Auditing SNOMED Relationships Using a Converse Abstraction Network

Duo Wei, MS[1], Michael Halper, PhD[2], Gai Elhanan, MD[1], Yan Chen, PhD[3], Yehoshua Perl, PhD[1], James Geller, PhD[1], Kent A. Spackman, MD, PhD[4]
[1]NJIT, Newark, NJ; [2]Kean University, Union, NJ; [3]BMCC (CUNY), NYC; [4]IHTSDO, DK

Abstract
In SNOMED CT, a given kind of attribute relationship is defined between two hierarchies, a source and a target. Certain hierarchies (*or subhierarchies*) *serve only as targets, with no outgoing relationships of their own. However, converse relationships—those pointing in a direction opposite to the defined relationships—while not explicitly represented in SNOMED's inferred view, can be utilized in forming an alternative view of a source. In particular, they can help shed light on a source hierarchy's overall relationship structure. Toward this end, an abstraction network, called the* converse abstraction network (CAN)*, derived automatically from a given SNOMED hierarchy is presented. An auditing methodology based on the CAN is formulated. The methodology is applied to SNOMED's Device subhierarchy and the related device relationships of the Procedure hierarchy. The results indicate that the CAN is useful in finding opportunities for refining and improving SNOMED.*

Introduction

Reliable design and accurate knowledge representation are desirable features of modern terminologies such as SNOMED CT (SNOMED, for short)[1]. However, due to the size and complexity of SNOMED, quality assurance is a demanding task.

In this paper, we focus on the issue of auditing the attribute relationships of a SNOMED hierarchy (or subhierarchy) with an eye toward finding opportunities for their refinement and improvement. A given attribute relationship (simply "relationship" hereon) is defined between a source hierarchy and a target hierarchy. A particular hierarchy may serve as a source for one relationship and the target for another. Certain hierarchies have no outgoing relationships of their own. We call such a hierarchy a *strict target hierarchy* (or *subhierarchy*, when appropriate).

Even though a strict target hierarchy has no relationships, it does exhibit *converse relationships*—i.e., those pointing in the opposite direction to the existing incoming relationships. While these relationships are not explicitly represented in SNOMED's inferred view, available, for example, through the CLUE browser, they are, however, often utilized in data retrieval tasks or in the formation of expressions in clinical environments. They can be employed in providing an alternative view of a source hierarchy's relationship structure. A new kind of abstraction network, called the *converse abstraction network* (*CAN*), is introduced to represent and display a hierarchy's concepts according to their distribution of converse relationships. This network is automatically derived from the underlying inferred view of the concept network. The CAN offers a unique perspective on the source hierarchy's relationships that differs significantly from the original design view and therefore can bring unexpected structural features to light.

We avail ourselves of this unique perspective by defining an auditing methodology that utilizes the CAN and is applicable to the source hierarchy. The methodology is applied to the Device subhierarchy (of the Physical Object hierarchy) and the device relationships of the Procedure hierarchy. Potential improvements to the relationship configuration discovered through this process are presented.

Background

SNOMED is a comprehensive terminology with 383,230 (July 2008 release) concepts (including inactive) arranged in 19 (IS-A) hierarchies, such as Clinical Finding, Procedure, and Physical Object. Its relationships are connections between concepts that serve as definitional or qualifying elements. As noted, an occurrence of a given relationship extends from a prescribed source hierarchy's concept to a target hierarchy's concept. Converse relationships are not maintained explicitly.[1]

More than half of SNOMED's hierarchies are "strict target hierarchies," with only incoming relationships. The Device subhierarchy is an example. The Procedure hierarchy targets it with five defining relationships: *procedure device*, *using access device*, *direct device*, *using device*, and *indirect device*. Each describes devices associated with a particular procedure. *Procedure device* subsumes the others in a role hierarchy.[1] Our current analysis involves converse relationships derived from SNOMED's inferred view. Specifically, we use a non-nested transform of SNOMED's original description logic

(DL) representation available from the relationships table.

The issue of auditing SNOMED has received considerable attention. Ontological and linguistic analyses have been applied to its content.[2] Algorithmic approaches based on SNOMED's native DL representation were used to find inconsistencies[3] and missed synonymy.[4] Formal Concept Analysis was employed to investigate semantic completeness.[5]

In our own work, we have formulated various auditing methodologies based on two hierarchical abstraction networks called the area taxonomy and partial area taxonomy.[6,7] Both are derived from a hierarchy that is the source of relationships. The former presents a high-level view of the distribution of relationships within the hierarchy. The latter refines that view with groupings of concepts having common ancestry. We note that these taxonomies and their accompanying auditing are not appropriate for a strict target subhierarchy such as Device. In this paper, we present a new abstraction network that is applicable in such circumstances.

