INTERNATIONAL HEALTH TERMINOLOGY
STANDARDS DEVELOPMENT ORGANISATION
UK Terminology Centre (UKTC)

NHS
Connecting for Health

THE CLINICAL TERMS VERSION 3
(THE READ CODES)

TEMPLATE FILE

APRIL 2008

Purpose of this document

This document is one of a series that, taken together, describe the contents, structure and function of Clinical Terms Version 3 (The Read Codes).

This introduction is intended to provide information on Clinical Terms Version 3. It is also a guide to the other available documents each of which is updated independently. For this reason, different chapters may have different version numbers.

INFORMATION

Distribution	On request
Author	UK Terminology Centre
Further Copies From	UK Terminology Centre Helpdesk Service Support Unit Prospect House, Floor 2 Fishing Line Road Redditch Worcestershire B97 6EW Tel: +44 (0) 1392 206248 Fax: +44 (0) 1392 206945 E-mail: datastandards@nhs.net Internet: http://www.connectingforhealth.nhs.uk/systemsandservices/data/readcodes
Date of Issue	April 2008
Reference Number	169v1.0

Published by the UK Terminology Centre

Table of Contents

Table of Contents....3
1. Introduction....4
2. Discussion of Different Types of Qualifiers....5
  2.1 Case 1: Qualifiers for Detail....5
  2.2 Case 2. Atoms which Describe the Internal Characteristics of a Concept....7
  2.3 Case 3: Qualifiers Describing Facts About the World....7
  2.4 Case 4: Qualifiers for Establishing the Context of Use of a Term....8
  2.5 Case 5: Qualifiers for Uncertainty and the State of Execution of an Action....8
3. Brief Overview of the Template Structure....8
4. Principles Behind the Template Structure....11
5. Introduction to the File Structure....13
  5.1 Overview of the File Structure....13
  5.2 Concept File....16
  5.3 Template File....16
  5.4 Analysing Databases Containing Qualifiers....20
  5.5. Inheritance....21
6. Support of the Template Structure....23
  6.1 The Advantages....23
  6.2 A Panacea for All Coding Evils?....24
  6.3 Atoms and The Future....25
Technical Annexe....26
  T. 1 Conceptual Model of Qualifiers....26
  T.2 Relationship between Conceptual Model and File Layout....30
  T.3 Summary of File Layouts....30
  T. 4 Release Format....34
  T.5 Implementation of the Man-machine Interface....34
  T. 6 References....35
  T.7 The CTV3 Document Series....36
Appendix: A....36
  A.1 Glossary....36

1. Introduction

The earlier Read Codes (4-Byte and 5-Byte Set) have been successfully used by many practitioners in primary and secondary care to record comprehensive summaries and even as the basis for a complete computerised medical record in some cases. Many professional groups including nurses, physiotherapists, dentists and general practitioners, as well as the different medical and surgical specialties, have expressed a need for a more comprehensive vocabulary if they are to record their everyday practice in an electronic patient record.

Three projects were therefore performed: the Clinical Terms Project, the Professions Allied to Medicine Terms Project and the Nursing Terms Project, with the aim of expanding the Read Codes to cope with specialists' requirements.

It rapidly became obvious in the course of these projects that the detail required does not sit happily in individual terms. Examples such as *Open reduction and fixation of fracture of the shaft of femur using a locked, reamed, intramedullary nail* [1]are unwieldy, awkward to read and difficult to select when included among picking lists of similar terms varying only by one or two degrees of detail. The strategy for dealing with this problem is to separate out some of the detail into **qualifiers**.

This document details the file structure of the **Template file** in Clinical Terms Version 3 (CTV3), which is the mechanism employed to manage the addition of qualifying details, the expression of semantic definitions (atoms) and the labelling of concepts with facts.

The document is split into two parts. First the method of representing qualifying information is introduced through examples. This is followed by a technical annexe, which includes more formal details. As an aid to understanding the document, Appendix A directs the user to the section in which frequently used notions are introduced. Examples in the document may be fictitious, but are generally drawn from the early pilot set of CTV3 and these may now be superseded.

[1] Less common core concepts can be qualified by other core concepts. This association is only possible using a "link attribute" e.g. Cataract – Associated finding: *Insulin-dependent diabetes mellitus*

2. Discussion of Different Types of Qualifiers

Concepts within CTV3 have a linguistic role of either attribute or non-attribute (see the document "Clinical Terms Version 3 – Main File Structure: Overview and Technical Description"). Non-attributes can further be divided into "core concepts" and "value only concepts". The former types (core concepts) are explicit clinical findings or procedures that form the core of a clinical record e.g. *Myocardial infarction, Cholecystectomy*. The latter types (value only concepts) relate to concepts describing detail that one would not record in isolation and always add detail to core concepts such as anatomical sites, organisms etc. (e.g. Skin of thigh, Mycobacterium tuberculosis). When this additional detail (using value only concepts[1]) is applied they are described as being **qualifiers** to core concepts.

Qualifiers are extra information that may be used along with any 'core concept' such as a procedure, diagnosis, investigation, or symptom. They are described using an **attribute** (e.g. Site) and a **value** (e.g. Hip joint structure). However, not all qualifiers have the same function or role and attribute-value pairs may describe other types of information. We can distinguish between:

- Qualifiers for detail
- Atoms which describe the intrinsic characteristics of a concept
- Facts about the world
- Qualifiers for establishing the context of use of a concept
- Qualifiers describing uncertainty or the state of execution of an action

2.1 Case 1: Qualifiers for Detail

The example given in the introduction, *Open reduction and fixation of fracture of the shaft of femur using a locked intramedullary nail* (reducing a fracture re-aligns bone fragments and various devices such as a nail placed in the middle of the shaft of the bone can keep it aligned afterwards), is typical of the kind of detail some specialists wish to record[2].

If we were to handle detail by including it in lengthy terms then:

a) Each complex term would be difficult to read, particularly when placed among many other similar terms in a picking list.

[2] Note on examples. The examples are drawn from an earlier developmental version of the template file and may not directly relate to the current content of CTV3 which continues to update in a controlled manner.

b) There would be a very large increase in the number of terms (and therefore in storage space requirements).

c) Because the task of generating all of these words would be so large, it would be difficult to ensure that every possible combination of characteristics has been included in the set of terms.

d) Search and analysis would be compromised. As the hierarchy becomes deeper to incorporate detail, it becomes increasingly difficult to retrieve certain categories of information. To illustrate this point, look at the imaginary hierarchy in Figure 1. This hierarchy is constructed by first splitting *Fixation of fracture* by the site of the fixation (Femur, Humerus, Tibia), then by the device used (Screw, Pin, Intramedullary nail, Plate) and then by the reaming of the device (Reamed, Unreamed). It is easy to retrieve all fixations of the femur (no matter which device was used), but difficult to retrieve all fixations using an intramedullary nail as these are dotted throughout the 'leaves' of the hierarchy.

   It is possible to get around this pattern by inserting new parents (e.g. *Fixation of fracture using intramedullary nail*) into the hierarchy. However, maintaining all the links in the dense mesh produced is demanding on resources and the result awkward to browse.

