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

NHS
Connecting for Health

THE CLINICAL TERMS VERSION 3
(THE READ CODES)

INTRODUCTION TO INFORMATION RETRIEVAL

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	175v1.0

Published by the UK Terminology Centre

Table of Contents

1. Executive Summary

This document is an overview of the methods that may be used to analyse clinical information stored in clinical information systems using Clinical Terms Version 3 (CTV3). It is aimed primarily at system designers, and assumes knowledge of the design of CTV3. The document acts as an introduction to a range of experiments performed by NHS Connecting for Health, Clarke-Ireland Associates and CAMS. These are seen as potential sources of information for system designers and highlight the advantages and disadvantages of some approaches to information retrieval, using a variety of different methods, database designs, and software and hardware platforms.

- Information retrieval with and without qualifiers can be efficiently performed in CTV3. The same conclusion was reached by the three groups involved in experimentation. The experiments covered searches for populations of patients (e.g. Find all patients with asthma) and searches questioning the record of a single patient (e.g. Does this patient have heart disease?).

- Information retrieval experiments were performed on a variety of databases of realistic size, both randomly generated and real containing between 100,000 and 1,200,000 events and up to 100,000 qualifiers.

- A variety of software and hardware platforms were used including ORACLE/UNIX/Sun, Microsoft Access/Windows/PC, Watcom/Windows/PC, Paradox/Windows/PC; and Interbase/Windows/PC.

- The experiments concentrated on using SQL, without using lower level code to ensure cross-platform compatibility.

- Four techniques for analysing hierarchies were examined in detail. Each had advantages in certain circumstances, depending on the software platform being used. Some methods may work better on one hardware/software platform than on another

- Close attention to the structure of queries was shown to improve performance significantly in the case of some database management systems.

- Different types of information system structure were tested. Flexibility can be obtained with a separate qualifier table, but faster performance with a fixed field structure although at the expense of flexibility. Hybrid forms may prove useful. Designs which transform a user's core concept-qualifier combinations into single codes were not explored.

- Though preliminary experiments on the efficiency of detection of equivalent forms (***Osteoarthritis*** qualified by ***Site: Hip*** is the same as the single term ***Osteoarthritis of the hip)*** have been successfully performed, the work needs to be extended to using the large database of qualifiers.

- Care needs to be taken when choosing a method for information retrieval. Each method has different requirements, and certain methods initially work particularly well on one hardware/software platform but more poorly on others. However close attention to the

structure of queries improves performance significantly.

- Care needs to be taken with the design of the clinical information database. Flexibility can be obtained with a separate qualifier table, but faster performance with a fixed field structure though at the expense of flexibility. Hybrid forms may prove useful. Designs which transform a user's core concept-qualifier combinations into single codes were not explored.

2. Analysing Information Stored in Clinical Information Systems

Knowledge about the clinical state of patients, of their disorders and operations, their signs, symptoms and investigations, and of their skills, abilities and social circumstances, can be stored in clinical information systems using Read Codes and terms. The chore of coding reaps benefits when the information contained in a patient's record can be re-used to inform a consultation, to create discharge or referral letters, to warn about potential drug interactions and contraindications when prescribing medication, to target 'at risk' groups of patients for interventions and investigations, to support clinical audit and to provide accurate information on healthcare activity to managers. Information retrieval describes the process of extracting information from a clinical information system to support such tasks as these.

This document is an overview of the methods that may be used to analyse clinical information stored in clinical information systems using CTV3. It is aimed primarily at system designers and assumes knowledge of the design of CTV3. The document acts as an introduction to a range of experiments performed by NHS Connecting for Health, Clarke-Ireland Associates and CAMS. These are seen as potential sources of information for system designers and highlight the advantages and disadvantages of some approaches to information retrieval, using a variety of different methods, database designs, and software and hardware platforms.

The body of work was performed to meet a need, expressed by system developers, both for advice as to the best ways to carry out the task of information retrieval, and as a demonstration that information could be retrieved from CTV3 systems in reasonable time.

A number of different experimental studies were carried out by several organisations:

- Preliminary scoping and analysis experiments -NHS Connecting for Health
- CTV3 analysis: hierarchy based searches - CAMS
- CTV3 analysis using a Query Language (SQL) - CAMS
- Read Coding information retrieval: analysis & reporting - Clarke-Ireland Associates, on behalf of NHS Connecting for Health

The main findings of these studies are summarised in this document.

3. Summary of Findings from Experiments

- Information retrieval with and without qualifiers can be efficiently performed in CTV3. The same conclusion was reached by the three groups involved in experimentation. The experiments covered searches for populations of patients (e.g. Find all patients with asthma) and searches questioning the record of a single patient (e.g. Does this patient have heart disease?).

