposed to changes in the relation between concepts and terms), but our methods also specifically deal with problems involve the relationship between terms and concepts. SNOMED-CT does include "cross-maps" between the SNOMED-CT ontology and external terminologies such as ICD-9 (and ICD-10 in the future), but the mappings (analogous to the *ConceptTermRelation* in our model) do not include equivalent attributes that would allow for the kind of change management we describe here [18,19,36,37].

One notable result in our experiments is that the recall performance of both methods are low, and this finding may be explained by the fact that the automated retrieval of cases (based on incomplete coding of cases by human coders) is measured against the responses of human expert judges who were asked about specific diagnoses. It is not hard to understand why Method 2's recall performance can be significantly better for terms whose codes have changed. Over time, human coders will become better at using the new (and correct) code for the diagnosis, even though there is bound to be some lag between the official start date of a code change in ICD-9-CM and full compliance with that change. A method that did not take into account code changes would miss more cases with the correct diagnosis, since the method would not know that cases with the new code should be retrieved. On the other hand, a method that took into account the new code would retrieve cases with the old code (prior to the date when the change is enforced) as well as cases with the new code (subsequent to the date when the change is enforced).

Precision was not measured in this study, because cases were retrieved with either method based on coding by the hospital coders with the relevant codes, even if the judges did not do so based on the limited abstracts they reviewed. However, the methodology we have developed has the potential to improve precision, because it takes code reuse into account. Code reuse is a type of terminology change that occurs in ICD-9-CM when the name associated with a code is changed in such a way as to change its meaning (the converse of a code change). Although none of the codes were affected by code reuse, this type of change does occur in ICD-9-CM. If one of the codes had been reused during the study period, a method that did not take into account code reuse would incorrectly retrieve cases coded with that code subsequent to its change in meaning, thereby increasing the number of false positive cases and reducing precision.

While ICD-9-CM is by no means representative of all terminologies, the pervasiveness of its use in healthcare, as well as the fact that many of the difficulties of handling "real world" terminologies also plague ICD-9-CM, made it a good candidate for this study on the effects of properly handling terminology changes on reuse of healthcare data. ICD-9-CM-encoded data is ubiquitous, as it is currently a part of reimbursement and reporting requirements, therefore, the results of this study are applicable to a broad range of areas such as quality assurance and clinical research. One limitation of the study is that ICD did not change as drastically as it has in past years, and so we were limited to studying code changes, and the measured difference in performance was relatively small.

We have also introduced a new method for handling changes to "Not elsewhere classifed"-type terms. While some of the techniques that we describe may have been adopted in other fields (notably temporal databases), and IHTSDO's work on SNOMED-CT (and to a lesser extent in the UMLS) provide the knowledge provide some of the knowledge necessary to support such techniques, to date our work has been the only one to address how such techniques could be used to address this problem.

When data are reused, the effects of terminology changes need to be taken into account. Our study shows that an ontology-based ICD-9-CM data retrieval method that does so performs better on recall than one that does not in the retrieval of data for terms whose codes had changed but which retained their original meaning.

Acknowledgment

References

[1] Gomez-Perez A, Fernandez-Lopez M, Corcho O. Ontological engineering. London: Springer-Verlag; 2004.

[2] Smith B. Ontology: philosophical and computational. In: Floridi L, editor. The blackwell guide to the philosophy of computing and information. Blackwell Publishers; 2003.

[3] Cimino JJ. Formal descriptions and adaptive mechanisms for changes in controlled medical vocabularies. Methods Inf Med 1996;35(3):202–10.

[4] Cimino JJ. Desiderata for controlled medical vocabularies in the twenty-first century. Methods Inf Med 1998;37(4–5):394–403.

[5] Cimino JJ. From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. J Am Med Inform Assoc 2000;7(3):288–97.

[6] Commission on Professional and Hospital Activities. International Classification of Diseases. Ninth Revision, with Clinical Modifications (ICD-9-CM). Ann Arbor; 1978.

[7] Spackman KA, Campbell KE, Cote RA. SNOMED RT: a reference terminology for health care. Proc AMIA Annu Fall Symp 1997:640–4.

[8] Musen MA. Modern architectures for intelligent systems: reusable ontologies and problem-solving methods. Proc AMIA Symp 1998:46–52.

[9] Jensen C. Temporal Database Management. Aalborg University; 2000.

[10] Inmon WH. Building the data warehouse. John Wiley & Sons; 1992.

