Exploring Complexity in Health: An Interdisciplinary Systems Approach
A. Hoerbst et al. (Eds.)

doi:10.3233/978-1-61499-678-1-466

Implementation of an Execution Engine for SNOMED CT Expression Constraint Language

V. M. GIMÉNEZ-SOLANO[a,1], J. A. MALDONADO[a,b], S. SALAS-GARCÍA[b], D. BOSCÁ[a] and M. ROBLES[a]
[a] *Institute for the Applications of Advanced Information and Communication Technologies (ITACA), Universitat Politècnica de València, Valencia, SPAIN*
[b] *VeraTech for Health, Valencia, SPAIN*

Abstract. The need to achieve high levels of semantic interoperability in the health domain is regarded as a crucial issue. Nowadays, one of the weaknesses when working in this direction is the lack of a coordinated use of information and terminological models to define the meaning and content of clinical data. IHTSDO is aware of this problem and has recently developed the SNOMED CT Expression Constraint Language to specify subsets of concepts. In this paper, we describe an implementation of an execution engine of this language. Our final objective is to allow advanced terminological binding between archetypes and SNOMED CT as a fundamental pillar to get semantically interoperable systems. The execution engine is available at http://snquery.veratech.es.

Keywords. Semantic interoperability, archetype, terminological binding, SNOMED CT subsets

1. Introduction

In health institutions generally coexist many different information systems, so information is stored in separate islands. When patient information is needed, querying only a part of the information is a potential risk for the patient. There are, thus, great benefits in the design and construction of semantically interoperable information systems.

Reaching a high level of semantic interoperability is not an easy task. There are three fundamental pillars on which it sits: on the one hand, information models, which consist of electronic health record storage and communication standards (HL7 CDA, EN ISO 13606 and openEHR reference models), and detailed clinical information models (archetypes, templates, visual interfaces, etc.); on the other hand, terminological models (clinical terminologies such as SNOMED CT, LOINC or ICD, among others). To achieve a high level of semantic interoperability it is necessary to bind information models to clinical terminologies. There exist two types of

[1] Corresponding Author: vigiso@upv.es. This work has been partially funded by the *Programa de Ayudas de Investigación y Desarrollo* (PAID) of the Universitat Politècnica de València and by the Spanish Ministry of Economy and Competitiveness and the EU FEDER programme through grants TIN2014-53749-C2-1-R and PTQ-12-05620.

terminological binding. First, semantic binding, which gives unequivocal meaning to the information structures contained into the information model by means of a link between an element of the model and a pre- or post-coordinated term of the terminology. Second, content binding, which constrains the set of possible coded values or model meanings of a data element within an information model.

Content binding requires a mechanism to specify subsets intensionally, in opposition to extensionally. One way to define such subsets is using a declarative language to interrogate the substrate of the terminology. The grammar of this language must allow specifying the terms to be selected and included into the subset and how they must be related. Moreover, their operators must be aligned with the logic model of the terminology. SNOMED CT Expression Constraint Language [1] is a recent development by IHTSDO that enables the intensional definition of sets of clinical meanings. In this work, we describe the design and development of an execution engine for this language. Our final objective is to enable advanced terminology binding in archetypes.

2. Methods

2.1. Storage of the SNOMED CT database

For the persistence of SNOMED CT we have used Neo4J, a graph-oriented database software which comes with a powerful language called Cypher to query data stored in the graph. We have implemented a Java loading module in order to import and transform into a graph the SNOMED CT text files delivered by IHTSDO. The last imported SNOMED CT release is INT 31/07/2015. It contains a total of 316833 nodes (SNOMED CT concepts) and more than a million and a half relationships between those nodes (both attributes and IS A specialization). Due to the breadth of the graph, the transitive closure has been calculated in order to expedite the query process (more than five million relationships in total).

Since the terminology SNOMED CT is graph-structured –directed and acyclic– it makes sense to store it in a graph-oriented database rather than traditional database model, such as relational. Graph databases bring a number of potential advantages, such as a higher performance when retrieving data from a query, or the flexibility to add new nodes and relationships to the graph without affecting existing queries thanks to the additive nature of the graphs [2]. Figure 1 shows a fragment of SNOMED CT content. Dotted IS A relationships indicate that there exist nodes in that path but they are not represented in order to simplify de figure.

