Structured Methods of Information Management for Medical Records

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Medicine

1993

William Anthony Nowlan

Table of Contents

List of Figures

Abstract

This thesis argues the need for, and presents the theory, design, and implementation of a formalism for the representation of medical concepts, that is a basis for recording detailed, structured medical records in computer–based systems (the Structured Meta Knowledge formalism – SMK). The motivations for this work are found in the PEN&PAD programme that is developing advanced clinical information systems for direct patient care.

Current techniques for the representation of medical concepts are based on coding and classification schemes. An analysis of these schemes shows them to be inadequate for this purpose. Classification schemes are enumerative representations of medical concepts and the relationships between those concepts. Increasing the scope of a scheme results in a 'combinatorial explosions' of terms, and the task of enumerating the relationships between those terms becomes unmanageable. The compositional scheme SNOMED overcomes some of these problems but lacks any formal semantics. An alternative approach is proposed.

SMK is a formalism for the representation of medical concepts that is:

- recursively compositional;
- constrained in which compositions are considered well–formed;
- generative with the representation of most concepts being implied by the model but not explicitly enumerated;
- capable of representing parsimonious models of terminology.

SMK is restricted to the representation of terminological knowledge with more general assertional knowledge specifically excluded. However SMK extends the usual definition of terminological knowledge to include limited forms of assertion covering i) the creation of elementary entities, ii) the assertion of subsumption, and iii) statements about terminology. These statements about terminology represent what it is 'sensible to say' in the domain and are the basis for constraining the representation to 'sensible medical concepts'. For example the knowledge that 'fractures occur in bones' means that 'fracture of the humerus' is sensible and 'fracture of the eyebrow' is not.

Theories are described for the well–formedness of a composition and the relationships between compositions, in particular that of subsumption.

SMK is implemented in a terminology engine and associated text–based tools. Problems remain with the current implementation of SMK, notably in the classifier. However the terminology engine forms the basis of working prototype clinical systems, and is in use by other workers as part of the GALEN Project.

The use of SMK for modelling medical terminology is demonstrated and its relationship to traditional classification schemes explored. Much of the utility of SMK derives from its generativity and ability to derive formal subsumption hierarchies based on the definitions of entities. There are recognised limitations imposed by the formalism and its definition may require extending. However it is proposed that SMK represents a significant improvement on the technique of enumerative classification.

An extension to SMK allows for the representation of the medical records of individual patients. There are two levels of instantiation in SMK i) from categories to individuals and ii) from individuals to occurrences. Categories are similar to classes in other representations. Individuals represent concrete things such as people and places. Occurrences represent observations of those individuals by an observer at a particular time and place. A medical record is thus represented as a network of occurrences.

A wider proof of concept requires the construction of large models of medical terminology with general utility, and the development of methodologies for co-operative work on those models. These tasks are within the remit of the GALEN Project.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of The University of Manchester or any other university or other institute of learning

Acknowledgements

I am greatly indebted to the members of the Medical Informatics Group, University of Manchester, both present and past. In particular I wish to thank Alan Rector, Bernard Horan, and Peter Crowther.

The work embodied in this thesis was supported by The Medical Research Council grant number SPG 8800091 and the UK Department of Health.

The Author

Anthony Nowlan obtained a B.A. in physics from Oxford in 1978. Following this he trained in medicine at University College London, qualifying in 1983. He subsequently worked in general and specialist hospital medicine obtaining Membership of The Royal College of Physicians in 1986. This was followed by a period as lecturer in epidemiology prior to joining the Medical Informatics Group, University of Manchester, as Clinical Research Fellow in 1988. Recently he has been appointed Senior Lecturer in Epidemiology and Medical Informatics in the Faculty of Medicine, University of Manchester.

His research interests are in the representation of medical terminologies, medical records, and the user–centred design of clinical information systems. He has published in all these areas.

Chapter 1 Purpose and Introduction

1.1 The purpose of this thesis

This purpose of thesis is to argue the need for, and present the theory, design, and implementation of a formalism for the representation of medical concepts, that is a basis for recording detailed, structured medical records in computer–based systems. The aims of the thesis are to:

- explain why present techniques are inadequate for the representation of complex systems of medical concepts;
- identify the requirements on a formalism that is a more adequate basis for the representation of medical concepts, and present the theories of that formalism;
- demonstrate that the formalism is capable of implementation and has useful properties for modelling medical terminology;
- present an appraisal of the current state of the formalism, its use, and its deficiencies;
- outline a set of related tasks that are part of the broader proof of concept for the formalism as a basis for the representation of medical terminologies and medical records.

This chapter begins by explaining the motivations behind the work embodied in this thesis. An example of medical language is then used to characterise why there is a need for formal representations. The chapter concludes by presenting the organisation of the thesis.

1.2 Motivations for the work

The work embodied in this thesis formed part of a larger programme of research on clinical information systems. This section outlines the aims of that programme and discusses some of the general characteristics of clinical systems.

1.2.1 Research on clinical information systems: PEN&PAD

The motivations for this work are to be found in the wider context of the PEN&PAD[1] programme of research [Nowlan 1990, Nowlan 1991b]. This programme aims to research, design, and develop prototype clinical information systems for use by clinicians in day to day patient care. The belief behind PEN&PAD is that clinical systems, integrated into clinical care, are the key to the successful and cost–effective use of information technology in health care. To achieve this integration clinical systems should:

- provide a sophisticated 'intelligent' user–interface that clinicians find useful and usable for patient care, covering both data entry and information presentation;
- support a comprehensive, detailed, and highly structured medical record;
- adapt to different models of clinical care and styles of practice within a single coherent framework.

[1] PEN–Practitioners Entering Notes, PAD–Practitioners Accessing Data

The formalism and its implementation described in this thesis form the foundations of the PEN&PAD clinical system and medical record. This requirement has been the single most important influence on the development of the formalism.

1.2.2 Characteristics of clinical information systems

Clinical systems are characterised by the need to represent detailed descriptions of individual patients. They are qualitatively different from more general information systems used for health care administration or epidemiological studies on populations of patients. A clinical system must cope with the scale, detail, and complexity of information required for clinical medicine. However at the same time a system needs to be simple and intuitive to use on a routine basis across a range of clinical settings. These requirements conflict and present a dilemma to developers of advanced systems for clinical care.

The premise within PEN&PAD is that this dilemma can only be addressed if in some way an information system's behaviour is guided by the *meaning* of the information it is manipulating. It is this relationship between behaviour and meaning which characterises the 'intelligent' interface. The system must concur with the clinician's perceptions of the clinical context. For example, a patient with leukaemia evokes in a clinician a different set of responses than will a patient with a sore throat. However such expectations are frequently and inappropriately transferred to computer systems. This is exemplified by the following three mutually contradictory statements that arose during discussions, with clinicians, of requirements on clinical systems:

> "obviously the patient having leukaemia is a big problem, so I want to know all about it"
>
> "it shouldn't trouble me with all the silly sore throats"
>
> "it should flag if they have had a lot of nasty sore throats recently because that could indicate an underlying problem such as an immune disorder or leukaemia"

Medical observations can be 'obvious', 'silly' or form a pattern to an experienced clinician, but they are nothing of the sort to a computer system. Most computer–based information systems can barely recognise that 'sore throat' and 'leukaemia' are different medical ideas, let alone have any interpretation of *why* they are different and what the consequences are. As obvious as it may be, computer systems are formal systems, without the benefits or otherwise of years of medical training, experience, and clinical intuition. Computer systems require formal representations of the medical concepts. This is the basic premise behind the work in this thesis.

1.3 What is to be represented: the shared medical model

How can we characterise what it is that needs to be represented? Medicine has a highly developed, structured, and widely shared system of understanding, that is derived from medical science, embodied in clinical practice, and reflected in medical language. This does not mean that medicine is an entirely rational and well understood discipline. This is far from the truth. Many medical processes are poorly understood, and the practice of medicine is socially and organisationally complex in ways that are not easy to formalise. Furthermore there are systems of medicine based upon theories of health and sickness quite different to those derived from the medical and biological sciences. Nevertheless a strong shared model of health and sickness provides an important framework for understanding practical problems and determining actions. An example will illustrate this

1.3.1 An example: thinking about 'stroke'

Consider the medical problem of 'stroke'. This is a clinical syndrome characterised by a rapid onset of impaired central nervous system function. It commonly involves an alteration of consciousness, loss of use of limbs or other parts of the body, impaired sensation, and high level cognitive dysfunction. The patient is struck down. Stroke, like other such common terms, covers a constellation of findings and underlying disorders involving a range of causes and processes. Two such disorders in the case of stroke are acute cerebral thrombosis and acute cerebral haemorrhage. However these descriptions are intended to be much more than labels for disorders. Their structure is precise and stimulate in a clinician a rich mix of ideas and implications.

Acute means that the onset of the condition is rapid, which will characterise both the tempo of the clinical manifestations and the details of the pathological findings. Cerebral indicates the affected organ is the brain. This is not just spatial information. It indicates that the critical functions of the central nervous system are likely to be damaged, and this suggests the type of clinical manifestations to be expected, and the assessments required. Finally there comes the pathological process. In one case this is thrombosis, the formation of a blockage within a blood vessel. This results in ischaemia which is an inadequate blood supply with consequent deprivation of oxygen and nutrients to part of the brain tissue, and the subsequent destruction of that tissue. In the second case the process is haemorrhage, the escape of blood from blood vessels into the surrounding brain tissue causing amongst other things pressure on that tissue and its eventual destruction. In both cases the result is the death of the affected part of the brain. However which of these has occurred may be important, particularly when considering the management of an individual patient. For example thinning the blood may be of use in some cases of a blood clot, but is inadvisable if the cause is bleeding. Likewise a more precise specification of the affected part of the brain indicates the type of impairment to be expected, and to some extent the prognosis for recovery. These descriptions are analysable and stimulate a complex set of ideas and relationships in the mind of the clinician.

This example is not intended to imply that the form of the description is a sufficient model of the clinician's perceptions, or the basis for direct advice giving on patient management and prognosis. The purpose is much more modest. If information systems are to aid clinical workers by allowing them to record, organise, and communicate their thoughts then it is necessary for those systems to represent some of the structure which helps shape that clinical thinking. What is required is a representation of what the medical concepts mean to clinicians. This is built upon an assumption of a shared medical model.

1.3.2 Medical limits on what can be modelled

It is important to realise that the challenge of representing medical concepts is not just a technical one. A formal representation can only be produced if there is a underlying medical framework. Some areas of medicine are well described by widely adopted systematic frameworks. For example in the field of tumour pathology, tumours can be benign or malignant, be composed of a variety of cell tissue types, and have a range of appearances. Important differences of opinion exist over the precise detail of such a framework, and those opinions will evolve as research proceeds. Such diversity is the life–blood of medical research. However there is general agreement that an analytical framework can exist. In contrast it is rather more difficult to develop a structured framework for psychosocial problems in general practice, covering marital discord, poor housing, school refusal, stress at work, and a poor relationship with an elderly dementing parent.

No representational technique can compensate for the lack of an underlying medical model. However it will be argued that the techniques currently in widespread use fail to capture those models that do exist.

1.4 Scope of the work in this thesis: the limits on context and modelling

Medicine is a large subject. It is all too easy for the task of representing part of medicine to spill over into that of representing all of medicine and biomedical science for all possible purposes. The work described in this thesis tries to avoid this problem by limiting its scope in two respects.

The first limitation is on the context of the work. The principle concern is with clinical discourse, the medical record, and 'what can be said'. The work is not trying to directly tackle physiological modelling, diagnostic reasoning, or therapeutic advice giving. These tasks are important but they are not the main focus of this thesis.

The second limitation is on the product of the work. The result of the work is an implemented formalism and not a large scale model of medical concepts. That is not to say the formalism was developed as a purely theoretical exercise. Quite the contrary. The prime requirement was to support PEN&PAD prototype clinical workstations that could be evaluated by clinicians in realistic test conditions. Thus several large models were developed to meet that requirement, and examples from those models will be discussed in this thesis. However no claim is made for the medical validity or more general utility of those particular models.

1.5 Organisation of the thesis

Chapter 2 explains the use of medical coding and classification schemes for representing medical information and analyses their inadequacies.

Chapter 3 discusses techniques from the field of knowledge representation that form the technical background to the development of the Structured Meta Knowledge formalism.

Chapter 4 describes the functional requirements for a medical terminology system based on the techniques identified in chapter 3 and focusing on the requirements from chapter 1.

Chapter 5 presents the Structured Meta Knowledge (SMK) formalism in relation to the functional description from chapter 4.

Chapter 6 is a preliminary account of the SMK modelling language.

Chapter 7 describe the implementation of the SMK formalism, the problems encountered, and some of the known limitations and omissions.

Chapter 8 presents an extended example of modelling in SMK and the relationship of this to traditional classification schemes.

Chapter 9 describes the extension of SMK to the representation of individual medical records.

Chapter 10 is the discussion and conclusions.

Chapter 2 Coding and Classification Schemes

This chapter presents an account of coding and classification schemes based on two exemplar classifications; the International Classification of Diseases and the Systematized Nomenclature of Medicine (SNOMED). The focus is on their representational properties and not their medical content. The aim is to explain why these schemes are in themselves inadequate as structured representations of medical concepts.

We begin with a general account of classifications and the changing demands being placed upon them. We then examine the International Classification of Diseases and its classificatory principles. This is followed by an analysis of SNOMED and in particular its compositional properties. We conclude with a summary of the main problems identified with the use of classification schemes.

Note on the background to this chapter

This chapter is based on an earlier analysis of coding and classification schemes. This analysis helped to identify why these schemes were inadequate for meeting the needs of the PEN&PAD clinical system, and moved the research in the direction of formal techniques for representing medical terminologies. The analysis was an important part of the background to the development of the formalism to be described in later chapters (Structured Meta Knowledge – SMK).

Since this analysis was performed the schemes mentioned have all recently completed or are undergoing a revision. However the conclusions of this chapter are unchanged.

2.1 The changing task: the move from populations to patients

Current methods of collecting and representing information in medical computing systems are deeply rooted in the epidemiological and statistical tradition. Within general practice for example, computers were initially used as tools for handling audit and preventive care information. Manual registers and logs, such as the basic age/sex or disease registers, were readily amenable to computerisation [RCGP 82]. The information was usually entered by administrative staff 'off-line' and in this respect the computer did not provide any significant improvement over manual systems. The benefits accrued when analyses were performed. It became possible to easily identify target groups such as 'all the young hypertensives in our practice'. Call and recall registers such as those for cervical cytology tests benefited from computerisation. The motivating force was a desire to apply the results of epidemiological research or clinical management policies to a practice population. Operating a few registers involves collecting and manipulating a small amount of information on a lot of people. The emphasis is upon uniformity to facilitate statistical analysis. In this regard the task is similar to that of an epidemiological study

As medical systems have developed the perspective has shifted from the population of patients to the individual. The established information tools have continued to be used on the implicit assumption that they can be adapted and extended to cope with the new tasks. This assumption is almost certainly false. Epidemiology seeks to iron out the effects of individuals and draw conclusions about communities. It is unlikely that tools devised specifically to abstract away from the individual towards uniformity are suitable for dealing with individual clinical information It is often the departure from the typical that matters most to both patient and clinician.

2.2 Current methods: nomenclatures and classifications

Historically the most important techniques for representing medical information in a structured form have been based on coding and classification schemes. To a large extent this is still the case today. The naming and classification of medical concepts, such as diseases, has been an important activity for over two hundred years [World Health Organization 77]. Nomenclatures, classifications, and coding schemes have found a use in most areas of medical activity and are often taken as being the 'naturally occurring form' of structured medical information. The problem of structuring information within a general practice computing system has been interpreted as equivalent to choosing the correct classification [GMSC-RCGP 88]. We shall begin with a general description of nomenclatures and simple enumerative classifications and then go on to illustrate some of the specific problems by looking at the International Classification of Diseases.

2.2.1 Nomenclatures.

A medical nomenclature is a collection of agreed terms or names for medical concepts, such as diseases. In principle it is a simple list of unique names each representing a single concept. A nomenclature does not in itself classify a concept, though the name may be derived from taxonomic principles drawing for example on pathology and anatomy. Most nomenclatures assign a unique 'code' to a single concept. The code may be a word or phrase, or a meaningless jumble of symbols. We shall use term to denote the combination of a code and a rubric

<term> : <code>–<rubric>

for example

a5f61–'acute myocardial infarction'

The precise choice of symbol is not crucial, rather it is the choice and definition of the concept it represents. Agreeing upon medically well defined concepts such as 'a disease of infectious aetiology' may be relatively straightforward. As concepts increase in complexity so does the scope for disagreement. What is 'stabbing pain in the chest worse on exercise', and does it merit inclusion and its own 'code'? Does it differ from 'sharp pain in the chest worse on exercise'?

2.2.2 Classifications

A classification is a representation of a set of concepts and the relationships between those concepts. It is an abstraction or simplification of the real world. Medical classifications are commonly uni–axial with single inheritance. This means that all the concepts reside in a single tree structure and each concept is related to a single parent. Each successive level down the hierarchy represents a greater refinement of a concept (Figure 2.1).

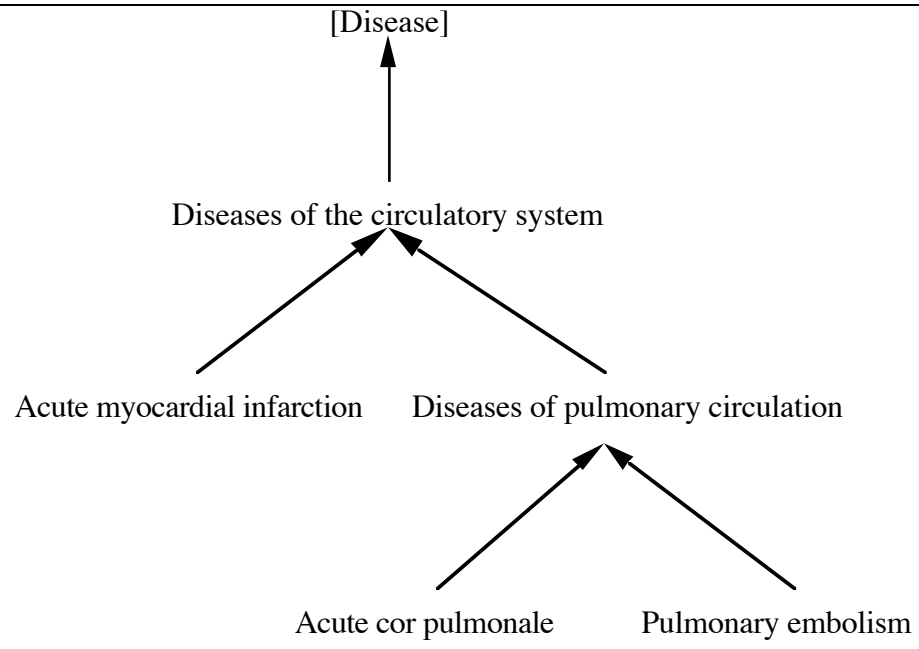

Figure 2.1 A graphical representation of a section of ICD-9, a mono–hierarchical, uni–axial classification.

2.2.3 Classification rules

The refinement of concepts requires the application of some form of classification rule to identify and separate concepts, typically into mutually exclusive subclasses. The hierarchy is read as meaning one disease is a type of another disease. For example in Figure 2.1, acute myocardial infarction is a type of disease of the circulatory system. What is implicit however is some understanding of what it means to be a disease of the circulatory system, and why acute myocardial infarction is one of these. It is also assumes that diseases of the circulatory system form a medically meaningful homogeneous class. Such rules and assumptions are not represented explicitly within the classification and typically they vary from concept to concept within the structure. The choice may also be deeply rooted in pragmatics and the relative medical importance attached to concepts. A classification does not represent fundamental 'medical truth', but rather it is goal-oriented [Wingert 89]. Both its choice of concepts and its structure are intimately related to the use for which it is intended.

2.3 An exemplar classification: the International Classification of Diseases (ICD)

2.3.1 History of the International Classification of Diseases

The International Classification of Diseases (ICD) is the oldest and most important classification still in use [World Health Organization 77]. It origins are to be found in the work of William Farr and Marc d'Espine, who each submitted a classification to the Second International Statistical Conference in 1855.

Farr classified diseases into five groupings:

- epidemic
- constitutional (general)
- local (by anatomical site)
- developmental

– those resulting from violence.

D'Espine on the other hand chose to classify by the nature of the disease:

– gouty

– herpetic

– haematic

– etc.

These two classifiers differed in both their choice of rubrics and their classification rules. The matter required a committee to resolve the dispute and in 1864 a total of 138 rubrics were adopted, and classified "in the style of Mr. W. Farr". Farr's principles of classification with particular emphasis on the anatomical site and aetiology of diseases remain evident in the classifications of today. This early disagreement upon classification principles serves as the model for most disagreements since, including much of the recent debate over the choice of a classification scheme for general practice [GMSC/RCGP 88] and subsequently for the UK National Health Service as a whole.

This early work developed into the International List of Causes of Death. In 1946 causes of significant morbidity were added to those of mortality, and from this emerged the ICD. The ICD is now in its ninth revision (ICD-9) with the delayed tenth revision just becoming available. The ICD is primarily concerned with clearly defined morbid and mortal conditions and is thus best suited to hospital and epidemiological usage.

2.3.2 The organisation of ICD-9

Medical concepts are assigned a code composed of three digits with an optional fourth digit following a decimal point. The classificatory relationship is represented through a mixed mechanism. The three–digit and four–digit codes relate in the obvious way. Higher level concepts are assigned ranges of codes. For example

Disease of the digestive system (520–579)
Other diseases of the intestines and peritoneum (560–569)
564 Functional digestive disorders
564.2 post gastric surgery syndromes

The codes are often described as 'not unique'. This means that several concepts may code to the same number, for example

564.2 post gastric surgery syndromes
dumping syndrome
post vagotomy syndrome
post gastrectomy syndrome

This is really a form of 'micro–classification' allowing a mapping from the larger nomenclature of diseases represented by words to that of the code numbers. All of the three specific dumping syndromes will 'code' in the same way. This reflects the assumption that for most purposes the differences between the three specific dumping syndromes do not matter. This may be true for epidemiological purposes but is inadequate for clinical use.

Limited cross-referencing is used to indicate a dual entry. A dagger (†) indicates inclusion by aetiology and asterisk (*) by site. For example

Inclusion by aetiology of disease:
Infectious and parasitic diseases
013.0 Tuberculous meningitis †

Inclusion by site of disease:
Diseases of the nervous system and sense organs
320.4 Tuberculous meningitis *

2.3.3 Classification principles within ICD-9

Figure 2.2 shows the three-digit categories within the first chapter of ICD–9 covering infectious and parasitic diseases.

1	intestinal infectious diseases
2	tuberculosis
3	zoonotic bacterial diseases
4	other bacterial diseases
5	poliomyelitis and other non-arthropod-borne viral diseases of central nervous system
6	viral diseases accompanied by exanthem
7	arthropod-borne viral diseases
8	other diseases due to viruses and chlamydia
9	rickettsioses and other arthropod-borne diseases
10	syphilis and other venereal diseases
11	other spirochaetal diseases
12	mycoses
13	helminthiasis
14	other infectious and parasitic diseases
15	late effects of infectious and parasitic diseases

Figure 2.2 three-digit categories within the first chapter of ICD-9, covering infectious and parasitic diseases
Note: the numbers used have no significance within ICD-9

Examination of these rubrics suggests the rules that have been used to classify the various infectious diseases. Figure 2.3 lists some candidate classification rules which appear to have been used. There are three points to note:

- the rules are not explicit and can only be inferred by reading the rubrics
- there is a strong model underlying the groupings, focusing mainly on causality
- the rules are combined in complex ways to form the classification

Example :

by causal agent [2,3,4,5,6,7,8,9,10,11,12,13]

by site of infection [1, 5,]

by being a zoonosis [3]

by having an arthropod vector [7,9]

by clinical features [6,15]

by mode of transmission [10]

by not being one of the above [4,5,8,9,11,14]

Figure 2.3 Suggested classification principles behind the classification of chapter 1 of ICD–9. The numbers refer to Figure 2.2

2.3.4 Goal-orientation

In the above example the choice of disease categories and classification rules closely reflects the purpose for which the classification was developed. The ICD like all classifications is strongly **goal-oriented**. It is primarily an epidemiological tool and focuses upon major aetiological factors, diseases of world-wide importance, and causes of serious morbidity and mortality. Embedded within its classification scheme are very strong views on disease models and prevalence. Those views are implicit and not open to inspection, nor can they be changed or adapted by the user of the scheme. The purpose for which it is to be used pervades its entire structure. A classification is not only a representation of medical concepts – it is a model of the goal sought by its use.

The ICD will only be satisfactory when used in an appropriate setting where the goals concur with its own embedded assumptions. It is not surprising that it has been considered inappropriate for use in general practice [RCGP 86, GMSC/RCGP 88]. What is remarkable is the continuing success of the ICD, not only in its native form but also as the basis of several other classifications. The embedded model of the ICD must be a reasonable reflection of some more general shared medical model.

2.3.5 The use of the classification structure: abstraction

The value of the hierarchical structure is in its support of abstraction. This is the process whereby lower level, more specific concepts can be brought together into higher level ones to allow conclusions to be drawn. ICD-9 is used to abstract from specific, and perhaps relatively rare diseases to broader disease concepts.