- Information retrieval experiments were performed on a variety of databases of realistic size, both randomly generated and real containing between 100,000 and 1,200,000 events and up to 100,000 qualifiers.

- A variety of software and hardware platforms were used including ORACLE/UNIX/Sun, Microsoft Access/Windows/PC, Watcom/Windows/PC, Paradox/Windows/PC; and Interbase/Windows/PC.

- The experiments concentrated on using SQL, without using lower level code to ensure cross-platform compatibility.

- Four techniques for analysing hierarchies were examined in detail. Each had advantages in certain circumstances, depending on the software platform being used. Some methods may work better on one hardware/software platform than on another

- Close attention to the structure of queries was shown to improve performance significantly in the case of some database management systems.

- Different types of information system structure were tested. Flexibility can be obtained with a separate qualifier table, but faster performance with a fixed field structure although at the expense of flexibility. Hybrid forms may prove useful. Designs which transform a user's core concept-qualifier combinations into single codes were not explored.

- Though preliminary experiments on the detection of equivalent forms (***Osteoarthritis*** qualified by ***Site: Hip joint*** is the same as the single term ***Osteoarthritis of the hip***) have been successfully performed, the work needs to be extended to using the large database of qualifiers.

3.1 Overview of the Experiments

Preliminary Analysis Experiments - NHS Connecting for Health

- Compares Range code method and tree-walking method using Oracle (Sun/UNIX) and Watcom (PC/Windows) on a random generated patient database of 1,200,000 events.

- Explores hierarchy and single qualifier (but not atom) searches.

- Note that scoping experiments were carried out prior to these preliminary experiments. These were designed to explore the issues and used a smaller patient database.

CTV3 Analysis: hierarchy based searches - CAMS

- Compares ancestor table method with Version 2-like code-dependent hierarchy search., providing timings.

- Explores ISAM (Clipper and Access) and SQL (Access) on a patient database of 100,000 events (random generated with weighted distribution).

CTV3 Analysis using SQL (CAMS)

- Discusses how queries to retrieve events from a patient database can be formulated using SQL.

- Uses ancestor method with examples of queries of simple hierarchical search and of qualifier search.

- Examines how qualifier searches may be adapted to match on atoms to ensure all equivalent forms of describing a concept can be detected in a search.

- Provides some timings (200,000 events, small number of qualifiers) on a generated patient database in Access.

Read Coding Information Retrieval: Analysis & Reporting - Clarke Ireland Associates for NHS Connecting for Health

- Provides timings for hierarchy and qualifier searches (but not atoms) using Interbase, Paradox and Access on PC/Windows platform on a random generated patient database (1,200,000 events and 100,000 qualifiers) and a converted database from general practice (600,000 events).

- Examines both Ancestor table and Range numbering methods.

- Looks at two designs for a patient database, one maximising speed (fixed number of qualifier fields in the event table) the other maximising flexibility (adding a separate qualifier table to the event table).

4. The Influence of Data Collection and Storage on Retrieval

A hierarchy of bicycles will be used to illustrate the different methods of information retrieval. Some concepts (e.g. *Mountain bike*) may be qualified by colour. Some have a particular colour by definition (e.g. *Red mountain bike*). In this case *Colour* = *Red* is an atom of red mountain bikes and is marked (A)

in the table below.

HIERARCHY	COLOUR	SUSPENSION	MATERIAL
Bicycle			Steel, Aluminium
Mountain bike	Red, Green, Blue	None, Front, Full	Steel, Aluminium
Red mountain bike	Red (A)	None, Front, Full	Steel, Aluminium
Red mountain bike with no suspension	Red (A)	None (A)	Steel, Aluminium
Red mountain bike with front suspension	Red (A)	Front (A)	Steel, Aluminium
Red mountain bike with full suspension	Red (A)	Full (A)	Steel, Aluminium
Road bike	Red, Green		Steel, Aluminium
Hybrid bike			Steel, Aluminium
Red hybrid bike	Red (A)		Steel, Aluminium
Red bike	Red (A)		Steel, Aluminium
Red mountain bike	Red (A)	None, Front, Full	Steel, Aluminium
Red mountain bike with no suspension	Red (A)	None (A)	Steel, Aluminium
Red mountain bike with front suspension	Red (A)	Front (A)	Steel, Aluminium
Red mountain bike with full suspension	Red (A)	Full (A)	Steel, Aluminium
Red hybrid bike			Steel, Aluminium

In CTV3, any concept may appear in more than one place in the hierarchy. *Red mountain bike* appears as a child of *Mountain bike* and *Red bike*.

