Downloaded from jamia.bmj.com on July 1, 2010 - Published by group.bmj.com

Implementing description-logic rules for SNOMED-CT attributes through a table-driven approach

Prakash M Nadkarni and Luis A Marenco

JAMIA 2010 17: 182-184
doi: 10.1136/jamia.2009.001792

Updated information and services can be found at:
http://jamia.bmj.com/content/17/2/182.full.html

These include:

Supplemental Material http://jamia.bmj.com/content/suppl/2010/03/04/17.2.182.DC1.html

Email alerting service Receive free email alerts when new articles cite this article. Sign up in the box at the top right corner of the online article.

Notes

To order reprints of this article go to:
http://jamia.bmj.com/cgi/reprintform

To subscribe to *Journal of the American Medical Informatics Association* go to:
http://jamia.bmj.com/subscriptions

Downloaded from jamia.bmj.com on July 1, 2010 - Published by group.bmj.com

Implementing description-logic rules for SNOMED-CT attributes through a table-driven approach

Prakash M Nadkarni,[1] Luis A Marenco[2]

▸ Supplementary appendix are published online only at http://jamia.bmj.com/content/vol17/issue2

[1]Center for Health Research, Geisinger Health Systems, Danville, Pennsylvania, USA
[2]Center for Medical Informatics, Yale University School of Medicine, New Haven, Connecticut, USA

Correspondence to
Dr Prakash M Nadkarni, Center for Health Research, Geisinger Health Systems, Danville, PA 17821, USA; PMNadkarni@geisinger.edu

Received 29 June 2009
Accepted 28 November 2009

ABSTRACT

Maintaining a large controlled biomedical vocabulary requires ensuring the content's internal consistency. This is done through rules, specified by the vocabulary's curators, which denote how the vocabulary's concepts should be defined. When individual organizations deploy such vocabularies, local concepts are typically added and linked to concepts in the main vocabulary: the process of maintaining and linking local content should follow the same rules. The operation of content-maintenance software can be facilitated by maintaining such rules in computable form. In this paper, we demonstrate how to implement computable rules for attribute usage in SNOMED CT using a table-driven approach where a given rule is expressed as one or more rows in a table and is consulted by generic code. This approach, which is tailored to database implementations, is computationally efficient and allows new attribute-definition rules to be created as data while needing minimal or no code modification.

INTRODUCTION: DESCRIPTION LOGIC, SNOMED CLINICAL TERMS CONCEPT HIERARCHY AND ATTRIBUTES

The current version of the Systematic Nomenclature of Medicine Clinical Terms (SNOMED CT),[1] now managed by the International Health Terminology Standards Development Organization (IHTSDO) is designed to use 'Description Logic' (DL) as the basis for controlled vocabulary use and maintenance. DLs are a family of knowledge-representation languages used for operating on terminological data. While less expressive than some alternative formalisms, such as first-order logic, DLs come with guarantees of computational tractability.[2] Individual DL implementations differ in the operations that they support. Operations such as concept union, negation, intersection and subsumption (the last is discussed below) apply generally. Concepts can also have *properties* and *roles* (also called *relationship types* or *attributes*), which are typically specific to a given terminology's problem domain: for example, SNOMED CT, which deals with clinical medicine, has attributes such as 'pathological process' and 'has specimen'.