[11] Jensen CS, Snodgrass RT. Semantics of time-varying information. Inf Syst 1996;21(4):311–52.

[12] Snodgrass R. The Temporal Query Language TQuel. ACM Tran Database Syst 1987;12(2).

[13] Oliver DE, Shahar Y. Change management of shared and local versions of health-care terminologies. Methods Inf Med 2000;39(4–5):278–90.

[14] Campbell KE, Cohn SP, Chute CG, Shortliffe EH, Rennels G. Scalable methodologies for distributed development of logic-based convergent medical terminology. Methods Inf Med 1998;37(4–5):426–39.

[15] Campbell KE, Das AK, Musen MA. A logical foundation for representation of clinical data. J Am Med Inform Assoc 1994;1(3):218–32.

[16] Noy NF, Musen MA. The PROMPT suite: Interactive tools for ontology merging and mapping. Int J Hum–Comput Stud 2003;59(6):983–1024.

[17] Noy NF. Semantic integration: a survery of ontology-based approaches. SIGMOD Record, Special issue on semantic integration. 2004;33(4).

[18] International Health Terminology Standards Development Organisation. Implementation Special Interest Group [accessed 07.12.10].

[19] International Health Terminology Standards Development Organisation. SNOMED-CT Data Structures Specification v1.0. <http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/Publications/Tools/SNOMED_CT_Release_Format_2_-_Data_Structures_Specification_v1.0.pdf> [October 3, 2010].

[20] American Medical Association. Current procedural terminology. Chicago, IL: The Association; Updated Annually.

[21] Iezzoni LI. Assessing quality using administrative data. Ann Intern Med 1997;127(8_Part_2):666–74.

[22] Spackman K. SNOMED RT and SNOMEDCT. Promise of an international clinical terminology. MD Comput 2000;17(6):29.

[23] World Health Organization. International classification of diseases, 10th Revision: <http://www.who.int/classifications/icd/en/>.

[24] World Health Organization. Ninth edition. International Classification of Diseases Index. Manual for the International Statistical Classification of Diseases. Geneva 1977.

[25] Jollis JG, Ancukiewicz M, DeLong ER, Pryor DB, Muhlbaier LH, Mark DB. Discordance of databases designed for claims payment versus clinical information systems: implications for outcomes research. Ann Intern Med 1993;119(8):844–50.

[26] Chapman WW, Dowling JN, Wagner MM. Generating a reliable reference standard set for syndromic case classification. J Am Med Inform Assoc 2005;12(6):618–29.

[27] Tu JV, Austin PC, Chan BT. Relationship between annual volume of patients treated by admitting physician and mortality after acute myocardial infarction. Jama 2001;285(24):3116–22.

[28] Vickrey BG, Rector TS, Wickstrom SL, Guzy PM, Sloss EM, Gorelick PB, et al. Occurrence of secondary ischemic events among persons with atherosclerotic vascular disease. Stroke 2002;33(4):901–6.

[29] Peterson ED, Shaw LK, DeLong ER, Pryor DB, Califf RM, Mark DB. Racial variation in the use of coronary-revascularization procedures. Are the differences real? Do they matter? N Engl J Med 1997;336(7):480–6.

[30] Aronsky D, Ransom J, Robinson K. Accuracy of references in five biomedical informatics journals. J Am Med Inform Assoc 2005;12(2):225–8.

[31] McGuinness DL, Harmelen Fv. OWL Web Ontology Language Overview 2004. <http://www.w3.org/TR/owl-features/>.

[32] International Health Terminology Standards Development Organisation. SNOMED-CT Data Structions Specification v1.0. 2010. <http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/Publications/Tools/SNOMED_CT_Release_Format_2_-_Data_Structures_Specification_v1.0.pdf> [03.10.10].

[33] McNemar I. Note on the sampling error of the difference between correlated proportions or percentages. Psychometrika 1947;12:153–7.

[34] Shrout PE, Fleiss JL. Intraclass correlations: uses in assessing rater reliability. Psychol Bulletin 1979;86:420–7.

[35] SPSS Inc. 233 S. Wacker Drive, 11th floor, Chicago, Illinois 60606; 2006.

[36] International Health Terminology Standards Development Organisation. Mapping SNOMED CT to ICD-10 Project [accessed 07.12.10].

[37] International Health Terminology Standards Development Organisation. Classifications and Cross-Mapping. <http://www.ihtsdo.org/fileadmin/user_upload/Docs_01/Members/Classifications___Cross-mapping.HazelB.pdf> [accessed 07.12.10].