What's in a code?
Towards a formal account of the relation of ontologies and coding systems (corrected version)

Alan L Rector MD PhD

School of Computer Science, University of Manchester, Manchester, England M13 9PL

citation: Rector AL. What's in a code: Towards a formal account of the relation of ontologies and coding systems.
In: Kuhn KA, Warren JR, Leong T-Y, (eds) Medinfo 2007. IOS Press; 2007. pp 730-734.

Abstract

Terminologies are increasingly based on "ontologies" developed in description logics and related languages such as the new Web Ontology Language, OWL. The use of description logic has been expected to reduce ambiguity and make it easier determine logical equivalence, deal with negation, and specify EHRs. However, this promise has not been fully realised: in part because early description logics were relatively inexpressive, in part, because the relation between coding systems, EHRs, and ontologies expressed in description logics has not been fully understood. This paper presents a unifying approach using the expressive formalisms available in the latest version of OWL, OWL 1.1.

Keywords:

knowledge representation, terminology, ontology, Electronic Health Records, OWL

Introduction

Coding systems, such as SNOMED-CT [1] and the NCI Thesaurus [2] are increasingly being developed using "ontologies" represented in description logics or languages based on them such as OWL. Other groups, such as the Open Biomedical Ontologies (OBO) consortium in the basic biological sciences, are developing what they overtly describe as "ontologies" [1], many of which are implemented in OWL.

A major benefit of using description logics and ontologies to represent coding systems is purported to be the ability to infer logical equivalence between sets of codes and to classify codes automatically. However, that promise is still far from being realised routinely in practice. We suggest here that one reasons for the difficulty is that the relationship between "ontologies" and coding systems has not been clarified. We put forward here a procedure that clearly distinguishes between the ontology as a logical representation about the world and the code as a data structure, and show how different questions can be answered by each.

Throughout this paper we shall use OWL 1.1[2] as our representation language in the simplified Manchester syntax described in [3]. Exemplar ontologies are available on the Web.[3]

Requirements

The first question for this paper is: "What should a code represent?" Our basic requirements are to be able to express:

1. Individual "pre-coordinated" codes – *e.g.* the code for "head injury".
2. Clinical complexes common in many coding systems such as "head injury with/without intracranial bleed"[4].
3. Syndromes in two senses: a) well defined invariant combinations of conditions – *e.g.* tetralogy of Fallot – and ill defined variable combinations of symptoms – *e.g.* chronic fatigue syndrome.
4. Composite "post-coordinated" code expressions (what HL7 refers to as "code phrases").
5. The logical equivalence, or not, of alternative combinations of codes – *e.g.* tof identify "intracranial bleed" whether it occurs singly or as part of the complex "head injury with intracranial bleed".
6. Negation – definitely not having a condition and "absence" in the sense of negative findings such as an absent pedal pulse.
7. Formation of arbitrary "value sets" – sets of codes for use in particular situations.
8. The clinical dialogue.

Finally, we require that these requirements be met within a uniform logical framework for what constitutes a code, so that an inference engine or "classifier" can infer the subsumption hierarchy according to well specified semantics.

[1] http://obo.sourceforge.net/

[2] http://owl1_1.cs.manchester.ac.uk/

[3] http://www.cs.man.ac.uk/~rector/ontologies/whats-in-a-code/

[4] In this paper we have sometimes substituted "bleed" for "haemorrhage" to conserve space in figures and formal expressions.

Framework: Data Structures and Ontologies

In a separate paper [4] we have argued that we need to consider information models at two levels:

- *Representations of the world* (or our conceptualisation of it) – "ontologies" and formal logical statements about patients, their disease and treatments, etc. The criteria for correctness is prediction of observations of the world.
- *Models of data structures* – "information models" – which we use to specify which data structures are valid. The criteria for adequacy is that it sufficiently constrains data structures that those produced by one system can be correctly processed by another.