Methods

We define a new kind of abstraction network, called the converse abstraction network (CAN), on the side of the target hierarchy of relationships in SNOMED's inferred view. Unlike the taxonomies of our previous work, the CAN is *not* a purely hierarchical structure reflecting logical concept subsumption and relationship inheritance. In fact, inheritance is not a characteristic of the converse relationships that we derive. Therefore, such inheritance is not reflected in the CAN. Moreover, the partial areas defined with respect to the CAN do not necessarily have the level of hierarchical cohesion found in the partial areas previously derived, as will be described.

Let us start with the definition of *converse relationship* with respect to SNOMED's inferred view. After that, we define the notions of *area* and *partial area* in the context of a SNOMED hierarchy. From these, we define the CAN. Lastly, we introduce an auditing methodology based on the CAN.

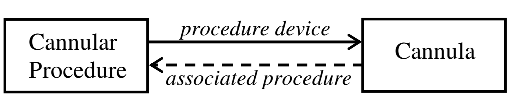

Figure 1. Example converse relationship

Consider the concepts *Cannular procedure* (from the Procedure hierarchy) and *Cannula* (from Device), connected by the relationship *procedure device* (Figure 1). We define the converse relationship of *procedure device* to be the relationship that reverses its direction, connecting *Cannula* to *Cannular procedure*. In this case, it is called *associated procedure* (see the dashed arrow in Figure 1). A converse relationship r' will have a name derived from its original relationship r.

We define two concept groupings for the converse relationships of a SNOMED target hierarchy. Let $r'_1, r'_2, \ldots, r'_n$ be converse relationships. We define the *area* of $r'_1, r'_2, \ldots, r'_n$ to be the set of concepts with exactly these converse relationships. An area is named by its unique set of relationships (written in braces). An example is the area {*used for access by proc*, *used by proc*} ("*proc*" short for *procedure*), a set of 48 concepts from the Device subhierarchy. One of its concepts is *Endoscope*, which is a target of two relationships, *using access device* and *using device*.

It is possible that some concepts within a hierarchy are not targets of any relationships at all. For these, we define an additional area, denoted ∅ (read "having no converse relationships"), to hold them. Collectively, the areas of a given hierarchy form a partition of that hierarchy. That is, each concept belongs to one and only one area.

The second grouping is derived from the first and is hierarchical in nature. Within an area ***A***, a concept is a *root* if none of its ancestors is also in ***A***. For each root *O* of ***A***, we define a set called the *partial area* containing *O* and all its descendants in ***A***. The partial area is denoted as *O*. For example, the concept *Endoscope* is a root of {*used for access by proc*, *used by proc*}. It and its 41 descendants (e.g., *Fetoscope*) in that area form a partial area.

In a taxonomy,[6,7] the subhierarchy residing in a partial area is completely connected. However, a partial area of a CAN may be disconnected. For example, *Ureteroscope* is in its grandparent *Endoscope*'s partial area. But its parent, *Urinary endoscope*, resides in an entirely different area, {*used for access by proc*}, thus upsetting the connectedness.

The areas and partial areas serve to give an indication of the converse relationship sources within a hierarchy and their associated hierarchical arrangement. For the purpose of visualization, we define a network structure based on the areas and partial areas. We refer to this directed network as the *converse abstraction network* (*CAN*). Each node of the CAN represents an area. Within an area node, we find embedded nodes, each of which represents a partial area. The edges of the CAN are defined between partial areas residing in different areas as follows. Let *O* be a root and let *P* be its parent. Recall that *P* resides in a partial area, say, L_P that must be in an area different from *O*'s. Then there exists an edge directed from partial area *O* to L_P. As examples, there are three partial areas *Urinary endoscope*, *Otoscope*, and *Rigid tracheoscope* in the area {*used for access by proc*}. The roots of the first two are children of *Endoscope*, and the root of the third is a grandchild of *Endoscope* via the parent *Rigid scope*. Thus, there is an edge from each of these three partial areas to the partial area *Endoscope*. The parent of *Endoscope* is *Scope AND/OR camera*, residing in the area ∅. As a special case, the edge in this circumstance goes from the partial area *Endoscope* to ∅. As it happens, the CAN is not a hierarchical network (e.g., a lattice).

