Codes, EHRs and semantic interoperability.

One often hears that coded data are essential for semantic interoperability and decision support. Coding is the use of symbolic, or alphanumeric identifiers to tag data items as referring to concepts or terms from an agreed vocabulary or ontology. Coding, may, in many circumstances have some value. But it also comes at a price. This article looks at the balance sheet to tease out the issues facing those making recommendations for electronic health records and semantic interoperability.
Although some more general aspects of terminologies are touched on, the focus is not on clinical terminologies and classifications *per se*, but rather the use of codes attached to, or used in lieu of words and terms.

History

Codes have been around since before the days of computers, but all digital computers must rely on codes for their very operation. Each instruction is represented by a combination of 0s and 1s. So too is every piece of data. With respect to data, codes can be applied at different structural levels. Thus, each character is represented by a code or codes, from the simple 127 common ASCII characters used to write this article, to the complex kanji, chinese and other characters and symbols that are represented by more complex coding schemes such as UNICODE. So, by combining character codes, we can represent and store words and phrases - i.e. text strings. The string “openEHR” is built from the ASCII codes:

```
01101111  01110000  01100101  01101101  01000101  01001000  01010010
```

In the early days of computing, memory, storage space and communications bandwidth were very limited. It made sense to not only code individual characters within a string, but even to code text strings themselves, by replacing the set of codes representing each character of the string by a single code representing the entire text string. Particularly so if the string was likely to be repeated in other locations. “Diabetes Mellitus” could be replaced by code 1101110011010001, for example. Or even by just a shorter string, say “DM”. It saved precious space and bandwidth. Codes were easier for computers to identify in searches and to place into predefined message structures.

But most of these barriers have evaporated over the years, as computer storage first increased a thousandfold, then a millionfold and onwards. The bandwidth of many of our network links have scaled by several orders of magnitude each decade. Now our programming languages and programmers can support sophisticated pattern matching through “regular expressions” and other advances which allow them to operate directly on text strings instead of codes. Still, the legacy of those old constraints live on in the specifications and the mindset of many current authors and standards development bodies.

The above history tells only part of the story. Surely there are other factors influencing the scene when it comes to representing clinical terms and data through the use of codes?

Standardisation and simplification

One argument cited for coding clinical text strings is to ensure agreement between communicating humans on a set of terms that describe their universe of discourse - the concepts related to health and healthcare, or some subset thereof. These are variously known as vocabularies, nomenclatures, value sets, termsets, codesets, terminologies, classifications, etc. The communicating parties agree on the set of valid or prescribed terms, and from then on they use the pre-allocated code as a reference to (proxy for) each term in

the prescribed set. The set constrains the permitted vocabulary for a specific scope and purpose. Thus the set can simplify the processing for that purpose. This raises two issues.

- Firstly, how do we manage each set, both at the specification and governance level, and also importantly at a technical implementation level?
- Secondly, why do we need the code as a proxy for the term? Why not just use the terms themselves? Let's address the second question first.

Concepts and language

Some termsets allow for concepts to be represented independently of the words used to describe each concept. This allows a concept to be described by more than one term. Synonyms and language translations are the two examples commonly cited. The concept "level of glycosylated haemoglobin in blood plasma" might legitimately be known and referred to as "hemoglobin A1c", "HbA1C", "HBA1C level", "glycated hemoglobin concentration", "la hemoglobina glucosilada" or a host of other terms.

Conversely, a term can describe more than one concept. A "left ventricle" could be a compartment of the heart or part of the brain.

Though not strictly necessary, codes can help handle the 'many to one' and 'one to many' relationships that might be needed, particularly in a large complex terminology where such requirements are certainly likely to occur.

Code and concept permanence

As medical knowledge expands and evolves, so to do the concepts and the language used to describe concepts, change. Sometimes, the concept morphs but the term used remains. Sometimes the concept remains the same but an alternative term is used. Sometimes the two occur simultaneously. One old concept is cleaved into two or more new concepts and each new concept given a new term. The original concept may even fade out of our daily lexicon. It is desirable, particularly in a longitudinal health record spanning many decades for the current viewers and users of the record to somehow be able to make sense of the concepts and language of yesteryear. But the terminology has to be designed and managed well, for these purposes, the electronic health architecture needs to support this, and the current (at the time of viewing) implementations could need access to the prior state of the terminology at the time the entries were made.