We argue that the task for information systems is: a) to begin with a representation of our understanding of the patient's situation in the world – the level of the ontology, b) to transform it into valid data structures – messages or EHR fragments – for storage and/or transfer to other systems, and then c) to re-interpret these data structures to derive representations of statements about the world – again at the level of the ontology Furthermore, we wish to perform these transformations with no, or only well defined, loss of information. These transformations are shown diagrammatically in Figure 1.

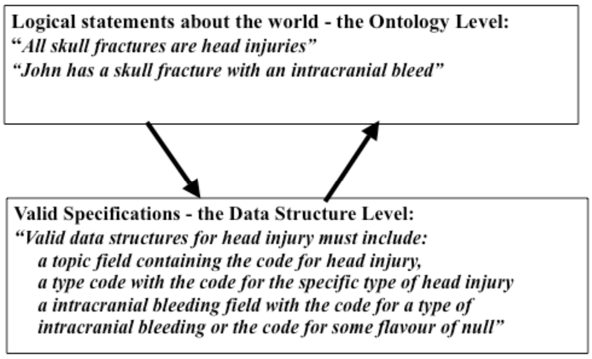
Logical statements about the world - the Ontology Level:
"All skull fractures are head injuries"
"John has a skull fracture with an intracranial bleed"
Valid Specifications - the Data Structure Level:
"Valid data structures for head injury must include:
a topic field containing the code for head injury,
a type code with the code for the specific type of head injury
a intracranial bleeding field with the code for a type of
intracranial bleeding or the code for some flavour of null"

Figure 1: Relation of ontology about the world and specification of valid data structures

Codes, data structures, and ontologies.

We argue that, although they may be derived from ontologies, "codes" are themselves data structures. An important reason to develop ontologies is to support coding systems, but the ontologies and the coding systems are distinct. Coding systems derived from ontologies may be thought of as "meta-models" of the underlying ontology – *i.e.* as models of the representation of the ontology in which each individual in the coding system represents the representation of a class in a particular formalisation of an ontology.[5] In hierarchical coding systems, the hierarchical relation – which we will here term "has_sub/is_sub_of" – reflects that subsumption (superclass-subclass) relationship in the underlying ontology.

The relationship between the ontology and the coding system derived from it is shown in Figure 2: each individual (dot) in the coding system represents a class (oval) in the ontology from which it is derived. The individuals (dots) in the ontology represent cases of the condition, or more precisely patient situations including those conditions.

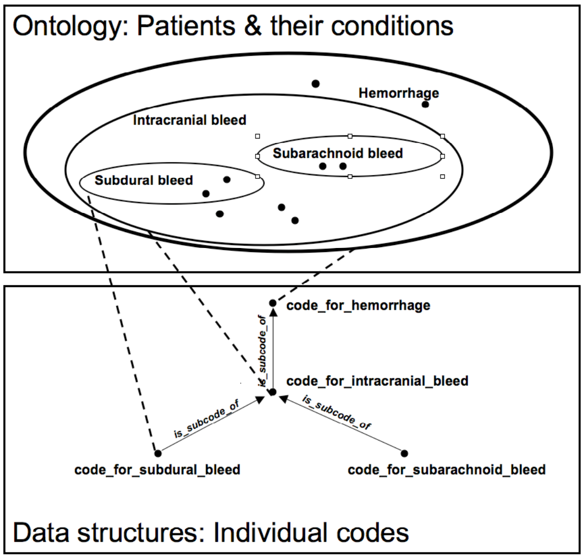

Figure 2: Relation of ontology to codes. Each class in ontology corresponds to an individual code.

Note that the classes in the ontology and the codes in the coding system have very different characteristics. The ontology is "open". There are an indefinite number of possible subclasses of cases in the ontology – and an indefinite number of different ways of classifying the world. By contrast, the coding system is "closed". It consists of enumerated lists of codes, in some cases with enumerated lists of possible qualifiers. For example, in the above schema, there might be many descriptions for intracranial bleeds expressible in the ontology, but the only two for which we have codes in this limited coding system are "subarachnoid bleed" and "subdural bleed".