The CAN provides a compact abstract view of the content of a hierarchy organized according to the concepts' sets of converse relationships and their IS-A arrangements. For example, there are 2,985 device concepts without any incoming relationships, and six medical balloon devices targeted by *using device* relationships.

The CAN's importance comes to light in the context of auditing relationships from one SNOMED hierarchy to another, target hierarchy. As in our previous work,[6,7] we are looking for unexpected structural features in the CAN that could possibly be manifestations of underlying problems. For example, the concepts in ∅ have no incoming relationships whatsoever. There are also general device concepts (e.g., *Catheter*) in small partial areas having many converse relationships, while their descendants (e.g., *Vascular catheter*) appear in partial areas with fewer such relationships. Such unexpected arrangements deserve attention from an auditor. In the auditing work, one needs to consider the original relationship targeting such (device) concepts and their related (procedure) source concepts. The goal in this is to find opportunities for refinement and improvement of SNOMED's relationship structure; or, in fact, to further validate the existing structure.

Results

The Device subhierarchy exhibits a total of five converse relationships, mentioned above, directed to the Procedure hierarchy. A portion of its CAN is shown in Figure 2. Overall, it has 22 areas and 260 partial areas. The number in parentheses in a partial area node indicates its number of member concepts. The CAN of the Device subhierarchy is not a pure hierarchical structure. In fact, one can see edges emanating from the same partial area (e.g., *Biliary T-tube*) pointing upward and downward. However, we do lay the CAN out in levels and color-code them according to the number of relationships of the various areas. For example, the green area {*used for access by proc*} is on level one with four partial areas (e.g., *Urinary endoscope*) and five concepts. The pink area with all five converse relationships is on level five. It has one partial area *Catheter*. If not all partial areas are shown for an area, then the numbers of concepts and partial areas are written in parentheses. For example, {*used for access by proc*, *used by proc*} has 48 concepts and six partial areas. The largest partial area is *Endoscope* (42).

A review of Device's CAN reveals many interesting structural features, enumerated in the following. (1) The vast majority of devices (2985, 78%) are not being pointed to by any procedures. (2) An edge pointed downward may exist from a partial area with fewer relationships to a partial area with more. E.g., *Urinary endoscope* (level 1) has an edge to *Endoscope* (level 2). (3) Many partial areas are singletons, meaning they contain only one concept each. (4) Some small partial areas are of a very broad nature, such as *Catheter* and *Drain*. (5) Certain partial areas include extremely high-level, non-specific devices, such as *Device* itself, which subsumes all the devices in all the CAN's partial areas. (6) Certain partial areas are pointed to by one or very few procedures. (7) Devices of a similar nature, such as *Venous catheter* and *Arterial catheter*, reside in different areas.

These features were used to focus the auditing efforts on certain concepts and relationships of the Procedure hierarchy (targeting Device), and thus provided opportunities to find potential errors that would probably not be detected directly from the Procedure hierarchy. In the following discussion, we provide examples pertaining to these observed features and review their value as indicators of potential errors or improvements in the modeling.

Discussion

The CAN exhibits properties that differentiate it from our previous partial area taxonomies.[6,7] For example, with partial area taxonomies, there is inheritance of relationships among partial areas along the *child-of* hierarchy. No such inheritance is guaranteed for the CAN. For example, *Vascular Catheter* is in the red area {*used for access by proc*, *acted on directly by proc*, *used by proc*} and has an edge directed to the *Catheter* partial area (in the pink, level-five area). Two of the relationships are not appearing for *Vascular Catheter*. In the figure, this is manifested by the edge pointing downward, while in the partial area taxonomy the *child-of* relationships point up to areas with fewer relationships.