The task of information retrieval in medicine is to find patients in a clinical information system who have had events of particular diseases, surgical procedures, symptoms, etc. In the examples here the events are of bicycles being bought by customers. An event is partially described by an object (e.g. *Red mountain bike*) which may or may not be associated with qualifiers. Each qualifier consists of an attribute (e.g. *Colour*) and a value (e.g. *Red*). Other information, such as the date the event occurred, is also important but omitted

here as it is not essential to the discussion of how information is retrieved when stored using CTV3 codes.

Other than special classes of information (e.g. dates and numbers) all objects, attributes and values have Read codes. They are also all organised into a hierarchy.

Read thesaurus

Bikes

Mountain bike

Red mountain bike, etc….

Attributes

Colour

Material

Suspension

Colours

Red

Green

Forest green

British racing green

Blue

All methods must retrieve only those events and patients that are expected according to the rules of CTV3 (that is no more and no less). But being correct is not enough. Each method must also be efficient and many factors affect this. These will be divided into four groups, though making a choice from one group may affect the freedom of choice in other groups.

The influence of data collection on information retrieval. Users can describe the same concept in different ways, using some or no qualifiers.

The influence of information system structure on information retrieval. The tables storing events and qualifiers may be laid out in different ways

The influence of methods of storing qualifying information on retrieval. Data may be stored for analysis as it is collected, with a mixture of core concepts and qualifiers. Alternatively the data may be transformed before storage to a more qualifier oriented form, or a pure core concept form (without qualifiers) to make it easier to analyse.

The influence of methods of representing hierarchies on retrieval. Whatever the balance between core concepts and qualifiers, other data transformations can optimise retrieval of hierarchically arranged concepts in different ways. These include augmenting the hierarchy table so that it becomes an ancestor table and introducing depth-first-hierarchy-numbering or path codes.

5. The Influence of Data Collection on Information Retrieval

By introducing qualifiers, CTV3 allows greater detail to be expressed. However a clinician can often choose whether to represent an event using either just a single core term or a combination of a more general core concept with one or more qualifiers. Though providing this choice of core concept is desirable for clinical users, it introduces extra complexity during information retrieval.

The query may be for events of a particular class, but the information collected as children of that class. For example the query may be for all people who have bought a *Mountain bike*. This clearly includes all people who have bought *red mountain bikes with front suspension*. The same issue arises for qualifier values. For example, the query may be for all people who have bought a *Mountain bike* which is *Green*, and this would need to include those that are coloured *British racing green* or *Forest green*.

Some pieces of information may only be collected using core concepts. The distinction between *mountain bikes* and *road bikes* may only be made using core concepts and not through using qualifiers. In clinical medicine the core concepts include all disorders, procedures, history and observations, patient's skills and abilities, and investigations. A bonus file with the file release defines all of these.

Some pieces of information may only be collected using qualifiers. Information about the material that the bike is made of (*Steel, Aluminium*) may only be collected as qualifiers in the example.

Some pieces of information may be collected using or not using qualifiers. For example, David Baker's bike may be described in several different ways:

Red mountain bike with front suspension

Red mountain bike qualified by *Suspension = Front*

Mountain bike qualified *by Suspension = Front* and *Colour = Red*

To recover a list of all people with a '*Red mountain bike with front suspension'*, it is necessary to recognise and retrieve all of its alternative or equivalent forms.

6. The Influence of Information System Structure on Information Retrieval

Different information system designs will suit different requirements. Some are examined here, continuing with the bicycle analogy. Information about the bikes bought by customers may be stored in an events table.

EventID	Customer	Object
0001	David Baker	Red mountain bike with front suspension
0002	Steve Peat	Red mountain bike with full suspension

More complex database designs allow for qualifying information to be stored. Extra fields may be introduced into the events table. This will be referred to as the **fixed field design** for qualifiers.

EventID	Customer	Object	Colour	Suspension	Material
0001	Tim Gould	Mountain bike	Blue	Front	
0002	Paul Lazenby	Red mountain bike		Full	Aluminium

Note that, as will be discussed later, the columns for colour, suspension and material may contain just extra qualifier information (as in the example above) or information intrinsic to the object as well (e.g. Paul Lazenby's bike may have been described as Colour = Red).

Alternatively events and qualifiers may be stored in two separate tables.

EventID	Customer	Object
0001	David Baker	Red mountain bike with front suspension
0002	Steve Peat	Red mountain bike with full suspension

EventID	Attribute	Value
0001	Material	Aluminium

Storing qualifiers in the events table has the advantage that a query needs only look at the one table to find Read coded information. These designs are optimised for speed of retrieval. Storing qualifiers in a separate table allows for greater flexibility. As new qualifiers are released by the UK Terminology Centre, there is no need to update the field structure of the patient database tables. Designers will want to consider hybrid models in which commonly searched qualifiers are placed in the events table, but all others are placed in a separate qualifiers table.