Ontological interpretation of codes:

In discussing the ontological interpretation of codes, it is useful to deal with two cases which, following SNOMED, we shall term "findings" and "observables". "Findings" are things that apply to only some patients – *e.g.* diseases, injuries, symptoms etc. The existence of a "finding" is significant information, regardless of its detailed description. "Observables" are characteristics of all patients but with different values or states – values measured by laboratory tests, examinations, or other means. For example, only some patients have head injuries. The presence of a head injury is information. By contrast, all patients have a serum potassium concentration; the information is in the *value* of that concentration.

Case 1: Findings and Clinical Situations

An "ontology" in the narrow sense, represents the fundamental entities in a domain and the relations between them. In

[5] This is very roughly the relation of the SNOMED-CT "distribution form" to the underlying description logic form

clinical medicine, these entities include anatomical structures, pathophysiological processes, organisms, cells, cognitive processes, etc. Authors such as Smith [5] advocate confining the word "ontology" to this narrow scope.

However, a major function of clinical knowledge is how these basic phenomena are organised into more complex entities with clinical significance – "macrocytic anaemia", "head injury without intracranial bleeding", "grade II stage 1 carcinoma of the breast", etc. Clinical coding systems typically reflect this higher level of organisation. To ascribe a code to a patient is to say that the patient has (or in some cases does not have) the complex conditions indicated.

OpenGALEN used the term "ClinicalSituation" to define classes of such complexes [6] . They correspond roughly to what SNOMED-CT has termed "Context Dependent Entities" and it has recently rechristened "situations"[6] and are related to what Smith terms "Spans" [5].

Figure 3 gives examples of a simple representation of pathophysiological entities and classes of clinical situations in OWL. Note that many of the classes of clinical situation contain only a single condition. "Wrapping" single conditions in "Situations" in this way may seem redundant. However, it provides the uniform structure needed to support uniform automatic classification as required. To each of these classes of situations there corresponds a code, hi_s_code, sfx_s_code, etc. organised in a hierarchy that mirrors the subsumption hierarchy for the classes of situations as indicated in Figure 2.

```
HI_S = Situation THAT includes SOME Head_injury.
SFx_S = Situation THAT includes some Skull_fracture.
…
ICB_S = Situation THAT includes SOME Intracranial_bleed.
SDB_S = Situation THAT includes SOME Subdural_bleed.
HI_ICB_S = Situation THAT includes SOME Head_injury AND
                   includes SOME Intracranial_bleed
…
Not_ICB_S =
      Situation THAT not (includes SOME Intracranial_bleed).
HI_Not_ICB_S =
      Situation THAT includes SOME Head_injury AND
                        NOT (includes SOME Intracranial_bleed).
…
```

Figure 3: Example definitions of "situations" in OWL (Abbreviated names correspond to labels in Fig 5)

Case 2: Observables, codes and values

"Observables" are qualities of patients that are present in all patients and whose values or states are determined by observation – often by means of laboratory tests or physical examination. Typically, observables are represented by a "code-value pair". However, as often noted, an observable plus its value – *e.g.* "<Serum potassium, elevated>" – can be equivalent to a finding – "elevated serum potassium". We therefore propose an interpretation in the ontology that makes this equivalence apparent following the example in Figure 4:

Whereas for findings a single code is interpreted in the ontology as a "Situation", for observables, it is usually a code-value pair, *e.g.* "<serum_potassium_code, elevated_code>", that is interpreted as a "Situation". However, not uncommonly there is also a code assigned to the entire Situation, especially when a qualitative symbolic value such as "elevated" is involved, *e.g.* in Figure 4, the class of situations involving elevated serum potassium, ESP_S, and the corresponding code esp_s_code. To determine if the code-value pair and single code are equivalent merely requires interpreting each according to their meanings in the ontology and then using the reasoner to determine if the two meanings are equivalent.