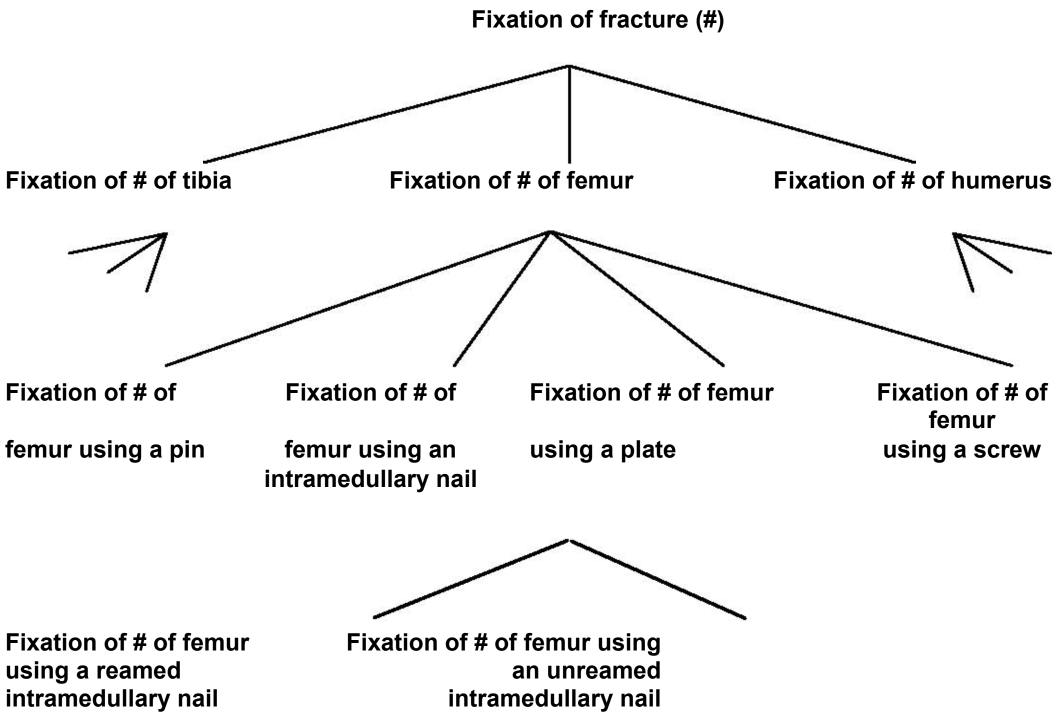

Fixation of fracture (#)
Fixation of # of tibia
Fixation of # of femur
Fixation of # of humerus
Fixation of # of femur using a pin
Fixation of # of femur using an intramedullary nail
Fixation of # of femur using a plate
Fixation of # of femur using a screw
Fixation of # of femur using a reamed intramedullary nail
Fixation of # of femur using an unreamed intramedullary nail

Figure 1: Illustrative hierarchy of concepts describing a fixation of fracture procedure

The approach adopted in CTV3 is to specify detail in the core concepts to only so far down the hierarchy and then to add qualifiers. The main procedures hierarchy, for example, contains the following ‘core concepts’:

- **Operations and procedures**
  - **Musculoskeletal procedure**
    - **Bone operation**
      - **Operation on fracture**
        - **Fixation of fracture**
          - **Fixation of fracture using pins**
          - **Fixation of fracture using wire**
          - **Fixation of fracture using screws**
          - **Fixation of fracture using plate**
          - **Fixation of fracture using intramedullary nail**
          - **Fixation of fracture using dynamic compression system**

The hierarchy is built up from a set of links between parents and children. *Operations and procedures* is a parent of *Musculoskeletal procedure*, which itself is a parent of *Bone operation*. Each parent may have several children and each child several parents (see the document “Clinical Terms Version 3 – Main File Structure: Overview and Technical Description”). Other detail, such as the locking and method of reaming of an intramedullary nail, the type of wire used and the site of the fracture is specified by **qualifiers**.

2.2 Case 2. Atoms which Describe the Internal Characteristics of a Concept

An **atom** is the expression of the intrinsic meaning of a concept with respect to value only concepts (and rarely to other core concepts). It does not provide additional detail, but describes some characteristic of the intrinsic meaning of the concept. Take the of *Simple mastectomy* (an operation, usually for cancer, in which the breast is removed), which can be mapped onto the description *Site: Breast*. This adds no extra detail in itself and so is not a ‘qualifier’. However, it is useful to have such descriptions, as they are helpful in analysis. For example, they help in searches for all operations involving the breast. We refer to these as #intrinsic characteristics’ or ‘atoms’ and this provides a semantic definition for a concept; this aspect is covered later in this document.

2.3 Case 3: Qualifiers Describing Facts About the World

A concept like *Amoxycillin 250 mg capsules* (an antibiotic) may require an additional statement that these are only available on prescription. This is a **fact**, which can only be identified through some knowledge of the world. It is certainly not intrinsic to the meaning of *Amoxycillin 250 mg capsules* as its status may change at any time. These are therefore not strictly qualifiers, although the distinction between Qualifiers of detail and Facts may be blurred. These too are covered in this document.

2.4 Case 4: Qualifiers for Establishing the Context of Use of a Term

Many core Read concepts, with or without qualifiers for providing extra detail, have no real meaning in isolation outside the context of the clinical record. Take the term **Normal blood pressure**. This may indicate:

a) a clinical finding (in a patient)

b) an expected finding (of a patient)

c) a goal to achieve (for a patient)

Providing three different codes for every symptom, sign, test result and disease would not be sensible, partly because there are actually rather more than three different contexts in which the term may be used. This kind of information is best supplied therefore as qualifiers. An initial set of context terms has been delivered by the Context of Care working group, for further testing. Further information is available in the document "Clinical Terms Version 3 – Context and Clinical Records".

2.5 Case 5: Qualifiers for Uncertainty and the State of Execution of an Action

Most interpretations are subject to uncertainty (e.g. possible, excluded, definite, probable) and procedures may be simply considered or else planned, scheduled and finally performed. An initial set of these terms has been delivered by the Context of Care group for testing. Further information is available in the document "Clinical Terms Version 3 – Context and Clinical Records".

3. Brief Overview of the Template Structure

The **template file** is the structure used in CTV3 to allow the controlled addition of qualifiers for detail (and context); atoms for the expression of the intrinsic characteristics of concepts; and facts. It has a basic structure in which two non-attributes are linked by an attribute. The focus non-attribute concept of the template is known as the **Object**, the additional non-attribute relating to the qualifier, atom or fact is known as the **Value** (not to be confused with *value only concepts*). The linking attribute concept is known as the **Attribute**. All concepts that have a role as an object, attribute or value are Read coded:

	Object	**Attribute**	**Value**
Example	*Fixation of fracture*	*Site*	*Shaft of femur*
Read Code	X601G	X9019	7NA83
Linguistic role of Read Code	Non-attribute	Attribute	Non-attribute

The template mechanism as described above is used to add additional qualifying detail; define intrinsic characteristics (atoms); and apply facts:

	Object	**Attribute**	**Value**
Qualifier	Fixation of fracture	*Site*	Shaft of femur
Atom	Meningococcal meningitis	*Causative agent*	Neisseria meningitides
Fact	Amoxycillin 250mg capsule	*NHS prescribing status*	NHS prescribable

The role of a value within a template should not be confused with value only concepts. A value only concept can act as either an object or a value (but never an attribute); likewise a core concept can act as either an object or a value (but never an attribute); an attribute concept can only act as an attribute. However, a core concept can never be a value to a value only concept. This is illustrated below, with examples of both qualifiers and atoms:

	Object	**Attribute**	**Value**	
	Core concept	Attribute concept	**Value only concept**	√
Qualifier	Rheumatoid arthritis	*Site*	Knee joint structure	
Atom	Meningococcal meningitis	*Causative agent*	Neisseria meningitidis	
	Core concept	Attribute concept	**Core concept**	√
Qualifier	Autonomic neuropathy	*Associated finding*	Diabetes mellitus	
Atom	Diabetic cataract	*Associated finding*	Diabetes mellitus	
	Value only concept	Attribute concept	**Value only concept**	√
Qualifier	Knee joint structure	*Laterality*	Left	
Atom	Right ventricle	*Laterality*	Right	
	Value only concept	Attribute concept	**Core concept**	χ

Formally a **qualifier** is the combination of an **attribute** (e.g. *Site, Laterality*) and a **value** (e.g. *Shaft of femur, Left*). Objects and attributes are always Read coded as stated above as are values unless they are dates or numbers (see later).

Potential qualifiers are grouped into templates, which describe the range of possible qualifiers that a drug, diagnosis, procedure, symptom (or any other medical concept) may take. The template for *Fixation of fracture using intramedullary nail* (an operation in which a long straight piece of metal is inserted into the middle of a shaft of bone to hold the fractured fragments in line) which is the **object** of discussion in this case, might include:

Object	**Attribute**	**Value**
Fixation of fracture using intramedullary nail	Site	Bone structure
Fixation of fracture using intramedullary nail	Type of nail	Flexible nail
Fixation of fracture using intramedullary nail	Type of nail	Locking nail
Fixation of fracture using intramedullary nail	Type of nail	Rigid nail
Fixation of fracture using intramedullary nail	Method of reaming	Hand
Fixation of fracture using intramedullary nail	Method of reaming	Powered method
Fixation of fracture using intramedullary nail	Method of reaming	Powered rigid method

Grouping the qualifiers that may be affixed to a core concept into templates makes entry of these into the medical record easier and is a formal mechanism of controlling additions that are allowable. Once the core concept such as *Fixation of fracture using intramedullary nail* has been chosen, qualifiers may be presented immediately to the user for selection, without further searching being necessary. Some of the values in the table may need to be browsed – for example the *Site*: of the fixation of fracture is described as *Bone structure*. Any child of *Bone structure* (e.g. different bones *Femur, tibia, Humerus* and different parts of bone e.g. *Head of femur*) is valid. This is possible because the value only concepts (as well as the core concepts) are arranged in a pure sub-type hierarchy (described in the document "Clinical Terms Version 3 – Main File Structure: Overview and Technical Description"). A good user interface should allow the user to type 'fem' when selecting a site and be returned *Femur* in the picking list (and <u>not</u> *Femora hernia* or *Female baby*).