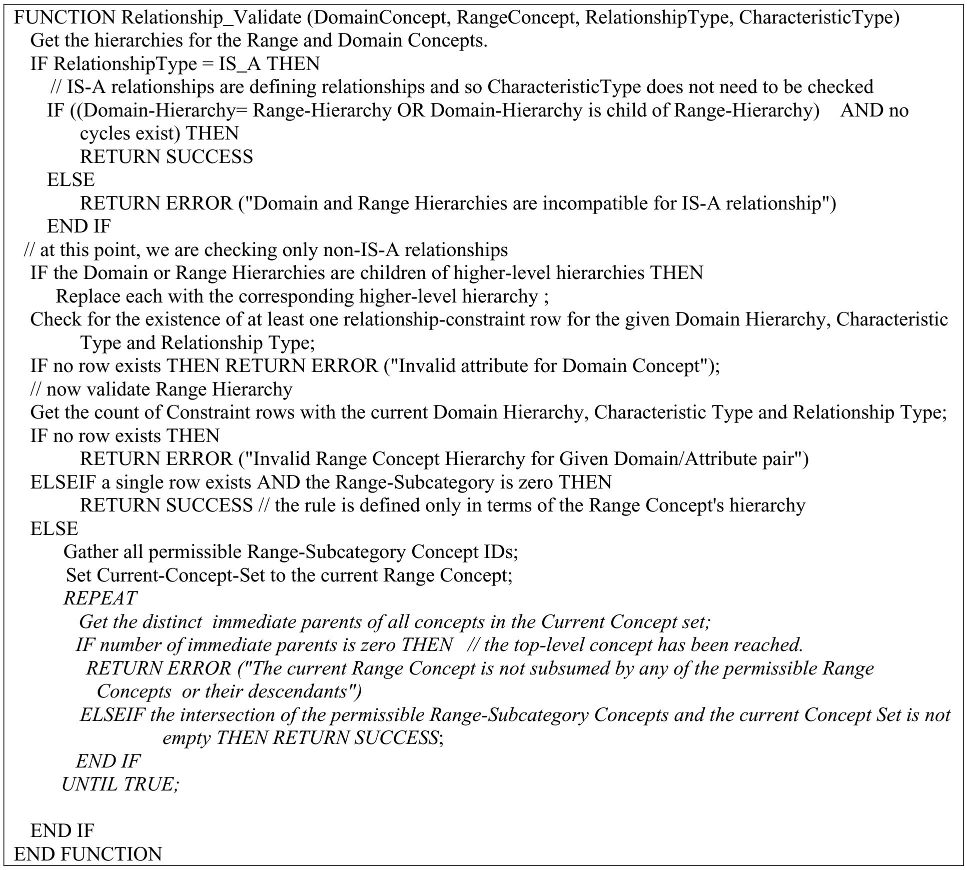
SNOMED CT is approximately tree-structured: all concepts descend from a 'SNOMED CT Concept', with 'child' concepts linked to parent concepts through 'IS-A' links, though many concepts descend from more than one parent. Every concept is labeled with the category that it belongs to, by using a parenthesized expression at the end of the concept's fully specified name, for example, 'Cellulitis (disorder)'. Categories, which are themselves concepts, lie in the upper part of the tree. Twenty of the 43 SNOMED CT categories are children of SNOMED CT Concept, but the rest are children of other hierarchies. For example, the categories 'Person', 'Occupation' and 'Life Style' are children of the category 'Social Concept', and 'Disorder' is a child of the category 'Clinical Finding'.

In a *relationship*, an *attribute* links a concept-pair. The type of relationship is defined by a 'Characteristic Type'. We consider two types in this paper: *defining relationships* specify concepts' *semantics*, and *qualifying relationships* specify allowable concepts that can be used to *refine* a concept's meaning. For concepts from a given hierarchy and for a given characteristic type, specific attributes apply. The hierarchy to which a defining attribute applies is termed the *Domain* (eg, 'Finding Site' applies to 'Clinical Finding') and the *Range* of an attribute is the set of concepts (and their descendants) that are permissible for that attribute's value, for example, the range of the attribute Finding Site for a Clinical Finding must be a member of the hierarchy 'Body Structure'. Chapter 4 of the 2008 SNOMED CT User's Guide[3] lists rules for permissible defining attributes for individual hierarchies, and the permissible Domains for each attribute. In about 60% of the cases, the Ranges are hierarchies: for the rest, one or more lower-level concepts, often belonging to separate hierarchies, constitute the Range. The set of rules is not complete, and may evolve over time. Further, some rules, for example, those related to measurement procedures, are documented but not currently modeled. Finally, rules are described currently only in prose: they are not computable. We describe a straightforward computable (tabular) representation and a table-driven algorithm that uses it for Relationship validation: this has been implemented in an operational SNOMED CT vocabulary server at both authors' institutions.