Ontological Level:

```
ESP_S = Situation THAT includes SOME
        (Serum_potassium THAT has_state VALUE elevated)
PQ5_S = Situation THAT includes SOME
        (Serum_potassium THAT has_quantity VALUE
                                        [5.1 mMolPerL]
```

Corresponding coded representation:

For code-value pairs: <serum_potassium_code, elevated_code>
For Situations: pq5_s_code, esp_s_code

Figure 4: Ontological representation and corresponding codes for example observable

Consequences: Addressing the requirements

Of the requirements in the introduction, the proposed framework meets requirements 1-3 directly. As indicated in the examples in Figure 3, pre-coordinated codes, complexes, and syndromes are treated uniformly. The extension to qualifiers and post-coordinated codes (requirement 4) is straightforward and omitted for reasons of space. Requirements 5-8 are more subtle and are discussed below.

Classification and equivalence (Requirement 5)

Is a patient who is assigned the code-value pair "<serum_potassium_code, elevated_code>" the same as a patient assigned the single finding code "esp_s_code"? Is a patient who is assigned separately the codes for "Situation THAT includes SOME Head_injury" (hi_s_code) and "Situation THAT includes SOME Intracranial_bleed" (icb_s_code) separately equivalent to a patient that has assigned the single code for "Situation that includes some Head_injury AND includes SOME Intracranial_bleed" (hi_icb_s_code)?

In the proposed framework, all such questions are answered by re-interpreting the codes as representations in the ontology and then comparing these representations, using an appropriate classifier where necessary. In the case of the equivalence of a code value pair and the corresponding finding code, the answer is obvious, since the interpretations as expressions in the ontology are identical – see Figure 4.

[6] Kent Spackman, Personal communication, 2006.

In the case of comparing several separate findings of single conditions with a single finding of those conditions combined, the answer follows naturally from the notion of a "clinical situation". If each patient can have only one "situation" at any one time (perhaps as observed by a given observer), then we need only form the conjunction of the criteria and compare the result. This can be done manually for simple lists, or the inference engine can be used for more complicated cases.

For example, to determine if patients with "head injury" and "no intracranial bleed" coded separately are equivalent to patients with the single code for "head injury without intracranial bleed", first interpret the codes to describe a patient or class of patients at a particular time:

```
(Patient THAT at_time SOME Time_point)
 has SOME
  (Situation THAT includes SOME Head_injury) AND
 has SOME
  (Situation THAT NOT (includes SOME Intracranial_bleed).
```

Because a patient can have only one situation at one time, the classifier will recognise that the second Situation is redundant and find that this is logically equivalent to:

```
(Patient at_time SOME Time_point)
  has SOME
    (Situation THAT includes SOME Head_injury AND
 NOT includes SOME Intracranial_bleed).
```

The axiom that each patient at a given time can have only one situation is captured by the generic axiom in OWL:

```
(Patient at_time SOME Time_point) has MAX 1 Situation
```

Note that this axiom holds at the level of the ontology but not at the level of codes. It is not true that a patient can be ascribed only a single code in an EHR or message. It is true that there can be only one clinical situation for a given patient at a given time (as determined by a single observer).

Dealing with negation (Requirement 6)

Negation of a finding – "not any"

Many codes represent classes of situations that involve a patient *not* have certain findings. Examples in Figure 3 include "No intracranial bleed" (Not_ICB_S) or "Head injury without intracranial bleed" (HI_Not_ICB_S). The classes of findings in these cases would best be defined by analogy with "patients who do not have *any* intracranial bleed".

Correct classification of negation manually without formal inference is difficult. The result of applying the classifier to an extended set of definitions based on Figure 3 is shown in Figure 4. This achieves the correct results automatically. For example, "No intracranial bleed" (Not_ICB_S) is a kind of "No subdural bleed" (Not_SDB_S) rather than vice versa. This is an example of the rule that negation inverts the kind-of hierarchy. If "B is a kind of A" – *i.e.* "all Bs are As" – then "NOT A is a kind of NOT B" – *i.e.* "all non-As are non-Bs".

However, note that the negation in Figure 3 applies to the entire criterion "includes SOME Intracranial_bleed" rather than to "Intracranial_bleed" itself. To say "Situation THAT not includes SOME Intracranial_bleed" means "the situation does not include *any* intracranial bleed", as intended. By contrast, to say "Situation THAT includes SOME NOT Intracranial_bleed" is to say that "the situation includes *something that is not* an intracranial bleed" – a different statement altogether.