Users do not have to choose any qualifiers – in the same way that users do not have to choose a core concept from the bottom of the hierarchy, but may settle for a higher level, more general concept.

Each attribute – value pair is labelled as being a qualifier, an atom (intrinsic characteristic) or a fact, through using an extra field. In the example below, a *Fixation of fracture using an intramedullary nail* always involves a Site: *Bone structure* so this is labelled as an <u>atom</u>; its *Nail type*: is extra detail so is labelled as a qualifier (Q). Lastly, the *NHS prescribing status*: *NHS prescribable* of *Amoxycillin 250 mg capsule* is a fact (F). It describes the fact that these capsules are only available on a medical prescription and cannot be bought over the counter by a patient at a local pharmacy.

Object	Applicable attribute	Applicable value	Characteristic status
Fixation of fracture using intramedullary nail	Site	Bone structure	A
Fixation of fracture using intramedullary nail	Nail type	Flexible nail	Q
Fixation of fracture using intramedullary nail	Nail type	Locking nail	Q
Fixation of fracture using intramedullary nail	Nail type	Rigid nail	Q
Amoxycillin 250mg capsule	NHS Prescribing status	NHS prescribable	F

4. Principles Behind the Template Structure

The qualifier file structure has been designed with certain principles in mind.

Clinicians need to represent all of the detail that health carers feel is important to code in a computerised medical record
The criterion for deciding whether or not to include an item of medical information as structured coded data is whether or not this information may be the subject of a search or analysis across medical records at some later date. Thus, knowing that a patient’s relative is on holiday until a certain date may be essential to the organisation of a patients discharge on a particular date, but there is no general requirement to retrieve all patients with a similar entry. Such detail is best recorded using free text.

Health carers need terms of different levels of detail
Health carers have quite different requirements for detail in different areas of their work. Cardiothoracic surgeons, for example, will wish to record great detail about cardiothoracic diagnosis and procedures, but may only require general terms for a patient’s past gastrointestinal history – just sufficient as a reminder perhaps. Where possible, detail that a generalist requires is placed in the core terms and other detail may be placed in qualifiers. This is a rule-of-thumb and inevitably judgemental decisions have to be made.

Clinicians wish to see and key on to natural clinical terms
There are many variants of fractures, depending on which site is involved, where the joints are etc. These variants can be captured using qualifiers to a simple concept such as **Fracture of wrist**. However, some complex concepts requiring complex clusters of qualifiers are clinically common and have eponyms e.g. *Colles' fracture*, which is a fracture of the metaphysis of the distal radius with no articular surface involvement and normal alignment of the distal radio-ulnar joint (that is a fracture involving one end of a bone in the forearm, near the wrist, which does not cut into or disturb the joint). These are natural clinical terms and must be included in the terminology.

Users should have some guarantee that records they produce can be shared by others
Most users prefer to adopt a standard unambiguous style when using Read Codes, so that their computer entries may be communicated to others (in discharge and referral letters for example), so that they can use standard reports for audit – allowing different users to compare data and so that they can ensure their own data is structured uniformly and so be amendable to analysis. **Templates** provide this capability as they are a shared resource, or a set of qualifiers that the health care profession has agreed to use in common, but the following two principles are also required to meet this aim:

We need to detect terms (or clusters of terms) that are equivalent
Occasionally there may be two or more ways of describing exactly the same medical concept. For example we can describe a *Streptococcal meningitis,* either by using a single concept or by using a higher level concept with a qualifier – *Bacterial meningitis – Causative agent: Streptococcus.* When this occurs a mechanism for detecting that these two constructs are equivalent is required, in case two health carers enter the information differently for different cases. Otherwise when asking a question of the database to retrieve information (e.g. all cases of *Streptococcal meningitis*) we might only recover a subset of the true number of cases, severely compromising analysis.

Terms and expressions must be unambiguous, if they are to be shared by health carers
The classical way of implementing qualifiers in a terminology or classification is simply to form an expression by associating a qualifying term such as a site code (e.g. *Skin of arm*) with a core concept (e.g. *Reconstruction with skin graft* – an operation to replace missing skin with skin carefully removed from elsewhere). Unfortunately, this can be misleading. Is the arm the harvest (donor) site or the recipient site for the skin graft? To ensure the meaning is clear, the role that the **value** (e.g. *Skin of arm, Skin of front of thorax*) plays in the expression is made clear using an **attribute** (e.g. *Donor site, Recipient site*).This way two sites might be specified. For example *reconstruction with skin graft* in which *Donor site: Skin of arm and Recipient site: skin of front of thorax.*

Entering detail should not place undue strain on the user in the consultation
If a user had to search separately for five terms to describe a single procedure (*Reconstruction with skin graft, Donor site, Recipient site, Skin of arm* and *Skin of front of thorax*), the flow of a consultation would be seriously affected. To circumvent this, the use of the template file allows convenient access to potential qualifiers once the core term (*Reconstruction with skin graft*) has been chosen and the range of possible attributes and values can immediately be offered on-screen for selection.

5. Introduction to the File Structure

The proposed file structure is introduced by way of a number of worked examples and technical detail is left to the technical annexe.

The files covered are:

- The Main File Structure (originally described as Version 3.0 and detailed in the document “Clinical Terms Version 3 – Main File Structure: Overview and Technical Description”).

- The Template file

5.1 Overview of the File Structure

Full details of the file structure are given in the technical section of the document. However, the next section, describes how the principles are achieved in the file structure. In summary, the files are as follows:

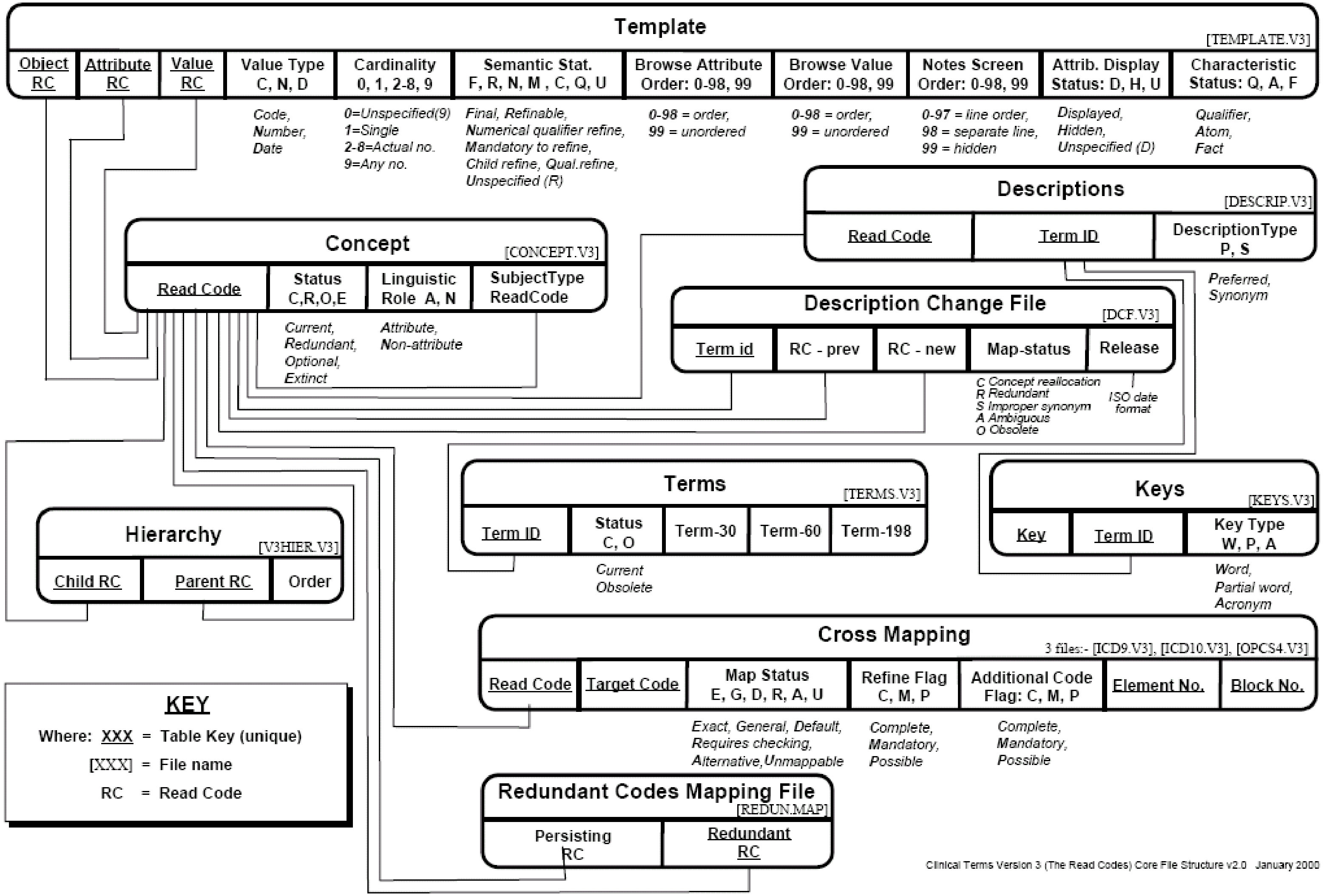
File	Brief description
Term file*	Contains the 30, 60 and 198 character versions of each term
Concept file*	Contains information about Read Codes such as whether the code is current, optional, redundant or extinct
Description file*	Links the terms to the Read Codes and defines whether the term is the preferred term or a synonym
Hierarchy file*	Contains links between parents and children that make up the hierarchy
Key file	Contains 10 character keys and links these to terms
Specialty files	One for each speciality, these contain codes that the specialty is particularly interested in

Cross-mapping files*	There is one file for each classification or coding system that is being mapped to. Described in a separate document “Clinical Terms Version 3 – Cross Mapping File”
Template file*	The mechanism by which “core concepts” are defined (semantic definitions or atoms) and associated with appropriate potential qualifiers and facts. Described in detail in this document
Redundant codes mapping file*	Contains the mapping between redundant Read Codes and the persistent equivalent (Read Coded) concept
Description Change file	Contains the mapping of terms and concepts whose statuses have changed between releases. Described in a separate document “Clinical Terms Version 3 – Managing Change: Description Change File”

* = Illustrated in the diagram on the following page

Details in the concept file are required for a full understanding of the qualifier scheme. Therefore only this file and the template file (the sole extra file in Version 3.1) are described below. Full details of these files are given in the technical annexe.

Clinical Terms Version 3 (The Read Codes) Core File Structure



Template [TEMPLATE.V3]

Object RC | Attribute RC | Value RC | Value Type C, N, D | Cardinality 0, 1, 2-8, 9 | Semantic Stat. F, R, N, M , C, Q, U | Browse Attribute Order: 0-98, 99 | Browse Value Order: 0-98, 99 | Notes Screen Order: 0-98, 99 | Attrib. Display Status: D, H, U | Characteristic Status: Q, A, F

Code, Number, Date

0=Unspecified(9)
1=Single
2-8=Actual no.
9=Any no.

Final, Refinable, Numerical qualifier refine, Mandatory to refine, Child refine, Qual.refine, Unspecified (R)

0-98 = order, 99 = unordered

0-98 = order, 99 = unordered

0-97 = line order, 98 = separate line, 99 = hidden

Displayed, Hidden, Unspecified (D)

Qualifier, Atom, Fact

Descriptions [DESCRIP.V3]

Read Code | Term ID | DescriptionType P, S

Preferred, Synonym

Concept [CONCEPT.V3]

Read Code | Status C,R,O,E | Linguistic Role A, N | SubjectType ReadCode

Current, Redundant, Optional, Extinct

Attribute, Non-attribute

Description Change File [DCF.V3]

Term id | RC - prev | RC - new | Map-status | Release

C Concept reallocation
R Redundant
S Improper synonym
A Ambiguous
O Obsolete

ISO date format

Terms [TERMS.V3]

Term ID | Status C, O | Term-30 | Term-60 | Term-198

Current Obsolete

Keys [KEYS.V3]

Key | Term ID | Key Type W, P, A

Word, Partial word, Acronym

Hierarchy [V3HIER.V3]

Child RC | Parent RC | Order

Cross Mapping 3 files:- [ICD9.V3], [ICD10.V3], [OPCS4.V3]

Read Code | Target Code | Map Status E, G, D, R, A, U | Refine Flag C, M, P | Additional Code Flag: C, M, P | Element No. | Block No.

Exact, General, Default, Requires checking, Alternative,Unmappable

Complete, Mandatory, Possible

Complete, Mandatory, Possible

KEY

Where: XXX = Table Key (unique)
[XXX] = File name
RC = Read Code

Redundant Codes Mapping File [REDUN.MAP]

Persisting RC | Redundant RC

Clinical Terms Version 3 (The Read Codes) Core File Structure v2.0 January 2000

5.2 Concept File

This file contains details of each Read-coded concept. Against each Read Code is stored its linguistic role, among other details. Two populations of concepts are distinguished by their linguistic roles.

First there are concepts with the status of **Attributes** (e.g. *Site, Laterality*) which can only be used to define some aspect of another concept which can be further detailed.

Secondly there are **Non-attributes** which include all other Read-coded concepts such as:

Core concepts E.g. Procedures, Disorders, Signs and Symptoms. The value added file (contyupe.v3) contains a listing of which sections of the Read Codes may be used as core concepts

Value only concepts E.g. Anatomical sites Arm, Leg, Left, Right. The value added file (contype.v3) contains a listing of which sections of the Read Codes must be used as value only concepts

5.3 Template File

This file lists:

- Potential **qualifiers** that may further detail a chosen concept;
- The '**atoms**' or intrinsic characteristics that describe the meaning of a concept;
- Any **facts** about the concepts.

The description in this section will largely refer to the handling of qualifiers, pointing out how the other two classes of information are handled where this is different.

Choosing Qualifiers

Object	Applicable attribute	Applicable value
Osteoarthritis	Site	Joint structure
Osteoarthritis	Severity	Mild
Osteoarthritis	Severity	Moderate
Osteoarthritis	Severity	Severe

Once a user has chosen a concept like *Osteoarthritis*, a list of attributes (*Site, Severity*) can then be offered and the user may wish to select values for one or more of these (e.g. *Hip joint structure, Moderate)*. Any descendant (e.g. *Hip joint structure*) of an applicable value concept (e.g. *Joint structure*) is also valid. Therefore once the user has chosen to see possible values that *Site* may take, the following might be offered:

Head and neck joint
Joint of vertebral column
Joint of shoulder girdle or upper limb
Joint of pelvis or upper leg
Joint of lower leg or tarsus
Joint of thoracic wall

A user might also type in a key 'Hip' at this point and reasonably expect a system to recover *Hip joint structure*.

Value Type – not all values are Read Coded

Attributes like other medical concepts, are given Read Codes – and in the 'real' template table it is the Read Codes that would be found, rather than the word *Site.* Most attributes (*Site, Laterality*) will take values that are also Read coded (e.g. *Neck of femur, Left*). Currently only attributes that relate to Read Codes value concepts have been released, but other developmental attributes (due for release in the future) e.g. *date of onset* require the values to be dates and others (e.g. *Numerical value of result*) will take numbers (e.g. 6.2). The field **value_type** within the template file of CTV3 defines whether the value of the attribute is Read coded (C), numerical (N), or a date (D). Formats for dates have not yet been identified, but will be ISO 8601 compatible. These formats need to be agreed along with National Policies including those relating to clinical message transfer. What appears in the **applicable_value** varies depending on the **value_type**:

In the simple case, as above, where the **value_type** is C, then the field with the Read Code of one of the set of possible values that the attribute might take. In the case of a numerical or a date value, the field is left empty.