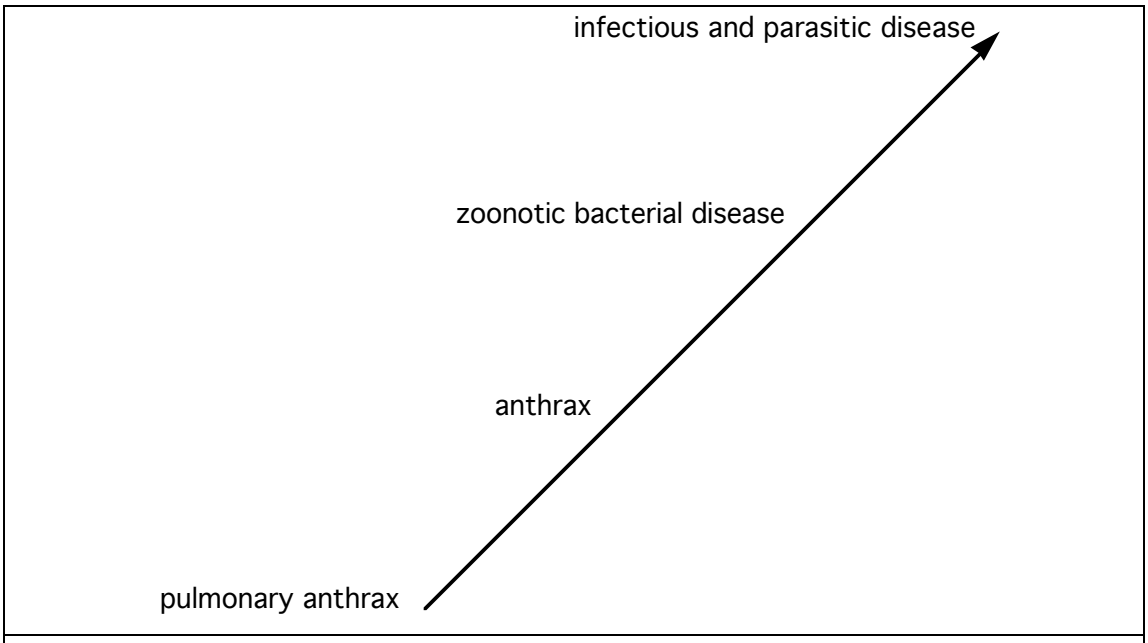

Figure 2.4. The use of abstraction in ICD-9

In Figure 2.4 pulmonary anthrax is in some sense *a kind of* anthrax, and thus *a kind of* zoonotic bacterial disease etc. The function of the abstraction within ICD is to facilitate statistical analysis at the appropriate level. Such analysis is typically intended to support causal hypothesising or to identify potential health problems within a community.

It is the goal-orientation of a classification which gives meaning to the abstraction. A scheme for disease classification based on principles of health care financing is unlikely to support abstraction suitable for the research of environmental factors in disease aetiology. If abstraction leads to a disease being described as a kind of 'expensive disease' then it is not obvious how this can give any insight into the relevance of sewage disposal as a public health problem. If however abstraction classifies it as a type of infectious gastrointestinal disease then useful conclusions may be drawn.

This example is perhaps extreme but highlights the problem found when classifications are transported from one task to another. In order to be appropriate for a task a classification must

- represent an appropriate choice of concepts
- embody a relevant abstraction.

A classification may be unsuitable for either or both reasons. If a classification appears to be failing in one of these respects then the usual response is to construct another classification. The result is a serious fragmentation of medical terminology with a lack of standards. This has resulted in a major secondary activity aimed at reconciling the numerous classifications in current use. The largest example of this activity is the Unified Medical Language System (UMLS) of the USA National Library of Medicine [Evans 1987, Evans 1988, Barr 1988].

2.3.6 Extending the coverage of a classification and the 'combinatorial explosion'

The 'codes' within a classification scheme are atomic, and it is not possible to form new codes by combining existing ones. Consider a scheme in which there are 1,000 diseases. If we introduce the idea of the severity of a disease, and limit this to three degrees of severity (mild, moderate, and severe), then the total number of terms representing diseases becomes:

1,000 disease x 3 severities + 1000 original terms = 4,000 terms

In order to increase the expressive power to cover the severity of a disease the number of terms has quadrupled. If we now introduce the progress of a disease (better, same, worse) the result is 16,000 terms.

Simple schemes are enumerative, and lack any compositional features. As a consequence they suffer from the 'combinatorial explosion'. Conceptually small extensions to the content produce an explosion of terms. The problem has been contained as long as classifications have been used for epidemiological and statistical purposes. However clinical descriptions require qualifiers and modifiers such as severity. This has created serious difficulties for coding schemes and is one of their major inadequacies.

2.4 Multiaxial nomenclatures and classifications: SNOMED

A traditional medical classification is a large structure and represents a major investment of effort in both its creation and maintenance. Its basic model of the world is written into its structure and as it grows it becomes increasingly difficult to adapt to new concepts or accommodate a new perspective on that world. The goal or purpose becomes more and more deeply embedded. The strong goal-orientation of a classification can be of great benefit when it is used in an appropriate setting, but it is also the major obstacle to adaptation and innovation. The dangers of the combinatorial explosion also act to limit what is expressed and further reduce their utility. An approach to this problem is to attempt a separation of the basic concepts within the domain from the more complex ideas and perspectives embedded in the hierarchical structure. Wingert et al described this as building a semantic model [Wingert 1989].

2.4.1 Organisation of SNOMED

"SNOMED is a systematized multi-axial nomenclature of medically useful terms hierarchically organized where possible"

So begins the introduction to the first edition of the Systematized Nomenclature of Medicine (SNOMED) [College of American Pathologists 1977 & 1982]. SNOMED adopts a multiaxial approach to the problems presented by the mono–hierarchical classification. Its origins are in the older and smaller Systematized Nomenclature of Pathology (SNOP). SNOMED allows the construction of complex terms from codes. These codes represent concepts mainly originating in the basic medical sciences. SNOMED is the largest and most complete multiaxial nomenclature available though alternatives and modifications have been proposed [Cote 1989].

SNOMED comprises six principle axes. The choice of axes is based on a particular view of biomedicine and the nature of man. The interpretation of the titles is broad:

Topography (T) – refers to tissues, organs, and bodyparts eg. muscle, liver, arm

Morphology (M) – describes abnormal changes in form including pathological anatomy eg. inflammation, neoplasia

Function (F) – encompasses all human functions and malfunctions eg. breathing, dyspnoea

Etiology (E) – classifies those agents relevant to disease causation including chemicals and micro-organisms

Disease (D) – many disease concepts and syndromes correspond to complex combinations of the T,M,F, and E fields. In recognition of this a disease classification was added which essentially corresponds to that of a more traditional classification such as ICD–9.

Procedure (P) – this classifies the types of actions performed by health care workers eg. operations, injections

2.4.2 SNOMED coding of terms

Each axis has an associated listing of concepts called a fascicle. Each concept is assigned an alphanumeric code. The first character of the code denotes the axes (T,M,E,F,D,P) and this is followed by up to five characters representing a duodecimal number. Every effort is made to avoid the use of the characters X and Y to represent decimal 10 and 11 respectively.

Coding in SNOMED is performed by combining concepts from one or more of the available axes. The example below, adapted from the introduction to SNOMED uses four axes, **T**opography, **M**orphology, **E**tiology, **F**unction.

T		M		E		F
Lung	+	Granuloma	+	M. tuberculosis	+	Fever
T-28000		M-44060		E-2001		F-03003

The implied semantic relationships between these axes are not stated within the published nomenclature. At this stage it is not possible nor desirable to be rigorous but a reasonable interpretation would be:

granuloma *in* **lung** *caused by* **M. tuberculosis** *together with* **fever**

The three relationships *in*, *caused by*, and *together with* are the basis of the language relating the four orthogonal mono–hierarchical classifications. The four codes define a unique point in a four dimensional SNOMED space. The purpose of the language is to allow the translation from the external concept, the thing to be coded, to a SNOMED term and vice versa. It specifies the relationships which must exist between the primitives within the external concept for it to be permissible to represent it by the SNOMED term.

The disease field does not fit simply within this model. One interpretation is to consider a disease as being similar to a morphological abnormality. The SNOMED term

T		E		F		D
Lung	+	M. tuberculosis	+	Fever	+	Tuberculosis
T-28000		E-2001		F-03003		D-0188

could thus be interpreted as:

tuberculosis *in* **lung** *caused by* **M. tuberculosis** *together with* **fever**

A second interpretation is one of equating a description with a disease. This is the example given in the first edition.

T		M		E		F		D
Lung	+	Granuloma	+	M. tuberculosis	+	Fever	=	Tuberculosis

T-28000		M-44060		E-2001		F-03003		D-0188

granuloma *in* **lung** *caused by* **M. tuberculosis** *together with* **fever** *which is* **tuberculosis**

This approach can serve as a means of linking the composite four field SNOMED term to a single concept within a simple mono–hierarchical classification such as ICD-9.

The two preceding examples appears to contain redundant information with 'tuberculosis' being mentioned several times. Two points however need to be considered. Firstly, for the disease tuberculosis there is a well understood physiological and aetiological model and the SNOMED term is relatively simple. The disease field however is particular important in representing complex diseases and syndromes not easily described clinically by a single aetiology, morphological abnormality, or functional disturbance eg. migraine. Secondly a person with medical knowledge will immediately infer that infection with M. tuberculosis means tuberculosis. This relationship can only be recorded explicitly within the SNOMED term. Some cross-referencing between diseases and other fields is present within SNOMED but this is not amenable to any general interpretation.

2.4.3 Sense and nonsense in SNOMED

According to the introduction that accompanies SNOMED, if the six axes of the first edition are considered orthogonal then we have

total number of concepts listed = 39,377

total number of possible terms = 6.7×10^{22}

However most of these terms will be medical nonsense. For example

T		M		E		F
Colon	+	Fracture	+	Donkey	+	Emotional state
T-67000		M-12000		E-4986		F-90000

This reads

fracture *in* **colon** *caused by* **donkey** *together with* **emotional state**

SNOMED is not a general classification of the things it is sensible to say in medicine. It cannot prevent the creation of medically meaningless utterances. It is a multidimensional space within which are the points corresponding to those sensible statements, but not all possible points correspond to a sensible concept. SNOMED is a framework for expression but is seriously impaired by the lack of rules for determining which compositions are sensible.

2.5 The structure within SNOMED fascicles

The above discussion has concentrated upon the relationships between SNOMED fields and terms. The organisation within each of the SNOMED fascicles deserves further analysis. This analysis will concentrate on the topography field.

2.5.1 Abstraction in SNOMED

Abstraction within classifications was discussed above. The idea of abstraction will now be further refined. Three basic types will be considered [Brodie 1984].

Generalisation

Aggregation

Association

Generalisation

In this form of abstraction similar concepts are considered as specialisations of a parent concept.

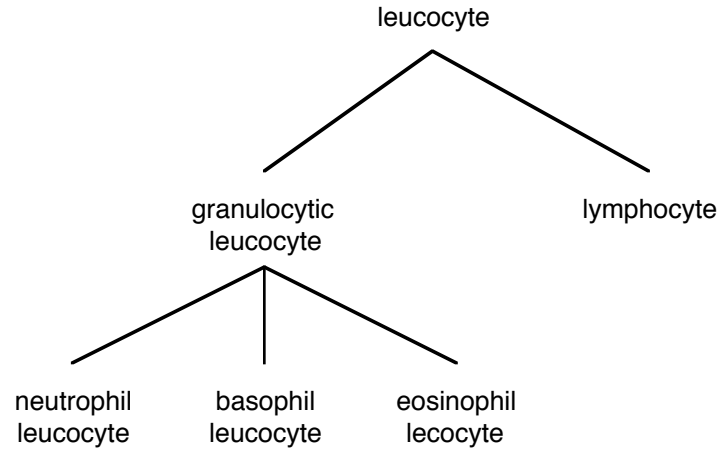

Figure 2.5 A generalisation/specialisation hierarchy for blood leucocytes

Generalisation establishes an *is-a* relationship between concepts (Figure 2.5). For example

eosinophil *is-a* granulocyte.

This is essentially the sole form of relationship within a simple classification such as ICD–9. The implicit classification rules are just those rules which define what it is that makes one concept a type of another. Granulocytes are leucocytes seen to contain granules on light microscopy, using an appropriate staining technique.

Aggregation

Aggregation is a form of abstraction in which the relationships between objects is in their forming a higher level aggregate object.

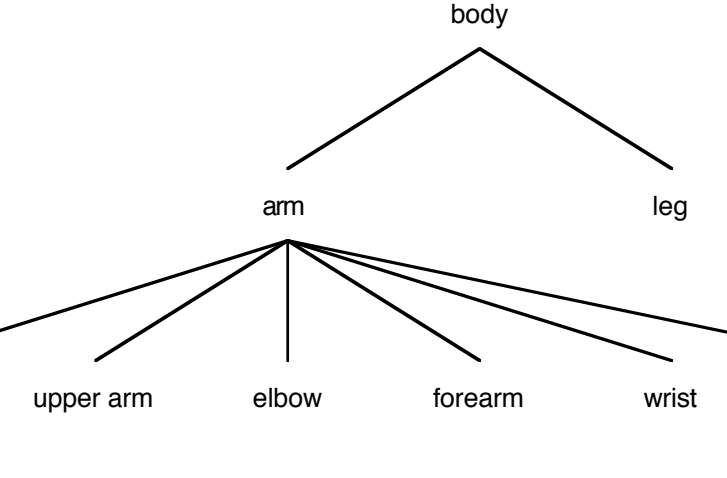

Figure 2.6 An aggregation/disaggregation hierarchy for the arm

Aggregation establishes an *is-part-of* relationship between concepts (Figure 2.5). For example

hand *is-part-of* arm

Association

This is the set membership relationship.

the elbow joint *is-a-member-of* all the bones of the body

This is described by some authors as grouping and its inverse as partitioning.

2.5.2 Relationships between topographical concepts in the SNOMED topography fascicle

This fascicle comprises ten sections (0 to X) covering the major organ systems and an eleventh (Y) for topographical regions.

0	General body topography; integumentary, haematopoietic and lymphatic systems
1	Musculoskeletal system and soft tissues
2	Respiratory system
3&4	Cardiovascular system
5&6	Digestive system
7&8	Genitourinary system and foetal structures
9	Endocrine system
X	Nervous system and special sense organs
Y	Topographical regions

In broad terms the code numbers are used to represent the classificatory structure, as with a simple classification. When the coding structure is examined, however, this statement becomes difficult to support without many qualifications and additional analyses. It will be proposed that the structure be perhaps best viewed as a collection of hierarchies. Examples will illustrate some of these points.

Figure 2.7 is an extract from Section 1 of the topography fascicle.

T-1X500	bone, NOS
T-1X501	long bone
T-1X502	short bone
T-1X503	flat bone
T-1X504	pneumatic bone
T-1X505	shaft of bone
T-1X506	proximal third of shaft of bone
T-1X507	middle third of shaft of bone
T-1X508	distal third of shaft of bone
T-1X510	periostium
T-1X511	subperiosteum
T-1X520	cortex of bone

(NOS - not otherwise specified)

Figure 2.7 Section of the SNOMED topography fascicle covering bones

An initial interpretation of the code numbers is to draw the hierarchy using a strict numerical rule, as can be done for a simple hierarchy such as ICD-9 (Figure 2.8).

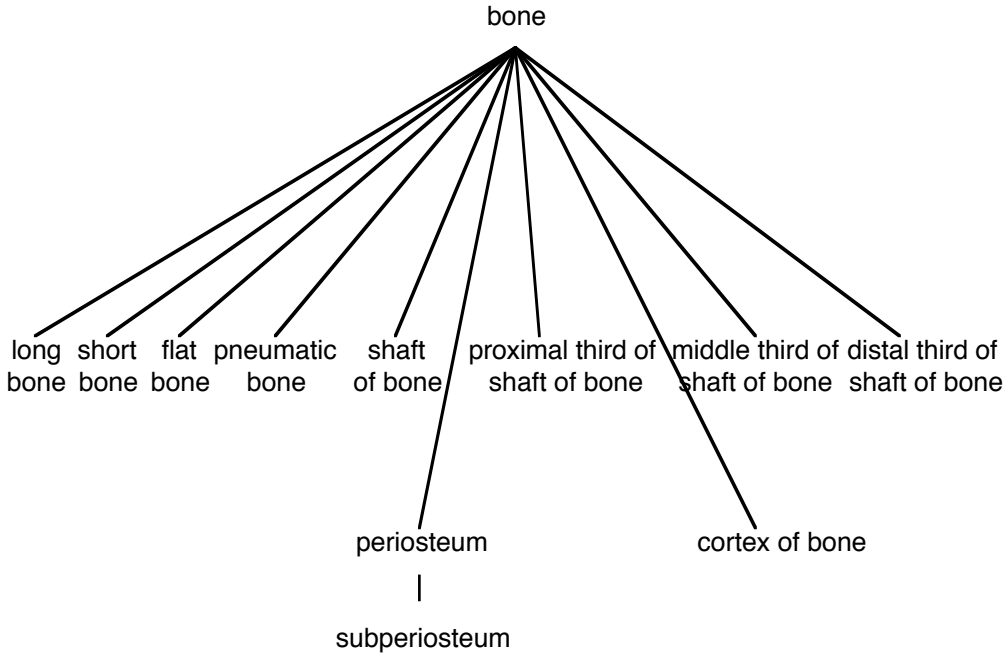

Figure 2.8. A strict numerical interpretation of the codes for part of the SNOMED topography fascicle

The interpretation of this structure is unclear. There are many types of relationship within it and the form of abstraction used is not clearly defined. It is reasonable to propose that a flat bone *is-a* bone. But is this case for a shaft of bone? It is hard to define a rule supporting the proposition that periosteum *is-a* bone. In an attempt to resolve this confusion the hierarchy needs to be disassembled. In the Figure 2.9 the structure has been re drawn as three mini–hierarchies in an attempt to identify the relevant types of abstraction operating between the concepts.

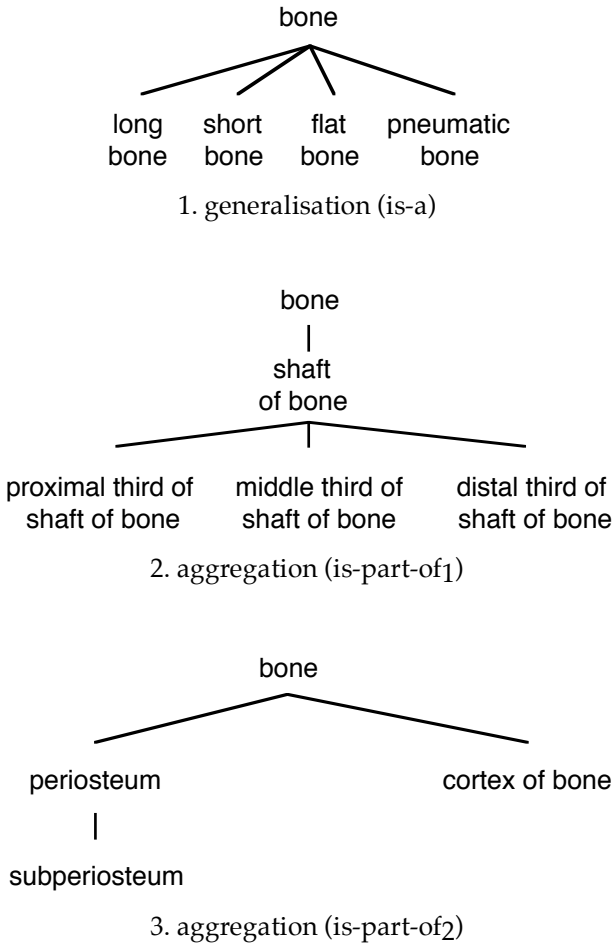

Figure 2.9 Three separate hierarchies derived from SNOMED terms in Figure 2.8

In this analysis aggregation occurs twice and this identifies another important relationship embedded within SNOMED. The first aggregation (is-part-of$_1$) groups the parts of a bone in a physical or spatial sense - the proximal, middle, and distal thirds. The second (is-part-of$_2$) groups some of the subtypes of tissue comprising bone. Thus there are two differing interpretations of *is-part-of,* the second of which the second is probably better described as an *is-constituent-of* relationship.

Specific anatomical structures

The above example considered the typical bone, its parts and subtypes. Probably the bulk of the topography fascicle is given over to specific named anatomical structures. In such sections the relationship between concepts is generally much clearer and tends to be aggregation. A section covering the bones of the lower extremity is shown in Figure 2.9.

T-11700	bone of lower extremity, NOS
T-11710	femur
T-11711	head of femur
T-11712	neck of femur
T-11713	greater trochanter of femur
.....	
T-11720	patella
.....	
T-11730	tibia
.....	
T-11740	fibula

Figure 2.10 Section of the SNOMED topography field for bones of lower extremity

The femur, patella, tibia, and fibula are all members of the set of bones of the lower extremity. This is association, and is generally only implicit within SNOMED and not explicitly coded. The code 'T-11710 bone of lower extremity, NOS' can be used to represent any *one* bone of the lower extremity but not *all* of them. This relationship is generalisation and not aggregation.

In the preceding extract from SNOMED, the femur has some of its parts enumerated. The abstraction here clearly involves aggregation. It is worth noting however that it is unusual for an anatomical structure to have all its parts enumerated and also any parts that are included may incompletely overlap.

2.6 SNOMED enhancements.

To the SNOMED coding scheme as described has been added three significant enhancements which are intended to make it useful for clinical descriptions. The relate to the creation of clinical records and can be appended to a SNOMED term

Systems information qualifiers

This is a list of phrases used to qualify an entire SNOMED term

HO history of
WD working diagnosis of
PH past history of
TR treatment required for
P problem
.....

Syntactic linkage symbols

These can be used to explicitly link one SNOMED term to another

DT due to
AW associated with
NL no link
.....

Time relations

There are five formats for recording time intervals. For example

Y001 = one year
D005 = five days.

2.6.1 Example of a clinical description in SNOMED

The following is an example of a complex pair of SNOMED terms:

Qualifier	T	M	E	F	D	Time	Link
HO	T28000	M32800	-	F75870	-	Y015	AW
	Lung	Emphysema		Cough chronic			
HO	-	-	-	F02850	-	Y025	NL
				Smoker heavy			

This example can be interpreted as:

history of **emphysema** *in* **lung** *together with* **chronic cough** *for* **15 years** *associated with* history of **heavy smoker**

The use of the links appears to provide a useful framework for clinical descriptions. However the mechanism aggravates the problems of interpretation found with the simpler terms. SNOMED is a major medical achievement in its choice of basic axes and the compilation of the nomenclatures within each axis. However it does not provide the immutable rules for representing any given medical statement.

2.7 Summary of nomenclatures and classifications and their inadequacies

Classification schemes are currently the commonest approach to the representation of medical concepts. They have their roots in epidemiological and statistical traditions. There are, however, several intrinsic properties that are obstacles to their use for clinical descriptions in computer–based systems:

1 simple classifications and nomenclatures are enumerative and thus as they are extended to clinical descriptions they suffer from the 'combinatorial explosion' of terms

2 classification rules are implicit and inconsistent, which reflects their strong goal-orientation making them unsuitable for multiple uses

3 the compositional scheme SNOMED overcomes some of the problems of the combinatorial explosion, but lacks formal rules for forming compositions and can thus produce nonsensical terms

4 within any single axis of SNOMED there is a failure to make important distinctions between different types of relationships

Chapter 3 Relevant Issues in Knowledge Representation

In this chapter we shall examine issues in knowledge representation which form the technical background to the Structured Meta Knowledge for the representation of medical concepts. The main theme of this chapter is the separation of knowledge into *terminologic*al and assertional kinds, with terminological knowledge at the heart of what we will require. A separate but related issue of 'functional approaches' to knowledge representation will also be outlined, prior to its use for presenting the requirements for a medical terminology system in chapter 4.

3.1 The need for knowledge representation techniques

Chapter 2 identified inadequacies in current approaches to medical terminologies. In this section we review those inadequacies and examine why knowledge representation is relevant to medical terminologies.

3.1.1 Current inadequacies

In the preceding chapter we examined specific representational problems with medical coding and classification schemes. These are that:

- simple coding schemes require the explicit enumeration of all possible terms and the relationships amongst those terms;
- compositional systems such as SNOMED provide few rules for ensuring that only medically sensible compositions are formed;
- classification is goal-oriented, manifested usually through a single hierarchical organisation, with ill-defined, implicit classificatory principles, and little or no relationship to any compositional features of the scheme.

These problems have worsened with the advent of computer-based medical information systems. The interpreter of the scheme is no longer a person. Intuition and background knowledge cannot be called upon to resolve or ignore 'obvious' ambiguities of meaning and imprecise classification. The introduction of computer-based clinical systems has proved to be both a major stimulus and serious challenge to the development of coding and classification schemes.

In response to these challenges most major coding schemes are introducing or extending their use of:

- compositional features through the use of qualifiers and modifiers, such as those for disease severity and the location of certain disorders. Codes are 'added' together or in some way related;
- multiple classification hierarchies derived from the various 'facets' of concepts, such as disease location or pathophysiology, or aimed at supporting specific health care specialist groups.