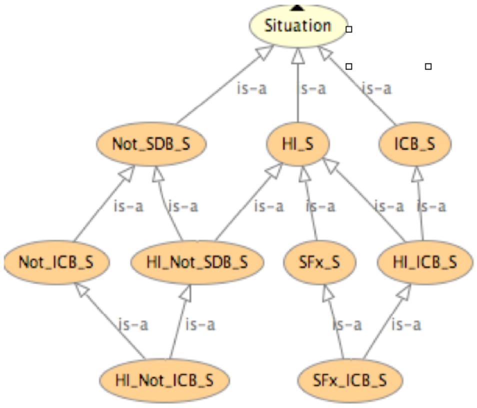

Figure 5: Fragment of logical classification of complexes in Figure 3as determined by the OWL classifier (abbreviations follow pattern of Fig 3).

Negative findings and negative observables – "some not"

In most cases negation follows the example of intracranial bleeding above. However, there is a group of what are often referred to as "negative findings" that cause confusion.

The classic example is "absence of pedal pulse". Such rubrics are fundamentally ambiguous. Does it mean "Absence of any pedal pulse" or "Absence of some pedal pulse". In the first case, we would expect it to imply the absence of all pedal pulses – *e.g.* "absence of dorsalis pedis pulse", "absence of posterior tibial pulse", etc. We would therefore expect the hierarchy to be inverted as for types of intracranial bleeding. "Absence of any pedis pulse" would therefore fall under "absence of dorsalis pedis pulse" in the hierarchy.

However, in the second case, if we mean "Absence of some pedal pulse", we would expect the reverse. The "absence of the dorsalis pedis pulse" is certainly an example of the "absence of *some* pedal pulse", so we would expect "absence of dorsalis pedis pulse" to fall under "absence of some pedal pulse". The hierarchy would not be inverted.

The cleanest way to deal with this case is to use the ontological interpretations as shown in Figure 6. The first, for the absence of *any pedal pulse,* is analogous to the usual negation of situations in Figure 5. The second, for the absence of *some* pedal pulse, deals with the special case of "negative findings".

```
Absence_of_any_pedal_pulse =
  Situation THAT NOT (includes SOME Pedal_pulse).
```

Figure 6: Absence of any vs Absence of some

The scope of the negation in the second case is critical. In the definition of "Absence_of_some_pedal_pulse" the negation is included in the definition of the property, "excludes" and affects just that property, whereas in the definition of "Absence_of_any_pedal_pulse" it negates the entire restriction “includes SOME Pedal_pulse”.[7] Unfortunately, no current description logic implements the constructor for negating properties, although it is known to be tractable [7]. OWL 1.1 implements a slightly weaker construct, disjoint properties, which, unfortunately is not sufficient for this case.[8] [9]

Theoretically, the OWL 1.1 solution is the best approximation that can be implemented currently, but because it has just become available, there is little experience with it in practice. An alternative construct, available in most formalisms with which there is more experience, is to regard “having a pedal pulse”, “having a dorsalis pedis pulse” etc. as “observables” with possible values “detectable” and “NOT detectable” as shown in Figure 6. This leads to correct classification but is arguably less faithful to the intended meaning.

```
Absence_of_some_pedal_pulse =
 Situation THAT includes SOME
  (Having_pedal_pulse THAT has_state SOME (NOT Detectable))
```

Figure 6: Alternative representation for negative findings

Value sets and faithfulness to the clinical dialogue (Requirements 7 and 8)

Whereas equivalence of meaning can only be addressed by reinterpreting the information structures, including codes, into the ontology, specifying valid value sets and a faithful representation of the actual clinical statements made can only be made in terms data structures and codes themselves.

Value sets are closed lists of codes or tightly specified code phrases. While we can talk about requiring “only the code for head injury and none of its subcodes”, at the ontological level, we cannot talk about the “all cases of head injury” without including all cases of all kinds of head injury. The process of specifying code sets and binding code sets to health records is discussed in detail in a separate paper [4].