Concept permanence is probably of even greater importance for statistical comparisons and research analyses that span long periods of time and patient cohorts. One only has to trace the history of the classification of, say, the various manifestations of hepatitis through successive versions of The International Statistical Classification of Diseases and Related Health Problems (most recent release is ICD-10), to appreciate the complexities and interaction of changing concepts and changing codes.

Codes and terminologies

Codes provide a mechanism for anchoring a term to a particular spot in a terminology where such terminologies provide relationships between concepts. In SNOMED CT, for instance, codes play a pivotal role in uniquely identifying concepts, and for uniquely identifying relationships between concepts via relationship types. Codes are also used to support other management and searching functions within the terminology.

Compound, multi-axis terminologies, like LOINC use codes to refer to legal combinations from a set of components. In the case of the LOINC terminology, a catalogue of laboratory tests are each given a code that describes the test in terms of 6 components, including the name of the component or analyte measured, its property (substance concentration, mass, volume), the timing of the measurement, the type of sample (serum, urine, etc.), the scale of measurement (qualitative vs. quantitative, etc.), and the method. The compound, or "pre-coordinated" codes, embody knowledge beyond that ascribed to the individual components, since only valid combinations of analyte, sample, scale etc. are constructed. Each compound concept can be processed as a single entity. If the processing system has access to the LOINC table, then each of the components of the compound entity can be also accessed separately.

In even more complex terminologies, such as SNOMED CT, the terminology can provide the ability to combine terms according to rules. SNOMED CT uses the codes attached to each atomic concept, a compositional grammar, and formal "concept model"s based on Description Logic for some key clinical topics such as clinical findings, to govern the production of compound concepts. Some of the compound, or pre-coordinated concepts, are already provided in the terminology. Given the existence of appropriate software, many more can be constructed by users of the terminology (post-coordination) for specific data entry requirements on an as-needs basis. Currently there is no standardised mechanism for labeling or coding these post-coordinated concept codes - they simply exist as expressions of codes together with their syntactic glue. The issues associated with using these SNOMED CT expressions in clinical systems and electronic health records are complex and significant. A mere hint of some of the issues can be gleaned from the extensive analyses conducted by Markwell [MAR2008] for the UK's National Health Service.

Codeset complexity

Codesets have been used for dealing with a vast range of differing size value sets for a considerable range of different purposes. Most of these codesets are local to a geographical region, local to individual manufacturers of clinical systems, or nationally mandated statistical data collections. The proliferation of codesets and the variability in the complexity of codesets combine in a way that inhibits semantic interoperability if they need to co-exist in a given clinical system.

There are many codesets that have been developed and intended for very specific data fields, and which are exhaustive for the intended field. They may only have 2, 3, 17, several dozen terms at most to cover. Examples of such simple codesets include administrative gender:

Code	Meaning
M	male
F	female
U	undifferentiated

Other codesets are larger, but still often only flat lists of terms and corresponding codes.

HL7 v3, for example, has some 250 'supported vocabularies', about half of which are managed by HL7, and half managed by organisations external to HL7. Even of those mostly

supposedly flat codesets internal to HL7, many are not stable from release to release in any sense, have contradictory definitions in different places, and have a plethora of different code forms and inconsistent information. Some codesets have a specialisation hierarchy implicit in the set. Some codesets have a specialisation hierarchy encoded into their codes. Some have a combination of both approaches. If HL7 International cannot manage their own codesets consistently and effectively, then how can systems trying to parse incoming HL7-based messages ever be expected to cope?

For many of these codesets, it is left to local implementers, national standards bodies, vendors and possibly even clinicians and others to decide if the codeset is appropriate for their scope of implementation. If not, then they must decide to either replace the set, modify it, or augment it with the codes and corresponding terms peculiar to their scope. The ongoing synchronisation often becomes an impossible treadmill of reaction to change, well beyond the control of the clinicians and clinical institutions trying to provide health care based on such an *ad hoc* approach.

A small number of large, well designed terminologies offer much for decision support. However, terminologies of this ilk, such as SNOMED CT are much more complex than simple flat codesets. One single release of SNOMED CT has millions of codes, pointing to concepts, terms, relationships. It's codes form a multipurpose polyhierarchy of concepts and terms, with multiple relationship types. It has mechanisms for extension, multi-language translations and subsetting for defined purposes. It's compositional grammar, as already mentioned, allows for terminological expressions to be constructed as needed, based on the concepts available. Even without SNOMED CT's significant and documented problems, implementing and harnessing all this power in any one real system is a profound challenge. Deploying it broadly and effectively across a range of systems to aid semantic interoperability is taking the challenge to even greater heights.