DESCRIPTION OF APPROACH

For a new Relationship to be valid, both Attribute and Range Concept must be permissible for a given Characteristic type and Domain concept. For IS-A relationships, we must also test for cycles: a cycle will exist if the child node in the relationship also happens to be in the parent's list of ancestors, as determined by recursive traversal of the hierarchy upward from the parent). Vocabularies of SNOMED CT's size are typically managed by vocabulary servers that use relational database technology. Most vocabulary servers support cycle check, but at least one widely used server (Apelon DTS, now available as open-source[4]) does not support attribute/range checks.

Downloaded from jamia.bmj.com on July 1, 2010 - Published by group.bmj.com



```
FUNCTION Relationship_Validate (DomainConcept, RangeConcept, RelationshipType, CharacteristicType)
  Get the hierarchies for the Range and Domain Concepts.
  IF RelationshipType = IS_A THEN
     // IS-A relationships are defining relationships and so CharacteristicType does not need to be checked
     IF ((Domain-Hierarchy= Range-Hierarchy OR Domain-Hierarchy is child of Range-Hierarchy)    AND no
          cycles exist) THEN
          RETURN SUCCESS
     ELSE
          RETURN ERROR ("Domain and Range Hierarchies are incompatible for IS-A relationship")
     END IF
 // at this point, we are checking only non-IS-A relationships
  IF the Domain or Range Hierarchies are children of higher-level hierarchies THEN
      Replace each with the corresponding higher-level hierarchy ;
  Check for the existence of at least one relationship-constraint row for the given Domain Hierarchy, Characteristic
          Type and Relationship Type;
  IF no row exists THEN RETURN ERROR ("Invalid attribute for Domain Concept");
  // now validate Range Hierarchy
  Get the count of Constraint rows with the current Domain Hierarchy, Characteristic Type and Relationship Type;
  IF no row exists THEN
          RETURN ERROR ("Invalid Range Concept Hierarchy for Given Domain/Attribute pair")
  ELSEIF a single row exists AND the Range-Subcategory is zero THEN
          RETURN SUCCESS // the rule is defined only in terms of the Range Concept's hierarchy
  ELSE
       Gather all permissible Range-Subcategory Concept IDs;
       Set Current-Concept-Set to the current Range Concept;
       REPEAT
          Get the distinct  immediate parents of all concepts in the Current Concept set;
         IF number of immediate parents is zero THEN   // the top-level concept has been reached.
           RETURN ERROR ("The current Range Concept is not subsumed by any of the permissible Range
             Concepts  or their descendants")
          ELSEIF the intersection of the permissible Range-Subcategory Concepts and the current Concept Set is not
                      empty THEN RETURN SUCCESS;
         END IF
       UNTIL TRUE;

  END IF
END FUNCTION
```

Figure 1 Pseudo-code for Validating a Relationship. Comments are indicated with the double-slash character (//). The computationally expensive step in the algorithm, the recursive traversal of the Relationship tables, is indicated by italics in the pseudo-code. Note that this step is postponed to the final part of the algorithm: the earlier part of the code uses opportunities to bypass the check by using the precomputed hierarchy values for the candidate Range and Domain concepts. Also of note, the step '*Get the distinct immediate parents of all concepts in the Current Concept set*' can be performed in a single SQL query.