The general solution for dealing with nested qualifiers is to introduce qualifier identifiers in place of the event identifier in the qualifier table. Nested qualifiers occur when a qualifier (e.g. *Aluminium*) can itself be qualified (it may be *Polished* or *Painted*).

EventID	Customer	Object	Qualifier pointer
0001	David Baker	Red mountain bike with front suspension	0001
0002	Steve Peat	Red mountain bike with full suspension	null

QualifierID	Attribute	Value	Qualifier pointer
0001	Material	Aluminium	0002
0002	Finish	Polished	null

The qualifier pointers (which point to qualifier identifiers) ensure that each qualifier is unambiguously associated with its values.

Nested qualifiers can be stored in a fixed table design, but only by denormalising the information. When each nested attribute (e.g. Finish) has only a single value (e.g. Polished), the representation is efficient, although adding new qualifiers at a later date requires the table structure to be changed by adding new columns.

EventID	Customer	Object	Material	Finish
0001	David Baker	Red mountain bike with front suspension	Aluminium	Polished
0002	Steve Peat	Red mountain bike with full suspension	null	null

If Aluminium frames could be both polished and painted, and David Baker's bike fell into this category, some information (e.g. David Baker, Red mountain-bike with front suspension, Aluminium) would need to be duplicated in the event table.

EventID	Customer	Object	Material	Finish
0001	David Baker	Red mountain bike with front suspension	Aluminium	Polished

0001	David Baker	Red mountain bike with front suspension	Aluminium	Painted
0002	Steve Peat	Red mountain bike with full suspension	null	null

Finally the need for storing qualifiers can be removed altogether if qualifier information collected by the user is compiled into more complex objects (also called core concepts), which are given new (local) Read Codes. This option is discussed in the section 'Information system designs which standardise format of information'.

6.1 Experiments on Clinical Information System Structure

The following studies used different database structures.

- Preliminary scoping and analysis experiments -NHS Connecting for Health
- CTV3 analysis: hierarchy based searches - CAMS
- CTV3 analysis using a Query Language (SQL) - CAMS
- Read Coding information retrieval: analysis & reporting - Clarke-Ireland Associates

The first study concentrated on searching objects (core concepts) in the event table. The second study introduced the search of core concepts and single qualifiers using a separate qualifier table. The third study explored a separate qualifier table in more detail. The final study compared timings of searches on an event table only, event and qualifier tables and a fixed-field qualifier containing table.

7. The Influence of Methods of Storing Qualifying Information on Retrieval

System designers have an alternative to retrieving information in different formats. They may choose to use a standard format for storing information. Information may need to be transformed from the format in which it was collected into the standard stored format.

7.1 Transforming Information to Qualifier Form

Designers may adopt a strategy of storing information only in qualifiers (where this is possible), simplifying the core concept that the user selected. For

example a user may enter information about David Baker's bike by choosing the core concept '*Red mountain bike with front suspension*', but the information might be represented in a system in the following way.

EventID	Customer	Object	Colour	Suspension	Material
0001	David Baker	Mountain-bike	Red	Front	null

Note that the distinct qualifier table could have been used in place of the fixed field format.

7.2 Transforming Information to Core Concept Form

At the opposite extreme, instead of transforming the information so that as much information is removed out to qualifiers as possible, the designer may seek to remove qualifiers altogether. For example a user might select *Hybrid bike* qualified by *Colour = Red*. This is equivalent to the concept *Red hybrid bike* which is stored in the information system.

EventID	Customer	Object
0001	Joe Towny	Red hybrid bike

Often though, CTV3 will not contain a core concept corresponding to the user's selections of core concept and qualifiers (e.g. if the user had selected *Red mountain bike with front suspension*' and qualified this by *Material = Steel*). A local code must be generated and the Read hierarchy extended by making the new core concept '*Red steel mountain bike with front suspension'* as a child of *'Red mountain bike with front suspension*'. This is the approach taken by automatic classification systems.

Note that there is no combinatorial explosion as only those concepts which the user requires to be stored in the information system are transformed and classified.

7.3 Disadvantages of Transforming Qualifier Information

Transforming information in these ways should speed up information retrieval, as it becomes unnecessary to search through both the atoms of a concept and its qualifiers to detect equivalent concepts. But the two difficulties discussed next must be considered.

Firstly the atoms (e.g. *Colour = Red*) of a concept (e.g. *Red mountain bike*) might change over time, in which case data in the event tables may need to be updated to reflect these changes. Change to atoms would occur if an error was found in the Read template file or if a new way of representing the meaning of concepts in terms of atoms was formulated. The latter should happen rarely and no change would be made without consultation with users and developers, and a window of opportunity for developers to react.