Codes and humans

As a general principle, codes are for computers not humans. Codes should work behind the scenes and not be exposed to users, particularly busy clinicians. They should not be deliberately exposed, unless absolutely necessary, to those who are only peripherally likely to understand their meaning, such as software developers, or data modellers. Writing standards and specifications for humans, that are littered with abbreviations and codes often dreamed up on a whim, that have to be understood, transcribed, embedded in program code, put into test scripts and test specifications and otherwise discussed and manipulated, and above all remembered, is fraught with danger. It is not sound engineering practice. It dramatically narrows the pool of experts who can understand and use the specifications, and risks misunderstanding and transcription errors and the resultant clinical errors that can ensue.

The above notwithstanding, there are places where codes and humans legitimately need to meet. These situations are where textural descriptions are too awkward to use. Common examples in daily life are things like postal codes and bus codes. It is far simpler to refer to postcode "5068", than "that area bounded by the Sunnybank River to the North, Franklin Bridge, Rainsford Rd and Elm St. to the east, holes 6-14, 17 and 18 of the Royal Plunkett Golf Course to the South, and ....", or to bus "J1E" instead of " the bus that departs from the corner of Edmund St. Walkerville and ...".

In health IT, examples might be genes and gene sequences, tumour staging, or the classification of diseases. In these circumstances, it is often easier for the humans involved to refer to these concepts by codes. Where codes are to be used by humans, it is sensible for the codes to carry additional meaning or representational hints to aid the humans

disambiguate the codes and reduce the chance of error during human processing and transcription. Thus in the bus code “J1E” , the ‘E’ might denote express. Similarly, where codes are to be used by humans, the shorter the code the less likelihood for error. In Australian hospitals, it is common practice for a patient’s identity to be verbally cross-checked by nurses prior to procedures, including administration of some drugs. This cross check usually uses the hospital’s own Unit Record Number for the patient, which usually has few digits and so is relatively human-friendly.

It is probably the history of human abbreviations in the early days of coding and messaging that has lead to the proliferation of a vast array of semi-interpretable “codes” creeping into what should only be computer-processable identifiers of many codesets. Even in the most recent versions of HL7, these are variously and conflictingly referred to as “mnemonics”, “codes”, “conceptIds”.

Codes in EHRs

EHR systems impose requirements on data far exceeding those required in messages. Data may come from a vast array of sources, including direct input by humans, messages from laboratories, pharmacies etc., referral and other documents from other healthcare providers, etc. Data may have to be available for decades. Data may have to be processed into different forms for different users and purposes - e.g. aggregation across time, and other variables. Data may be needed to be searched using search criteria expressed at a variety of levels of detail. Data may have to be presented in different forms to humans whose medical knowledge varies considerably.

Codes can help in this process, but they can also hinder. They can hinder, because they are always at least one step away from the human meaning conveyed by the code, and so their processing is **critically** dependent a) on the availability of the code system that can provide the link to the term or meaning of the code, b) on the quality of the code system and underlying terminology c) on the capability of the processing system to deal with problems when the links can’t be resolved or generate conflicts, d) on the ability of the EHR system to handle evolution of the coding scheme or terminology over time.

Code Usage

When small termsets such as gender are used within a given language realm, what possible gain is there by replacing the value “male” by a code such as “1”, or “M”? There certainly is plenty to lose!! Why should every information system that receives such a code have to deal with this? Humans can understand “male” easily. Computers can process “male” easily. Humans cannot understand the code “1” in any meaningful way!. Computers cannot process “1” in any meaningful way, other than perhaps saying that “1” (male) is less than “2” (female)! Is this the intention of the sender of such coded data - to obfuscate and compromise patient safety? The code is absolutely useless without access to the accompanying meaning - e.g via some code table. Who can guarantee that that access will always be available? Why place such a burden on every clinical system needing to process gender for absolutely no benefit.? It is far more important to give the clinicians definitional information about the meaning of appropriate terms in the particular context of the data field. Does this refer to administrative or physical gender?

The more small codesets that information systems have to deal with, where disambiguation of multiple-meaning terms is not required, the less likely we will have of achieving a

reasonable level of useful information exchange. We should not be blindly advocating that all data be coded. We should stop and think of the ramifications of such recommendations.