SNOMED has always adopted a compositional approach, but is being extended with more attention to structure of compositions [SNOMED III]. The NHS Centre for Coding and

Classification is planning to introduce groups of qualifying codes into the Read Clinical Classification [NHS CCC personal communication]

The realisation of the need for compositionality and multiple hierarchies is an important step in the development of medical coding schemes. However this realisation is not in itself a solution but rather a reformulation of the underlying problems, and raises a new, potentially more difficult and problematic set of questions.

3.1.2 Increasing complexity

Qualifiers such as those for the severity and progression of a disease appear to forestall the full effects of the combinatorial explosion, but what are the rules for combining these? There is the now familiar problem of preventing nonsense

penicillin + severe

but is it possible to apply several qualifiers simultaneously

bronchitis + severe + worsening

and if so what are we to make of

bronchitis + severe + mild.

If multiple hierarchies are introduced then

- how are they to be coordinated? Are they true multiple hierarchies or merely independent alternatives, with no guarantee that the use of two or more together will give coherent answers?

- what, if any, is their relationship to the use of compositions? Can 'bronchitis+severe' itself be classified?

The use of more powerful compositions and multiple hierarchies appears to relieve inadequacies in coding schemes. However the price for these benefits is an unavoidable increase in complexity. The development of coding schemes is now grappling with the fundamental trade–off between expressive power and technical complexity, a problem familiar to workers in the field of knowledge representation.

A proliferation of *ad hoc* mechanisms for tackling this complexity poses a threat to the utility of coding and classification schemes greater than that from their original shortcomings. It is true these shortcomings are a major obstacle to the development of advanced medical information systems but they are to a large extent self evident. Users can form judgements as to their importance in particular situations, and choose to limit their objectives or provide work–arounds which they understand. However a collection of *ad hoc* technical methods for combining codes and hierarchies, subject to many interpretations, and buried in the program code of complex computer systems will inevitably lead to incorrect and unpredictable behaviour. There is no escaping the need for a principled interpretation of what compositions and hierarchies mean. This meaning is not medical in and the heads of clinical professionals, but formal and defined in relation to the coding and classification scheme. This need has *de facto* taken the tradition of medical coding and classification into the field of knowledge representation.

3.2 Issues in knowledge representation

We shall now discuss some key issues in knowledge representation that have shaped the approach to medical terminology embodied in the SMK formalism. The topics to be discussed are:

- compositional data structures such as semantic networks and frame–based languages, and problems in their interpretation;

– the distinction between two broad types of knowledge – terminological knowledge and more general assertional knowledge;

– the limitations of terminological knowledge and the need for a less restricted definition.

The final issue is on a different but related theme:

– 'functional' approaches to knowledge representation that move away from data structures to descriptions of *what* systems do rather than *how* they do it.

3.3 Compositionality, data structures, networks, and frames

Compositionality requires relating or linking concepts together. This is the essential feature of the family of representational techniques for composing descriptions such as semantic networks and frames (for an overview see [Ringland 1988]). It is a characteristic of these representations that relationships or links can be specified, above and beyond the '+' of our earlier example. They can be named and distinguished one from the other. Thus we could rewrite our example schematically as

bronchitis–severity–severe

It would now appear clear what the relationship means; 'severe' is the 'severity' of the 'bronchitis'. Unfortunately this is not the case. The use of suggestive naming conventions has disguised a fundamental problem of interpretation. It is true we can now distinguish amongst relationships, but we have translated the uncertainty over the interpretation of '+' into a dual uncertainty over the pair of links '-'. We shall now discuss two distinct interpretations of those links.

3.4 Two interpretations: assertional versus definitional

To understand the two interpretations we shall consider some example data structures. In this discussion we consider frame and semantic network languages to be essentially equivalent. In a semantic network there are objects (nodes) which are linked by labelled arcs to create a complex data structure. For example in a semantic network style we may sketch a structure concerned with cancer (figure 3.1).

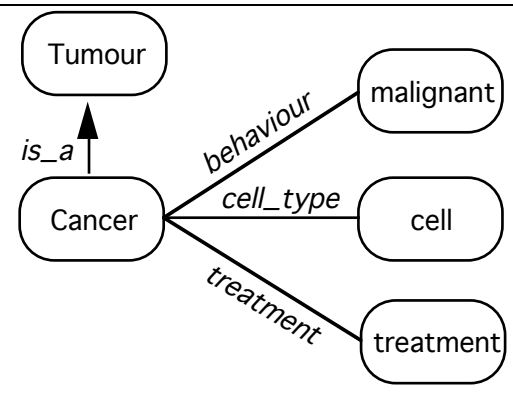

Figure 3.1 Some knowledge related to cancer in a semantic network

In a frame–based language we may have something of the form

```
cancer:
            isa:        tumour
      behaviour:        malignant
       cell_type:       cell
      treatment:        treatment
```

In this example *behaviour* is a slot and *malignant* is the filler of that slot. The differences between the network and frame–based forms are essentially notational.

On the face of it these would appear to be useful structures associating important information with the concept of cancer, and a reader can draw conclusions as to what this information means. This however is the problem. The interpretation of a link is left to the user of the data structure. We haven't defined a principled interpretation of the fundamental structure itself, which is necessary if it is to *mean* anything in a formal representation, as opposed to meaning something in the head of a person. The question over the formal interpretation of such structures is at the heart of Woods' landmark paper 'What's in a link' in which he addresses critical issues in the interpretation of semantic networks [Woods 1975]. This paper set a standard for work in this area, and much has followed, notably that of Brachman and Levesque. The following discussions follow this work closely.

There are many detailed issues of interpretation but the most important of these is the distinction between two particular readings of the links in such a data structure. The first sees them as assertions about the concept. The second interprets them as a structured description of what is meant by the concept, that is its definition. We shall consider these in turn.

3.4.1 First interpretation: links as assertions

The first interpretation takes the links to be assertions or statements *about* the concept. The example frame could be interpreted as corresponding roughly to the sentence:

'every cancer is a tumour and behaves malignantly and is composed of some sort of cell and is treated with some sort of treatment'

This has been and continues to be an important interpretation. However as a mechanism for making assertions it raises two important points:

1 Which properties are essential and which are incidental? Are some of the properties listed necessary for cancer to be cancer? It would not make much sense medically if the concept were not malignant. However knowledge of the precise cell–type or treatment used is not critical. Even if the cell–type were unknown we would still be satisfied that we are dealing with cancer. Being malignant is essential, whereas the cell type is in some sense incidental to being a cancer. This is not to imply that the precise cell type of a cancer is not medically important, but it is not essential to understanding the idea of cancer.

   The inevitable consequence of this interpretation is that true composite descriptions cannot be expressed. For example to form the idea of 'squamous cell cancer' would require the creation of a squamous–cell– cancer frame and an assertion that the cell type is epithelial–cell. But this again fails to recognise that the cell type is now essential to the definition of the concept.

2 As a mechanism for representing assertions it makes the handling of incomplete knowledge very difficult. For example the sentence 'either chemotherapy or surgery or radiotherapy is used to treat cancer' is hard to capture.

The latter point is important. The expressive power of a representation is more or less dependent upon its ability to represent incomplete knowledge [Brachman 1985]. We shall consider this in more detail when looking at what a terminological system can not represent, but it is the former difficulty with definitions which is our main concern.

3.4.2 Second interpretation: links as definitions

The second interpretation of the links sees them as part of the structure of a concept. The links are not statements about the concept cancer but constitute its structure in the sense that they form an 'Aristotelian' definition of the concept cancer. Thus the frame

cancer:	
isa:	tumour
behaviour:	malignant

would be interpreted as meaning that cancer is a some sort of shorthand for the phrase

'malignant tumour'

The linguistic distinctions here are significant. In the assertional interpretation the structure corresponds to a sentence whilst in the structural interpretation it translates to a noun phrase.

Woods put all this succinctly in his example of a node in a semantic network for which there are links to 'telephone' and 'black' [Woods 1975]. Is this node to be interpreted as an assertion

'telephones are black'

or as an intensional definition

'black telephone'

Woods also pointed out that the first interpretation does not in itself indicate the quantification associated with the assertion. Does it mean all telephones, some telephones, most telephones, telephones in general unless there is information to the contrary, or what?

The point here is not that one or other interpretation is correct, rather that they are different. By considering the problem in terms of data structures such as semantic networks and frames we are left open to ambiguities of a fundamental kind, confusing two quite different interpretations. In the case of cancer it would be usual to consider the links to *tumour* and *malignant* as essential to what we mean by cancer, while those for *cell_type* and *treatment* can tell us something about *Cancer*. However this distinction is not based purely on the type of information. We have seen that the *cell_type* is essential to our definition of a 'malignant epithelial cell tumour', usually called *Carcinoma*. In principle any relationship could form part of the definition of a concept or be used in some assertional way to describe a concept. The critical thing is that we must be able to unambiguously distinguish between the two uses.

3.5 Separating terminological and assertional knowledge

The distinctions between the definitional and assertional interpretations has been the focus of extensive study and motivated the development of several knowledge representation systems. In the work of Brachman and Levesque [Brachman 1979] and Brachman [Brachman 1983a, 1985] a clear distinction was made between the two forms of knowledge

terminological knowledge – that which corresponds to the intensional interpretation

assertional knowledge – other more general facts about a concept

On the basis of this distinction a variety of systems have been developed. Some such as the representation language KL–ONE [Brachman 1979] and its derivatives have concentrated on terminological knowledge alone and its utility for knowledge representation. Others, in particular the more general hybrid knowledge representation system KRYPTON, have explored the relationship between terminological and assertional knowledge [Brachman 1983a], employing different approaches to each of these.

Making clear the distinction between the two was not the sole motivation for the separation. It is generally believed that the representation of terminological knowledge is possible using a more restricted formalism than that for assertional knowledge, and hence is computationally more tractable [Brachman 1984, Nebel 1990, Rector 1986]. This is an important issue but first we shall consider KRYPTON in more detail.

3.5.1 The T–Box and A–Box

The classic distinction made in the KRYPTON system was that between the part of the system responsible for the representation of the terminological knowledge, the T–Box, and that which handled the more general assertional knowledge, the A–Box [Brachman 1983a].

3.5.2 The T–Box

At its core the language of the T–Box provides for

- the representation of concept definitions
- the determination of the subsumption relationship between two concepts, that is whether or not one concept is a more general or specific form of another.

There is also a limited ability to define primitive types. These are concepts which are considered to have no necessary and sufficient definition. They are a way of dealing with some of the problems with concepts considered to be natural kinds, such as 'dog' or 'femur'.

Expression forming operators within the T–Box language are used to compose the definitions of more complex concepts in terms of simpler ones. For example the operation **ConjGeneric** forms a new concept as the conjoin of two others. It is then possible to use this as the definition of a symbol. Using Brachman's example we can define what is meant by *bachelor*

(**ConjGeneric** *unmarried–person man*)

There are other operations which make use of roles which are similar to attributes. These extend the scope of what may constitute a definition. Two such operations are:

1 Value-restricted generic

(**VRGeneric** person child bachelor)

'a person any child of which is a bachelor'

2 Number–restricted generic

(NRGeneric person child 1 3)

'a person whose number of children is between 1 and 3 inclusive'

3.5.3 Subsumption

These definitions are often called structured descriptions and are the basis for determining the subsumption relationship between two concepts. In the above example, given the definition of *bachelor* we would expect the system to conclude that *man* subsumed the concept *bachelor*, because its definition is more general than that for *bachelor*.

Subsumption is the basis of classification which is a fundamental means of reasoning within terminological systems. It can be used for constructing a taxonomic hierarchy based upon the comparison of each newly defined concept with those which have been previously defined. The classifier within KRYPTON constructs and maintains just such a taxonomy of concepts [Schmolze 1984]. The problem of the computational tractability of subsumption has been extensively studied and amongst other things appears to be critically sensitive to what is allowed to constitute a definition [Brachman 1985, Patel–Schneider 1989, Nebel 1990]. We shall need to return to these issues in more detail in later sections.

3.5.4 The A–Box

The A–Box within KRYPTON is in principle of less relevance to the development of a medical terminology system, but is important both in what it says is excluded from the T–Box and in how it relates to the T–Box.

The language of the A–Box within KRYPTON is essentially that of a first order predicate calculus language, and hence contains sentence forming operators such as **Not**, **Or**, and **ThereExists** (unrestricted negation, unrestricted disjunction, and existential quantification respectively). These have been explicitly excluded from the T–Box. For example if we extend Brachman's example to include the bachelor's prospects of marriage, the following proposition may prove useful

$$\exists x,y \text{ bachelor}(x) \wedge \text{spinster}(y) \wedge \text{friend}(x,y) \wedge \text{not}(\text{family\_relation}(x,y))$$

'there exist a bachelor and a spinster who are friends and not related'

3.5.5 Relating the T–Box and the A–Box

As suggested by the last example the relationship between the T–Box and the A–Box can be thought of as the A–Box comprising first order sentences for which the predicates (bachelor, spinster, friend, family_relation) are to be found as terms in the T–Box (figure 3.2).

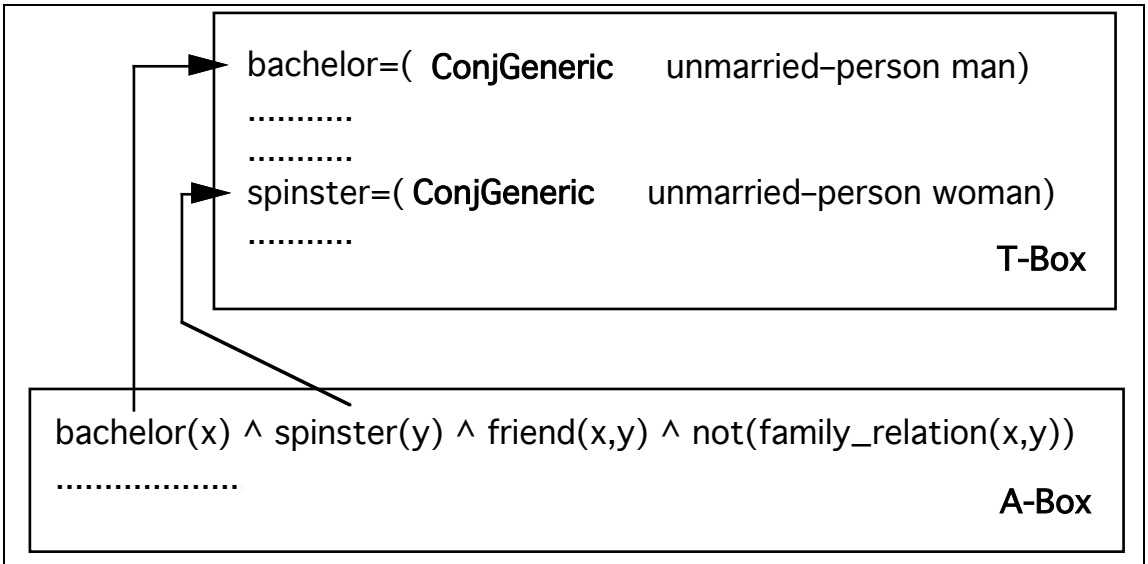

Figure 3.2 The relationship between T–Box and A–Box

There are options over the practical issues of relating the two. One approach is to expand the theory in the A-box by making assertions corresponding to the definitions in the T-box and then do standard first order reasoning over the resultant theory. This essentially gives two representations of the same knowledge. In the A–Box a bachelor becomes

$$\text{bachelor}(x) \Leftrightarrow \text{man}(x) \wedge \text{unmarried–person}(x)$$

Unfortunately this fails to distinguish those facts derived from terms in the T–Box and those more arbitrary facts which happen to have the same logical form This reintroduces the original problem of confusing the two interpretations.

In general the T-box places conditional dependencies amongst the normally independent predicates. The most important of these is of course subsumption. For example if in the TBox malignant_melanoma is subsumed by cancer then the two propositions

$$\text{malignant\_melanoma}(x)$$

$$\text{not}(\text{cancer}(x))$$

are inconsistent in the A-box.

3.6 Extending terminological knowledge

The definition of terminological knowledge embodied in the T–Box of KRYPTON is extremely limiting in what it permits. In this section we shall propose necessary extensions to that definition in order to meet the requirements of medical terminologies.

3.6.1 Language restriction and loss of utility

A 'pure' T–Box language has been shown to seriously limit its utility for practical knowledge representation, particularly medical knowledge [Haimowitz 1988, Doyle 1989]. The main justification for restricting the language is that in the worst cases the system should not take an unreasonable amount of time to complete its inferences, and in particular that as the size of the knowledge in the system grows the time taken should not grow exponentially. This was the reason for excluding from the T–Box 'incomplete' knowledge, represented by unrestricted negation, unrestricted disjunction, existential quantification, and similar constructs.

This requirement for tractability in all cases has been questioned as a useful measure of the utility of a knowledge representation system [Doyle 1989]. Computational issues are important but should not be the sole determinant of the shape of a formalism. The argument is that a tractable but inadequate formalism is just as useless as an intractable one. But can formalisms be found which perform well in 'typical' cases, despite possibly being intractable in the worse case? Furthermore can other techniques and heuristics be used to cope with any practical problems that arise? Answers to these questions are an important part of the proof of concept for any proposed terminology system based on an extended T–Box[2], but are almost certainly empirical, and depend on the pattern of usage of the particular system.

3.6.2 The need for extensions to the T–Box

We propose extensions to the definition of terminological knowledge beyond that in the T–Box, based on two observations:

1 it is not possible, certainly within a medical system, to explicitly express and define all relevant terminological information. There is a need for the system to *know* some information which it cannot *understand*, particularly the primitives of the domain and some forms of subsumption;

2 it is critical that compositionality is constrained, permitting only descriptions of concepts which make medical *sense*. To achieve this requires some limited knowledge *about* concepts.

We shall deal with each of these in turn.

3.6.3 Primitives and the assertion of subsumption

The representation of many domains, and certainly medicine, will require the use of a large number of primitive concepts, such as 'Parkinson's disease', and 'femur'. Such concepts have no simple definition. A lot is known about them but it is hard to give a definition that does not either adopt an arbitrary perspective, or require a great deal of detailed biomedical knowledge. If such a concept is represented as a primitive, without a definition, it cannot be classified which seriously limits the utility of the system. A user may know and understand what sort of thing the concept is, but for reasons the user can not or does not wish to express explicitly within a model. Hence as a minimum it will be necessary to allow the user to assert subsumption. This was a clear conclusion from experience with the knowledge representation scheme NIKL and its derivatives [Haimowitz 1988, Doyle 1989]. It is also quite obvious when working with any medical coding and classification scheme. There is in fact limited support for this within KL–ONE through the use of primitive specialisations, but there is little doubt that the use of asserted subsumption needs to be made more general.

There is a closely related issue which is the idea of a description being always true of a concept but not essential to its definition. For example we may wish to assert that all cancers are severe,

---

2 The term C–Box (conceptual) is suggested to describe such an extended T–Box [Rector 1992]

with the same strength of belief as if this were part of the definition of cancer, but at the same time not wish to require that a concept be defined as severe before we could conclude it was a kind of cancer. We are saying something about cancer in a very strong sense but not going as far as placing it in its definition. In particular this assertion should:

1 be indefeasible – mild cancer is always a contradiction;

2 contribute to the classification of the concept – cancer is a kind of severe disease.

Referring back to Woods' telephone, it is now possible to assert that 'telephones are black', but this is universally quantified. All telephones are black without exception. This can be thought of as asserting that telephone and black–telephone are the same thing. This type of assertion will be considered in detail when describing the Structured Meta Knowledge formalism.

3.6.4 Constraining the system to representing only 'sensible' concepts

The T–Box of KRYPTON provides a mechanism for expressing the definitions of concepts and arranging those concepts in a taxonomic hierarchy based upon subsumption. However the T–Box does not place constraints on which compositions are permitted. These constraints are essential to the type of medical system we shall describe, and are the key to a generative as opposed to a merely compositional system. The goal in relation to large medical terminologies is to provide a parsimonious means of expressing the knowledge that provides those constraints. We wish to provide relatively few terminological facts from which the system can infer the existence of many concepts. These terminological facts are assertions, but of a limited type. They are statements about what it is sensible to say, not what is true in general of a concept.

Obviously there is a grey area here and distinguishing those facts which are terminological in nature from those which are incidentally true is a matter of judgement. The distinction depends on what is needed to recognise whether or not a definition is sensible. For example we know that 'bones can fracture' and thus to speak of a 'fracture of the humerus' makes sense, whereas a 'fracture of penicillin' does not. However an explicit enumeration of all sensible concepts is not possible. We therefore seek to capture the generalities of the underlying shared medical model, and enumerate only the exceptions. Much of the utility of SMK is derived from its use of this type of knowledge.

3.7 The functional approach to knowledge representation

We now move on to a separate but related issue. As well as making the T–Box/A–Box distinction, the work on KRYPTON adopted a functional approach to knowledge representation [Brachman 1985]. Experience has shown that even when the interpretation of a data structure is clearly stated as being definitional (a T–Box), major problems can remain. System developers have specific problems they wish to solve and terminological languages have limited capabilities. Thus there is always a temptation to extend the interpretation of the data structures in an attempt to work around those limitations. The response to these 'abuses' was to avoid letting users near the data structures in the first place. This was done by describing the system in *functional* not structural terms. This functional approach to knowledge representation places the emphasis on *what* a system does and not *how* it does it.

3.7.1 Examples of the 'misuse' of data structures

Experience in developing the SMK formalism within the PEN&PAD programme supports the need for a functional approach. In the earlier stages SMK suffered from several T–Box/A–Box confusions, but more subtle 'misuses' occurred. A typical example arose from the question 'how many kinds of something are there in a system'. To rephrase Brachman's example of 'how many kinds of rock are there?' [Brachman 1983a], we shall ask the same question of pneumonia.

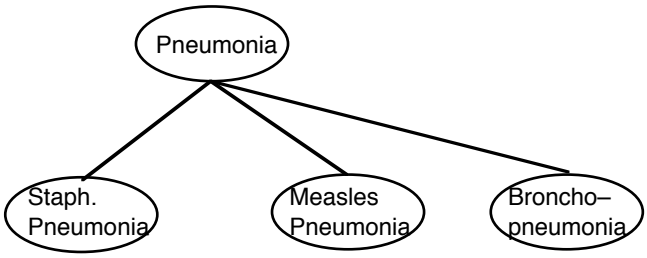

Figure 3.3 How many kinds of pneumonia are there?

Based on the graph in figure 3.3 the answer to the question would seem to be three. However this answer is based on viewing a graphical data structure and applying a particular interpretation. If we are considering a compositional system, and it is possible to describe the severity of diseases and whether or not lung conditions are bilateral, then there is no reason why the graph could not contain further types of pneumonia (figure 3.4).

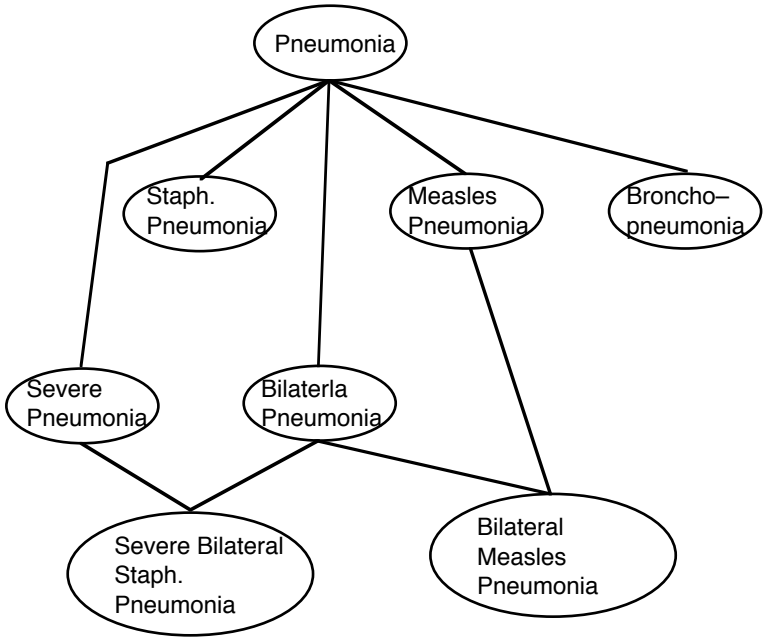

Figure 3.4 More types of pneumonia

Likewise the process could continue to cover more and more pneumonias. The point is that the answer to the question should not be derived by simply counting the nodes in a graph. Compositionality may imply the existence of many more concepts than have been explicitly placed within the graph.

Exactly this problem arose within the development of the PEN&PAD clinical workstations. How do you answer the question 'what are the kinds of pneumonia' in order to produce the contents of a menu offered to a user? Consideration of such problems has produce a view of interacting with the terminological system which is better put as 'what can I go on to say' rather than the less clear form 'what are the kinds of'. The view of the system shifts from one of a data structure to one of a descriptive tool.

3.7.2 The functional approach: defining systems by their operations

In KRYPTON the response to such problems was a move away from nodes, arcs, and frames to descriptions made purely in terms of the operations which can be performed, giving the user no direct access to the data structures used to represent the information [Brachman 1985]. KRYPTON is defined in terms of three types of operators:

1 Compositional operators such as **ConjGeneric** described earlier.

2 **TELL** operations which add to the knowledge in the system.