Secondly if a complex concept is constructed and given a local Read Code, it may have to be placed in several different places in the hierarchy, otherwise it would not be found in all searches. If a customer bought a '*Red road bike'*, then once this was given a local Read Code, it would need to be placed in the hierarchy both under '*Red bike'* and under '*Road bike'*. In other cases, a new explicit concept might have to be inserted into its correct place in the hierarchy, between two existing concepts. Lastly, if the atoms (semantic definition) of a concept change, then the new concept would need to be reclassified in the hierarchy. The process of finding the extra/new places in which a concept should be placed in a hierarchy may be managed automatically by referring to the atoms of a concept, but it is not a trivial task to implement it efficiently. This is the research area of automatic classification, and found in such work as the GALEN project.

7.4 Maintaining Qualifier Information in the Form Entered by a User

This may be the preferred method for many suppliers as it conserves the way that information was entered and analyses information in just this format. However, it is important to deal with the problem of equivalence i.e. that two different Read Code expressions found in the computerised record may mean the same thing. For example:

Red mountain bike with front suspension

Red mountain bike qualified by Suspension = Front

To pick up both forms in a query, the search must include both atoms (found in the template table supplied as part of the Read Code release) and qualifiers (using the event and qualifier tables). An introduction to queries for this can be found in the document 'Version 3 analysis using SQL' from CAMS.

One crucial issue is how the question can be efficiently executed. Is it necessary to do three searches corresponding to the three ways that the purchase of a red mountain bike with front suspension could have been stored? Though this will produce a correct answer, many hierarchy nodes will be revisited - the search will therefore be inefficient. It will also be necessary for the user to find these three forms, which will require some expertise. However there is a remedy.

The search can be made more efficient by climbing up the hierarchy from the query node (e.g. *Red mountain bike with front suspension)* to the highest node(s) which when correctly qualified is precisely the same as the query node. This is the node *Mountain bike*. The query can therefore be executed as a query for all events of purchasing a *Mountain bike* qualified by *Suspension = Front* and *Colour = Red*.

HIERARCHY	COLOUR	SUSPENSION	MATERIAL
Bicycle			Steel, Aluminium
Mountain bike	Red, Green, Blue	None, Front, Full	Steel, Aluminium
Red mountain bike	Red (A)	None, Front, Full	Steel, Aluminium
Red mountain bike with no suspension	Red (A)	None (A)	Steel, Aluminium
Red mountain bike with front suspension	Red (A)	Front (A)	Steel, Aluminium
Red mountain bike with full suspension	Red (A)	Full (A)	Steel, Aluminium
Road bike	Red, Green		Steel, Aluminium
Hybrid bike			Steel, Aluminium
Red hybrid bike	Red (A)		Steel, Aluminium
Red bike	Red (A)		Steel, Aluminium
Red mountain bike	Red (A)	None, Front, Full	Steel, Aluminium
Red mountain bike with no suspension	Red (A)	None (A)	Steel, Aluminium
Red mountain bike with front suspension	Red (A)	Front (A)	Steel, Aluminium
Red mountain bike with full suspension	Red (A)	Full (A)	Steel, Aluminium
Red hybrid bike			Steel, Aluminium

Note that when climbing up the hierarchy from *Red mountain bike with front suspension* we also reach the parent *Red bike*. But *Red bike* cannot be qualified in such a way that it is equivalent to *Red mountain bike*. This search node can therefore safely be ignored. How though can a search routine know that *Mountain bike* can be qualified to be equal to *Red mountain bike with front suspension* whereas *Red bike* cannot?

NHS Connecting for Health has proposed to release a version of the hierarchy table with an extra column containing an equivalence flag (previously referred to as the completeness flag) which indicates whether a parent can be equivalent to the child if suitably qualified. The flags for the example hierarchy would be set as follows, where E = potentially Equivalent and N = canNot be equivalent:

Parent	Child	Equivalence Flag
Bicycle	Mountain bike	N
Bicycle	Road bike	N
Bicycle	Hybrid bike	N
Mountain bike	Red mountain bike	E
Red mountain bike	Red mountain bike with no suspension	E
Red mountain bike	Red mountain bike with front suspension	E
Red mountain bike	Red mountain bike with full suspension	E
Hybrid bike	Red hybrid bike	N
Red bike	Red hybrid bike	N
Red bike	Red mountain bike	N

The algorithm for generating the query is therefore to check the parent(s) of the concept being queried and if these can be equivalent to the query node to substitute the parent for the child in the query, adding the atoms of the child node that are different to those of the parent into the query. The parent(s)' parents are then examined and so on until an N flag is reached. The end-result is that the query is gradually refined as follows:

Red mountain bike with front suspension

Red mountain bike qualified by *Suspension = Front suspension*

Mountain bike qualified by *Colour = Red* and *Suspension = Front suspension*

The final result is an efficient and correct search that is automatically generated from the user's initial specification and which also saves the user from the task of searching out alternative ways in which the information may have been stored. The same technique can be used for transforming information into qualifier-form and core-concept form (when it is used in reverse).