One ramification is that we are forced to build code maps between many different standards and coding systems in order to meet the coding requirements demanded by each system. There is no longer room of consideration being given to the importance of insisting on an appropriate code for each data item. Instead, software developers and implementers and message "integrators" are left trying to force square pegs into round holes. Continuing the simple gender example above, we have many examples such as the following internet discussion forum snippet:

```
> We have a case where a HIS system has added some definitions to their
> possible values for patient sex. They are:
>
> M = male
> F = female
> T = transgender
> U = Undifferentiated
> ? = Unknown
>
> However, DICOM only supports:
>
> F - Female
> M - Male
> O - Other
>
> And I found this table in HL7 2.4:
>
> 3.4.2.8 PID-8 Administrative sex (IS) 00111
> Definition: This field contains the patient's sex. Refer to
> User-defined Table 0001 - Administrative sex
> for suggested values.
> User-defined Table 0001 - Administrative sex
> Value Description
> F Female
> M Male
> O Other
> U Unknown
> A Ambiguous
> N Not applicable
```

Evaluation of a specific terminology or codeset is often undertaken in isolation of its use. Criteria such as Cimino's 12 desiderata [CIM1998] are often considered for this task. Cimino [CIM2006] further augmented these structural requirements with desirable characteristics to support the purpose of a terminology, citing the following:

1. *Terminologies should support capturing what is known about the patient.*
2. *Terminologies should support retrieval*
3. *Terminologies should allow storage, retrieval and transfer of information with as little information loss as possible.*
4. *Terminologies should support aggregation of data.*
5. *Terminologies should support resuse of data.*
6. *Terminologies should support inferencing.*

Whilst not exhaustive, these useful criteria can also be used to judge the utility of the codes used to underpin the functioning of the terminology. But judging a terminology, and the coding thereof, should be undertaken in the context of the entire clinical information system(s), the clinicians and other users of the data, the flows of the data from system to system, and all of

the other information and terminology components that are also involved, both now and into the future. A truly daunting assessment task.

The cost of codes

The consequence of requiring so many codes and coding systems is that comprehensive electronic health record systems need many code tables for each implementation; they need to maintain versions of those code tables; they need the capability to process the code tables; they need to process the versioning of the code tables; they often need mapping tables to map between code tables; they need the capability to parse and interpret and map based on the peculiarities of both the source and target coding systems; they may need to hold multiple versions of the mapping tables; they need the capability to process the versioning of the mapping tables; they need the capability to map between different versions of different mapping tables. And in almost every case, the maps are not one to one. Compromises and arbitrary decisions are made on an institution by institution, map by map and code by code basis.

Another cost of the variability in the formulation of coding systems is the difficulty in providing generic tools. In Australia, there are published examples informing general practitioners how to access key hidden information locked in their patient records - information important for managing their patient's health, referencing arcane codes and SQL queries peculiar to one particular system, using one particular coding system at one particular point in time.

```
SELECT CM_PATIENT.PATIENT_ID, CM_PATIENT.SURNAME, CM_PATIENT.FIRST_NAME,
MAX(MD_PATHOLOGY_ATOM.RESULT_DATE) AS MaxResultDate, MAX(VISIT.VisitDate) AS
MaxVisitDate FROM MD_PATHOLOGY_ATOM RIGHT OUTER JOIN MD_PATHOLOGY ON
MD_PATHOLOGY_ATOM.PATHOLOGY_ID = MD_PATHOLOGY.PATHOLOGY_ID RIGHT OUTER JOIN
CM_PATIENT ON MD_PATHOLOGY.PATIENT_ID = CM_PATIENT.PATIENT_ID FULL OUTER JOIN
VISIT ON CM_PATIENT.PATIENT_ID = VISIT.PatientNo
WHERE DateDiff(yy, VISIT.VisitDate, GetDate()) < 1 AND (CM_PATIENT.DECEASED_DATE
IS NULL) AND (CM_PATIENT.GENDER_CODE = 'M') AND (DATEDIFF(yy, CM_PATIENT.DOB,
GETDATE()) > 50)
AND (DATEDIFF(yy, CM_PATIENT.DOB, GETDATE()) < 74)
AND (MD_PATHOLOGY_ATOM.LOINC = '2857-1' OR MD_PATHOLOGY_ATOM.LOINC IS NULL)
GROUP BY CM_PATIENT.PATIENT_ID, CM_PATIENT.SURNAME, CM_PATIENT.FIRST_NAME
```

In the above example, one keen clinician wanting to send recall notices to patients deemed to be candidates for Prostate Specific Antigen (PSA) tests, has delved into the bowels of his patient records, determined the relevant database tables, determined that LOINC has been used to code test names, determined the LOINC code (from some 40,000+ codes) historically used by the particular pathology lab in their HL7 message for the PSA test, determined how gender is coded in this specific clinical system, built and run the requisite SQL query; and hopes that nothing changes next time the query is run! A great piece of detective work, but clearly not an acceptable nor sustainable way to empower clinicians with usable, semantically interoperable electronic health records that meet their requirements.