3 **ASK** operations which inquire of the system.

For example there is a **TELL** operation to define a symbol in some knowledge base (KB) producing a modified knowledge base (KB*)

$$\textbf{TELL}: \text{KB} \times \text{SYMBOL} \times \text{TERM} \rightarrow \text{KB}$$

$$\textbf{TELL}: \text{KB} \times \textit{bachelor} \times (\textbf{ConjGeneric}\ \textit{unmarried–person man}) \rightarrow \text{KB}^*$$

An example of an **ASK** operation is the test for subsumption, **ASK1**

$$\textbf{ASK1}: \text{KB} \times \text{TERM} \times \text{TERM} \rightarrow \{\text{yes, no}\}$$

$$\textbf{ASK1}: \text{KB} \times \textit{man} \times (\textbf{ConjGeneric}\ \textit{unmarried–person man}) \rightarrow \text{yes}$$

There are other operations defined for KRYPTON. Many return answers other than yes or no, and some are operations on the A–Box.

Defining the operations in this way does not define their interpretation by the system, but it provides a helpful level of abstraction. This approach will be used to define the basic operations required of a terminology system prior to describing the interpretation of those operations according to the theories of SMK.

In concluding this section it is interesting to note anecdotally that simple coding schemes are not immune from these difficulties. Access to the structures of a coding scheme has been known to result in the code for 'tattoo' being edited to 'tat', meaning 'tired all the time'. Obviously the implementation had provided TELL as well as ASK operations.

3.8 Summary of relevant issues in knowledge representation

The main points from this chapter or listed below:

1 Techniques exist for representing the relationships between concepts using a variety of data structures.

2 Such structures can be interpreted as either intensional definitions or statements about concepts.

3 This distinction between terminological and assertional knowledge is embodied in the T–Box/A–Box division within knowledge representation systems.

4 Terminological knowledge is of most relevance to handling medical terminologies, but requires extending to include some forms of assertions in order to satisfy important requirements.

5 The functional approach to knowledge representation describes a system by the operations it supports and not the data structures it uses. This provides a useful level of abstraction for stating requirements.

Chapter 4 A Functional Description of a Medical Terminology System

Chapter 1 considered the current approaches to handling medical terminologies through the use of coding and classification schemes. This has identified important problems with these schemes such as absent or inadequately constrained compositional features, and rigid, ill–defined hierarchical relationships. However despite these technical shortcomings they do represent one of the most complete and organised statements of what a medical terminology should cover and provide a yardstick against which we can measure new approaches.

Chapter 2 examined issues in knowledge representation that are relevant to the representation of medical terminologies. The focus was on the representation of terminological knowledge that was characterised by looking at examples from several knowledge representation schemes. However terminological knowledge, as strictly defined for example by Brachman in the work on the T–Box, was seen to be too limiting and required extending for use in medical terminology systems.

The strategy underlying the development of the medical terminology system based upon Structured Meta Knowledge (SMK) has been to see the coding and classification schemes as representing the problem, and the field of knowledge representation as offering possible solutions. SMK is trying to tackle the similar problems as coding and classification systems, but it is adopting a different, hopefully more sophisticated, set of solutions. The goal is a principled semantic representation scheme for medical concepts which captures the essence of the strong underlying medical model. Traditional classification schemes such as ICD–9 are medically rich but representationally impoverished. The medical knowledge that went into their creation has been lost or locked away inside simple structures. It is essential that those structures are opened up and the knowledge represented explicitly.

In this chapter we shall describe a **medical terminology system** based on the ideas surrounding SMK but by no means exclusive to it. These ideas are quite straightforward, but the emphasis is on clarity. Mindful of the problems that can arise by focusing on the structural aspects of knowledge representation, we shall try to provide a functional description of a basic terminology system following the style outlined in the previous chapter. This is intended to act as a bridge between the world of problems and that of computer–based solutions.

4.1 Motivations for a functional description

The phrase 'terminology system' has been chosen to emphasise that we are considering an artefact that supports functionality, and actually *does* something. This notion is closely allied to the use of computers to provide that functionality, and its is a natural way to speak of a computer system. SMK is implemented as a large and complex computer programme. It interprets the terminological facts in a particular model, and on that basis it provides answers to questions. It is possible to reproduce on paper all the terminological facts the system uses to draw its conclusions, but it would be quite impossible for a person to draw all the same conclusions by reading those facts. The data in an SMK terminology model is not in a form which is directly useful without a computer–based interpreter.

4.1.1 The changing nature of coding schemes and terminologies

Until recently a coding had to be suitable for unaided interpretation. Thus a coding scheme does not *do* anything – it just *is*. However coding schemes have been evolving rapidly, and the situation has changed. There are three arguments in favour of a functional perspective on what

is required to deliver a medical terminology. All three relate to the inescapable fact that the data structures of the scheme must be formally interpreted:

1 All major coding and classification schemes are now intended primarily, and often exclusively, for use within computer systems. This is illustrated by comments by developers and users such as "the codes runs very slowly", or "its lists of choices are too long", or "it is difficult to find things". Clearly these are not criticisms of a data structure *per se*. They indicate that the view of the coding scheme is inextricably bound up with that of some computer–based system.

2 Traditional schemes are increasing in complexity with the introduction of modifier codes, qualifiers of certainty, and multiple hierarchies. This makes them increasingly dependent upon interpretation by a computer–based systems. However it makes no sense at all for the choice of interpretation to be left to the individual system designer. This can lead to incorrect or incompatible interpretations, which would at best undermine prospects for communication and systems integration, and at worst be medically wrong.

3 The features provided by a computer–based terminology system are intended to form part of larger information systems, such as a medical record system. For this to be possible the larger information system must know what it can expect in functional terms from the terminology. It is interested in *what* it does and not *how* it does.

Traditionally the test of a coding and classification scheme has been whether are not *people* find it sensible and useful. This is now no longer the case. The test is whether there is an interpretation by a machine of the knowledge in the scheme which results in the *machine* being judged to be sensible and useful by a person or another machine. This statement may sound trite, but the failure to recognise this fundamental shift in the relationship between user and scheme is, we believe, at the heart of many of the current tangles over the development of medical terminologies.

4.1.2 Experience with the development of SMK and the need for a functional description

There have been several recurring themes and problems during the development of SMK which, at their heart, have been about determining what it is supposed to be doing. Early in the development there were no clear distinctions between the data structures represented as a network of objects, the interpretation of those structures by the computer, and the provision of functionality to external agents based on that interpretation. These three layers are sketched in figure 4.1, as the 'Terminological Black–box'. The recognition of the need for these layers was an important point in the development of SMK. The subject of this chapter is the first layer, the functional description.

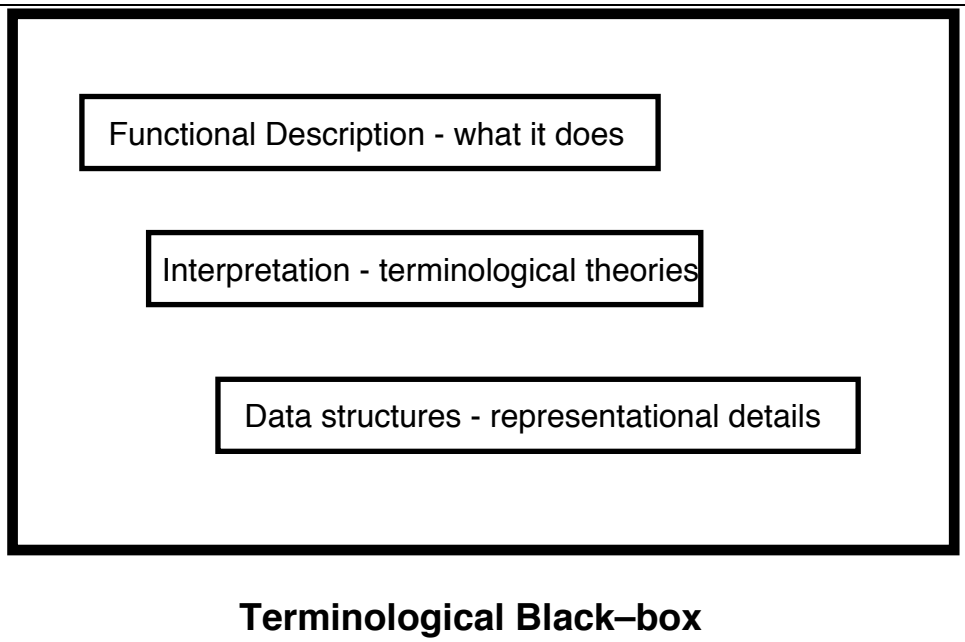

Figure 4.1 The three conceptual layers of the terminology system

4.2 A Functional Description of a Terminology System

We shall now give a functional description of a basic terminology system based on the ideas incorporated in SMK, but by no means exclusive to it. At this stage it is intended to be independent of the precise formal interpretation of the functions described, and certainly it is hoped to be independent of structural details. It must be emphasised that at one level whether the terminological black–box contains codes, arcs, nodes, frames or whatever is irrelevant. How things are done becomes important when we consider the practicalities of building an interpreter and constructing a useful, large, and consistent terminological model which meets the functional description. However at each stage illustrative examples will be given which relate to both traditional coding systems and more sophisticated compositional approaches, with the introduction of some examples showing the style of SMK.

In trying to produce a functional description there is the question of the appropriate level of abstraction. We could at the highest level state that our system must support the relevant medical tasks for which it is intended. This of course is the ultimate evaluation of its usefulness but is insufficiently precise for our purposes. It would amount to requiring the system to posses intuition. On the other hand just making available a series of primitive operations on data structures such as nodes and arcs defeats the whole purpose of the approach. A more appropriate level views the system as providing certain useful terminological functions which relate to those we could expect of a coding and classification scheme interpreted by an experienced user, but it carries them much further in scope and specificity.

The functional description is intended to be a basic description or set of requirements. It will later be used as the framework for presenting the interpretation of those functions embodied in SMK, that is the rules and constraints which comprise the underlying theory. The details of the interpretation are not presented in this chapter.

4.3 General issues: compositionality, constraints, generativity, and parsimony

Any questions which can be asked of a terminology system are with respect to some particular set of concepts it represents and the relationships amongst them. The representation of a system of concepts we shall call a model. The word entity will be used to mean the representation of a single concept within a model. The terminology system can be thought of as embodying a model of terminology and a user can refer to concepts within that model, ask questions about those concepts and their relationships to each other, and also add to what is

known about them. There are several cardinal features that are required of the system. It must be:

Compositional

It must be possible to form concepts by combining or relating two or more concepts together. The result of a composition must itself be a entity and compositionality must be recursive ie. there can be compositions of compositions.

Constrained

It must be possible to constrain which compositions are considered sensible, and the system should only permit those sensible compositions to be formed. The inverse view of a constraint is that it is necessary to sanction compositions before they are considered sensible.

Generative

The system should not only be capable of deciding if a composition is sensible, it should also be able to infer or generate the set of all sensible compositions based on the constraints. Note that this is a capability of the system itself, and not merely that someone or something could generate compositions if they wished. Generativity is dependent upon compositionality but is a much stronger requirement.

Parsimonious

This requires that the number of constraints which need to be given to the system is few compared to the number of compositions the system can generate. The whole exercise would have been futile if it was necessary to enumerate all sensible compositions in order to provide the constraints. The goal is to capture in the generalities underlying the strong model of medical of terminology and use inference to generate the compositions. Note however that generativity is a prerequisite to parsimony but is not sufficient. If there is no strong, regular underlying model of medicine, or it is ignored, then parsimony is impossible, and the system will degenerate to an enumerative one.

Most coding and classification schemes are not compositional, the exception being SNOMED, but even here the result of a composition is not a 'code' and hence it cannot be fully and recursively compositional. Enumerative schemes are trivially constrained but are not generative and hence not parsimonious. SNOMED is essentially unconstrained and thus generativity and parsimony as defined above are absent.

4.4 Operations on the terminological system

We shall now consider the fundamental operations required from the medical terminology system. The descriptions of these operations will be quite informal. The main concern is for clarity and not definitional rigour, The aim is to set the background for the discussion of SMK in subsequent chapters. We shall consider three broad types of operation to:

– compose an expression that refers to an entity within the system;

– ask a question of the system;

– add a constraint or sanction by making a terminological statement

To these we shall add operations which relate the formal meaning of expressions and operations to an external interpretation as a phrase or sentence in 'natural' language. This is the means by which a user can elucidate the meaning of what the system is saying and judge whether or not it makes sense.

4.4.1 Expressions and compositional operators

The first requirement is for a language with which to compose expressions that refer to entities within the model. There are two components to the compositional language

– identifiers which refer to atomic, primitive entities within the terminology model

– compositional operators for combining atoms to form complex expressions which refer to structured entities within the model

Most simple coding schemes comprise only identifiers of atoms and provide no compositional operators. The set of possible expressions is thus defined by the valid code numbers of the terms in the scheme. Thus for example 'A1234' could be thought of as a trivial expression referring to an entity (term) in the Read Clinical Classification.

In a compositional system there have to be operators which combine entities together. SNOMED is compositional but there are no formal operators for combining individual codes into a SNOMED term. In knowledge representation systems, for example KRYPTON's T-Box discussed earlier, there are compositional operators such as VRGeneric and ConjGeneric. In SMK the principle operator is which:, used to form expressions such as

Fracture which: hasLocation *Humerus*

intended to mean 'fracture of the humerus'.

It is essential to recognise that, in this description of a terminology system, the use of an expression does not assert the existence of the concept to which it ultimately refers. Expressions form the arguments to operations that ask questions of the system or add to its knowledge. An expression may or may not refer to an entity within the model but simply using the expression has no effect at all on that model.

4.4.2 Operations which ask questions of the system

Given the notion of expressions in the language we can now go on to look at a series of operations which ask questions of the system. These add no new knowledge to the terminology model (TM), and the system is left unchanged by such operations. There are three principle operations covering well–formedness, equivalence, and subsumption. Each is of the form

operation?(TM, <expression>) → {yes, no}

The use of a question mark (?) indicates that an operation is interrogatory.

4.4.3 Well–formedness

The system must be able to determine if an expression corresponds to an entity that is consistent with the model. If for example there are type constraints placed on which combinations of entities are allowed then one would be asking if a particular combination met those constraints.

For any expression the system can be asked if with respect to some terminology model, TM, that expression is well–formed or ill–formed

well_formed?(TM, <expression>) → {yes, no} (1)

Again it must be emphasised that the operation **well_formed?** is not asserting that the expression is well–formed. It is merely asking if it is well–formed, and the answer will depend on the constraints in the model, and the rules for interpreting those constraints.

Well-formedness is a trivial notion for simple coding schemes which contain only atoms (terms). The operation corresponds to testing directly if the atom is one of those about which the system has been told. In systems which have a compositional style, most notably SNOMED, it is the absence of such an operation which creates serious problems and results in the potential for nonsensical combinations.

The theory for well–formedness in SMK is a major aspect of the formalism. Its goal is that if we have a model TM1 which knows for example that fractures occur in bones, the system should be able to conclude the following

well_formed?(TM1, *Fracture* **which** *hasLocation Humerus*) → yes

and

well_formed?(TM1, *Fracture* **which** *hasLocation Penicillin*) → no

All other operations to be described are only defined for expressions which are well–formed[3].

4.4.4 Equivalence

It is possible for several different expressions to mean the same thing medically. For example if we could compose an expression roughly interpreted as 'heart attack of the heart' then we would like this to be equivalent to 'heart attack'. Likewise 'fracture of a long–bone in the humerus' means the same as 'fracture of the humerus'. In a terminological system we will wish to resolve problems of tautology, redundancy, and other variants.

For any pair of expressions it can be asked if they are equivalent under interpretation by the system i.e. do they correspond to the same entity

equivalent?(TM, <expression1> <expression2>) → {yes, no} (2)

For simple coding systems this corresponds to a trivial test of whether or not two atoms are identical.

In compositional systems it implies the existence of rules or theories for transforming any expression into a canonical form. Two expressions are equivalent if they transform to the same canonical form. Within SMK the theories of the canonical form are some of the most important and complex in the interpretation.

4.4.5 Subsumption

In a terminological system subsumption (the is a kind of relationship) is the single most important relationship between two entities. One entity subsumes another if by necessity it is more general than the other. It can also be understood as a subset relationship between the properties of the former and the latter. For example the entity corresponding to the idea of 'cough' subsumes that for 'severe cough' because the latter is a cough which is also severe. The extensional interpretation of subsumption corresponds to statements such as

'all cases of lung cancer are also cases of cancer and cases of lung disease'.

For any pair of expressions it can be asked if the former subsumes the latter

subsumes?(TM, <expression1> <expression2>) → {yes, no} (3)

Subsumption permits the derivation of a subsumption hierarchy, which represents a partial ordering amongst entities.

Traditional coding and classification systems are on shaky ground over the question of subsumption. The classificatory relationships generate some form of hierarchy though this is not based on any straight forward or uniform notion of one thing being 'a kind of' another. As discussed in chapter two it is usual to find numerous 'classification rules' acting together to form groupings that are medically complicated. The hierarchical relationship is closer to that of 'broader than' and 'narrower than' as used in bibliographic thesaurii. The problem with classification schemes is aggravated by the use of a single hierarchy with a fixed number of levels.

In simple classifications all hierarchical relationships are asserted. In compositional systems the question is what if any is the relationship between the structure of entities and their subsumptions. This is an important aspect of terminological knowledge representation schemes, but does not at present form any part of schemes such as SNOMED.

---

[3] In the current implementation of SMK the test for well–formedness is implicit in the system, that is in general presenting the system with an ill–formed expression is an error.

The rules for determining subsumption between entities in SMK are very important, and the goal here is answers to questions of the form

subsumes?(TM, *Fracture*, *Fracture* **which** *hasLocation Humerus*) → yes

i.e. a fracture of the humerus is a kind of fracture.

4.4.6 Decompositional and generative operations

All the preceding operations have had simple yes/no answers. These are the essential operations and have been the main focus of the work but clearly others will be required. These operations will produce answers other than yes or no and are of the form

operation?(TM, <arguments>) → set of <expression>

They are broadly of two kinds:

1 decompositional operations which enquire as to what is known about an entity other than subsumption, based upon its composition and relations to other entities eg.

of?(TM, <relationship> <expression>) → set of <expression (4)

For example if we ask what is the location of a 'fracture of the humerus' we would expect the answer 'humerus'.

2 generative operations which produce entities based on the idea of what can be said that is represented by the constraints on the system eg.

generate?(TM, <relationship> <expression>) → set of <expression>(5)

For example we could ask in what ways it is possible to further describe a fracture and one answer may be by its location. Going on from this we may then ask what are the possible locations for a fracture and we would expect to be provided with a variety of bones.

Such operations have no counterpart in traditional schemes. A traditional scheme can only report back what it has been explicitly told. These operations in SMK underpin its use in supporting predictive data entry.

4.4.7 Operations which add knowledge , constraints, and sanctions to the system

The next set of operations are those which add to the terminological knowledge within the model. These operations change the model and therefore the answers it gives to questions. They are quite different to the ask operations. These operations are of the form

operation(TM,<arguments>) → TM*

The result of this operation is to change the model TM into a new model TM*.

Three main operations will be described covering the creation of atomic entities, the assertion of subsumption, and the assertion of a non–subsumptive terminological relationships.

4.4.8 Creation of atomic entities

Many entities in a terminological model have no sufficient definition. For example it is difficult to come up with a definition for arm which did not adopt a single fixed view of the concept or required an inappropriate degree of anatomical detail. Such concepts are simply given to the system as atoms. Thus it must be possible to tell the system of a new concept, identified by some suitable unique identifier

atom(TM,<identifier>) → TM* (6)

This is one of the two principle operations for constructing a straightforward coding scheme. It corresponds to adding a new term. However, such schemes are merely enumerative. SMK starts with atomic creation but builds on this through composition.

4.4.9 Defining a conventional subsumptive relationship

It is necessary to be able to assert a subsumptive relationship between two entities.

is_kind_of(TM,<expression1> <expression2>) → TM* (7)

These means that by convention one entity is a kind of another, but the reasons for this relationship are not represented explicitly in the model.

This has to be done when creating an atomic entity if it is to minimally relate to anything at all. With this in mind we can thus define a variant of **atom** combined with **is_kind_of** which creates and places a new entity as a kind of some other entity referred to by an expression

atomic_kind_of(TM,<identifier> <expression>) → TM* (8)

This is akin to the creation of a primitive type as a kind of some other concept in KL–ONE.

Traditional coding schemes are confined to facts related to hierarchical relationships/conventional subsumption, within the provisos discussed earlier. In compositional systems it is possible in principle for subsumption to be derived formally from the structure of entities and there is less of a need for conventional subsumption. The inferring of subsumption is one of the major benefits to be gained from a structured, formal approach. There will however remain situations when a formal model of subsumption is either impossible or unnecessary and here conventional subsumption can be used. Its use need not be confined to the definition of new atoms.

4.4.10 Non–subsumptive terminological statements

There are many other possible relationships between entities other than subsumption. For example we can speak of fractures being located in bones. The danger of course is of being drawn into general issues of knowledge representation, and strict limits need to be placed on what sorts of facts are permitted. The model will be confined to terminological statements which are essential to determining whether or not compositions are well–formed. For example in order to conclude that 'fracture of the humerus' is sensible we have to know something like 'fractures occur in bones' and that the 'humerus is a bone'. The subsumption relationship can tell us that the humerus is a bone. We are concerned here with the former statement.

Statements such as 'fractures occur in bones' are the explicit constraints or sanctions to be placed on expressions so that the system can conclude whether or not they are well–formed. Hence the precise form of these terminological statements is closely related to the theory of well–formedness within the system. Traditional coding schemes have no equivalent of these statements.

In SMK these terminological statements are made as sanctioning statements. The assumption by the interpreter is that nothing is sensible unless knowledge can be found which say it is. These are called 'possibility' statements and correspond to operations in schematic notation of the form

possible(TM,<expression1><relationship><expression2>) → TM* (9)

for example

possible(TM,*Disease hasSeverity Severity*) → TM*

In this notation <relationship> refers to an entity which can relate together two other entities. The example can be interpreted as the statement 'it is possible to talk about diseases having a severity'. Note that possibility here has nothing to do with uncertainty. We are quite certain that diseases can have a severity. The choice of 'possible' is perhaps a little unfortunate, and sensible may be a better choice, but its use is deeply ingrained in the culture associated with SMK.

The use of these terminological statements is a complex issue in SMK and will be discussed at length in subsequent chapters.

4.5 External interpretation of entities

So far we have considered formal operations which manipulate entities within a model, and later we shall be analysing the various formal conditions which the model must satisfy. However the goal of a terminology system is not just to produce a formally self–consistent model. It must give a user some purchase on practical problems. Hence, expressions in the formal terminological language and the operations on the system must be capable of external interpretation, usually through the use of natural language expressions.

Thus in the functional description of the system we shall place a requirement that it can produce language phrases capable of external interpretation. These languages phrases need not be elegant prose, but they must be adequate to complete the relationship between the users concept, the representation of that concept as an entity in the model, and the interpretation of that entity by the user.

4.5.1 Entities and phrases

We shall at the moment confine our system to producing language phrases to represent entities to the user, and for a terminological system they will always be noun phrases.

For any expression in the terminological language the system can produce an external representation as a noun phrase

phrase?(TM, <expression>) → <phrase> (10)

For example we could expect noun phrases such as 'fracture of the femur' or 'malignant melanoma'[4].

For a simple coding scheme the external phrase corresponds to the rubric of the term. For the moment will shall put aside the complications of synonyms and alternative natural languages. For a simple scheme, CCS, we may have

phrase?(CCS, A1234) → 'pneumonia'

The phrases for atoms must be given to the system. A compositional system is further required to generate a phrase based on the precise form of the expression. This will depend upon the phrases for the atoms and the compositional operators. For example in schematic form

phrase?(TM, *Fracture* which: hasLocation *Humerus*) →
'fracture which has location humerus'

4.5.2 Operations and sentences

The responses to the interrogatory operations performed on the system can be thought of as sentences being proposed by the system. For each operation the system is asked to perform there should be a corresponding sentence based upon the operation, its arguments, and its result.

sentence?(<operation>) → <sentence> (11)

The choice of words is not particularly important. The key point is that there should be some formal and uniform set of principles for producing an external interpretation for what the system is saying. Example interpretations for the three main operations are:

well_formed? : <phrase> {is, is not} **a valid concept**

equivalent? : <phrase1> {means, does not mean} **the same as** <phrase2>

[4] a cancerous tumour of the pigment forming cells in the skin

subsumes? : all <phrase2> {are, are not} **a kind of** <phrase1>

Thus for example if we have the following

well–formed?(CCS, A1234) → yes

and

phrase?(CCS, A1234) → 'pneumonia'

then we could expect the system to produce a sentence of the form

sentence?(**well–formed?**(CCS, A1234)) → 'pneumonia is a valid concept'

4.6 External evaluation of the terminology system

4.6.1 Tests of sensible behaviour

The overall test of the terminology system is whether or not its answers to questions make sense to an appropriate user. We can imagine a pseudo–operation with respect to some medical domain D which determines whether or not a sentence produced by the system is sensible. For example a qualified user reads it and makes a judgement. For the current discussion we shall say that as a result of such an external test any sentence is either sensible or nonsensical.

sensible_statement?(D, <sentence>) → {sensible, nonsensical}

For example if CCS has been compiled correctly then we should expect

sensible_statement?(Medicine, 'pneumonia is a valid concept') → sensible

This is not of course an operation we perform on the terminology system – it is a 'thought' operation. For example 'malignant melanoma' and 'fracture of the humerus' are judged, by some suitably appointed person, to be medically sensible phrases, while 'benign cancer' and 'fracture of the blood pressure' are nonsensical.