2.2. SNOMED CT Expression Constraint Language logical model

There are three types of SNOMED CT Expression Constraints: simple, refined and compound. To define a simple expression constraint, a constraint operator is applied to a focus concept in order to select itself and its descendants ("<<"), only its descendants ("<"), self and ascendants (">>"), only ascendants (">") or members of a reference set ("^"). Note that it is also possible to reference all SNOMED CT substrate with the "*" symbol. To define a refined expression constraint it is necessary to apply a refinement –which is preceded by the symbol ":"– to a simple expression constraint, comprising one or more attribute-value pairs. An attribute is a relationship between concepts of two hierarchies and may be preceded by a cardinality ("[min..max]"), a reverse operator ("R") or a constraint operator ("<<, <"). There are several options for the value of the attribute, namely: a simple expression constraint, a compound expression constraint, a refined expression constraint (leading to expression constraint nesting) and a numerical or textual value. Attribute-value pairs may be combined with conjunction ("AND" or ",") and disjunction ("OR") operators to establish sets of pairs, groups or both. Groups have a special processing and they are predefined in the logical definition of the concepts. Moreover, groups may be preceded by a group cardinality. Finally, compound expression constraints can defined by means of conjunction, disjunction or exclusion ("MINUS") between simple, refined or both expression constraints.

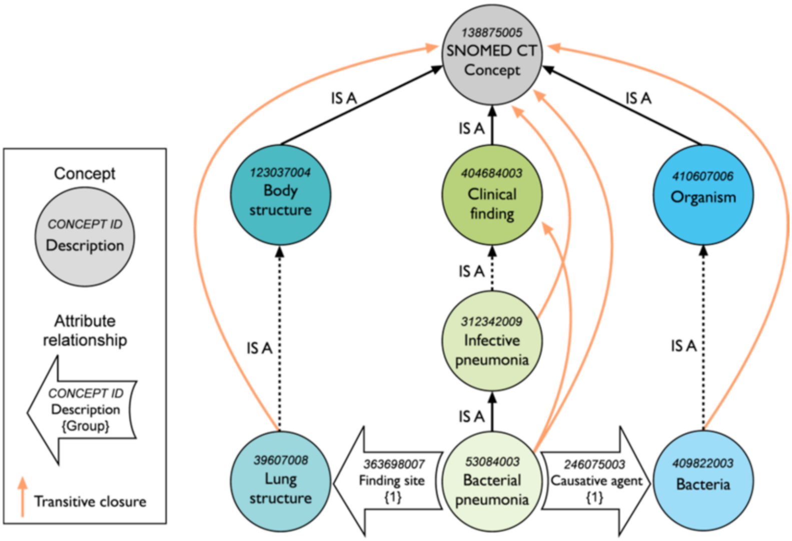

Figure 1. Example of some SNOMED CT nodes and relationships

Table 1 shows a numbered list of expression constraint examples with a textual definition of each subset. For instance, expression 1 is a simple expression constraint where the operator "<" (only its descendants) is applied to the focus concept 404684003 |Clinical finding|. Expression 4 is a refined expression constraint where the refinement is applied to the descendants of the focus concept 404684003 |Clinical finding| and consists of two attribute-value pairs, one of them (363698007 |Finding site|) with a cardinality of "[2..*]" (greater or equal than two). Finally, expression 6 uses the reverse operator ("R") in order to obtain the descendants or self ("<<") of the focus concept 105590001 |Substance| which cause any clinical finding (note that the expression constraint would be meaningless without the reverse operator since any descendant of the concept 105590001 |Substance| cannot be caused by any concept).

3. Results

We have developed SNQuery, a SNOMED CT Expression Constraint execution engine. It is implemented in Java and is available at http://snquery.veratech.es. SNQuery execution engine validates the syntax (both brief and full syntax) of input expressions.

Note that the descriptions of the concepts in the expression constraints are optional and the engine does not take it into account. The Machine Readable Concept Model (semantic rules) is basic to perform the semantic validation of expressions (e.g. the finding site of a clinical finding cannot be a substance) is under development at this moment by IHTSDO. As a consequence, the semantic validation, although implemented, is not available until a stable MRCM is ready. The engine returns the list of concept identifiers, along with its Fully Specified Name, that satisfy the expression constraint.

Table 1. Examples of SNOMED CT Expression Constraints

No.	Definition	Expression Constraint
1	Subset of all descendants of *clinical finding*.	< *404684003 |Clinical finding|*
2	Subset of all *clinical findings* located in the *pulmonary valve structure* or any of its descendants.	< *404684003 |Clinical finding| : 363698007 |Finding site| = << 39057004 |Pulmonary valve structure|*
3	Subset of all *clinical findings* located in the *pulmonary valve structure* or any of its descendants and morphologically associated with any type of *stenosis*.	< *404684003 |Clinical finding| : 363698007 |Finding site| = << 39057004 |Pulmonary valve structure|, 116676008 |Associated morphology| = << 415582006 |Stenosis|*
4	The same subset of example 3 but with at least two locations.	<*404684003 |Clinical finding|: [2..*] 363698007 |Finding site|=<<39057004 |Pulmonary valve structure|, 116676008 |Associated morphology| = << 415582006 |Stenosis|*
5	The same subset of example 3 but with an associated morphology different to any type of *stenosis*.	< *404684003 |Clinical finding| : 363698007 |Finding site| = << 39057004 |Pulmonary valve structure|, 116676008 |Associated morphology| != << 415582006 |Stenosis|*
6	Subset of all *substances* which are causative agents of any *clinical finding*.	<<*105590001|Substance|: R 246075003 |Causative agent|=<<404684003|Clinical finding|*