Similarly, on the one hand, we want the *meaning* for a patient of the two statements – “This patient has a head injury” and “this patient has an intracranial bleed” – to be the same as the meaning for a patient of a single statement combining the two conditions. On the other hand, for purposes of clinical responsibility and documenting the clinical dialogue, we want to keep track separately of each statement made about the patient in the form in which it was made. For example we may want to record who made which statement, when, why, etc. This can only be done at the level of the codes and data structures where the statements are distinct, not at the level of the ontology where their meanings are indistinguishable.

Conclusion

This paper presents two key ideas: a) That codes should be regarded as individual data structures which can be interpreted in terms of meanings in a separate ontology about patients and their conditions, and b) that the ontology should be structured in two layers: a layer of kernel concepts – conditions, anatomy, etc – and a layer of “clinical situations” which describe classes of patient states at a particular times as viewed by particular observers. It suggests that both codes for “findings” and code-value pairs for “observables” should be interpreted as representing classes of situations in the ontology. It shows how these notions can be used to provide a unified framework for dealing with the questions of equivalence of the meaning of negation. An account of patients’ situations, the conditions for validity of data structures conveying information on those situations, and the dialogue about those patients’ care requires taking account of both the level of the ontology and the level of data structures and codes.

Acknowledgements

This work supported in part by the UK Department of Health “Connecting for Health” programme, the UK MRC CLEF and CLEF-Services projects (G0100852), the JISC and UK EPSRC projects CO-ODE and HyOntUse (GR/S44686/1) and the EU Funded Semantic Mining Network of Excellence. The HL7 Terminfo working group stimulated and contributed to many of the ideas presented here.

References

1. Stearns M, Price C, Spackman K, Wang A. SNOMED clinical terms: overview of the development process and project status. In: *AMIA-2001*. Henley & Belfus; 2001. p. 662-666.
2. Golbeck J, Fragoso G, Hartel F, Hendler J, Oberthaler J, Parsia B. The National Cancer Institute's thesaurus and ontology. J Web Semantics 2003;1(1):75-80.
3. Horridge M, Drummond N, Goodwin J, Rector A, Stevens R, Wang H. The Manchester OWL syntax. In: Cuenca Grau B, Hitzler P, Shankey C, Wallace E, editors. *OWL: Experiences and Directions (OWLED 06)*; 2006; Athens, Georgia: CEUR; 2006. p. http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS//Vol-216/submission_9.pdf.
4. Rector A, Qamar R, Marley T. Binding ontologies & coding systems to electronic health records and messages. In: Bodenreider O, editor. *Formal Biomedical Knowledge Representation (KR-MED 2006)*; 2006; Baltimore: CEUR; 2006. p. 11-19.
5. Smith B. The logic of biological classification and the foundations of biomedical ontology. In: Westerstahl D, editor. 10th In *Int Conf on Logic Methodology and Philosophy of Science; 2004*; Oviedo Spain: Elsevier-North-Holland; 2004.
6. Rector AL, Rogers JE. Ontological and practical issues in using a description logic to represent medical concept systems: Experience from GALEN. In: Barahona P, Bry F, Franconi E, Henze N, Sattler U, eds. *Reasoning Web*. Heidelberg: Springer-Verlag; 2006. p. 197-231.
7. Lutz C, Sattler U. Mary likes all Cats. In: International Workshop in Description Logics (DL2000); 2000; 2000. p. 213-226.

Address for correspondence

Alan Rector, School of Computer Science, University of Manchester, Manchester M13 9PL, England. Email: rector@cs.manchester.ac.uk

[7] In standard predicate logic notation, the difference is between $\neg\exists y.\ PP(y)\ \&\ includes(s,y)$ and $\exists y\ .\ PP(y)\ \&\ \neg\ includes(s,y)$.

[8] The difference between “includes = NOT excludes” and “DISJOINT includes, excludes” is that in the case of true negation, we can infer that something not included is excluded, where in the case of mere disjointness we cannot.

[9] Corrigendum: Disjoint properties should *not* be used because, although it is not possible both to include and exclude the same fact, it is not possible to infer from excludes SOME X that NOT includes SOME X