Having emphasised the need for sentences as links to external tests it is now convenient to omit this explicit stage when deciding if the result of an operation corresponds to a sensible terminological idea. Furthermore we shall assume that the domain is some appropriate medical domain. Hence we shall combine the terminology system operation

sentence?(<operation>) → <sentence>

with the evaluation operation

sensible_statement?(D, <sentence>) → {sensible, nonsensical}

to give a single evaluation operation

sensible?(<operation>) → {sensible, nonsensical}

This translates to the question 'does the answer provided by the system in response to the operation correspond to a sensible thing to say within the domain'. This is the obvious question and the proceeding explorations may appear to have been rather long–winded. They were done however to make quite explicit what the system can and cannot do. Suggestive naming conventions make it easy to read too much into expressions and operations.

4.6.2 External interpretation of the terminology system: correctness and completeness

The basic operations outlined above allow a preliminary definition of what is required for a terminology system to be considered correct and complete with respect to the domain it is intended to represent, that is what it means to be a sound representation of the terminological ideas it is supposed to model.

Correctness

Every possible operation which can be performed by the system for any set of arguments should correspond to a sentence which is judged by an external test to be sensible. This is the test that the system is correct.

Completeness

All possible sensible sentences should be produced by the system. This is the test that the system is complete. There are limitations on this requirement:

1 The requirement is only placed on those operations that are consistent with the model i.e. the system cannot be expected to enumerate all those things which are not true in medicine.

2 There must be an externally defined notion of all the sensible sentences

The prime concern is completeness with respect to well–formed expressions. A model intended to cover a domain should be capable of generating well–formed expressions representing all the concepts in that domain. 'All medical concepts' is an empirical notion. It might for example rely on expert opinion or alternatively, it might be based on corpora such as medical records, bibliographic material, or standard medical nomenclatures.

It is less clear what it meant by 'all medical subsumptions'. It is difficult to imagine the task of determining all the *is kind of* relationships in a domain. If experts were capable of completing this task it may not be necessary to develop formal techniques. Clearly however the system has to be sufficiently complete, for example as tested against existing classifications and for purposes of medical audit. Similar considerations apply to the operation testing equivalence.

The requirements for completeness and correctness with respect to well–formedness is captured by the phrase 'all and only sensible medical concepts should be represented by the system'.

4.7 Summary of chapter

A functional approach has been used describe the requirements for a medical terminology system. The emphasis is on what the system does, that is its functionality, and not how it does it procedurally. Distinctions were also made between the functional description of what the system does, the interpretation of those functions within the system, and the underlying data structures.

The functional description was presented as a series of operations on the system covering composition of expressions, the asking of questions, and the telling of knowledge.

Finally consideration was given to the external interpretation of the system and the requirements for correctness and completeness with respect to the medical domain it is intended to model. The principle requirement is for 'all and only' the concepts of the domain to be represented.

Chapter 5 The Structured Meta Knowledge Formalism (SMK) and Its Satisfaction of the Requirements From the Description of a Terminology System

In the previous chapter we outlined a functional description of a basic medical terminology system. In this chapter we shall show how that description is satisfied by the theories embodied in the Structured Meta Knowledge formalism (SMK). The style will be semi–formal. It concentrates on conveying the essence of what SMK is trying to achieve rather than on rigorous proof or exhaustive definition.

Before beginning the account, it is important to understand what is being described. SMK is a formalism and not a specific model of medicine. The examples used will refer to specific concepts such as Cough and hasSeverity, but these are only examples. The concept Cough is not a part of SMK per se, it belongs to some example model represented using SMK. There are a handful of fundamental primitives within SMK with names such as *TopThing* , but these represent basic properties of the formalism and are not specific for medicine.

Organisation of this chapter

The sections of this chapter relate to the functional description of a terminology system presented in chapter 4 as follows:

Section 5.1 deals with the representation of concepts as entities and section 5.2 with the definitions of entities. These two sections address the compositional operators and the test of equivalence (sections 4.4.1 and 4.4.4).

Section 5.3 deals with subsumption, both conventional (asserted) subsumption and the rules for determining formal subsumption based upon the definitions of entities (sections 4.4.5 and 4.4.9).

Sections 5.4 and 5.5 explains the use of triples to represent non–terminological statements. These are part of the basis for sanctioning those expressions that are well–formed (section 4.4.10).

Section 5.6 deals with the rules for canonising descriptions (criteria) and testing the coherence of criteria sets (contradictions, tautologies, etc). This is part of determining well–formedness.

Section 5.7 presents the consolidated requirements for well–formedness based on the sanctions described in sections 5.4 and 5.5 and the coherence tests of section 5.6 (section 4.4.3).

Section 5.8 deals with the naming of entities and surface linguistics (section 4.5).

Section 5.9 considers decompositional and generative operations (section 4.4.6).

Section 5.10 then summarises the functional description and its satisfaction by SMK.

5.1 The representation of concepts in SMK: Entities

SMK is a compositional system for the representation of medical concepts based on structured descriptions. The name SMKobject is used to refer specifically to the representation of a concept within SMK. In this section we shall outline the basic characteristics of SMKobjects , and the principles of structured descriptions in SMK

5.1.1 Entities and relationships

SMKobjects within SMK can be divided into two fundamental kinds which we shall call entity and relationship:

- an entity corresponds to a node in a semantic network or a frame in a frame–based language;
- a relationship corresponds to an arc in a semantic network or a slot/role in a frame–based language.

5.1.2 Elementary SMKobjects: elementary entities and attributes

SMKobjects may be elementary (primitive) or complex (composite). This is true for both entities and relationships and hence there are two kinds of elementary SMKobject:

elementary entity - atomic concept eg. *Cough*

attribute - elementary relationship eg. *hasSeverity*

Elementary SMKobjects have no sufficient definition within a model. They can be placed in relation to other concepts but their existence is simply asserted. Elementary SMKobjects are distinguished from other entities by the assigning of a unique identifier. Hence the two schematic operations for defining new elementary SMKobjects according to their identifiers

new_elementary_entity: <identifier>

new_attribute: <identifier>

In principle an identifier can be of any suitable form, as long as it is unique. However for ease of reading we shall use meaningful character strings (symbols) such as *Fracture* and *hasLocation*. An italic font indicates an SMKobject, with an uppercase initial letter for an entity and a lowercase for an attribute. It must be emphasised that the words used to form the identifier contribute nothing to the formal meaning of a entity.

Attributes are used to describe relationships between pairs of entities. All relationships within SMK are potentially invertable, and thus every attribute has an inverse, for example *hasLocation* and *isLocationOf*. The two kinds of elementary entity are shown diagrammatically in figure 5.1

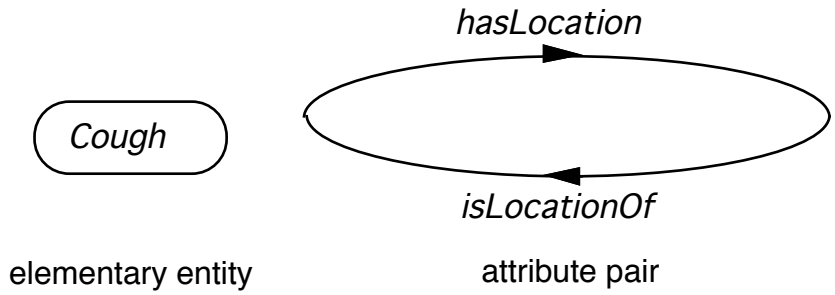

Figure 5.1 Elementary SMKobjects – entity and relationship

The definition of an attribute as an elementary relationship may appear slightly odd. We speak of them as being used to describe pairs of entities, but within SMK they are much more than labels on arcs. Attributes and all relationships are first class objects like entities, and must for example form a hierarchy. An attribute in SMK is akin to an 'non specialised' relationship, that is relates 'something' to 'something'. It is only distinguished from all other such attributes by the use of an arbitrary identifier. This is in complete analogy with an entity.

5.1.3 Complex entities and expressions: prototypes and the role of criteria

The definition of a new complex entity is formed by the addition of descriptions to the definition an existing entity. SMK has a straightforward notion of a description which corresponds to an attribute–entity pair. This attribute–entity pair is called a criterion.

<criterion> : <attribute>–<entity>

Examples of criteria are

hasLocation–Bone

hasCause–Virus

hasSeverity-Severe

Criteria are not in themselves first class objects. They form part of the structure of entities, although they have many of the properties of first class objects. For example subsumption is defined between criteria. They are important constructs in understanding the behaviour of SMK and specifying the rules of the formalism, most of which concern the manipulation of criteria.

We shall use the schematic operator **which:** to indicate an expression extending the structured description of an entity

<entity> which: <criterion>

For example

Cough which: *hasSeverity–Severe*

Pneumonia which: *hasCause–Virus*

These expressions refer to the intensional definitions of entities corresponding to the concepts 'severe cough' and 'viral pneumonia' respectively. The links here are to be interpreted structurally and they are in complete analogy with Woods' 'black telephone' as discussed in chapter 3 [Woods 1975]. A complex entity which corresponds to such an expression is called a prototype.

The **which:** operation may be repeated to define a more complex concept, for example

(*Cough* which: hasSeverity–*Severe*) which: *hasProgress–Worse*

or more concisely

Cough which:
hasSeverity–Severe
hasProgress–Worse

This corresponds to 'a severe, worsening cough'. It is an important principle of SMK that the order in which a description is created should not affect the final outcome[5]. Hence

[5] There are limitations on the current implementation which cannot always correctly resolve arbitrary orders of criteria if there are dependencies amongst those criteria.

Cough which:
hasProgress–Worse
hasSeverity–Severe

indicates the same entity as in the previous example.

It is important to note the difference between two obviously related but distinct prototypes, for example

Fracture which: *hasLocation–Humerus*

and

Humerus which: *isLocationOf–Fracture*

One is the converse of the other, but they are distinct. The former is a 'fracture of the humerus' and the latter 'a humerus that is fractured'. Although linguistically they connote similar ideas they are formally different , in that one is a kind of injury and the other a kind of bone. This distinction is very important in SMK and the failure to recognise this is a frequent source of confusion.

Note that SMK meets the requirement to be fully or recursively compositional, that is compositions can be used within compositions, for example

Humerus which: *isLocationOf–*(*Neoplasm* which: *hasBehaviour–Malignant*)

5.2 The canonical form of an entity and identity

So far we have seen that criteria can be applied sequentially to an entity to form increasingly complex prototypes. Some of the important theories of SMK concern the transformation of apparently differing expressions into the same entity. For example a 'humerus which is the location of a fracture of the humerus' is clearly the same as a 'fracture of the humerus'. The key to this is the definition of the canonical form for an entity. There are several difficult questions surrounding the definition of the canonical form which centre on the relationship between the use of a canonical form to determine equivalence and subsumption, and a more extended interpretation which takes into account aspects of the structure of the model which sanctions the well–formedness of an entity. Another aspect of this problem is whether the canonical form refers to a declarative definition of the entity or an expression involving operators in some concrete syntax (eg. which:) that is evaluated by some interpreter. We shall consider these questions in more detail later, but at the moment will shall concentrate on the definitional form which is sufficient to determine identity.

5.2.1 Definition of the canonical defining form and the determination of identity

Any entity, E, can be uniquely and sufficiently defined by a base type, B_e, which is always an elementary entity, and a defining criteria set, D_c

$$E \equiv E(B_e, D_c) \qquad (1)$$

The use of the subscripts denote that B_e is elementary and D_c is in its canonical form. For example

(*Fracture*, {*hasLocation–Humerus*, *hasSeverity–Severe*})

Every complex entity is thus derived by specialisation of some elementary entity. The base type represents the limit on what it is possible to define explicitly within the model ie. the base type embodies the indefinable criteria. The rules for canonising an expression, and in particular the criteria set, D_c, are in the main straightforward, though some aspects represent the more difficult corners of SMK. They will not be discussed until subsumption has been covered in more detail but it is worth pointing out here that the order of the criteria in the canonical defining criteria set is immaterial.

Note that an elementary entity can be thought of as being its own base type and having an empty defining criteria set. An elementary entity is defined by its identifier.

The canonical form is central to determining equivalence. If two entities have the same canonical defining form then they are identical, and SMK will only allow one such entity to be present in a model.

5.3 Subsumption

Subsumption is the most important relationship in SMK. As is the case with subsumption in other similar representations the relationship forms a partial ordering amongst entities. A system of entities can thus be thought of as forming a graph and we will speak of an SMK network, of which the backbone is the subsumption hierarchy. This hierarchy is a true multiple hierarchy.

Within SMK there is a requirement that for all SMKobjects there exists at least one other SMKobject by which it is subsumed. The only exception to this is the top entity, known by tradition as *TopThing*. No entity is permitted to subsume itself, so if the entities are viewed as forming a graph then this graph must be acyclic.

There are two mechanisms by which subsumption arises within SMK. It may be asserted, when it is called conventional subsumption, and it may be derived from the definitions of the entities, when is called formal subsumption. These shall be discussed in turn.

5.3.1 Conventional subsumption

This is the assertion of subsumption and is essential for placing elementary entities within the network. For example it can be asserted that the elementary entity *Fracture* is subsumed by *Trauma*, figure 5.2.

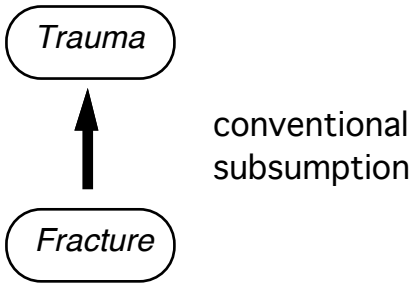

Figure 5.2 The assertion of conventional subsumption

Conventional subsumption represents the addition of knowledge to the network. The subsumption relationship could not have been derived by any process of inference over what is already known. However within SMK it is not restricted to the placing of new elementary entities. Both elementary entities and prototypes can be a conventional subsumer or subsumee. The use of conventional subsumption at any time other than when placing a newly defined elementary entity potentially changes the properties of existing entities and thus challenges the global coherence of the network. It would be extremely easy to produce contradictory or ambiguous situations, for example by asserting that 'severe cancer' was a kind of 'mild disease'. The definition of global coherence remains one of the least well developed areas of work on SMK. In the current style of usage of SMK the potential problems are alleviated by often banning or restricting the use of some of the constructs with, for example, conventional subsumption being usually limited to having a 'leaf entity' as the subsumee.

The schematic operation addSub: performs conventional subsumption[6]

Trauma addSub: *Fracture*

We shall denote the denote the test for conventional subsumption between two types by ≤c

$$E_1 \leq c\ E_2 \Rightarrow \{true\ false\} \qquad (2)$$

5.3.2 Formal subsumption

This is a preliminary discussion of formal subsumption. We shall begin by considering the relationship between the definitions of entities, before considering the effects of assertions.

5.3.3 Formal subsumption between criteria

The key to defining formal subsumption between entities is the definition of formal subsumption between criteria. The basic definition of subsumption between two criteria c_1 and c_2 is straight forward. If the two criteria are defined as

$$c_1 \equiv attribute_1\text{–}value_1$$
$$c_2 \equiv attribute_2\text{–}value_2$$

then subsumption denoted by ≤ is defined by

$$c_1 \leq c_2 \text{ if } (attribute_1 \leq attribute_2) \text{ AND } (value_1 \leq value_2) \quad (3)$$

It amounts to whether both the respective attributes and values subsume each other. For example

hasLocation–Bone ≤ *hasLocation–Humerus*

assuming that the model includes *Humerus* is a kind of *Bone*.

This definition of criterial subsumption is not quite complete. Later we shall look at a more complex definition which allows for special relationships between attributes such as *hasLocation* and *isPartOf*. This more complex definition is needed to co-ordinate different relationships and for example capture the idea that a 'fracture of the shaft of the humerus' is also a kind of 'fracture of the humerus', without requiring that the 'shaft of the humerus' itself be a kind of 'humerus'.

5.3.4 Formal subsumption between sets of criteria

It is now straight forward to define subsumption between criteria sets which is the key to the definition of formal subsumption between entities. Given two set C_1 and C_2 then:

$$C_1 \leq C_2 \text{ iff for all } c_i \text{ in } C_1 \text{ there exists a } c_j \text{ in } C_2 \text{ such that } c_i \leq c_j \qquad (4)$$

This is a specialised type of subset relationship in which for all the criteria in one set there is the same or a more specialised version in the second set. For example if we assume the obvious relationships then

{*hasLocation–Bone hasCause–Bacterium*}
≤
{*hasLocation–Humerus hasCause–Staphylococcus hasSeverity–Severe*}

---

6 Within the current interpretation of SMK conventional subsumption is the only assertion which is not represented by a distinct object. For the purposes of completeness we propose a pseudo–SMKobject S which represents that E_1 subsumes E_2

$$S(E_1 \leftarrow E_2)$$

5.3.5 Formal subsumption between definitions

Subsumption between criteria sets can be applied to the canonical defining form of entities to determine formal subsumption. Thus if we have two entities defined as $E_1(B_{e1}, D_{c1})$ and $E_2(B_{e2}, D_{c2})$ then we can define a function for formal subsumption $\leq f$

$$E_1 \leq f\ E_2 \text{ if } (B_{e1} \leq B_{e2}) \text{ AND } (D_{c1} \leq D_{c2}) \qquad (5)$$

This definition recognises that subsumption for all entities comprises two parts. There is the formal part represented by the rules for subsumption between criteria sets, and the conventional part contained within the requirement for subsumption between the base types.

For example

Fracture which: *hasLocation–LongBone*
$\leq f$
Fracture which: *hasLocation–Humerus*

and

Disease which: *hasCause–InfectiveAgent*
$\leq f$
Hepatitis which: *hasCause–Virus*

The second example assumes that *Hepatitis* is a kind of *Disease* and *Virus* is a kind of *InfectiveAgent*.

The ability to infer formal subsumption between two entities is one of the most powerful features of SMK. However the above definition is not quite complete. The possibility of conventional subsumption and other forms of assertions involving prototypes as well as elementary entities means that the conventional subsumption is not simply confined to that part of the definition concerned with elementary entities. In the taxonomic hierarchy conventional and formal relationships are woven together, and there are things which may be known about entities which we wish to include in the determination of subsumption. This was one of the important extensions required to the more restricted T–Box discussed in earlier chapters.

For the moment we shall give a simple recursive definition of total subsumption ($\leq$) based on the formal ($\leq f$) and conventional ($\leq c$) components.

$E_1 \leq E_2$ if (6)
$E_1 \leq c\ E_2$ OR
$E_1 \leq f\ E_2$ OR
there exists an E_3 such that $(E_1 \leq E_3)$ AND $(E_3 \leq E_2)$

5.3.6 Co-ordination of subsumption with the part–whole relationships

There are several aspects of medical terminology which require co-ordination between relationships or axes. The best example of this is the relationship between the partative and the location relationships. The long bones of the body have a central shaft and the *Humerus* is such a bone, hence we can speak of

Shaft which: *isPartOf–Humerus*

This is a kind of *Shaft* not a kind of *Humerus*. It *isPartOf* the *Humerus*. However it is the case that a fracture of the shaft of the humerus is a kind of fracture of the humerus (and not in some sense part of it). If we ask the question 'does the patient have a fracture of the humerus', and there is a fracture of the shaft, then we would expect the answer yes. A disease located in part of something is also located in all of it. In this example the attribute *hasLocation* is specialised across the attribute *isPartOf*. This is in addition to the general specialisation of all attributes across the subsumption relationship.

The role of specialisation of one attribute by another is best considered through the way it affects the subsumption relationship between criteria. We have already seen part of the definition of subsumption between criteria

$$c_1 \leq c_2 \text{ if } (\text{attribute}_1 \leq \text{attribute}_2) \text{ AND } (\text{value}_1 \leq \text{value}_2)$$

This however takes no account of specialisation across other types of relationship. If we have a criterion c_2 defined as

$$c_2 \equiv \text{attribute}_1\text{–}(E\{\text{attribute}_3\text{–value}_3\})$$

that is the criterion attribute$_3$–value$_3$ is necessarily true of E, and the specialisation relationship between the two attributes

$$\text{specialises}(\text{attribute}_3, \text{attribute}_1)$$

then

$$c_1 \leq c_2 \text{ if } (\text{attribute}_1 \leq \text{attribute}_2) \text{ AND } (\text{value}_1 \leq \text{value}_3)$$

For example

hasLocation–Humerus
≤
hasLocation–(*Shaft* which: *isPartOf–Humerus*)

and hence

Fracture which: *hasLocation–Humerus*.
≤f
Fracture which: *hasLocation–*(*Shaft* which: *isPartOf–Humerus*)

The great value of this form of specialisation is that it acts to co-ordinate the subsumption relationship with other relationships, most notably *isPartOf*, without requiring the two to be collapsed into one as is often the case in knowledge representation schemes.

5.4 Constraints on expressions and sanctioning: statements as triples

We now come on to discuss one of the key requirements on the terminology system described in chapter 4: that it should be capable of producing 'all and only' medically sensible concepts. This requires that the representation can derive the constraints on what does and does not make sense. The operation to achieve this was sketched in section 4.4.10 as

possible(TM, Disease hasSeverity Severity) → TM*

This is read as 'it is possible for things which are diseases to have a severity'.

Note that the statement above is actually a sanctioning statement rather than a constraint. The approach taken within SMK is that nothing can be said unless there is a statement that says it is possible. The result is that all compositions are forbidden unless they have been sanctioned.

5.4.1 Complex relationships: triples

Within SMK terminological statements are represented as triples which are relationships with a complex or defining structure of the form

topic–attribute–value

The schematic operation **triple:** is used to make a statement

<entity> triple: attribute–value

or if the attribute–value pair is considered as a criterion then

<entity> triple: <criterion>

For example

Disease triple: *hasSeverity–Severity*

The canonical defining form of a triple simply requires that each of its constituent parts be in their canonical form. Hence a triple T is written as

$$T \equiv T(E_C–c_C) \qquad (7)$$

with the subscript indicating the canonical form of the entity and criterion. As before this is the basis for determining identity and SMK only permits one such triple to be present within the model.

A graphical notation is frequently used to denote SMK statements as shown in figure 5.3. Although this is highly suggestive of data structures this type of notation is often useful in understanding the role of triples.

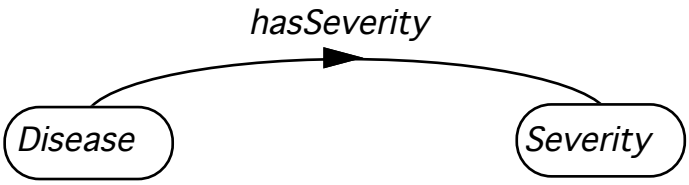

Figure 5.3 A triple between two entities

5.4.2 Sanctioning of descriptions by triples

A triple is an assertion but of a particular kind, representing what it is sensible to say. It does not change the properties *per se* of the entity it describes, but instead it determines which new descriptions it is possible to form by elaborating the definition of that entity. The presence of a triple can permit the addition of a criterion to the definition of an entity to form a more specialised, distinct concept. For example, consider the expression

Pneumonia which: *hasSeverity–Severe*

In trying to decided whether or not this expression is well–formed, one requirement is that the criterion *hasSeverity–Severe* is applicable to the entity *Pneumonia*. This is determined by the presence of a suitable triple which sanctions its application. If we assume that triples are inherited, that is statements apply not only to the entities to which they directly refer but also all those pairs which are subsumed by those entities, then the application of the criterion can be sanctioned by a previously made statement such as

Disease triple: *hasSeverity–Severity*

This is sketched diagramatically in figure 5.4.

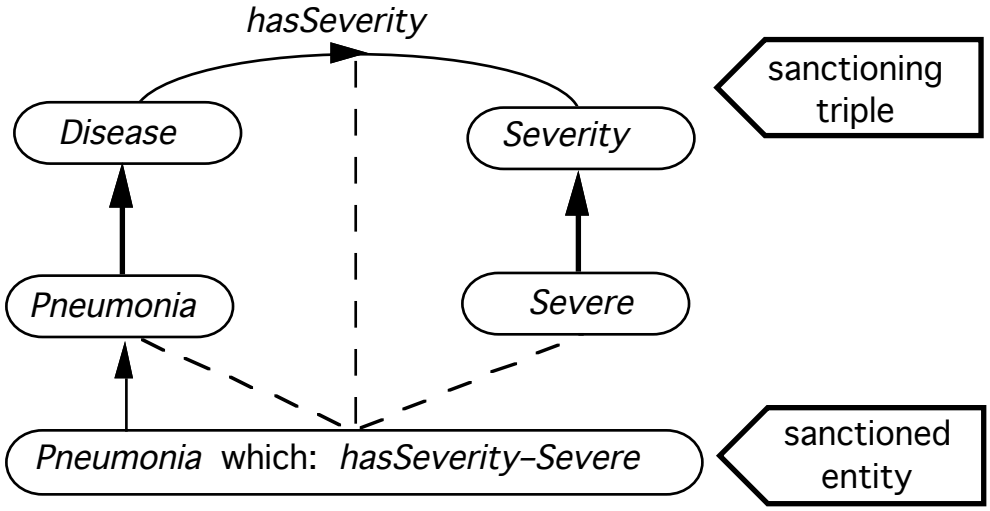

Figure 5.4 Sanctioning of a description by the presence of a triple

In general the sanctioning of the application of a criterion can be defined by

sanctioned(E which: c) if (8)
there exists a triple T such that
$T(E_i–c_i)$ AND
$(E_i \leq E)$ AND
$(c_i \leq c)$

This is the mechanism for constraining which things are sensible and hence parsimony. The goal is a parsimonious model with relatively few sanctioning statements compared to the number of possible sanctioned prototypes.