Choosing Qualifiers of Qualifiers

Once a qualifier has been chosen (e.g. *Site: Hip joint structure*), the user should then be offered anything (e.g. *Laterality*) that might qualify the selected value (e.g. *Hip joint structure*). This relates to qualifiers of qualifiers, sometimes referred to as "nested qualifiers". These qualifiers also appear in the template file. There is no theoretical restriction on the number of qualifiers that may be nested in this way.

Object	Applicable attribute	Applicable value
Osteoarthritis	Site	Joint structure
Hip joint structure	Laterality	Side

Note that the children of *Joint structure* include *Hip joint structure* (also *Knee joint structure, Shoulder joint structure,* etc.) and the children of *Side* are *Left, Right, Unilateral and Bilateral.*

If a user describes a case of Osteoarthritis affecting a left hip joint the expression to be added to the medical record would be *Osteoarthritis* qualified by *Site: Hip joint structure* in turn qualified by *Laterality: Left*. Alternatively, the user might have picked the more detailed core concept Osteoarthritis of hip which has an atom *Site: Hip joint structure* which can be further qualified by *Laterality: Left.*

Characteristic Status – what kind of information is in the template?

The three broad classes of information held in the template file are distinguished by the characteristic status flag. Characteristic status may be:

> **Q (for Qualifier)**: the table contains a qualifier that might supply extra detail that a user might choose to further describe a concept. For example, *Fixation of fracture using an intramedullary nail* may be qualified by the revision status.
>
> **A (for Atom)**: the table contains information about the meaning of the concept being described. We refer to these as intrinsic characteristics or atoms of a concept which form part of its semantic definition. These may only be selected as qualifiers if they themselves are further qualified or if one of their children is chosen. Selecting just the atom would add nothing extra to the original concept. For example, the concept *Osteoarthritis of hip* will have the atom *Site: Hip joint structure*. Adding this as a qualifier to *Osteoarthritis of hip* would not add any useful information. However, we might qualify *Hip joint structure by Laterality: Left* in which case we would be adding new information.
>
> **F (for Fact)**: the table contains facts about the concept. This is used only for historical drug dictionary data (pre July 2006 data, prior to cessation of monthly maintenance of CTV3 drugs), where information such as the legal category and availability of drugs is held in the form.

Object	Applicable attribute	Applicable value	Characteristic status
Fixation of fracture using an intramedullary nail	Revision status	Primary	Q
Fixation of fracture	Revision	Revision	Q

using an intramedullary nail	status		
Osteoarthritis of hip	Site	Hip joint structure	A
Digoxin 125 micrograms tablet	Legal category	Prescription Only Medicine	F

Semantic Status – guidance on when to choose qualifiers

We have seen examples of situations in which the value of a qualifier (e.g. *Bone structure of femur*) may be further refined wither by choosing a child or by further qualifying it (e.g. *Laterality: Left*), but there are some situations it is undesirable to allow the value in a qualifier to be further refined.

The **semantic status** flag describes which operations can be carried out on the value:

> **R (may be Refined)**: the usual case as when choosing a value (*Symmetrical, Asymmetrical*), for the *Distribution* of joints affected in a case of *Osteoarthritis*.
>
> **F (is Final and must not be refined)**: occasionally it is either not possible to further refine a value or it does not make sense to do this. In the case of a *Total thyroidectomy*, it does not make sense to offer the different aspects of thyroid structure to the user as all of the thyroid is removed. Therefore the Attribute + Value pair *Site: Thyroid structure* is marked with semantic status = F.
>
> **M (mandatory to refine)**: in these cases the value is an excellent way of grouping a set of values, but not informative enough itself. Atoms (Intrinsic characteristics) fall into this class. One of the children of the applicable value must be chosen or one of its own qualifiers must be chosen before the attribute and value can be considered as a qualifier. In the case of *Osteoarthritis of hip joints*, it would not be useful to choose *Site: Hip joint structure* as a qualifier, but if *Hip joint structure* were further detailed, for instance by choosing the *Laterality*: (attribute) of the hip joint to have the value (L*eft, Right, Unilateral or Bilateral*), then the user would have specified a useful qualifier.
>
> **Q (only Qualifier may be refined)**: only offer the user qualifiers of this value to choose and not its children.
>
> **C (only Children may be refined)**: only offer the user children of this value to choose from and not qualifiers of this value.
>
> **N (only Numerical qualifiers may be refined)**: only offer the user the ability to add a numeric (and ***Units*** and ***Mathematical sign***) to this value and not other qualifiers, nor any children.

U (Unspecified): only used in early pilot sets, when semantic status has not been specified (not currently released).

Object	Applicable Attribute	Applicable Value	Value Type	Semantic Status	Characteristic Status
Pilonidal abscess	Causative agent	Bacteria	C	R	Q
Osteoarthritis of hip	Site	Hip joint structure	C	M	A
Hip joint structure	Laterality	Side	C	M	Q
Total Thyroidectomy	Site	Thyroid structure	C	F	A

Additional Fields

The template file also contains a number of additional fields. These have been included to assist in determining how qualifiers should be presented on-screen and to determine how many values can be chosen. These additional fields are:

Browse Attribute Order,

Browse Value Order,

Notes Screen Order,

Attribute Display Status and

Cardinality

At the time of writing, these fields have not yet been fully implemented and they are not described in this document.

5.4 Analysing Databases Containing Qualifiers

A key issue has dogged those few terminologies that have defined the use of qualifiers or modifiers. When these are introduced, then a fundamental requirement of any coding scheme – that of supporting the capability of recovering information at a later date when analysing a database of patient cases – may be put in jeopardy. This occurs if there is no **information model** describing a way in which health carers can share a common format. For example, if there are different ways of recording *acute appendicitis* users may not be aware of all of the possible forms of saying this and there is potential to miss cases when searching a database.

The problem arises because there is a need to satisfy two conflicting requirements: to record clinical terms (such as a *Colles' fracture*) and to describe fine degrees of detail. *Colles' fracture* is simply one variant of a range of fracture that is particularly common and has an eponym. To describe many other variants we need to qualify a more general concept like *Fracture*.

Similarly, certain forms of *Osteoarthritis,* because they are commonly used, have been made explicit as terms (e.g. *Osteoarthritis of hip*), but to express other rarer forms, a qualifier must be added to *Osteoarthritis*.
Any child of *Joint structure* (and therefore any joint) may qualify *Osteoarthritis.* Now, as in the appendicitis case previously described, there is a potential problem in that a user may record a case of osteoarthritis affecting a hip joint in two different ways, either using the single term or qualifying the basic term *osteoarthritis*, but when posing a query to a medical records system, the user will wish all such cases to be retrieved, no matter in what form users have volunteered the information.

To achieve this we need first to be able to translate from one form to the other. This is possible because in the CTV3 qualifiers have precisely the same structure (attributes and values) as atoms which describe the meaning of more complex concepts. For example, *Osteoarthritis of hip* is defined as containing the atom *Site: Hip joint structure*, which matches precisely the simpler concept of *Osteoarthritis* qualified by *Site: Hip joint structure.*

object	**Attribute**	**Value**	**value type**	**semantic status**	**characteristic status**
Osteoarthritis	Site	Hip joint structure	C	R	Q
Osteoarthritis of hip	Site	Hip joint structure	C	M	A

System designers may wish to use this resource in different ways. The document "Clinical Terms Version 3 – Introduction to Information Retrieval" describes how an information system using qualifiers may be analysed.

5.5. Inheritance

The following fragment of hierarchy show the parents of a Colles' fracture:

Read Thesaurus
Clinical findings
Disorders
Accidents, Poisoning and Injury
Injury
Injury of Body Structure
Musculoskeletal Injury
Bone Injury
Fracture
Fracture of bone of upper limb
Fracture of radius

Fracture of distal end of radius
Colles' fracture
Smith's fracture
Volar Barton's fracture, etc.

All fractures may be classed as closed or open – which we may capture in the following template.

object	applicable attribute	applicable value	value type	semantic status	characteristic status
Fracture	Communication with wound	Open injury	C	R	Q
Fracture	Communication with wound	Closed injury	C	F	Q

An open wound is one in which the skin is broken and bacteria may get in from the outside to infect the site of the fracture. In a closed fracture, the skin is unbroken. Every child of *Fracture* inherits these qualifiers unless the attribute is restated against a child with a new set of values. For example, against the (now optional) concept of a *Closed Colles' fracture*, the template contains the atom *communication with wound: Closed injury* and not the alternative.