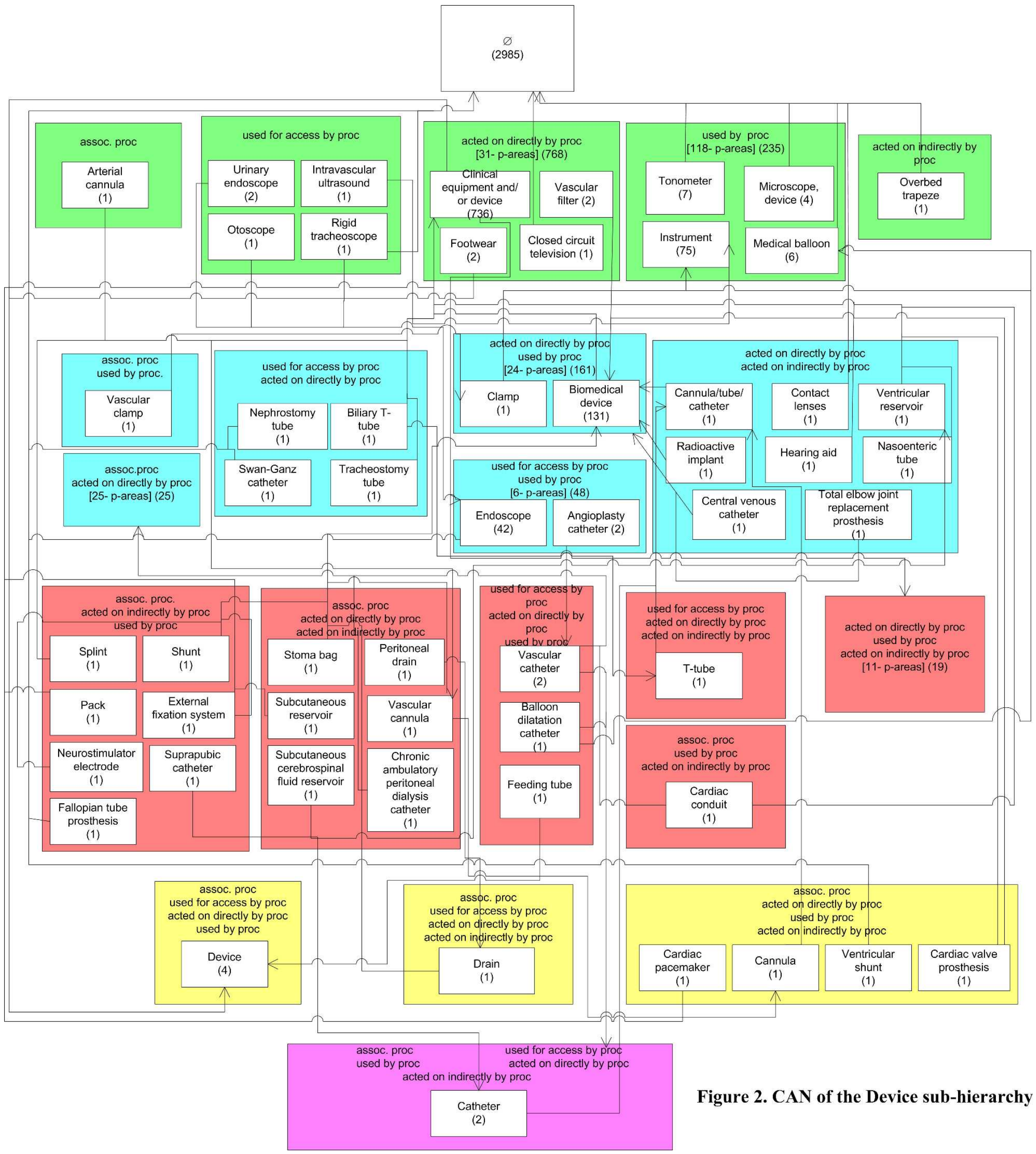

Figure 2. CAN of the Device sub-hierarchy

An unexpected observation, at least to a SNOMED novice, is that most devices (78%) are in the area ∅, and have no relationships targeted at them. However, this can be explained by two SNOMED qualities: *sufficient definitions* and *refinability*. Sufficient definitions can be achieved without the use of the most specific concept as an attribute value or with specific relationship types. For example, *Intracavitary brachytherapy* is a procedure that does not have any device relationship. However, the procedure achieves sufficient definition by using another relationship, *method* to *Brachytherapy – action*. Due to this, *Brachytherapy implant* does not have any incoming relationships and resides in ∅.

Additionally, the Device subhierarchy does not use inheritance of attribute values but instead relies on the notion of *refinability*. A refinability value is assigned to every relationship type between a pair of concepts, usually at some ancestral level. As a result, many descendant concepts will reside in ∅. For example, *Charnley total hip prosthesis* resides in ∅ along with its siblings. However, the procedure *Total hip replacement* points to the parent device, *Total hip replacement prosthesis*. Thus, the procedure achieves sufficient definition while allowing the device to be refined as needed by the procedure's descendants.