5.5 Levels of statements and qualifiers: conceivable, grammatical, possible, and necessary

5.5.1 Qualifiers

So far we have concentrated on statements representing what it is possible to say. However within SMK there are several additional sorts of terminological statement which can be made. These statements are needed to provide levels of sanctioning and hence flexible control over the acquisition of knowledge and the creation of prototypes.

The various statements are distinguished by a qualifier which specifies the level to which they belong. Our definition of a triple thus needs extending to include the qualifier, q

$$T \equiv T(E_C–cc: q) \quad (9)$$

In the case of the possibility statements we have discussed the qualifier is not surprisingly possible.

For example the operation triple: is extended to

Disease triple: *hasSeverity–Severity*: possible

Note that qualifiers are not entities but form part of the definition of a triple. There are four qualifier levels within the current version of SMK. These are

- conceivable
- grammatical
- possible
- necessary

There is a precedence amongst the qualifiers and in order to make a statement at one level there must already be present within the model a suitable triple, usually inherited, at the preceding qualifier level. This ability to create statements at several distinct qualifier levels provides for more appropriate control over the representation of the terminological knowledge in a model. The role of each qualifier will be considered in turn.

Conceivable

A conceivable statement is simply the definition of the attribute itself and reflects the view of attributes as elementary statements of a totally non–specific form. The creation of a new attribute makes it conceivable for a new distinct type of relationship to occur, but says nothing beyond that. This view establishes the conceptual link between attributes as elementary relationships defined by an identifier, and triples which are structured relationships defined by their composition. Users have no direct access to the conceivable qualifier

Grammatical

A grammatical statement is similar to basic type constraints on attributes in other languages, and represents what makes 'grammatical sense'. For example

Disease triple: *hasLocation–BodyPart*: grammatical

is sensible because it is 'disease of a body part' makes grammatical sense, whereas the following is not sensible

Disease triple: *hasLocation–Drug*: grammatical

The use of grammatical statements evolved from the type constraints which were placed on attributes in earlier versions of SMK. The use of grammatical statements has major advantages over the use of type constraints on attributes in that it:

- produces a unified view of constraints
- allows for multiple distinct grammatical statements to be made between pairs of entities – type constraints on attribute are a property of the attribute alone and not the topic–attribute–value triple.

The grammatical statements represent the first level at which a specific statement makes any sense at all. They provide a high level schema to guide subsequent knowledge acquisition. They are also useful in guiding questioning of the model. For example it is sensible to ask about 'diseases of the arm' in general without wishing to specify which ones in detail. It makes no sense at all however to ask about 'diseases of drugs. It is important to note however that a grammatical statement cannot be used in a simple way to sanction a which: operation.

Possible

Possibility statements, as described earlier, are the representation of what it is sensible to say. They represent the bulk of the detailed terminological knowledge within a model.

Necessary

Necessary statements represent things which are necessarily and indefeasibly true about an entity but which do not form part of its sufficient definition. They are very strong assertions about an entity. For example

Cancer triple: *hasSeverity–Severe*: necessary

states that it is indefeasibly true that all cancers are severe, and thus for example that 'mild cancer' is a contradiction. Being severe becomes a necessary property of cancer but it is not required in a sufficient definition. Another view is that the statement asserts that 'cancer' and 'severe cancer' are one and the same thing. Following the assertion that 'cancers are severe' it becomes tautologous to speak of 'severe cancer' The effects of necessary statements are discussed in more detail in section 5.6 within the discussion of the coherence of criteria sets.

5.5.2 Constraints on making a statement

The precedence amongst qualifiers determines whether or not a statement can be made. The rule for sanctioning the making of a statement is, as would be expected, essentially the same as that for the sanctioning of a **which:** operation with the added dimension of the qualifier. It requires a statement at the preceding qualifier level, hence

sanctioned(E **triple:** c q) if (10)
there exists a triple T such that
$T(E_i–ci: qi)$ AND
$(E_i \leq E)$ AND
$(c_i \leq c)$ AND
$next\_higher(q_i,q)$

In this definition $next\_higher(q_i,q)$ is true if q_i is the qualifier of the next higher precedence to q, for example grammatical immediately precedes possible. This definition is not quite complete because it also requires that there is no inherited triple of the same or lower precedence. For example it is not permitted to make a possibility statement if there is already an appropriate possibility statement, perhaps one more general, within the model. This is considered tautologous and therefore prohibited.

5.5.3 Subsumption between triples

Subsumption between triples is defined simply by

$T(E_i–ci: q_i) \leq T(E_j–cj: qj)$ iff $(E_i \leq E_j)$ AND $(c_i \leq c_j)$ AND $(q_i \leq q_j)$ (11)

In this definition the test $(q_i \leq q_j)$ simply means that one qualifier must be of any higher precedence than the other and not necessarily of the next higher precedence.

With subsumption defined for triples they can be thought of as forming a hierarchy. Because a statement is only sanctioned if an appropriate triple of the next higher qualifier precedence is present, the immediate 'parent' of every triple is of the next higher qualifier precedence (figure 5.5).

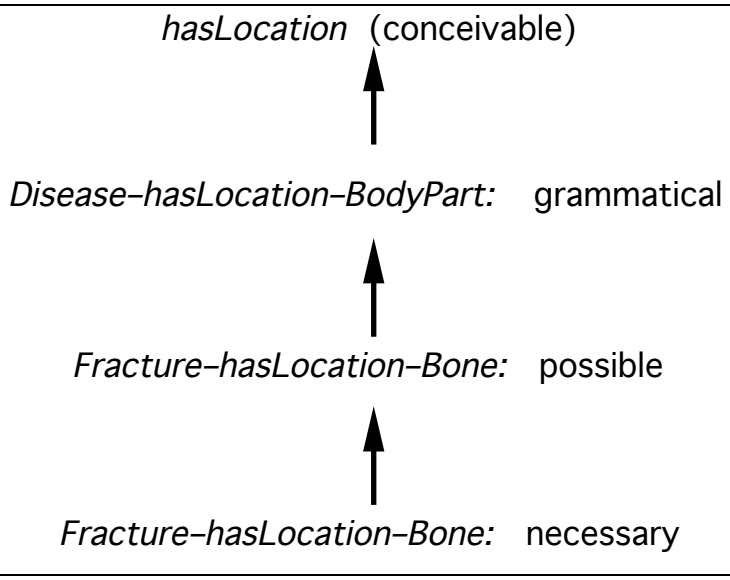

Figure 5.5 A hierarchy of triples across the four levels of qualifiers

Note that the interpretation of attributes as conceivable statements means an attribute always subsumes all those triples which are formed by its use. There is a fundamental SMK primitive entity *TopAttribute* which subsumes all attributes and hence all triples. This is in turn subsumed by *TopThing* which means that the triples meet the requirement to be within a complete subsumption hierarchy of all SMKobjects.

5.5.4 The reciprocal nature of statements

Grammatical and possible statements are reciprocal. This means that making a statement is equivalent to also making the inverse statement. Thus for example the statement

Disease triple: *hasLocation–BodyPart:* grammatical

implies the statement

BodyPart triple: *isLocationOf–Disease*: grammatical

This must be the case because if it is correct to say one it must be correct to say the other. This is shown in figure 5.6.

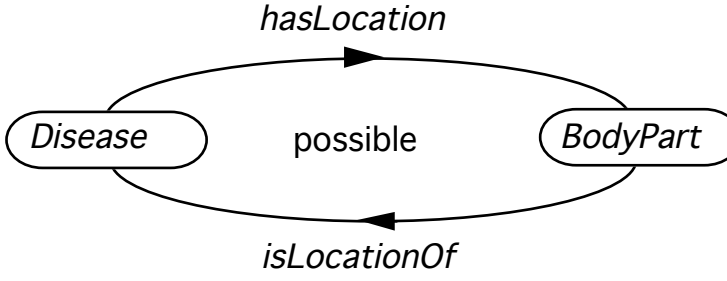

Figure 5.6 The symmetry of triples in SMK

The combination of the two triples is a very interesting object. Within the current interpretation of SMK this combination is itself an SMKobject, and the two triples are considered as directed views of that SMKobject, or more graphically it two 'ends'. This means it is possible to describe the entire mutual relationship. In the example given above this is the relationship of 'locating', which comprises one thing having a location and the other being a location.

Necessary statements are not reciprocal. For example we may wish to say that all fractures are located in a bone but certainly not that all bones are the location of a fracture. This reflects the fundamentally different role of these statements and suggest that in future developments they should be dealt with in a separate but related way to the others.

5.6 Coherence of expressions, complete criteria sets, and the canonical forms of criteria and criteria sets

In the previous section we have looked at the role of statements in sanctioning the creation of definitions, which is part of what is required to decide if an expression is well–formed. In order to complete this definition, and that of formal subsumption, we need to look in detail at the set of all the criteria which relate to an entity and the rules for deciding if that set is coherent. This is the basis for preventing concepts such as 'mild severe cough' and 'broken arm in the leg'. We begin with the determination of the necessary criteria of an entity.

5.6.1 Inheritance of criteria and complete criteria sets

The subsumption relationship is the basis for determining all of the properties of an entity, which in the context of the current discussion means all the criteria which are inherited by an entity. Within SMK the canonical defining criteria of an entity are also indefeasibly true of any entity which it subsumes, that is criteria are indefeasibly inherited over the subsumption relationship.

For example if we remodel our view of *Fracture* and assert that it is subsumed by the complex prototype Trauma **which:** *hasLocation–Bone*, then it is indefeasibly true of *Fracture* that it is located in *Bone* (figure 5.7).

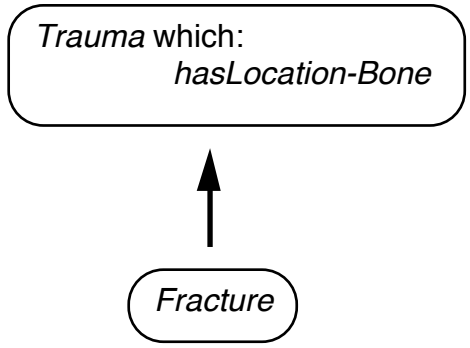

Figure 5.7 The inheritance of a criterion through conventional subsumption

Note that the entity Fracture is still completely defined as an elementary entity by the identifier Fracture, but it is also necessarily true that it is located in *Bone*. Hence there is the notion of two sets of criteria:

complete criteria set:	all those criteria which are necessarily true of an entity
defining criteria set:	only those criteria, assumed to be canonised, which together with the base type are sufficient to uniquely identify the entity

The defining criteria set is always a subset of the complete criteria set.

In figure 5.7 the conventional subsumption has had the effect of asserting that the criterion *hasLocation–Bone* is indefeasible true of *Fracture*. Criteria in the complete criteria set which are not part of the definition of an entity arise because of the use of assertions of some form. This will hold true for all entities. Thus extending the example to a prototype we have

Fracture which: *hasSeverity-Severe*

base type:	*Fracture*
defining criteria set:	{*hasSeverity-Severe*}
complete criteria set:	{*hasSeverity-Severe, hasLocation–Bone*}

The complete criteria set of an entity is the union of its defining criteria set and the set of all those criteria it has acquired, either directly or via inheritance, which are true because of an assertion somewhere in the hierarchy.

Necessary statements also elaborate the complete criteria of an entity and are indefeasibly inherited. They represent very strong assertions of necessary criteria. In an earlier example a necessary statement was used to assert that *Fracture* must be in a *Bone*. This achieves a similar if not identical effect to the last example when *Fracture* was created as an elementary kind of *Trauma* which: *hasLocation–Bone*. This comparison is shown diagrammatically in figure 5.8.

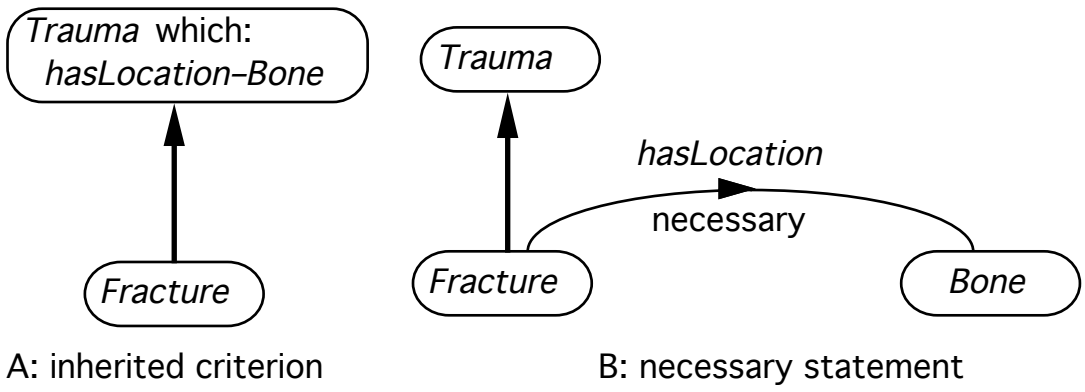

Figure 5.8: Two models comparing the use of conventional subsumption and necessary statements

This illustrates the essential equivalence of the effect of conventional subsumption from a prototype (A) and a necessary statement (B), both of which result in *hasLocation–Bone* being an indefeasible property of *Fracture*. The use of a necessary statement can however often avoid the creation of an entity (*Trauma* which: *hasLocation–Bone* in the above example) purely to support an assertion about some other entity. This whole area will be illustrated in more detail when looking at some longer examples and issues of 'modelling style' in chapter 8.

5.6.2 Coherence and cardinality

Within SMK there is a requirement that the complete criteria set of an entity be coherent. This involves the derivation of a canonical form for criteria and criteria sets and a basic test of coherence.

The ultimate determinant of coherence is the cardinality of the attributes that go to form the criteria within a set. Cardinality is used here in the database sense of the number of permitted distinct values for any particular relationship. In the current interpretation of SMK an attribute can only have one of the two cardinalities, one or many. This extends the definition of an attribute to include its cardinality

<attribute> : <identifier>:<cardinality>

The cardinality of a criterion is that of its attribute and is the basis for deciding whether or not a criteria set is coherent. For any canonical criteria set the cardinality of each and every criterion

must be respected. For example if the attribute *hasLocation* is defined as having a cardinality of one then a criteria set containing two criteria such as {*hasLocation–Arm hasLocation–Leg*} is considered incoherent because there are two distinct criteria with that attribute. Note however that the inverse attribute, in this case *isLocationOf*, may be defined with a cardinality of many, in which case the set {*isLocationOf–Fracture isLocationOf–Cancer*} is coherent.

5.6.3 Transforming criteria sets to a canonical form

The count of the criteria with a given attribute is not sufficient to determine if a criteria set is coherent. It may be the case that two criteria with the same attribute are present, but they are related in such a way that one is redundant. For example {*hasLocation–Arm hasLocation–Limb*} has two criteria for the attribute *hasLocation*, but *Arm* is a kind of *Limb* and hence it would seem desirable to reduce this to the coherent form {*hasLocation–Arm*}. This is an example of re–writing or transforming a criteria set to its final canonical form before testing for coherence based on cardinality.

We wish to be able to transform any set of criteria to a canonical set which respects the cardinality of each individual criterion. If this cannot be achieved then the set is incoherent. These transformations will outline an algebra for expressions and criteria.

The first and most obvious need is to resolve multiple criteria, if at all possible, aimed at removing those which are redundant. Given the definition of criterial subsumption we give a preliminary definition of the rules for re-writing criteria. If within a criteria set one criteria subsumes another then the former says nothing additional to the latter and can be 'deleted' from the criteria set.

Thus for a set of criteria S={c_1, c_2, ... c_n} if for some criterion c_i in S there is a criterion c_j also in S such that c_i subsumes c_j then delete c_i from S

For example because *Humerus* is a kind of *Bone* then

{*hasLocation–Bone hasSeverity–Severe hasLocation–Humerus*}
⇓
{*hasSeverity–Severe hasLocation–Humerus*}

and

{*hasLocation–Bone hasLocation–Femur hasLocation–Humerus*}
⇓
{*hasLocation–Femur hasLocation–Humerus*}

In the first example the resulting set is coherent with respect to cardinality whereas the second remains incoherent.

5.6.4 Joins of criteria

The situation of one criterion subsuming the other is actually a special case of the more general situation of being able to form a criterion by combining the properties two criteria. The new criterion has all and only the characteristics of the original two criteria. We shall call this operation join and denote it by the modified plus sign ⊕. The result of a join is illustrated by the following example

hasLocation–Humerus ⊕ *hasLocation*–(*Bone* which: *isLocationOf–Cancer*)

⇓

hasLocation–(*Humerus* which: *isLocationOf–Cancer*)

The result is both a humeral location and a location in a cancerous bone. This has been achieved by joining the most specific aspects of the two criteria into a third. In general then for two criteria defined as

$$c_1 \equiv a_1\text{–}E_1$$

$$c_2 \equiv a_2\text{–}E_2$$

the criterial join is defined by

$$c_1 \oplus c_1 \Rightarrow (a_1 \oplus a_2)\text{-}(E_1 \oplus E_2) \qquad (12)$$

5.6.5 Joins of entities

The entity $(E_1 \oplus E_2)$ in the previous definition is the join of the definitions of E_1 and E_2, that is the entity whose definition encompasses those of both E_1 and E_2 and no more. If one entity subsumes another than this simply reduces to the more specialised of the two. However it can be more complicated for example as shown by the entities in the preceding example

Humerus ⊕ *(Bone* which: *isLocationOf–Cancer)*

⇓

Humerus which: *isLocationOf–Cancer*

This operation is similar to the ConjGen operation mentioned earlier in relation to KL–ONE, but is more tightly defined and must satisfy certain constraints. It also has similarities with the maximal join operation as used in the conceptual graph representational scheme [Sowa 1984].

If we write E_1 and E_2 as canonical definitions then we define the join of two entities as

$$E_1(B_{1e}, D_{1c}) \oplus E_2(B_{2e}, D_{2c}) \Rightarrow E_3((B_{1e} \oplus B_{2e}), (D_{1c} \oplus D_{2c})_c) \qquad (13)$$

For the elementary bases B_{1e}, and B_{2e} the new base is simply the more specific of the two. For example

Bone ⊕ *Humerus ƒi Humerus*

For the pair of canonical defining criteria sets, D_{1c} and D_{2c}, it is the canonical set derived from the join of the two. Note that this implies a recursive definition.

For elementary entities the join will be ill–formed unless one subsumes the other. For example

Femur ⊕ *Humerus*

is incoherent. The exception is if some rather odd entity has been defined as being subsumed by these two concepts, that is it is an 'arm–and–leg bone'. If the elementary base of an entity is thought of as the elementary indefinable criterion then there is a universal requirement that the cardinality of such an elementary criterion is one. Something cannot have two bases. This provides a unified view of descriptions in SMK.

Likewise for more complex situations

(*Fracture* which: *hasLocation–Humerus*) ⊕ (*Fracture* which: *hasLocation–Femur*)

is clearly incoherent.

In the definition of the criterial join (13) the attributes a_1 and a_2 where shown as being distinct. In fact as SMK is currently formulated they are assumed to always be the same. This is the whole basis for attempting the join in the first place. Attributes do however form a hierarchy and in principle it could be necessary to test for a join between criteria with non–identical attributes. However cardinality is not clearly defined in these situations and at present it is assumed not to happen.

5.6.6 Exteriorisation of embedded criteria

The preceding considered the handling of multiple criteria within a set. There are however situations where 'embedded' criteria lead to redundancy or incoherence when only a single criterion appears to be involved. For example the following is tautologous

Femur which: *isLocationOf–(Fracture* which: *hasLocation–Femur)*

and should clearly reduce to the simpler form

Femur which: *isLocationOf–Fracture*

This requires 'deleting' the embedded criterion *hasLocation–Femur* from the value of the inner criterion.

More general redundancy is also undesirable and

Bone which: *isLocationOf–(Fracture* which: *hasLocation–Femur)*

should again be transformed to the same simpler form

Femur which: *isLocationOf–Fracture*

In this example it is also necessary to refine *Bone* to *Femur* as well as 'delete' the criterion.

Embedded criteria may also result in an expression being incoherent. For example

Humerus which: *isLocationOf–(Fracture* which: *hasLocation–Femur)*

that is the arm which is the location of a broken leg, is clearly as nonsensical as is its converse form

(*Fracture* which: *hasLocation–Femur*) which: *hasLocation–Humerus*

The problem of testing whether or not the expression is coherent involves testing its converse expression for coherence. However the transformations needed to resolve conflicts in situations such as the one above are somewhat more complex.

The problem arises when a criterion has embedded in its value another criterion whose attribute is the inverse of its own attribute. This places restrictions on the type of entity to which it can be applied. Consider a criterion of the form c=a–V where V is such that one of the criteria c_i which is true of V is of the form c_i=a′–Y where a′ is the inverse of a.

For example

c= *isLocationOf–(Fracture* which: *hasLocation–Humerus)*

where

V= *Fracture which: hasLocation–Humerus*
c_i= *hasLocation–Humerus*
a= *isLocationOf*
a′= *hasLocation*
Y= *Humerus*

If this criterion is used in an expression

E which: c

then the rule is that this must be transformed to

E_j which cj

where

1 E_j is the most specialised entity derived from E and Y. This is just the join operation defined above

$$E_j \equiv (E \oplus Y)$$

2 c_j is the criterion a–V_j such that V_j is the most generalised form of V which can be derived by attempting to 'delete' the criterion c_i. This operation is a generalised exteriorisation of a criterion. We shall denote it by Θ. It is defined by

$$V_j \equiv V \Theta c_i$$

such that V_j is well–formed and is the most general possible entity for which

$$V_j \text{ which: } c_i \Rightarrow V$$

In uncomplicated cases this amounts to 'deleting' a criterion from a definition, and the Θ operation acts as an inverse **which:**.

Thus if E is *Bone* then we have

$$E_j \equiv \textit{Bone} \oplus \textit{Humerus fi Humerus}$$

$$V_j \equiv \textit{Fracture which: hasLocation–Humerus } \Theta \textit{ hasLocation–Humerus} \Rightarrow \textit{Fracture}$$

and hence

$$E \text{ which: } c \Rightarrow E_j \text{ which } cj \Rightarrow \textit{Humerus} \text{ which: } \textit{isLocationOf–Fracture}$$

In general if we denote that some criterion, $c_i \equiv$ a′–V is true of an entity X by X{c_i} then the whole exteriorisation of embedded criteria is given by

$$E \text{ which: } a–(X\{a'–V\}) \Rightarrow (E \oplus V) \text{ which: } (X \Theta (a'–V)) \qquad (14)$$

It is possible however for dependencies amongst criteria to complicate the Θ operation. For example if Y is defined as

Y ≡ *Neoplasm* which:
hasBehaviour–Malignant
hasSpread–Secondary

and assuming that according to the model only malignant neoplasms can be secondary then

$$Y \Theta (\textit{hasBehaviour–Malignant}) \Rightarrow Y$$

In this example entity Y is left unchanged because is not possible to delete the criterion *hasBehaviour–Malignant* without the remaining entity being ill–formed. It would be a non–malignant secondary neoplasm which in any sensible model is forbidden. There is thus a conditional dependency between the criterion hasSpread–Secondary and the criterion hasBehaviour–Malignant . Likewise if a criterion is necessarily true of an entity then it cannot be coerced to a more general form. For example the criterion *hasLocation–Bone* cannot be 'deleted' from the entity *Fracture*.

The operation for generalised exteriorisation has proved to be one of the more difficult to implement. It is manageable at present if their are no conditional dependencies amongst the criteria. If however the situation is like the case of a secondary malignant neoplasm described above then at present a warning is given and the attempt at deletion is abandoned. In principle it should be manageable, but its implementation would benefit from more explicit recording of the dependencies.

The above description has implicitly assumed that the cardinality of the embedded criterion is one. If it is not the case then there is no need to insist on the generalised exteriorisation of criteria. For example

ChestPain which: *isAggravatedBy–*(*Exercising* which: *aggravates–Cough*)

cannot be resolved because *ChestPain* and *Cough* are elementary and disjoint and thus the result of ⊕ is ill–formed for this pair. However the cardinality of *aggravates* is many and hence the embedded form is itself well–formed. The concept is however a little odd being 'chest pain which is made worse by the sort of exercising which aggravates a cough', and it may be the case that such definitions are sufficiently peculiar to merit some form of warning when encountered.