Until The UK Terminology Centre releases the new version of the hierarchy table populated with the equivalence flag, users will need to do the job of specifying a search at the highest point in the hierarchy at which the search is valid (i.e. this part cannot be done automatically). In the example given the user should specify the search to start from *Mountain Bike*. If qualifiers are specified in the search, users must also make sure that these are specified from the highest point in their hierarchy which is valid.

7.5 Experiments on Standardising the Format of Information

The experiments performed so far have examined the efficiency of both qualifier and core concept-only structures. However, no experiments have looked at the process of transforming a user's clinical information into these structures.

8. The Influence of Methods of Representing Hierarchies on Retrieval

Four methods for representing and searching hierarchies are presented below. These are tree-walking over the hierarchy file, path codes, range numbering and the ancestor table. The techniques may be used for searching either for the children of core concepts or the children of attributes and values. Therefore in a query containing several qualifiers (e.g. retrieve all purchases of *Mountain-bikes* where *Material = Aluminium* and *Colour = Green*) the chosen technique would be used several times to recover all the children of *Mountain-bike, Aluminium* and *Green* (including *British racing green*).

8.1 Tree-walking Method

This method uses the hierarchy file distributed as part of the release files for analysis without requiring modification.

Parent	**Child**
Bicycle	Mountain bike
Bicycle	Road bike
Bicycle	Hybrid bike
Mountain bike	Red mountain bike
Red mountain bike	Red mountain bike with no suspension
Red mountain bike	Red mountain bike with front suspension
Red mountain bike	Red mountain bike with full suspension
Hybrid bike	Red hybrid bike
Red bike	Red hybrid bike
Red bike	Red mountain bike

A query collects the children of the query node, then those children's children and so on to the leaves of the hierarchy. Any patient (customer) with an event (bicycle) in the collected set of children is included. The general form of the query in ORACLE SQL, which unusually allows a connect-by statement that recurses across child links, is as follows:

```
SELECT eventid FROM events
 WHERE object IN(
        SELECT DISTINCT Child
        FROM Hierarchy
        CONNECT BY PRIOR Child = Parent
        START WITH Child = <QueryReadCode>)
```

Advantages and Disadvantages of the Tree-walking Method

Tree-walking requires minimal resource over and above the Read Code hierarchy file, which is a compact way of storing a hierarchy. It operates directly on the Read Code in the events and qualifiers tables, requiring no extra indexing or annotation.

The lack of optimisation means the method performs inefficiently in many environments, although its performance in Oracle was excellent.

Experiments on the Tree-walking Method

This method was explored in the preliminary analysis experiments by NHS Connecting for Health.

8.2 Path Code Method

Traditional classification and coding schemes such as ICD-9, ICD-10 and Read 4-byte and Version 2, are designed in such a way that a concept's place in the hierarchy can be represented by a path code. For example:

HIERARCHY

Bicycle (…..)

Mountain bike (A….)

Red mountain bike (AA…)

Red mountain bike with no suspension (AAA..)

Red mountain bike with front suspension (AAB..)

Red mountain bike with full suspension (AAC..)

Road bike (B….)

Hybrid bike (C….)

Red hybrid bike (CA…)

Red bike (D….)

This works particularly efficiently when the hierarchy is of fixed depth (traditionally 4 or 5 levels), and the maximum number of children at any level must be limited to the number of characters used in the scheme (e.g. A-Z, a-z and 0-9). The scheme may be adapted to the hierarchy structure used in CTV3, which is a directed acyclic graph that allows multiple children and any number of levels in the hierarchy. To deal with multiple children, concepts (e.g. *Red mountain bike*) may need to have multiple path codes associated with them (e.g. AA.. and DA…) as in the example below. The children, grandchildren etc. of a concept can be automatically inferred from the concepts path codes.

HIERARCHY

Bicycle (…..)

Mountain bike (A….)

Red mountain bike (AA…) (DA…)

Red mountain bike with no suspension (AAA..) (DAA..)

Red mountain bike with front suspension (AAB..) (DAB..)

Red mountain bike with full suspension (AAC..) (DAC..)

Road bike (B….)

Hybrid bike (C….)

Red hybrid bike (CA…) (DB…)

Red bike (D….)

Red mountain bike (DA…) (AA…)

Red mountain bike with no suspension (DAA..) (AAA..)

Red mountain bike with front suspension (DAB..) (AAB..)

Red mountain bike with full suspension

(DAC..) (AAC..)