The basic DL operation involved in Relationship validation is *subsumption*: that is, checking, by following a chain of IS-A links, whether one concept is an ancestor (or descendant) of another. An ancestor *subsumes* a descendant: for example, 'Malignant neoplastic disease (disorder)' subsumes 'Primary malignant neoplasm of female breast (disorder)'. Subsumption is effected through recursive query of the Relationships table. For a large number of relationships (1.2 million+in SNOMED CT, July '08 release), recursive table traversal is potentially expensive computationally, and must be used as a last resort. The database design optimization we employ to minimize traversal is standard for tree structures: additionally record, along with each concept, the highest-level ancestor that will be consulted frequently. Here, we record the *hierarchy concept ID* for every SNOMED CT concept explicitly using an additional column in the Concepts table. We precompute this value through a batch script that matches the trailing parenthesized expression in the concept's fully specified name against a list of hierarchy description-ID pairs. This allows a subsumption-checking shortcut: if the two concepts being checked belong to unrelated hierarchies (eg, one is a Product and the other is a Clinical Finding) one concept cannot subsume the other. (While chapter 6 of the *SNOMED CT technical implementation guide* also mentions other alternatives, such as computing and storing the entire chain of parent concept IDs along with each concept, we have found that this simple approach achieves a reasonable space-for-time tradeoff.)

Even when traversal is mandated, traversal direction is important. In a mostly hierarchical vocabulary, it is much more efficient to go from descendant to ancestor than vice versa, because the number of concepts that need inspection will typically decrease or stay unchanged with each successive search level. For the validation check, we only need to move upward from a candidate concept, and test at each level if the target concept is encountered.

Representing rules

We record rules ('constraints') in a *Relationship_Constraints* table with the integer columns: CharacteristicType, Domain_Category, RelationshipType, Range_Category and Range_Subcategory. All columns together constitute the primary key. In the MS-Access based user interface for creating and editing rules, which is a front end to an MS SQL Server schema, the first four columns are presented as pull-downs. CharacteristicType's values are defined in the SNOMED CT User Guide (0=Defining, 1=Qualifier, 2=Historical, 3=Additional). RelationshipType (Attribute) is based on the set of (65) distinct Relationship Types extracted from the Relationships table. Domain and Range Category columns are based on the list of unique hierarchies. Range_Subcategory records either zero (if the rule defines the Range only at the Hierarchy level) or a Concept ID if the rule specifies a non-hierarchy concept and its descendants. In the latter case, we still precompute and record that concept's hierarchical ancestor in the Range Category to allow the subsumption-check shortcut. If a rule specifies multiple concepts for the Range, we create one row for each Range concept. This table also stores IS-A relationship information between hierarchies, to allow the following check: *a concept can be a direct 'child' of another, that is, linked by an IS-A relationship, only if both concepts belong to the same hierarchy, or if the child concept's hierarchy is a child of the parent concept's hierarchy*. Thus, a 'disorder' may descend from a 'clinical finding', but not vice versa. Strictly speaking, IS-A hierarchy-relationship information is redundant with that already in the Relationships table: however, the *Relationship_Constraints* table's small size (currently 117 rows capture all of the specified rules for

Downloaded from jamia.bmj.com on July 1, 2010 - Published by group.bmj.com

defining and qualifying attributes) allows in-memory caching and rapid search.

VALIDATING A NEW RELATIONSHIP: MODIFIED ALGORITHM

The modified algorithm for validating a Defining or Qualifying Relationship is described in figure 1.

Testing of the framework

In the course of compiling the rules and testing with data, we found some anomalies in the July '08 SNOMED CT Release which merit curatorial verification. For the domain 'Clinical Finding', it is stated that for the attribute 'Pathological Process', the range should be 'Autoimmune'. This rule is possibly overly specific: autoimmune processes are not the only pathological ones. Another overly specific rule applies to the Domain 'Clinical Findings', Attribute 'Severity' while the Domain is stated to be 'Severities', this precludes the use of disease staging (eg, tumor staging), and the range should possibly be either redefined to be the slightly more general concept 'Ranked Categories', or else multiple ranges should be specified.