In the template file distributed by NHS Connecting for Health, the inheritance is already done so that every child of *Fracture* will have the attribute *Communication with wound:* and each of the two values against it. An 'uninherited' version of the template file (implicit file) is included as a value added file (VAF) for those users who wish to inherit qualifiers at run-time.

To inherit values correctly, it is important that the hierarchy contains only subclass (or 'kind of' or 'type of') links. Indeed this is one of the powerful advantages in organising the hierarchies in such a formal way. If this were not the case problems would arise; for example, if a "part of" hierarchy was allowed as illustrated below:

Bones
Bones of forearm
Radius
Bone marrow of radius
Osteoclasts of the radius

Examining this hierarchy reveals that assumptions using the hierarchy cause potential errors: a fracture can involve a *Bone, Bone of forearm*, or the *Radius* but cannot be true of *Osteoclasts of the radius* (osteoclasts are specialised bone cells only viewable under a microscope). The problem is that *Osteoclasts of the radius* are not a 'type of' *radius*, but a part of the *Radius*. Inheritance usually only works across 'type of' links. This requirement has influenced the way that the hierarchy is being constructed in the Clinical Terms Projects. If 'part of' links are required in the future these will be placed in the template file, like all other relationships between concepts which are not the special relationship ‘subclass’.

Currently there is one exception to the rule sub-type hierarchy that occurs in the history and observation chapter of CTV3. This can be currently detected by identifying a change of the subject_type (**concept file**) of parent and child. This discrepancy currently relates to a type of concept (Findings category type) that requires an addition of a qualifier (either numerical or interpretative) to formulate a clinical finding. For example *Blood sugar level* requires the addition of a numerical value as well as units to create a valid finding. Although inheritance of qualifiers can occur successfully across this type of boundary change there are no plans for further areas of this kind. Furthermore rationalisation of this area into a pure hierarchy of finding categories is currently under consideration.

6. Support of the Template Structure

Previous academic work and released products have highlighted both the requirement for qualifiers and the potential pitfalls to be avoided.

There is a general agreement that to be able to capture the detail required for an adequate description of patient care, there is a need for qualifiers. It is simply not practicable to combine all detail into single concepts (and terms) as these become long-winded, unreadable, difficult to analyse and difficult to maintain. Similarly there is general agreement that if qualifiers are introduced, then an information model must be allied to the terminology to ensure that a query offered to a stored database of clinical records can retrieve all of the desired records. In CTV3, the template file performs both of these functions.

The following key points briefly summarise the advantages of the approach taken which is then followed by a critique of some of the decisions taken and describe some of the issues that a system developer will have to deal with compared to the implementation of earlier versions of the Read Codes and more simple coding schemes.

6.1 The Advantages

The general goals of the CTV3 file structure are to:

- Increase the ability of the Read coding scheme to deal with detail necessary for an accurate coded clinical record;

- Provide better support for the retrieval of information from medical record systems;

- Improve the ability and accuracy of mapping to target coding systems and classifications;

- Build in a natural path for evolution and upgrade from earlier versions of the Read Codes, so that designers can rely on compatibility with the past and see how future development will proceed;

- Remove optimisations from the file structure, which should remain conceptually clear. Optimisation is left up to system designers.

The CTV3 file structure should also:

- Make possible the design of good human-computer interfaces so the user may rapidly select qualifiers.

- Lead to the design of more sophisticated search mechanisms for information stored in clinical information systems, through the provision of a mapping from complex concepts onto a flatter representation of their atomic components.

6.2 A Panacea for All Coding Evils?

What are the drawbacks of the scheme and could Read Codes Version 2 have done the job as well or better?

If current users of the Read Codes (4-Byte Set or 5-Byte Set: Version 1 or Version 2) wish to take up CTV3 then their system developers will need to rebuild the patient record database. However, all the information they have previously collected using the older versions will be retained in their CTV3 database, viewable and available during analysis of case records (described in the document “Clinical Terms Version 3 0 Incorporation of Earlier Versions of the Read Codes (The Superset)”).

Implementing qualifiers requires more storage space both to store the Read Code release and for patient notes recorded. The term identifier (5 characters) is longer than the old term code (2 chars) and any chosen qualifiers will require extra Read Codes and term identifiers. This will be of far greater concern to current users who may still have simple hardware (e.g. small hard disk). The increase in storage requirements has come about because of the detail that specialist wish to include in the codes. Indeed introducing qualifiers has made this expansion possible as the simple ‘enumerative’ scheme of Version 2 would require far more space to do the same job.

More design work is required of system developers, particularly if data input is to be a rapid and non-traumatic experience for users and if analysis of medical records in clinical information systems is to be efficient. On the other hand, greater opportunities are created for those wishing to explore these aspects.

The greater complexity of the file structure requires more advanced quality assurance routines if the codes are to be released without errors. However, these routines become easier to design as less is implicit within the meaning of the codes and the NHS Connecting for Health can adopt a more consistent authoring style. More than 500 quality assurance rules are applied before each release, most of these running on a daily basis.

CTV3 cannot cope with every possible statement that a clinician wishes to put in a clinical record. For example, when describing a dressing apraxia, an occupational therapist may wish to say that a patient puts his shirt on over his left foot. However, it is doubtful whether it is important to code all of this information as it is unlikely that anyone will want to retrieve these particular cases. What must be termed or coded is the fact that a patient is unable to put on a garment because of an apraxia. The rest of the detail that is peculiar to this individual might sit well in free text.

When detail is placed in qualifiers, some of the naturalness of medical language is lost, but there is also much evidence to show that simple structured data presentation allows information to be more easily assimilated.

Occasionally the list of values for an attribute may be over complete. For example, though in theory any artery might to used to construct a coronary artery graft, it would be highly disadvantageous to the patient to use the middle cerebral artery (an essential artery in the brain). However, it is usually not practical to record all such exclusions. It is arguably more dangerous for a user to enter a plausible value that is wrong than an implausible one, as only the latter is easily spotted.

6.3 Atoms and The Future

A major function of the template file is to define which qualifiers may be used with which core terms. A second function, which will become more pronounced, is the description of a concept in terms of its ‘atoms’. We have already seen how a *Colles’ fracture* may be defined in terms of such elements as a fracture of bone, sited at the wrist, with normally aligned fragments of bone. Even when fully populated, the template file will only contain those atoms necessary to ensure that during analysis all concepts can be retrieved whether stored as a detailed core concept (e.g. *Colles’ fracture*) or as a simpler core concept (e.g. *Fracture*) with qualifiers.

Eventually, the user might have access to a complete description of each concept in terms of atomic characteristics. For example, the surgical method which is implicit in a *Fixation of fracture using an intramedullary pin* might be defined as an atomic characteristic (e.g. *Method: Fixation*). At this point, rather than relying on the hierarchy as a mechanism for search, the user will be able to specify analyses in terms of a set of characteristics. For example, ‘retrieve all patients who have had a reduction and fixation not using an intramedullary pin’. This is described in more detail in the document “Clinical Terms Version 3 – Introduction to Information Retrieval”.

Technical Annexe

In this section, the conceptual model underlying the qualifier files is outlined; the file structures are collected and repeated, then followed by a brief discussion of some implementation issues.

T. 1 Conceptual Model of Qualifiers

T.1.1 Logical Data Structure

The diagram below describes the relationship between entities relevant to qualifiers. The data structure is described as an object model. The Relation ‘has subclasses’ identifies one or more partitions of the class. Each partition contains exclusive classes; for example, a concept is either a **Read coded concept** or a **Non-Read coded concept,** but cannot be both. The entities shown in the diagram (e.g. **Read coded concept, Non-Read coded concept**, etc) are described in section T.1.2 on the next page.

In the diagram, boxes are entities. Interconnecting lines between boxes represent relationships. The cardinality of each relationship (many-to-one, one-to-one, etc.) is given by the “crow’s feet”. Therefore, in the diagram, one **Attribute for object** may relate to several **Object-attribute-value triples**. The asterisk on the relation **plays role of** indicates that either a **Non-Read coded concept** or a **Non-attribute** may play the role of a **Value**.