Red hybrid bike (DB…) (CA…)

Where the range of Read Codes in the path to be searched is between <StartQueryReadCode> and <EndQueryReadCode> then the basic pattern of the SQL code to deal with a single path search is as follows:

```
SELECT * FROM Events

    WHERE Events.Object >= '<StartQueryReadCode>'

        AND Events.Object <= '<EndQueryReadCode>'
```

In order to deal with the problem that each Read Code may need to be indexed on several path codes, more changes must be made to the database structure. These are not discussed here.

To increase the number of children of a concept that a path code can represent, the character set used in the path code can be augmented to include other printing and non-printing characters. To deal with more levels (the CTV3 hierarchy can have any number of levels, the current maximum found being 15 levels deep), longer path codes can be used.

Advantages and Disadvantages of the Path Code Method

For greatest efficiency, each concept in the event and qualifier database needs to be indexed on each of the path codes. This incurs a space penalty.

As the hierarchy may change at each release, the path codes need to be recalculated. This is a significant disadvantage because the path codes stored in each patient record, upon which these are indexed, therefore need to be rewritten.

As with any method that involves rewriting elements of the patient record, it is necessary to ensure that the original Read Codes and terms chosen by a clinician cannot be altered in error.

Experiments Performed using the Path Code Technique

Basic experiments suggest that reasonable performance can be obtained using this method, but the method has not been explored in detail. A simple exploration of the path code technique (without dealing with the problem of structuring the database to cope with the multiple path codes) is published in the document below, mainly as a comparison for Ancestor table search.

Version 3 analysis - hierarchy based searches (CAMS)

8.3 Range Number Method

The technique of range numbering involves numbering each concept in the hierarchy from the top down - these are the bracketed numbers in the example shown below.

The end result of applying the method is that each Read concept is given a hierarchy number. In the example, the number is found in brackets e.g. *Mountain bike* has the hierarchy number (2). Then the children of that concept can be represented as one or more ranges of hierarchy numbers (e.g. 2-7 for *Mountain bike* in the example).

Each concept is only given one number, no matter how many places it occurs in the hierarchy. In the example, *Red mountain bike* has the number (3). But where it occurs in several places in the hierarchy, its parents may be associated with several ranges (e.g. the ranges 3-6 and 9-10 against *Red bike*) as the natural sequencing of the numbering is broken.

HIERARCHY

Bicycle (1), 1-10

Mountain bike (2), 2-6

Red mountain bike (3), 3-6

Red mountain bike with no suspension (4), 4-4

Red mountain bike with front suspension (5), 5-5

Red mountain bike with full suspension (6), 6-6

Road bike (7), 7-7

Hybrid bike (8), 8-9

Red hybrid bike (9), 9-9

Red bike (10), 3-6 & 9-10

Red mountain bike (3), 3-6

Red mountain bike with no suspension (4), 4-4

Red mountain bike with front suspension (5), 5-5

Red mountain bike with full suspension (6), 6-6

Red hybrid bike (9), 9-9

If each event and qualifier Read Code is indexed with its hierarchy number, and a mapping from each Read Code onto the hierarchy ranges that correspond to its children is held in a resource table, then searching for children can be achieved efficiently. For example:

TABLE Idrange

Id	Idlow	idhigh
1	1	10
2	2	6
3	3	6
4	4	4
5	5	5
6	6	6
7	7	7
8	8	9
9	9	9
10	3	6
10	9	10

The basic format of an SQL query for events using the above table is as below. It is assumed that the events table contains a field (objectID) for the range number of the object.

```
SELECT eventid
FROM event, idrange
WHERE idrange.id = <QueryEventID>
        AND event.objectID BETWEEN idrange.idlow AND
idrange.idhigh
```

Advantages and Disadvantages of the Range Number Method

Hierarchy numbering proves the most efficient method of numbering Read concepts for information retrieval in some hardware/software scenarios.

Hierarchy numbers have some useful properties. Multiple ranges tend to coalesce together at higher levels (e.g. *Red bike* where 9-9 and 10-10 coalesce to 9-10). The ranges against the top node always coalesce to 1-N, where N is the number of concepts in the system.

Range numbering can be used throughout the Read hierarchy for objects (core concepts) and attributes and values of qualifiers.

The way that the hierarchy is numbered affects the efficiency of information retrieval. For example Disorders and History & observations overlap in some places. In the reported experiments, History & observations were numbered first, leaving Disorders fragmented (approximately 200 ranges were found against the concept Disorders). In reality it would be sensible to number

Disorders first because these are often aggregated in queries (e.g. retrieve all patients with chronic cardiovascular and respiratory conditions for influenza immunisation). In contrast, History & observations are typically searched for locally (e.g. retrieve all patients with a Peak flow < Expected peak flow). However leaving the numbering of Disorders fragmented proved a useful index of maximal inefficiency for the purposes of these experiments and a contrast with the efficient numbering of Procedures.