Finally, while determining the IS-A subsumption rules for hierarchies, we found a cyclical relationship in the data. While the top-level concept 'Regimes and therapies' is a child of the top-level concept 'Procedure' (understandably, because Therapies are a specific kind of Procedure), the concept '229319000|Mobilizing of Body part (regime/therapy)' has six procedure children, for example, '173995005|Mobilization of intestine'. This may be due to a lack of classification consistency: either the parent concept should be a procedure, or all its children should be regimes/therapies. (Re-running the algorithm on the July '09 SNOMED CT release, we found that this error had been fixed by moving 'Mobilization of Intestine' under '74923002|Mobilization (procedure)'. The original curation error possibly occurred because 'mobilization' is polysemic, referring either to surgical mobilization, a procedure, or to non-surgical, physical-therapy mobilization (attempted increase in motion range).

DISCUSSION

Table-driven methods are known to be simpler, more readily modifiable and more efficient than compound logic statements,[5] especially as the latter becomes progressively detailed over time: this is likely to happen with SNOMED CT, whose constraint rules for attributes are expected to evolve extensively with time. For example, it is intuitively obvious that certain relationship types in SNOMED CT, such as 'part of', apply to certain hierarchies (eg, body structures) but not others: these rules, however, have not yet been officially codified. An advantage of table-driven approaches is that new permissible-attribute/range rules can simply be added as new rows in the Constrains table. Table-driven approaches are widely used for decision-support scenarios such as detection of drug-drug interactions in Computerized Physician Order Entry/Pharmacy systems, where rules/constraints that are structurally homogeneous would number in the tens of thousands if expressed in procedural/rule form.

Our approach is a concrete implementation of a subset of the abstract logical model of IHTSDO's SNOMED CT Machine Readable Concept Model.[6] This model is not currently public because it is still in draft form and evolving. However, access to it is available to IHTSDO members (membership is free and can be applied for by email). The draft model's class diagram splits the Relationship-Constraints table into several classes in a highly normalized design: this becomes necessary as the list of constraints becomes large. The full draft model is highly sophisticated, considering the possibility of complex constraints based not just on individual relationships, but sets of relationships (eg, if relationship type A exists for a particular concept-pair, then a record with relationship type B must also exist). Currently, the draft model does not accommodate the possibility of tailoring the constraints to the Characteristic Type of the relationship, as ours does: such an enhancement, however, can be readily effected.

Appendix 1, available as an online data supplement (http://jamia.bmj.com/content/vol17/issue2), contains (1) a detailed schema diagram, (2) a URL to a downloadable MS-Access database containing the Rule data, a user interface for creating them, (3) source code for Microsoft SQL Server with a Visual Basic.NET stored function implementation.

Funding This research is supported in part by CDC Grant U10 OH008225-03, NIH Grant R01 DA021253 and institutional funds from Geisinger Health System.

Competing interests None declared.

Provenance and peer review Not commissioned; externally peer reviewed.

REFERENCES

1. **International Health Terminology Standards Development Organization.** SNOMED Clinical Terms (SNOMED CT). 2009. http://www.snomed.org (accessed Feb 1 2009).
2. **Russell S,** Norvig P. *Artificial intelligence: a modern approach*. 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 2002.
3. International Health Terminology Standards Development Organization. Chapter 4: Attributes used in SNOMED-CT. SNOMED Clinical Terms (SNOMED CT) User's Guide; 2008.
4. **Apelon Inc.** Apelon Distributed Terminology System. 2009. <sourceforge.net/projects/apelon-dts/> (accessed Jun 4 2009).
5. **McConnell S.** *Table-driven methods. Code complete*. 2nd ed. Redmond, WA: Microsoft Press, 2004.
6. **International Health Terminology Standards Development Organization (IHTSDO).** Working Project Groups: SNOMED machine-readable concept model. 2009. http://www.ihtsdo.org/about-ihtsdo/governance-and-advisory/working-groups/project-groups/ (accessed Oct 20 2009).