However, from a user perspective, as in a decision-support system or other terminology-driven systems, such an arrangement may be perceived as deficient. If one wants to select a specific endoscope for a gastrointestinal procedure while that procedure is sufficiently defined with the device *Endoscope*, one may be able to select, say, *Otoscope* as the device. And since the Device subhierarchy does not have any outgoing relationships, the devices cannot be defined by the body systems or organs they act upon.

The CAN also highlights the fact that a partial area may have a downward edge directed to another partial area with fewer relationships. For example, *Urinary endoscope*, its child *Nephroscope***,** and its sibling *Otoscope* reside on level 1. However, all three are children/descendants of *Endoscope* and have other siblings that reside on level 2 along with *Endoscope*. Moreover, *Urinary endoscope* and *Otoscope* are each pointed at by only one procedure. With taxonomies,[6] we have seen such small partial areas as being indicative of possible errors, and one might expect that a child have at least as many device relationship types as its parent. Regarding *Urinary endoscope*, it might be more appropriate for urinary procedures with relationships currently using *Endoscope* to have the more specific target, instead. This would result in *Urinary endoscope*'s movement into the *Endoscope* partial area. However the current structure is still sufficient by SNOMED criteria.

While the notion of sufficient definition may explain the use of higher-level device categories, some may seem at too high a level. For example, *Removal of Kantrowitz heart pump* points to *Device* via *direct device*. While acknowledging refinability, this assignment seems overly general since *Device* roots a significant subhierarchy. As is the case with other fully specified procedure concepts, such as *Open insertion of Hickman central venous catheter*, the procedure should point to either *Heart pump* or the more specific *Kantrowitz heart pump*. However, these device concepts do not exist in SNOMED. Their omission suggests a needed refinement.

The Hickman example highlights another observation. While the fully specified Hickman procedure above uses the explicitly specified *Hickman catheter* device, its sibling, *Open insertion of Broviac central venous catheter*, does not. The Broviac device is missing. In this case, two "parallel" concepts are modeled differently and offer an opportunity for further refinement.

In this discussion, we made an effort to illustrate various kinds of problems exposed by the alternative view offered by the CAN. Unlike our previous work[6,7], this study did not unearth a large number of errors. This is not surprising since this part of SNOMED received comprehensive scrutiny by its editors. This is one possible (and potentially the most preferred) result of an auditing effort. However, we would like to emphasize that the CAN is not proposed as an all-inclusive auditing method but rather as an additional tool in an auditor's toolbox. The abstraction view[6,7] is structural-based and helps expose anomalies that might not be uncovered otherwise. It is complimentary to other methods such as DL-based auditing methodologies.

Conclusion

Converse relationships, derived from relationships in SNOMED's inferred view, have been used in the construction of a new kind of abstraction network, the CAN, for a strict target hierarchy. An auditing methodology for such a hierarchy's incoming relationships whose basis is the CAN was presented. The results of applying this methodology to the Device subhierarchy indicate that the CAN is a useful auditing vehicle that can bring various aspects of the relationship structure to light and aid an auditor in refining and improving SNOMED.

Acknowledgment

This work was partially supported by the NLM under grant R-01-LM008912-01A1.

References

1. IHTSDO: SNOMED CT. March 1, 2009 http://-www.ihtsdo.org/our-standards/snomed-ct..
2. Ceusters W, Smith B, Kumar A, Dhaen C. Ontology-based error detection in SNOMED-CT. In: Proc. Medinfo 2004. San Francisco, CA; 2004. p. 482–486.
3. Schlobach S, Huang Z, Cornet R, Van Harmelen F. Debugging incoherent terminologies. J. Autom. Reasoning. 2007;39:317–349.
4. Cornet R, Abu-Hanna A. Auditing description-logic-based medical terminological systems by detecting equivalent concept definitions. Int J. Med. Informatics. 2008;77(5):336–345.
5. Jiang G, Chute CG. Auditing the semantic completeness of SNOMED CT using formal concept analysis. JAMIA. 2009;16(1):89–102.
6. Wang Y, Halper M, Min H, *et al*. Structural methodologies for auditing SNOMED. JBI. 2007;40(5):561–581
7. Halper M, Wang Y, Min H, *et al*. Analysis of error concentrations in SNOMED. In: Proc. 2007 AMIA Annual Symposium. Chicago, IL; 2007. p. 314–318.