5.6.7 Consolidated requirement for canonisation and coherence

Given a proposed definition

E which: c

we can now summarise the requirements for coherence

1 the complete criteria set of the entity must be in its canonical form

2 the canonical complete criteria set of the entity must be coherent with respect to cardinality, and the entity can have only a single base type

5.7 Consolidated requirements for being well–formed

Having now covered the process of sanctioning and coherence it is possible to pull together the full requirements for an expression in SMK to be well–formed. An expression which is attempting to apply a criterion to an entity must be such that

1 the criterion is sanctioned by a suitable possibility statement

2 the resultant entity has a coherent complete criteria set

For example we would hope that given a sensible set of statements the first requirement would fail the following for the reasons shown

Fracture which: *hasLocation–Penicillin* – no grammatical statement

Fracture which: *hasLocation–Lung* – no possibility statement

The first requirement however would pass

(*Fracture* which: *hasLocation–Humerus)* which: *hasLocation–Femur*

because fracture of the humerus is a kind of fracture and the femur is a kind of bone. However this case fails the coherence test. The cardinality of *hasLocation* is one.

5.8 Naming and surface linguistics

Another set of important requirements discussed in the functional description of a terminology system concerned the need to provide an external representation of the concepts, in a form which could be elucidated by human users. This amounts to something which can be read and understood, though it is not impossible to imagine a conceptual structure which produced images or sounds.

5.8.1 Names

In SMK the external representation is tied to the ability to name concepts. It is important to understand at the outset the distinction between an identifier and a name. Identifiers apply only to elementary entities and although as mentioned earlier we have used suggestive naming

conventions in the examples, there is no reason why identifiers cannot be meaningless symbols or 32–bit integers. The only requirement is that they be unique, and if a model is to have validity across many systems, be agreed. Names however can be given to complex as well as elementary entities and are intended to suggest meaning to a reader. The operation to assign a name is schematically

<entity> name: <name>

A name can also be thought of as a shorthand for an entity, particularly if it is the name of a complex entity. For example

Neoplasm which:
hasBehaviour–Malignant
hasCellType–EpithelialCell

could be given the name *Carcinoma*. It is important to recognise that giving a name to an entity does not affect in any ways its formal properties. The entity named *Carcinoma* will still be subject to all the constraints of the formalism. What it does do however is make the link between a formal definition as an entity and the concept a user would recognise by the word 'carcinoma'.

Within the current formulation of SMK names are required to be unique. Hence a name can also be used to refer to an entity, and this will be exploited extensively when looking at the implementation of the terminology engine and associated tools which provide the concrete implementation of the operations described schematically in this chapter. Names are also a very convenient way of referring to large complex descriptions.

5.8.2 Public names

There is another form of naming supported by the current version of SMK. Public naming allows a string of characters to be assigned to an entity.

<entity> publicname: <string>

This public name again has no effect on the internal meaning of the concept, and further more is not required to be unique. It is used to generate phrases but cannot be used in any straightforward way to refer to an entity. For example

Neoplasm which:
hasBehaviour–Malignant
hasSpread–Secondary
hasCellType–EpithelialCell

may be given the public name 'secondary carcinoma'

Public naming represents the strongest separation between the underlying formal conceptual meaning and surface linguistic representation.

5.8.3 Production of phrases

The generation of external phrases is currently possible on one of several bases

1 Using only the names of elementary entices and the phrase forming operation which, for example

"Neoplasm which hasBehaviour Malignant, hasSpread Secondary, hasCellType EpithelialCell"

2 Using the names of complex as well as elementary entities where these are defined, for example

"SecondaryEpithelialCellCancer"

3 Using the public names of entities where these are defined, for example

"secondary carcinoma"

The current implementation of external representations and surface linguistics within SMK is naive. More sophisticated mechanisms including grammars are needed to produce phrases and sentences which are closer to natural or medical language.

5.9 Operations to be included

There are several operations which were not discussed in the preceding chapter.

5.9.1 Deriving the properties of an entity

The **of:** operation returns the values of those criteria which are necessarily true of an entity

<attribute> of: <entity> ⇒ {<entity}

For a given attribute this operation returns the values of all of the criteria for that attribute within the canonical complete criteria set of the entity

$$a\ of:\ E \Rightarrow \{V_i \mid E\{a{-}V_i\}\}_C \qquad (16)$$

For example on the basis of earlier examples

hasLocation of: *Fracture* ⇒ {*Bone*}

For an attribute with a cardinality of one the set will always contain a single value or be empty.

5.9.2 Determining the applicable attributes for an entity

The operation **refiningAttributes:** determines which attributes are applicable to an entity

refiningAttributes: <entity> ⇒ {<attribute>}

This returns a set of attributes by which it is possible to further describe , or specialise an entity. It corresponds to returning the attributes of the set of possibility triples relevant to that entity. It is one aspect of 'what it is sensible to say'. For example is some simple model we may find

refiningAttributes: *Fracture* ⇒ {*hasLocation hasSeverity*}

5.9.3 Generating prototypes

The operation **refineBy:** attempts to refine an entity by describing it further using a specific attribute

<entity> refineBy: <attribute> ⇒ {<entity>}

This is somewhat more tricky to define than the previous two extended operations. It is easy to fall into the 'how many rocks are there' trap as discussed in the chapter 3. The result of this operation should only depend on the knowledge in the system and not on which particular prototypes happen to be represented as objects in some physical data structure. On the other hand an operation which returned all possible refinements is not that useful. The challenge is to get a definition which permits for progressive refinement, but will eventually achieve closure. The definition of this is incomplete but the principles are as follows.

First identify the criterion for the relevant attribute that is to be the refined. There are three possible situations:

- there is a criterion in the complete criteria set
- there is no criterion in the complete criterion set but there is a suitable possibility triple

– there is no suitable criterion

This procedure to identify a criterion is relatively straightforward, the difficult part is to define a rule for refining that criterion. For example consider the operation

Fracture refineBy: *hasLocation*

The first step will identify the criterion *hasLocation–Bone* as the one being in need of refinement. But what are the refinements of this criterion? It would seem obvious that we require some kinds of bones such *Humerus* and *Femur*. *Fracture* which: *hasLocation–Humerus* is a perfectly good refinement of *Fracture* by *hasLocation*. However the 'some kinds of' question is exactly the trap pointed out by Brachman with his 'some kinds of rocks' example discussed in chapter 3. For example why not include in our list of bones *Humerus* which: *isLocationOf–Cancer*?

The current approach to this problem is to require that the refinement does not invoke additional formal refinements. The main interest is in following the knowledge in the system not in over stimulating its generative properties. Hence the refinement tries to follow assertions in the form of conventional subsumptions. This process usually locates elementary entities. This is the used in the PEN&PAD predictive data entry system [Nowlan 1991].

It is not clear at the moment whether or not closure is possible for an attribute independently of refinement by a second attribute. Experience so far of the fact that dependencies exist amongst criteria, for example only malignant tumours can become secondary, suggests it is not a straightforward matter. However a sound definition of closure will be essential for testing the completeness of a system.

5.10 Summary of the satisfaction of the functional description

To conclude this chapter we shall review the interpretation in SMK of the key requirements for a terminology system

Compositional operators

Elementary entities are referred to by an identifier and complex entities are denoted by the use of the operator **which:**

There are rules for canonising descriptions.

Well–formedness testing

This is defined by an expression being:

1 sanctioned by a suitable possibility statement;

2 coherent with respect to cardinality following canonisation.

Equivalence testing

Is defined by the canonical defining form of an entity which is dependent on the rules for canonising criteria and criteria sets.

Subsumption testing

The test for subsumption (≤) involves testing both conventional (≤c) and formal (≤f) subsumption. One attribute may specialise across another and this serves to co-ordinate the subsumption relationship with others, most notably the partitive relationships.

Creation of atomic concepts

Elementary types and attributes are permitted through the use of **new_elementary_entity:** and **new_attribute:**.

Defining a subsumptive relationship

Conventional subsumption can be asserted by **addSub:**.

Non-subsumptive terminological relationships

These are statements made using the **triple:** operator and belong to one of four layers. A statements can only be made if sanctioned by an appropriate pre–existing triple.

External interpretation of concepts

This is based on the use of naming, with the **name:** and **publicname:** operations.

Chapter 6 The SMK Modelling Language

This chapter introduces the SMK modelling language and presents some simple examples of its use. A more substantive model and its relationship to traditional classification schemes will be examined in chapter 8. The implementation of the compiler and the SMK terminology engine will be discussed in chapter 7.

The first section of this chapter covers the basic syntax of the SMK language and the main operations for constructing models. The next section describes the entities that are created as the basis for every model. The final section gives a few examples of the use of the language to construct simple models.

6.1 The SMK language

The SMK language is not elaborate and the concrete syntax is very close to schematic operations described in chapters 4 and 5. It was devised to help test the SMK terminology engine and support the modelling required by PEN&PAD.

6.1.1 The syntax of SMK operations

All statements in the language are of the form

<entity> keyword <argument(s)>.

The keyword is the name of a defined SMK operation such as **newSub**. A bold font indicates an operation. The operand is always an entity. The list of arguments may be empty but is fixed in number and type for a given operation. Not all arguments need be entities.

The result of every operation is either:

- an entity
- a set of entities
- the non–entity or empty set
- an error

Thus all operations can be thought of as expressions which evaluate to an entity and can be used as operands and arguments. For example the operation to create a new elementary entity as a kind of an existing entity is the **newSub** operation

Trauma **newSub** Fracture

The result of this operation is the new entity *Fracture*. Hence to compose a severe fracture we can write

(Trauma **newSub** Fracture) **which** hasSeverity Severe

Rounded brackets, (), indicate the usual precedence. Note that in this syntax no hyphen is used between the attribute and value of a criterion.

The other elements of the statement are words, for example *Fracture* and *hasLocation*. These are interpreted as names (or identifiers), and can refer to elementary or complex entities.

6.1.2 The compositional operator 'which'

Every expression presented by the compiler to the underlying terminology engine is required to be well–formed, otherwise it is an error. The schematic operation well–formed? discussed in the functional description is implicit in the use of the compiler and underlying terminology engine. Thus the operation **which** as used above does not just compose an expression. It also tests well–formedness and returns the corresponding entity, or generates an error.

It is usually the case that only a small subset of all possible well–formed expressions will be represented or reified as objects in the data structures within the terminology engine. The compositional operator **which** will either return a previously reified entity or reify a new entity as appropriate. Which of these has occurred is transparent to the user. This is a very important principle of SMK. The possibility statements have the effect or creating a large 'virtual' network of entities, and it is this virtual network that the operations interrogate.

6.1.3 The main SMK operations

A description of the operations available in the current version of the language is given in appendix 1. The principle operations are briefly described below.

Creation of an elementary type

The **newSub** operation defines a new elementary entity as a kind of an existing entity

<entity> **newSub** <identifier> (1)

For example

Symptom **newSub** Cough.

Creation of an attribute

Attributes are created in pairs:

<attribute> **newAttribute** <id1> <id2> <inheritance> <cardinality> (2)

This defines a new attribute as a kind of an existing attribute. The arguments <id1> and <id2> are the identifier of the new attribute and its inverse. The inverse is created as a kind of the inverse of <attribute>. The argument <inheritance> has not been discussed previously. It is intended to denote whether or not the attribute is inherited. Currently all attributes are inherited hence it always has the value allAll. The argument <cardinality> is one of the four possible combinations of one and many. For example

Attribute **newAttribute** hasLocation isLocationOf allAll manyOne.

This creates the attribute *hasLocation* with a cardinality of one, and the inverse attribute *isLocationOf* with a cardinality of many.

Conventional subsumption

Conventional subsumption is asserted using **addSub**

<entity1> **addSub** <entity2> (2)

This makes <entity2> a kind of <entity1>. For example

(Disease **which** hasSeverity Severe) **addSub** Cancer.

Note that this operation differs from **newSub** in that no new entity is defined. The entity *Cancer* must already be defined.

Creation of a triple

This operation inserts a triple

<entity> **triple** <criterion> <qualifier> (3)

For example

Disease **triple** hasSeverity Severity possible.

Note that because triples are always bi–directional the statement

Severity **triple** *isSeverityOf Disease* possible

has the same effect on the model, though it returns the inverse triple to the previous example.

Naming

There are two naming operations. The first assigns a unique name to an entity

<entity> **name** <name> (4)

Here <name> is a word (symbol). A name can later be used to refer to the entity in any statement. For example

(Neoplasm **which** hasBehaviour malignant) **name** Cancer.

The second naming operation is 'public naming', a phrase associated with its use by the PEN&PAD interface

<entity> **public** <string> (5)

This assigns a character string to an entity. The public name is not unique and cannot be used to refer to the entity in an operation. It can be used to print a name for an entity. For example

Cancer **public** 'cancerous growth'

6.1.4 Other components of the syntax

The followed are also used to write statements

. a period denotes the end of a statement

() rounded brackets indicate precedence

[] square brackets indicate a list which is to be expanded by the compiler prior to the evaluation of any operations. An expandable list can be used anywhere in a statement to replace a single element. The result is that operations are performed for the Cartesian product of all the lists used in the statement. For example

Drug **newSub** [Aspirin Penicillin Morphine].

performs the **newSub** operation three separate times, once for each identifier. The result of the whole statement is itself an expandable list of the results of the individual operations, that is [Aspirin Penicillin Morphine].

Likewise

Disease **triple** hasSeverity Severity [grammatical possible].

is equivalent to the two separate **triple** operations.

Disease **triple** hasSeverity Severity grammatical.
Disease **triple** hasSeverity Severity possible.

<> Angle brackets indicate a list which is not to be expanded by the compiler. The list is passed directly as an argument to the operation. For example

Fracture **which** <hasLocation Humerus hasSeverity Severe>.

results in the single entity representing 'severe fracture of the humerus'. This is in contrast to the use of square brackets

Fracture **which** [hasLocation Humerus hasSeverity Severe].

which results in a list of two distinct entities, 'fracture of the humerus' and 'severe fracture'.

; a semicolon indicates the cascading of operations onto a single operand. For example

Disease **newSub** Trauma; **triple** hasLocation BodyPart grammatical.

creates the new entity *Trauma* but performs the **triple** operation on the original operand *Disease*.

"" paired double quotes indicate a comment.

Consecutive spaces, tabs, and newlines in any quantity are dealt with as a single separator. Any formatting through the use of tabs and newlines is purely for ease of reading and has no effect on the evaluation.

6.2 Fundamental entities of SMK

Several fundamental entities are created as part of the initialisation procedure of the SMK terminology engine. These entities are the fundamental primitives of the formalism and the starting point for all model. A hierarchical tree of these entities is shown in figure 6.1.

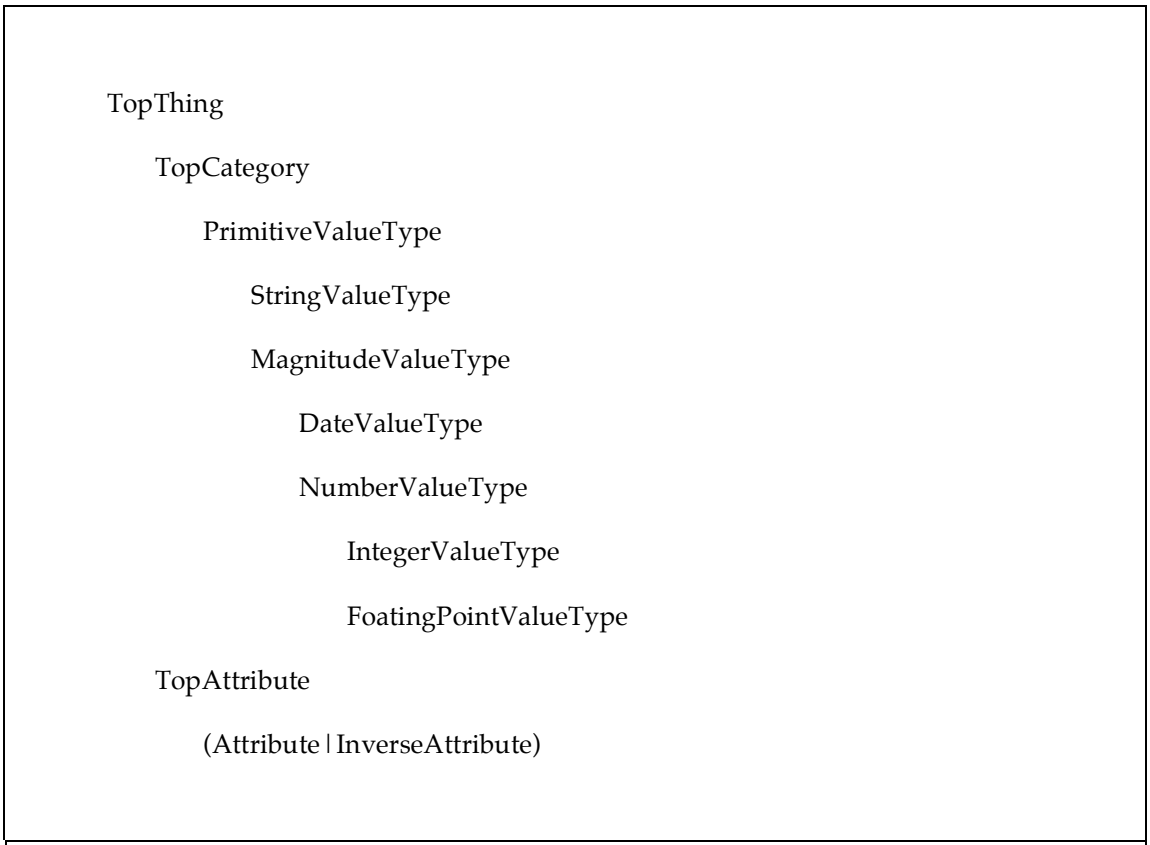

Figure 6.1 The fundamental primitive entities of an SMK network

The roles of the fundamental primitives are:

TopThing – the very first entity and subsumer of all other entities

TopAttribute – the parent of the first true attribute (Attribute | InverseAttribute) – subsumes all attributes and hence all triples

TopCategory – subsumes all symbolic entities

PrimitiveValueType – as well as symbolic concepts SMK supports some primitive values, the main ones being numbers, strings, and dates. These primitive values such as the number 3 and the date '25 June 1992' are entities but are dealt with by primitive mechanisms. Their main use is in the medical record (see chapter 9 and appendix 1 for more details). Primitive value types subsume all the relevant primitive values. For example *IntegerValueType* subsumes the entity representing the integer 3.

Non – this does not appear in the hierarchy. *Non* represents the non–existent entity or the empty set of entities. Nothing is known about *Non* and it has no relationships. *Non* may be the result of an operation.

6.3 Examples of simple SMK models

We shall take a preliminary look at the use of the SMK language based on a few very simple examples. An extended example is the subject of chapter 8. We begin defining a few medical concepts and defining a clinical modifer. The final example is a model of fractures and illustrates some of the basic features of formal subsumption. The medical content of all the examples is trivial.

6.3.1 Creation of some high–level medical concepts and a clinical modifier

Figure 6.2 shows the source text for a few high–level medical concepts using only **newSub** and **newAttribute**. The attribute *DescriptiveAttribute* is used as an 'abstract' attribute. No triples will be created using this attribute. It represents the concept of an attribute used for clinical descriptions as opposed to one used to describe for example a laboratory examination.

```
(TopCategory newSub MedicalThing)

    newSub [Condition TopographicalSegment ClinicalModifier Drug].

Condition newSub [Disease Symptom Sign].


Attribute newAttribute DescriptiveAttribute InverseDescriptiveAttribute allAll manyMany.

DescriptiveAttribute newAttribute ModifierAttribute InverseModifierAttribute allAll
manyOne.
```

Figure 6.2 An example high level set of entities in an SMK model

The modelling of simple clinical modifiers and descriptors tends to follow a pattern. An example for severity is shown in figure 6.3 and has the following stages:

(1) define an attribute – *hasSeverity*

(2) define the abstract modifier– *Severity*

(3) define the specific values – *mild*, *moderate*, *severe*

(4) state to which things they apply – in this example any *Condition*

Note that the grammar and possible triples apply to the same entity. This is quite common for entities that are clinical modifiers. Having added the knowledge in figure 6.3 to the model it is now possible to speak of the severity of kinds of *Condition*.

(1) ModifierAttribute **newAttribute** hasSeverity isSeverityOf allAll manyOne.

(2) ClinicalModifier **newSub** Severity.

(3) Severity **newSub** [severe moderate mild].

(4) Condition **triple** hasSeverity Severity [grammatical possible].

"This statement will now be well–formed and return an entity"

Disease **which** hasSeverity severe.

"The following is ill–formed because of the cardinality of hasSeverity"

(Disease **which** hasSeverity severe) **which** hasSeverity mild.

Figure 6.3 The definition of Severity as a modifier of Condition and the sanctioning of severe diseases. The numbers in parentheses () are annotations

6.3.2 A model of fractures

Figure 6.4 shows an example for bones and fractures. There are several new constructs to note. The entity *Fracture* is defined as an elementary kind of *Trauma* **which** *hasLocation Bone*. It is thus indefeasibly true of a *Fracture* that it is in a bone. The attribute *hasLocation* is specialised across the attribute *isPartOf* through the use of the **specialises** operation as discussed in section 5.3.6.

```
"Make a few bones"
        TopographicalSegment newSub Bone.
        Bone newSub LongBone.
        LongBone newSub [Humerus Femur].
"Define the attributes and that hasLocation is specialised across isPartOf"
        DescriptiveAttribute newAttribute hasLocation isLocationOf allAll manyOne.
        DescriptiveAttribute newAttribute hasPart isPartOf allAll manyOne.
        isPartOf specialises hasLocation.
Make a shaft and say that long bones have them"
        TopographicalSegment newSub Shaft.
        LongBone triple hasPart Shaft possible.
"Create trauma"
        Disease newSub Trauma.
"Say that bones can be traumatised"
        Trauma triple hasLocation Bone possible.
"Generate bone trauma and create Fracture"
        (Trauma which hasLocation Bone) newSub Fracture.
"Use names for convenience"
        Fracture which hasLocation Humerus) name FractureOfHumerus.
        Fracture which hasLocation (Shaft which isPartOf Humerus)
                                        name FractureOfShaftOfHumerus.
        Fracture which hasLocation (Shaft which isPartOf LongBone)
                                        name FractureOfShaftOfLongBone.
        (Fracture which
                <hasLocation (Shaft which isPartOf Humerus)
                hasSeverity severe>)
                                        name VeryBadBreak.
"The following are all ill–formed and will generate errors"
        (Fracture which hasLocation Humerus) which hasLocation Femur.
        Humerus which isLocationOf (Fracture which hasLocation Femur).
```

Figure 6.4 An example model for fractures showing the effects of refinement of one attribute across another

A section of the subsumption hierarchy resulting from the source text in figure 6.4 is shown in figure 6.5. *FractureOfShaftOfHumerus* is subsumed by both *FractureOfShaftOfLongBone* and that *FractureOfHumerus*. The second of these is because *hasLocation* is specialised across *isPartOf*.

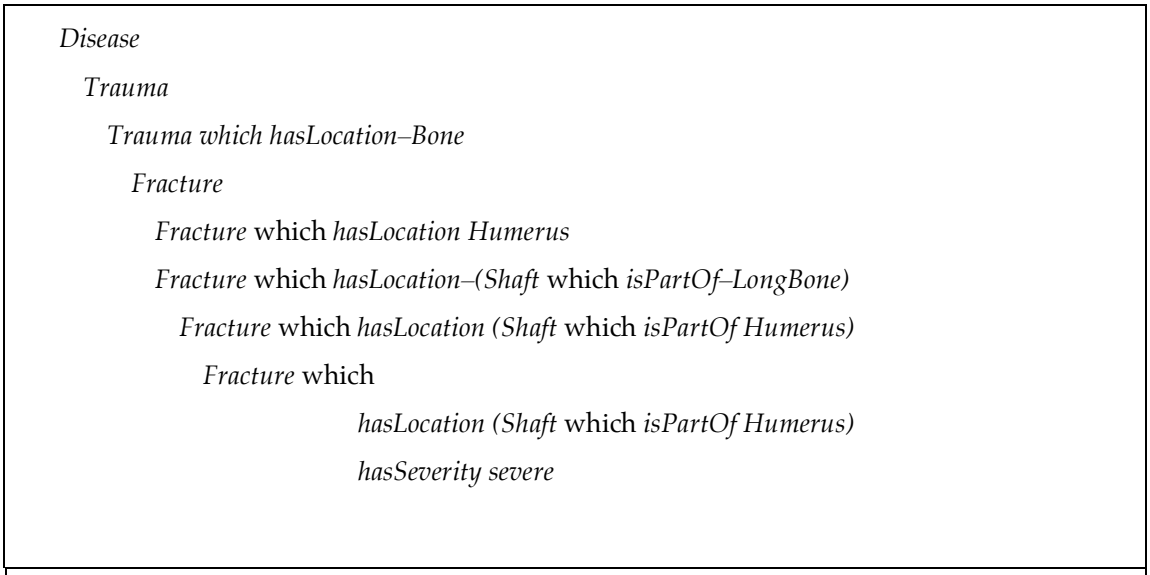

Figure 6.5 Part of the subsumption hierarchy for some types of fractures resulting from the model created in figure 6.4

6.4 Summary

SMK is made available through the use of the SMK language and compiler. Simple operations are used to construct models starting from the fundamental primitives of SMK.