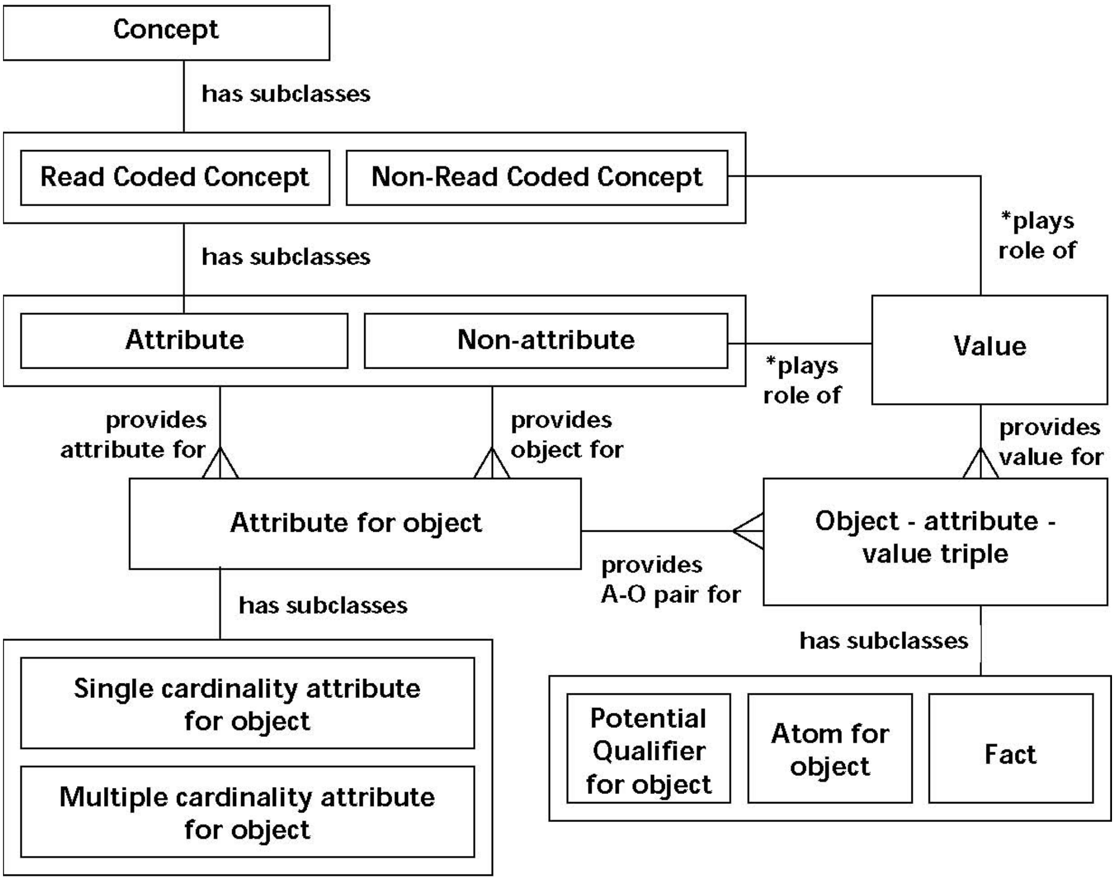


Figure 2: Diagram describing entities relevant to qualifiers, facts, links and intrinsic characteristics

The model includes all elements of the file structure important to understanding the interrelationships between attributes, objects and values. Items in the template file that relate only to display and choice of qualifiers, etc. are not modelled.

T.1.2 Entity Descriptions

Concept
Concepts include anything that is relevant to the practice of health care and that we might wish to represent in a clinical information system.

Read coded concept
A concept that is associated with a Read Code.

Non-Read coded concepts
A number or a date (includes time) that is not associated with a Read Code.

Non-attribute
A Read coded concept that describes an item of health care interest in the world. **Non-attributes** may play the roles of **object** or **value** in **object-attribute-value** triples.

Often mentioned in discussion of the Read Codes is the informal notion of a **Core concept** (not shown in the diagram). This is one of a specified population of **non-attributes** that includes drugs, investigative and therapeutic procedures, laboratory tests, signs and symptoms and disorders. The other class of non-attributes is **Value only concepts**; these provide additional detail (of atoms or facts) relating to core concepts and can not be recorded independently e.g. anatomical sites, organisms. A single Read Code represents each core concept and value only concept.

Attribute
A Read coded concept that defines the role that a value is playing when qualifying an **object**. E.g. the role that the value *Skin of arm* plays when qualifying the object *Reconstruction with skin graft* might be determined by the attribute *Donor site*:. Another possible role would be *Recipient site*:.

Value
Values may have Read Codes (when they are drawn from the class of **non-attributes**) or may be **Non-Read coded concepts** (value is therefore a polymorphic type). These other non-Read coded concepts have not currently been released, but two types of these have been distinguished: **numerical values** (e.g. *2*) and **dates**.

Object-attribute-value triple
Concepts may be linked together using a single syntactic form, the object-attribute-value triple, to express potential qualifiers for objects, facts about objects and intrinsic characteristics (‘atomic meaning’) of objects. Each of these classes is separately described below.

Potential qualifier for object
An attribute-value pair (e.g. *Site: Radius*) that is a potential qualifier of an **object** (e.g. *Fracture*).

Atom of object
Some attribute-value pairs (e.g. *Site: Bone structure of femur*) describe an aspect of the meaning of the object they are associated with (e.g. *Fracture of femur*). These may only act as qualifiers if the value is itself further qualified (e.g. *Bone structure of femur* is qualified by *Laterality: Left*) or if a child of the value (e.g. *Neck of femur*) is chosen.

Fact
A combination of an **attribute, object** and **value** that expresses some knowledge about the object. Facts in the historical CTV3 drug dictionary data (pre July 2006) cover the legal category and availability of drugs, for example.

Attribute for object
The combination of an attribute and an object.

Single cardinality attribute for object

A subset of **Attribute for object** pairings in which the potential set of values is mutually exclusive.

Multiple cardinality attribute for object
A subset of **Attribute for object** parings in which the set of values is not mutually exclusive – the attribute may take several values. In the file scheme, this class is further subdivided into those A**ttribute for object** pairings that take a maximum of 2, 3 … up to 8 pairings and those that have no ceiling.

T.1.3 Relationship descriptions

Has subclasses
Divides a class of concepts into two subclasses.

Plays role of
Non-Read coded concepts (e.g. numbers and dates) and **non-attributes** may both play the role of a value.

Is attribute in, Is object in, Provides A-O pair for, Provides value for
Each object-attribute-value triple consists of a value and an attribute for object paring, which itself consists of an attribute and a non-attribute.

T.1.4. Summary of the rules

The rules below further constrain the relationship between each entity.

- A concept in the Read Code system must be either a Read coded concept or a non-Read coded concept.
- A Read coded concept must be either an attribute or a non-attribute.
- A value must be either a non-attribute or a non-Read coded concept.
- Each value may belong to zero, one or several object-attribute-value triples.
- Each attribute may belong to zero, one or several attribute for objects.
- Each non-attribute may belong to zero, one or several attribute for objects.
- Each attribute for object must have one and only one attribute and one and only one non-attribute.
- Each attribute for object must belong to one or several object-attribute-value triples.
- Each object-attribute-value triple must have one and only one attribute for object and one and only one value.

- Each object-attribute-value triple must be either a potential for object, or an atom of object or a fact.

- Each attribute for object may be either a single cardinality attribute for object or a multiple cardinality attribute for object.

T.2 Relationship between Conceptual Model and File Layout

Each **Read coded concept** is list in the concept file. The linguistic role field determines whether it is an **attribute** or a **non-attribute.**

Object-attribute-value triples are represented in the template file, except for a subclass of facts (on pack sizes, manufacturers and NHS cost) which are represented in the EAN mapping file (see the document “Clinical Terms Version 3 – The Drug and Appliance Dictionary” concerning EAN mapping file and cessation of monthly maintenance of CTV3 drugs).

The subclass of Object-attribute-value triples (whether it is a **potential qualifier for object, atom of object** or **fact**) is described in the template file. It is distinguished by the characteristic status field.