Yet to be addressed

There are still some significant areas related to representing clinical concepts that need further, substantial research and which may affect the decisions we make about the coding of data, including:

- linking meaning in clinical guidelines to meaning in data. Guidelines need to be written by humans, yet processable by computer. If they end up in a coded form in a computer, we must have the tools to reverse the coded form of the guideline for clinical use. Because of the patient specific context required for the application to a specific patient, is there any

realistic option to linking guidelines to data other than through openEHR archetypes? The links cannot simply be done through coded terminology.

- similarly for assisting the classification of patient data for research and reporting using ICD or similar classification systems. The linking of contextual patient data means that codesets and mapping tables are insufficient.
- Gaining better understanding on the value of storing information pertaining to real entities and events (an *ontology of reality* perspective) vs storing codes for concepts in patient records. (an *ontology of use* perspective). Barry Smith and others [CEU2006] have argued that we should uniquely identify (in order to refer to) each real instance of a bone fracture (for example) of each patient, rather than some generic concept of a bone fracture, thus allowing us to track and disambiguate bone fracture instances.
- human interfaces - the way in which text data representing clinical context is captured and displayed is an area needing far more research. How we translate from concepts to terms, from codes to words and vice versa in every system interface is critical to ensuring clinical safety. We need consistent, coherent, repeatable, reliable solutions to these, not a miss-mash of a myriad different approaches, constrained by the nuances of individual coding schemes and vendor architectures. The UK's National Health Service project on Common User Interface is a good start in this direction.

Summary

So, at the end of this short treatise, how does the balance sheet look? Is the answer to code, or not to code?

1. In the context of electronic health records and semantic interoperability, codes as identifiers of concepts, are primarily for assisting computer processing of those concepts.

2. In particular, codes can certainly assist in the process of language translation and interoperability across national boundaries.

3. The coding of data, in and of itself, offers very little. Systems need to be able to make use of the codes. Dramatic changes to today's clinical systems are required in order to supply the benefits that coded data offers. This is very expensive.

4. The proliferation of the many small codesets that abound today, subverts interoperability. Variations in coding schemes; the potential for overlapping or conflicting meaning; the management and versioning issues attendant with the codesets - all are barriers to EHR systems that acquire their data from many sources.

5. For searching of EHRs and for decision support, a single comprehensive terminology and terminology architecture is highly desirable - something offering the potential power of an improved SNOMED CT. Clinical systems based on such a complex terminology require the use of codes.

6. The use of closed, proprietary coded terminologies and the notion of semantic interoperability are mutually incompatible. Ubiquitous semantic interoperability requires ubiquitous access to the codes and the terminology by all participating systems.

For want of a better cliche, "semantic interoperability" is a journey, not a destination. It is a long, slow, expensive journey that will probably never end. As with most journeys, it is cheaper and wiser to make the right steps, at the right time, in the right order. It is sensible to avoid steps that will later need retracing. The journey should not start with a mad rush to "code" data as fast as we can, particularly if it means every system is beholden to a raft of

separate, inconsistent coding schemes. Far better to apply some sound architectural principles and at least sufficient engineering to ensure that as far as possible we take steps in the right direction and take steps that we won't inevitably have to retrace.

Useful links and references

- [CIM1998] Cimino J, Desiderata for Controlled Medical Vocabularies in the Twenty-First Century , Methods of Information in Medicine, 1998 (http://www.mayo.edu/imia-wg6/conf/doc/cimino.pdf)
- [CIM2006] Cimino J, In defense of the Desiderata http://www.dbmi.columbia.edu/cimino/Publications/2006%20-%20JBI%20-%20In%20Defense%20of%20the%20Desiderata.pdf
- [CEU2006] Ceusters W and Smith B, Strategies for Referent Tracking in Electronic Health, Journal of Biomedical Informatics, 2006:39(3):288-98
- [MAR2008] Markwell D, Terminology Requirements and Principles, NHS publication, 2008 (http://www.ehr.chime.ucl.ac.uk/download/attachments/3375121/TerminologyBindingRequirementsAndPrinciples_v1.0.pdf)
- Rector A *et al*, and various publications at http://www.semantichealth.org/
- openEHR Architecture Overview http://www.openehr.org/releases/1.0.1/architecture/overview.pdf