The major disadvantage of range numbering is that, as the hierarchy table changes at each release, the hierarchy number of many concepts will change from one release to the next. As the hierarchy number must be stored in the patient record for maximum advantage, this means changing this number in records at each release. Clinical information systems with large numbers of patient records will take time to update. The more frequent drug updates might have their own range of hierarchy numbers to limit necessary updates.

An extension of range numbering which removes the need to update range numbers with each release has been proposed. The idea here is that, when creating hierarchy numbers gaps may be left (the increment could be as much as 1,000) between numbers and any overflow beyond this could be managed by adding new concepts as new numbers outside the original ranges. The end result is that the number of ranges associated with each concept tends to increase, so gradually reducing the efficiency of the process. However at any (opportune) time the original efficiency can be recovered through a re-index using new hierarchy numbers.

As hierarchy numbers may change over time, these should be stored in the patient record in such a way that there is no danger of the rest of the clinical record being rewritten -the Read Code, term identifier, term rubric and free text should not be updated. This is clearly necessary if the clinical record is to act as a useful repository of information from clinical and medico-legal points of view.

Experiments Performed using the Range Number Technique

Range numbering was explored in more detail in the studies:

- Preliminary analysis experiments - NHS Connecting for Health
- Read Code information retrieval analysis & reporting - Clarke-Ireland Associates

In particular, a document describing the latter study extends exploration of the basic idea to analysis of qualifiers, both using separate qualifiers table and a fixed field database design. The efficiency of the extension to the technique of not requiring updating of hierarchy numbers is explored.

8.4 Ancestor Method

The ancestor table method augments the hierarchy table to map all concepts onto their children's' children (recursively) as well as mapping each concept onto itself. The resulting table for the example scenario is shown below.

ANCESTOR	CHILD
Bicycle	Bicycle
Bicycle	Mountain bike
Bicycle	Red mountain bike
Bicycle	Red mountain bike with no suspension
Bicycle	Red mountain bike with front suspension
Bicycle	Red mountain bike with full suspension
Bicycle	Road bike
Bicycle	Hybrid bike
Bicycle	Red hybrid bike
Bicycle	Red bike
Mountain bike	Mountain bike
Mountain bike	Red mountain bike
Mountain bike	Red mountain bike with no suspension
Mountain bike	Red mountain bike with front suspension
Mountain bike	Red mountain bike with full suspension
Red mountain bike	Red mountain bike
Red mountain bike	Red mountain bike with no suspension
Red mountain bike	Red mountain bike with front suspension
Red mountain bike	Red mountain bike with full suspension
Red mountain bike with no suspension	Red mountain bike with no suspension
Red mountain bike with front suspension	Red mountain bike with front suspension
Red mountain bike with full suspension	Red mountain bike with full suspension
Road bike	Road bike
Hybrid bike	Hybrid bike
Hybrid bike	Red hybrid bike
Red hybrid bike	Red hybrid bike
Red bike	Red bike
Red bike	Red hybrid bike

A query for all events involving a mountain bike being bought therefore requires little more than a join on the ancestor and events tables.

```
SELECT COUNT(Event.EventID)
FROM Ancestor, Event
WHERE Ancestor.Ancestor = <QueryReadCode>
        AND Ancestor.Child = Event.Object
```

The SQL can be extended to cope with multiple qualifiers, nested qualifiers

and detecting equivalent forms of the same concept (see experiments section).

Advantages and Disadvantages of the Ancestor Table Technique

This technique analyses Read Codes directly, and does not require any intermediate number or code to be stored in the patient database. No updates are required to the content of the patient database when new Read Code releases are installed.

The technique proves the fastest on some hardware/software platforms, though it can be slow on others. Optimisations of the SQL code can significantly improve performance on these platforms though (see Clarke-Ireland Associates report).

The size of the ancestor table will increase significantly as the average number of parents per concept increases, partly because the method does not have the advantage of coalescence that the range number technique has. Expansion of the number of multiple parents in the hierarchy table is planned to happen gradually in CTV3 as the hierarchy is made more complete.

To offset this space disadvantage, each patient record only needs to be indexed on the Read Codes contained in it. Some other methods (e.g. path code) need multiple index entries per Read Code.

Experiments Performed using the Ancestor Table Technique

Ancestor table timings are explored in:

1. Version 3 analysis - hierarchy-based structures from CAMS

2. Version 3 analysis using a Query Language (SQL) from CAMS

3. Read Coding information retrieval (analysis & reporting) from Clarke-Ireland Associates.

The third paper examines the application of the technique to dealing with multiple qualifiers in both fixed field databases and separate event and qualifier tables, while the second paper examines the adaptations required for dealing with nested qualifiers and for detecting equivalent forms, but only in a patient database structure with a separate qualifier table.