For execution of expression constraints we translate them into a set of Cypher queries. The Cypher queries are executed against the SNOMED CT substrate and the subset of concepts intensionally defined by the expression constraint is returned –in tabular form.

These results are linked to the IHTSDO SNOMED CT browser and can be ordered –by identifier and description– and be exported to different formats. SNQuery web interface contains an example section with some expression constraints extracted from the language specification that can be easily executed or used as basis for building new expressions. SNQuery displays a list with the last ten executed expression constraints and offers the possibility of re-execute them. It is also possible to enhance the expression constraint by adding or replacing the existing terms with Fully Specified Names. Finally, the engine can convert the expression constraint into brief and full syntax.

3.1. SNQuery execution times

An analysis of the SNQuery execution times has been carried out on a Windows Server 2008 R2 Datacenter, Intel Core i7-2600K 3.40 GHz and 16 GB RAM. Table 2 shows the execution times (in milliseconds) of the six examples in Table 1 plus a query against all SNOMED CT substrate (i.e. "*"). Specifically each execution time has been calculated as the average of running ten times each query. Description time is the time required to recover the fully specified name of the resulting concepts, therefore it is linear with its number. It should be noted that SNQuery uses an eager-loading approach even for “*” case. The most obvious conclusion is that execution time does not only depend on the number of retrieved concepts but also on the complexity of the expression constraint and the size of the hierarchies involved.

Table 2. Execution times (in milliseconds) of SNQuery engine

Query (see Table 1)	**1**	**2**	**3**	**4**	**5**	**6**	**“*”**
Results (concepts)	*103226*	*116*	*22*	*2*	*88*	2422	*316833*
Evaluation (ms)	**3518**	**909**	**1670**	**2064**	**13353**	**5198**	**7160**
Descriptions (ms)	3520	96	47	37	71	575	7594
Total (ms)	7038	1005	1717	2101	13424	5773	14754

4. Conclusions and future work

The development of an execution engine for the SNOMED CT Expression Constraint Language facilitates the intensionally definition of subsets of concepts. The availability of these subsets is useful to bind content between clinical information models such as archetypes and SNOMED CT, which is a necessary step to achieve a high level of semantic interoperability of electronic health records (EHR). Graph databases in general support queries that traverse the graph to an undefined depth and over undefined relationship (connections) [3]. These features make them a good option for managing and querying SNOMED CT.

Our next step is to integrate the execution engine into the software platform LinkEHR [4] for the modeling and normalization of EHR based on archetypes. We would like to extend this platform with advanced terminology binding, supporting both semantic and content binding between archetypes and SNOMED CT.

But the usefulness of these expression constraints does not end here. It is expected, as a continuation of this work, to enrich archetypes with advanced data consistency rules. Currently, it is not possible to define rules involving more than one archetype entity. These rules may be useful, for example, to check whether the value of a node is in accordance with the value of another node, or to specify the value of a node as the result of applying an operation on a set of nodes. Furthermore, it would be also possible to specify rules involving the result of a SNOMED CT constraint expression in order to define semantic constraints or conceptual abstractions inside archetypes.

References

[1] International Health Terminology Standards Development Organisation (IHTSDO). *SNOMED CT Expression Constraint Language Specification and Guid*e, v1.0, 2015.

[2] Robinson I, Webber J, Eifrem E, Graph Databases: new opportunities for connected data. O’Reilly, Second Ed. pp. 8-9, Jun. 2015.

[3] Campbell W.S., Pedersen J, McClay J.C., Rao P., Bastola D. and Campbell J.R., An alternative database approach for management of SNOMED CT and improved patient data queries*, Journal of Biomedical Informatics*, vol. **57**, no. 1, pp. 350-357, Oct. 2015.

[4] Maldonado J.A., Moner D, Boscá D, Fernández-Breis J.T., Angulo C, and Robles M, LinkEHR-Ed: a multireference model archetype editor based on formal semantics., *Int. J. Med. Inform.*, vol. **78**, no. 8, pp. 559-70, Aug. 2009.