The cardinality of **attribute for object** pairings (**single cardinality attribute for object and multiple cardinality attribute for object**) is also represented in the template file. It is designed in the cardinality field. This is a de-normalisation that means the cardinality of an A for O pairing is repeated for each potential value of the pairing.

The nature of each **value** (whether it is a **non-attribute** or a **non-Read coded concept**) is described in the value type field of the template file.

T.3 Summary of File Layouts

The file layouts are listed here. They are discussed in the text of the first section of the document. The following key holds:

char(n)	is a fixed length string n characters long.
M	means that the field is mandatory, i.e. will always contain a value (or not null in SQL terminology). It does not mean that a system designer must use the field.
underlined fields	denote the unique key for a table. Some tables have concatenated keys.

T.3.1 Concept File

read code	char(5)	M	The unique clinical concept identifier

concept-status	char(1)	M	Denotes the current applicability of the concept
			C = current
			R = redundant
			O = optional concept
			E = extinct

linguistic_role	char(1)	M	The linguistic role a concept may play. Attributes and non-attributes are not interchangeable. Attributes may only be used in the template file (although they are also arranged hierarchically and thus appear in the hierarchy file).
			A = attribute
			N = non-attribute

subject_type	char(5)	M	Read Code for the category of a code being described. These include: Procedure type Clinical findings type Drug type Attribute type

T.3.2 Template File

object	char(5)	M	The Read Code of the concept whose characteristics (potential qualifiers, atoms or facts) are being described.
applicable attribute	char(5)	M	The Read Code for the attribute.
applicable value	varchar(5)	M	The value that the object attribute rates If value_type = C, then this is the Read Code of a value the applicable attribute takes. If value_type = N, or If value_type = D, then this is blank.

value_type	char(1)	M	Defines the nature of the value:
			C= Coded (Read) value
			N = Numerical value
			D = Date value

cardinality	char(1)	M	Specifies whether the attribute may have more than one value in the context of this template. 0 = Unspecified 1 = Single value 2…8 = maximum of 2…8 values 9 = Any number of values The default behaviour of 'Unspecified' (0) should be 'Any number of values' (9).
semantic_status	char(1)	M	Specifies whether the characteristic should or should not be further refined. Refining refers either to selecting a child of the value or a qualifier/linkage. F = Final. Either should not be refined or cannot be refined. R = may be Refined. The user may choose to refine this value if there are children or qualifiers. M = Mandatory. Value must be refined if selected as it is just a useful class name for a set of values. C = only refine Children. Q = only refine Qualifiers. N = only refine Numerical qualifiers. U = Unspecified. The default behaviour of unspecified should be 'may Refine' (R).
browse-attribute_order	char(2)	M	A number between 00 and 99

			defining the order in which attributes should appear when presented on screen for users to browse or choose from. 00 denotes first item. 01 denotes second item, etc. All items numbered 99 are unordered. These should be placed after any ordered items.
browse_value_order	char(2)	M	A number between 00 and 99 defining the order in which values should appear when presented on screen for users to browse or choose from. 00 denotes first item. 01 denotes second item, etc. All items numbered 99 are unordered. These should be placed after any ordered items.
notes_screen_order	char(2)	M	A number from 00 to 99; when 00 to 97 it denotes which qualifiers, once chosen, should always accompany the core term on screen and the order in which these should be displayed. 00 denotes first item. 01 denotes second item, etc. All items numbered 98 should appear on a separate screen-line as the attribute links two prime concepts. All items numbered 99 should not routinely appear on screen unless the user specifically requests to see this detail. Most qualifiers fall into this category.

attribute_display_status	char(1)	M	A flag indicating whether, once a qualifier has been chosen by the user, the attribute needs to be shown on screen when the user is looking at the qualifier. D = Display. In some cases it is essential to display an attribute on-screen to ensure the value is unambiguous. H = Hide. In other cases the attribute may be omitted as display of the value alone is considered to be unambiguous. U = Unspecified. The default behaviour should be to display the attribute.

characteristic_status	char(1)	M	A flag indicating whether the Attribute_value pair in a template entry is:
			Q = potential qualifier for object.
			A = Atom of object.
			F = Fact.

T. 4 Release Format

In release form, the files contain variable length fields, delimited with the vertical bar character”|” (ASCII value 7C hex). The bar will not be surrounded by spaces and will not be an allowable character in any of the data fields.

T.5 Implementation of the Man-machine Interface

NHS Connecting for Health aims to provide the raw material to facilitate the selection of qualifiers at the user interface. This is a major motivation behind linking attributes and values to the concepts they might qualify. Typically we would expect that once users have selected a core term (e.g. *Fracture of bone)*, they should then be presented with a list of attributes (e.g. *Site, Communication with Wound, Severity*) from which they might choose.

After selecting an attribute (e.g. *Site*), a list of permissible values might be presented to the user (e.g. a picking list containing the children of *Bone Structure such as Bone, Epiphysis of Bone, Metaphysis of Bone*). Once a user has chosen a value (e.g. *Femur*), then any children or qualifiers (e.g. *Laterality*) of the chosen value should next be offered.

When an attribute has a single value, but that value has several children it will be advantageous to present the user with both the parent(s) and child concepts. When an atom has children or qualifiers, these should be offered to the user, depending on its semantic status. However, the user cannot specify the atom alone as a qualifier.

The set of values in a picking list may be very large (e.g. *Bacteria*). There are several ways to handle this. The user might simply be shown immediate children and then browse steadily down the hierarchy in the usual way (e.g. picking *Bacillus*, then *Coliforms* then *E. Coli* at successively more detailed levels).

Alternatively, the user may wish to type in a string directly such as 'Coli' when confronted with the children of *Bacteria* to bypass the need to browse down the hierarchy. This might produce a picking list containing *Coliforms* and *E. Coli* (but not *Colitis* (a disorder) or *Colistin* (a drug) which are not descendants of Bacteria).

T. 6 References

1. Campbell KE, Musen MA. Representation of clinical data using SNOMED III and conceptual graphs. Stanford University, Stanford, CA, Knowledge Systems Laboratory Report KSL-92-13, November 1992.

2. Cote RA, Rothwell DJ, Palotay JL, Beckett RS, Brochu L. The Systematized Nomenclature of Human and Veterinary Medicine - SNOMED International (4 volumes), College of American Pathologists, April 1993.

3. Evans DA, Cimino JJ, Hersh WR, Huff SM, Bell DS. Toward a medical-concept representation language. J Am Med Informatics Assoc, 1994, 1:207-217

4. Read Codes File Structure Version 3: Overview and Technical Description, NHS CCC Technical Report, 1993

5. Rector AL, Nowlan WA, Kay S. Unifying medical information using an architecture based on descriptions. In: Miller RA, ed. Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care (SCAMC 90), Washington DC, 1990, 190-194.

6. Sanger N, Lyman M, Nhan NT, Tick LJ. Medical language processing: applications to patient data representation and automatic encoding.

1994, IMIA conference on Natural Language and Medical Concept Representation, to appear in Methods of Information in Medicine.

T.7 The CTV3 Document Series

Several other documents, each covering a different aspect of CTV3, are available. These include:

Clinical Terms Version 3 – Introduction

Clinical Terms Version 3 – Contents

Clinical Terms Version 3 – Main File Structure: Overview and Technical Description

Clinical Terms Version 3 – Cross Mapping File

Clinical Terms Version 3 – Incorporation of Earlier Versions of the Read Codes (The Superset)

Clinical Terms Version 3 – General Practice Flagged Subset

Clinical Terms Version 3 – Managing Change: Description Change File

Clinical Terms Version 3 – Drug and Appliance Dictionary

Clinical Terms Version 3 – Introduction to Information Retrieval

Clinical Terms Version 3 – Information Retrieval – Experiments

Clinical Terms Version 3 – Context and Clinical Records

Appendix: A

A.1 Glossary

The terms used throughout the document are explained in the text in some detail. Rather than redefining those here, pointers are given to the sections in which these concepts are first introduced.

Term	Section in which concept is introduced
Atom	2.2, 5.2
Attribute	3, T.1.2

Cardinality 5.2

Characteristic status 5.2

Concept 5.2

Child 2.1

Context 2.4

Fact 2.3, 5.2

Intrinsic characteristic 2.2

Object 3, T.1.2

Parent 2.1

Qualifier 2.1, 5.2

Template file 3

Uncertainty 2.5

Value 3, T1.2