An extended example of modelling is discussed in chapter 8, and the full source text for this model is in appendix 3. Full details of the SMK operations and the syntax of the compiler are to be found in appendix 1.

Chapter 7 Implementation of SMK

This chapter describes the implementation of SMK in a terminology engine which is intended to meet the requirements defined in chapters 4 and 5. The implementation is a prototype but is the basis for the PEN&PAD clinical workstations.

We shall begin by outlining the background to this implementation, and in particular the problems with earlier work in the wider context of the PEN&PAD workstation programme. This will be followed by a description of the main architecture of the SMK terminology engine and the associated tools, and a discussion of some outstanding problems in the implementation of the key theories within SMK.

7.1 Background: early work, implementations, and problems

The earlier implementations of what developed into SMK formed part of the general development of the PEN&PAD prototype clinical workstations. The high level objectives of PEN&PAD are to research and prototype the user–interface, in particular the use of predictive data entry, and the summarisation and presentation of information.

PEN&PAD is trying to unify the use of at least four types of information:

- abstract terminological knowledge about medical concepts - *what it is sensible to say*
- medical records containing data about individual patients - *what has been said*
- assertional knowledge about clinical practice - *what is usually said*
- pragmatic knowledge about interface choices, and behaviours - *how it is to be said*

It is this unified framework which underlies the idea of the intelligent user–interface.

During the early development of SMK and PEN&PAD in general several themes emerged that helped shape the current implementation.

Inappropriate focus on data structures

> The implementation was overly concerned with the particular data structures (nodes and arcs). There was an initial failure to distinguish between those arcs which formed the intensional definition of an entity (criteria) and those which represented substantive assertions about concepts (triples).

Relationship to the object–oriented implementation language

> The implementation language, Smalltalk–80™, is object–oriented. Maintaining a principled relationship between the entities of SMK and the classes of the programming language was difficult. Both 'languages' have hierarchies and ideas of inheritance, and this caused confusion in the early stages of work. The problem was aggravated by the scope of the implementation and initially many of the issues at stake were poorly understood.

Scattered implementation of the theories of SMK

> The focus on data structures together with the object orientation of the programming language resulted in the implementation of the theories of SMK being scattered widely across the various implementation classes. There was no part of the implementation which corresponded in any straight forward way to the terminological theories of SMK.

Interface control and the need for behaviour

The use of the system to control an interface had a major effect on what was needed. One of the arguments behind the technical approach to PEN&PAD is that an interpretation of the medical meaning of the information being manipulated is essential to the support of a 'sensible' interface. Interfaces however are not just about *meaning* they are also about what is supposed to *happen* which implies a behavioural element. An entity in SMK places the emphasis on meaning and little if any on the notion of behaviour. In programming languages such as Smalltalk the purpose of the inheritance hierarchy is to allow in part the sharing of structure, but more importantly the sharing and inheritance of behaviour (see [Snyder 1990] for a discussion of inheritance in programming languages). Given the absence of any means of expressing behaviour in SMK the mechanisms within Smalltalk (methods) were used to represent procedural information.

The problem is that the behaviour required of the interface depends mainly on what an entity means in SMK rather than to which implementation class it belongs. For example the interface may be expected to behave differently if the topic is 'cancer' as opposed to a 'cold'. It is exactly this link which underlies the notion of the intelligent interface, but it was this which created some of the most serious implementation difficulties. Behaviours need to be defined and inherited across the SMK taxonomy, which they could not.

Attempts to overcome this particular problem in earlier implementations resulted in two types of 'work–around'

1 A proliferation of specific implementation classes with specific behaviours. The choice of which implementation class to use to represent an entity was dependent on the meaning of that entity. In the ridiculous extreme there would be a one to one mapping between entities and implementation classes with a distinct implementation class for each concept in medicine.

2 The use of relatively fewer classes but with behaviours of Byzantine complexity which had to be able to distinguish between the meanings of entities. Hence the computer code would contain references to specific concepts in medicine.

This problem confused the work on the main terminological component and needed to be separated from that work.

Having recognised these problems a completely new implementation was developed which began with a limited set of objectives, concentrating on the key terminological operations, and conforming to the theories of well–formedness, and subsumption embodied in SMK. This was to provide a solid foundation for the subsequent development of the PEN&PAD system.

7.2 The SMK Terminology Engine version 2 and tools

A schematic architecture of the implementation is shown in figure 7.1. It comprises two distinct elements:

1 the SMK terminology engine and its three sub–components

2 the tools which provide a basic text based user–interface for the creation of models, and simple inspectors for examining the structure and relationships between entities.

7.2.1 The SMK Terminology Engine

The terminology engine is the implementation of SMK. It comprises three components which broadly correspond to the conceptual levels of the terminology system discussed in chapter 4; the functional description, the interpretation of the terminological theories, and the data structures.

1 The SMK Operations

The *functional description* of SMK is embodied in the SMK Operations Interface to the terminology manager. This defines what the terminological system does through the operations available to an external user of the system. Note that the user in this context is not a

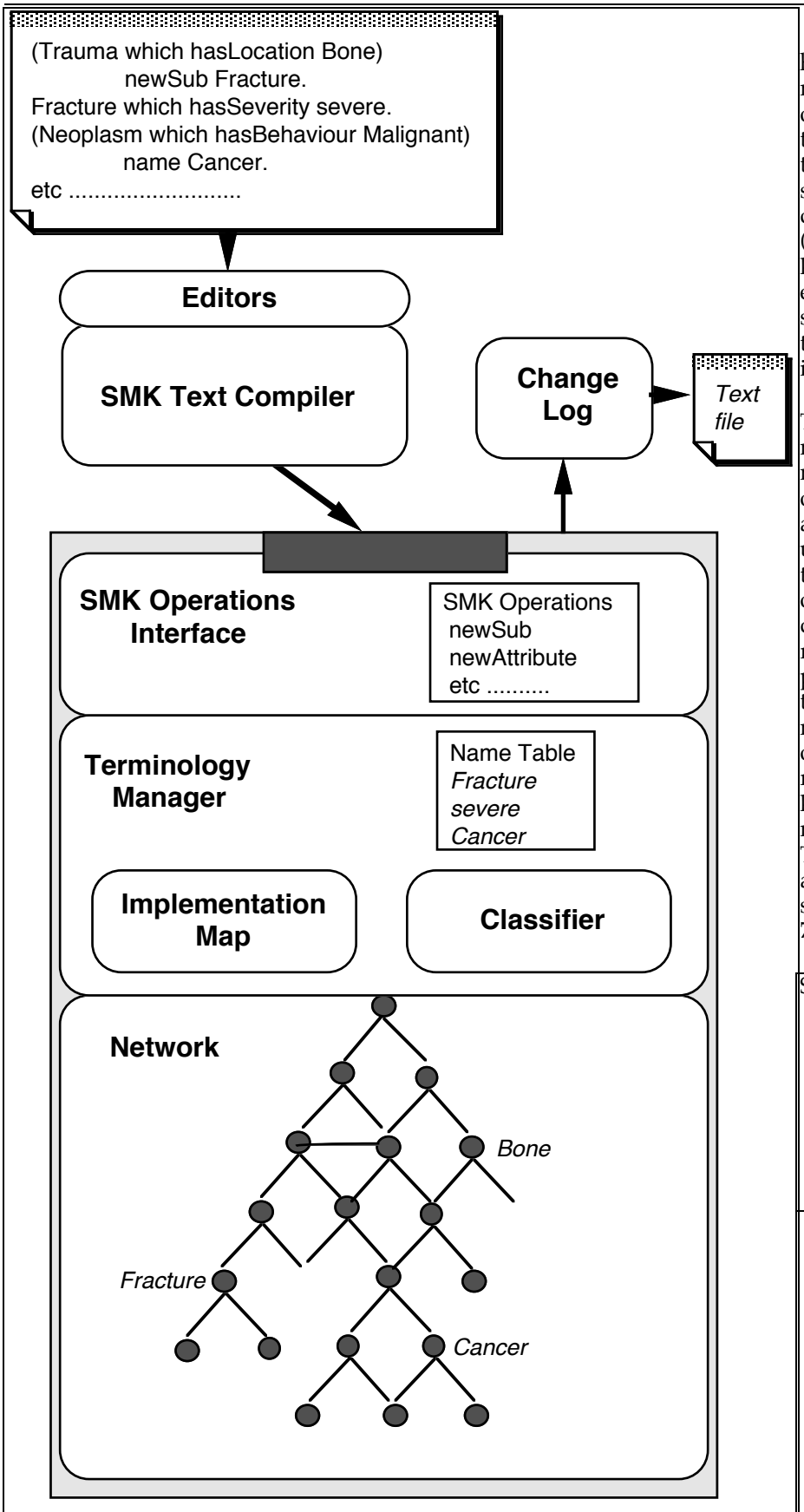

Figure 7.1 Schematic architecture of SMK terminology engine (outlined in the grey box) and related tools

human user. It means another object wanting to treat the terminology system as a complex data type (object). All knowledge editing tools should relate to the engine via this interface.

The terminology manager maintains a table of operations that are available for use by external tools. All the operations are defined in relation to and performed by the terminology manager. An operation is recognised by its keyword which must be unique. The structure of an operation is shown in figure 7.2.

SMKOperation :
keyword:
operand:
arguments:
selector:
adds knowledge:

Figure 7.2

$ t r u c t u r e o f a n SMK operation

An operation is executed by requesting the terminology manager to perform the Smalltalk method

```
performSmkOperation: anOperation
forReceiver: anOperand
arguments: anArrayOfArguments
```

This derives the relevant Smalltalk selector from the definition of the operation and the corresponding method is performed using the given operand and arguments.

2 The Terminology Manager

The *terminological theories* which characterise SMK, such as what it means to be well–formed or the rules for subsumption are embodied in the terminology manager[7] and associated subcomponents. These objects manipulate and maintain the network of entities

- the classifier deals with the canonisation of expressions, coherence, sanctioning of expressions, and maintaining the static hierarchical pointer structure within the network according to the rules of subsumption.

- the implementation map coordinates the relationship between SMK entities and Smalltalk classes. It will be discussed when considering SMK typologies in section 7.5.

- the name table is the mapping from symbolic names to entities represented as objects. It is a key part of the process of providing an external interpretation of entities.

3 The Network

The static *data structure* of the system is implemented as a network of objects. The internal structure of an object is used to represent the properties of an entity, such as its intensional definition, the set of triples which refer to it, and any conventional subsumptions. As well as the internal structure there is also a complete hierarchical pointer structure between the objects which is interpreted as the subsumption hierarchy within SMK. Once an entity has been placed in the network and the pointers computed, its subsumption relationships are available without again resorting to the basic test of subsumption. This process is managed by the classifier.

7.2.2 Tools

There are several basic tools associated with the implementation of the terminology engine. These tools are not comprehensive knowledge editing tools for constructing large terminology models. The motivation for their development was the need to test the implementation of the terminology engine and provide basic support for modelling in PEN&PAD.

SMK Text Compiler

This is a basic text parser and compiler which uses the SMK language described in chapter 6. The text compiler is made available through three text editing tools

SMK File Editor – edits a single file

SMK File Browser – displays lists of files and their contents

SMK Workspace – a scratch pad for temporary use

SMK Change Log

This a basic logging mechanism which records in a text file all those operations that have added terminological knowledge to a particular system. It follows the syntax of the text compiler and is thus in essence a decompiler. It produces a text file which can be compiled by the SMK Text Compiler to restore the knowledge in the original terminological model starting from scratch.

---

[7] For those readers who are already familiar with the implementation of SMK the Terminology Manager described here is exactly what was previously called the Network Manager. The original name did not indicate that the specific theories of SMK are implemented in this and associated objects.

It is particularly useful when a tool other than the text compiler is used to add knowledge to the system. For example a direct manipulation graphical browsing tool has been developed as part of the wider programme of work. This interacts with the terminology engine using the standard SMK operations interface and hence a standard text change log is generated. In this way the change log together with the text compiler provide a primitive import–export facility for moving knowledge between systems, relatively independently of the precise tools used to create that knowledge.

SMK Entity Inspector

This is a tool for inspecting an entity or set of entities. It displays a hierarchy of the entity and all its 'parents' in the subsumption hierarchy. Any of these can be selected and the internal structure of the object then inspected.

7.3 Implementation of the Terminology Manager and associated components

The terminology manager is the implementation of the main theories embodied in SMK. There are three main subcomponents to the terminology manager; the classifier, the name table, and the implementation map. The last of these will be discussed in section 7.8 when describing the instantiation or reification of entities as objects.

7.3.1 Terminology manager (network manager)

There is one terminology manager for any network. Conceptually the terminology manager encapsulates the network. The current implementation of SMK is restricted to a single network within any one Smalltalk image, but in principle it would be perfectly possible to support multiple distinct networks.

In particular the terminology manager coordinates the

- interface to external users (SMK operations interface)
- creation of entities through the implementation map
- identification and naming of entities through the name table
- classifiers which handle the coherence and classification of entities

7.3.2 Naming and the name table

The name table maps symbols (*Cough*) to objects representing entities. It takes care of identifiers as well as names and ensures that taken together they are unique. The name table is used by external agents to refer to entities, via the operations interface. The inverse mapping of entity to name is represented in the individual objects.

7.3.3 The classifier: self–consistency, sanctioning, and classification

The name classifier is somewhat of a misnomer because a classifier handles the determination of canonical forms and well–formedness as well as the problem of classification. The types of information required for each of these tasks is similar, concentrating on the relationships between criteria, and thus they were brought together for the sake of efficiency. A variant of the classifier handles the insertion of triples which can be viewed essentially as a classificatory process. However we shall deal primarily with the requirement to sanction and classify a description.

Evaluation of expressions

A new instance of a classifier is created to deal with the evaluation of every expression. A classifier behaves like a proto–entity and goes about its task by trying to prove it can exist and finding its place in the network. Entities which have been instantiated or installed in the network are described as reified. A classifier has all the properties that it requires to behave like a reified entity for the purpose of performing its job. For example subsumption testing may be performed between a reified entity and a classifier. This polymorphism is one of the useful features of the object–oriented programming environment.

At the start of the evaluation of an expression intended to represent an entity, a newly created classifier encapsulates the entity which is to be described. This is the entity *EpithelialCellNeoplasm* in the example

EpithelialCellNeoplam **which**
hasBehaviour–Malignant
hasSpread–Secondary
hasClinicalStage–StageIV

This is stage 1 in figure 7.3.

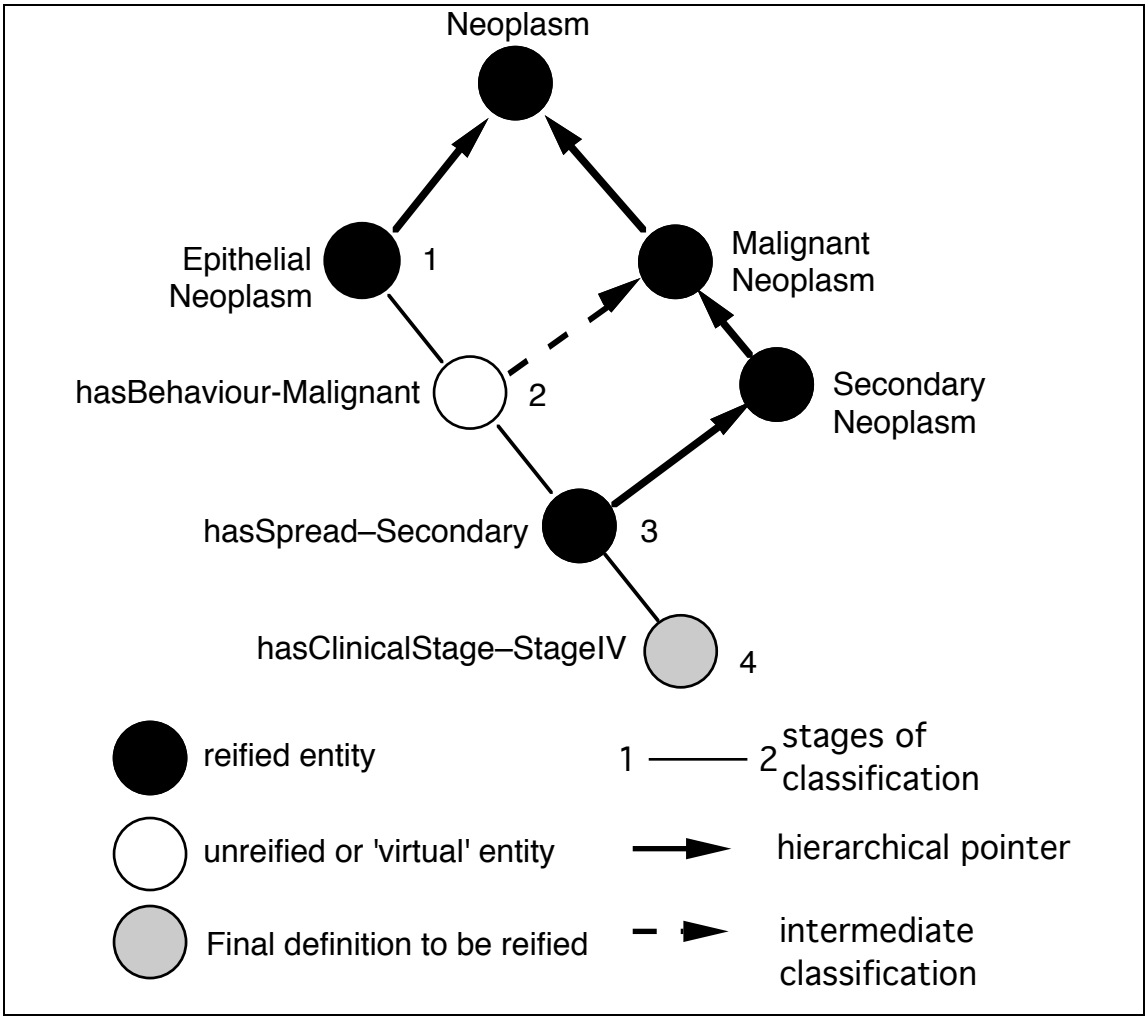

Figure 7.3: Progressive stages of evaluation of a complex expression by the classifier

The classifier then attempts to create a new definition by applying the criterion *hasBehaviour–Malignant*. There are three steps to this procedure which embody most of the underlying theory of SMK

i determine the *canonical form* of the necessary criteria set formed by adding the new criterion, and check it is *coherent*

ii *sanction* the new criterion by finding a suitable source triple, probably inherited from another entity

iii *classify* the resulting definition to determine the set of concepts which subsume the new definition. At the end of this step the classifier itself is classified.

This takes the classifier to stage 2 in figure 7.3. No reified entity is found in the network at the 'location' of the classifier's definition and so the classifier retains the extended definition and classification, and behaves as a 'virtual entity'. This allows the classifier to discover all the knowledge that is pertinent to an intermediate entity without having to reify it within the network. This is essential to keeping the network sparse. The classifier at stage 2 is subsumed by *MalignantNeoplasm* which, for example, holds the statement that malignant neoplasms may spread and be secondary. Thus the classifier is able to repeat the above process and sanction the next criterion to arrive at stage 3. There is a reified entity at this point and the classifier thus encapsulates it and proceeds to the last stage of the description.

At stage 4 no reified entity is found and the evaluation is complete. The definition contained within the classifier is then used to reify that entity. The formal subsumptions are completed and the entity installed in the network. This entire process is atomic and when completed the entity is fully installed. If a reified entity is found at the final stage then it is simply returned by the classifier.

The process described above outlines the means by which a classifier handles expressions. The third stage of the process, the actual classification itself, is the one which has caused the most difficulties and is undoubtedly still incomplete. There are two elements to this process

- the test for subsumption
- the search strategy which must ensure that all relevant entities are tested

The first implementation of the classifier was mainly concerned with being complete and performed an exhaustive test of all entities in the network using a naive implementation of subsumption. This worked well but was felt not to be a practical proposition for large networks. Two optimisations were performed.

7.3.4 Optimisation of the test of criterial subsumption

Most of the theories of SMK involve testing subsumption between criteria. The implementation tries to optimise this by:

- ensuring criteria are unique: that is there is only one object in the implementation corresponding to a given attribute–value pair;
- maintaining a complete hierarchy amongst those criteria based on the rules for criterial subsumption.

In this way criteria are handled very much like first class entities. However are 'confined' to the terminology engine and only emerge to external users as part of the structure of entities. A criterion can never be the result of a user operation. There is an overhead on maintaining this structure but it makes subsequent testing efficient.

7.3.5 Optimisation of the search strategy

The aim of this optimisation is to simplify the problem of determining the set of entities *subsumed* by a newly reified entity. The principle is that the classifier should always be inserting the new entity *between* entities or as a *true leaf* entity. To ensure this requires the presence of a 'top prototype' for every attribute. This is an abstract prototype which relates *TopThing* to *TopThing* and represents the most general form of entity whose description includes the relevant attribute. For example

TopThing which: *hasLocation–TopThing*

can be read as 'anything which has a location of any sort'. It will subsume everything which contains a criterion for *hasLocation*. The usual sanctioning mechanisms are bypassed to reify this abstract entity. When a new description is formed by the addition to an existing description of a criterion for *hasLocation*, it can be guaranteed that there is at least one entity which will subsume the new definition as a result of the criterion for *hasLocation*.

In general the set A of entities that a new entity E must subsume, can be determined from the set B of entities that the classifier has determined subsume E. All the members of A will be

already subsumed by at least one member of the set B. In this way no entities are 'lost' to the classifier. Graphically this is rather like being able to 'pick–up' the network based on any attribute and be guaranteed to have hold of all the relevant entities for that attribute (figure 7.4).

The special 'top prototypes' for each attribute are marked as being abstract and are usually invisible to the user. It is worth noting here similarities to the techniques used in implementing conceptual graphs. The search algorithms employed require a complete lattice to be maintained and hence many intermediate concepts are created by the system to ensure that this requirement is satisfied [Sowa 1984].

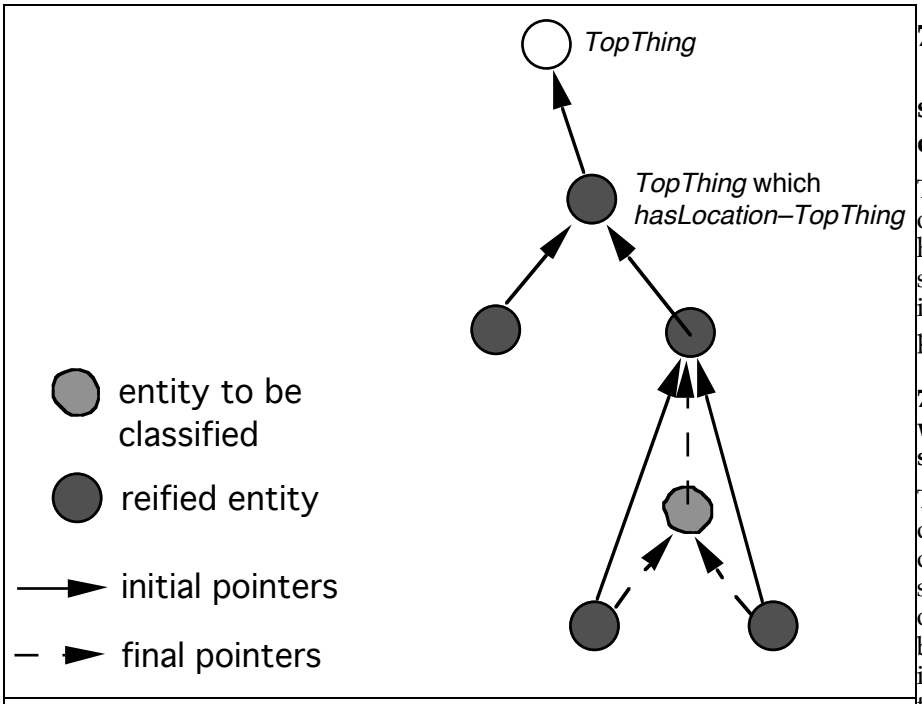

Figure 7.4 Subhierarchy for hasLocation rooted on the abstract prototype *TopThing* which *hasLocation TopThing*, and the insertion of a new entity into the hierarchy

7.4 Problems with the classifier

The strategies outlined above have proved successful but identified problems remain.

7.4.1 Problems with criterial subsumption

These affect the correctness of the classifier. The separate hierarchy of criteria can become inconsistent with the main hierarchy of entities. As a consequence the classifier may fail to derive a subsumption relationship correctly. The problem arises with some assertions, most commonly conventional subsumption. The consequences of these assertions are not fully carried through into the hierarchy of criteria. This issue is a matter of judgement. The hierarchies could be maintained side by side but the overhead is now probably too large. It is felt that a compromise is required which maintains the uniqueness of criteria but allows them to derive their subsumption relationships efficiently from the main hierarchy.

7.4.2 Problems with the search strategy

These affect the completeness of classification. The difficulty is primarily with maintaining the complete and distinct subsumption pathway for each criterion that goes to form the definition of an entity. In certain situations two or more subsumption pathways can degenerate into a single pathway. Several examples will illustrate this.

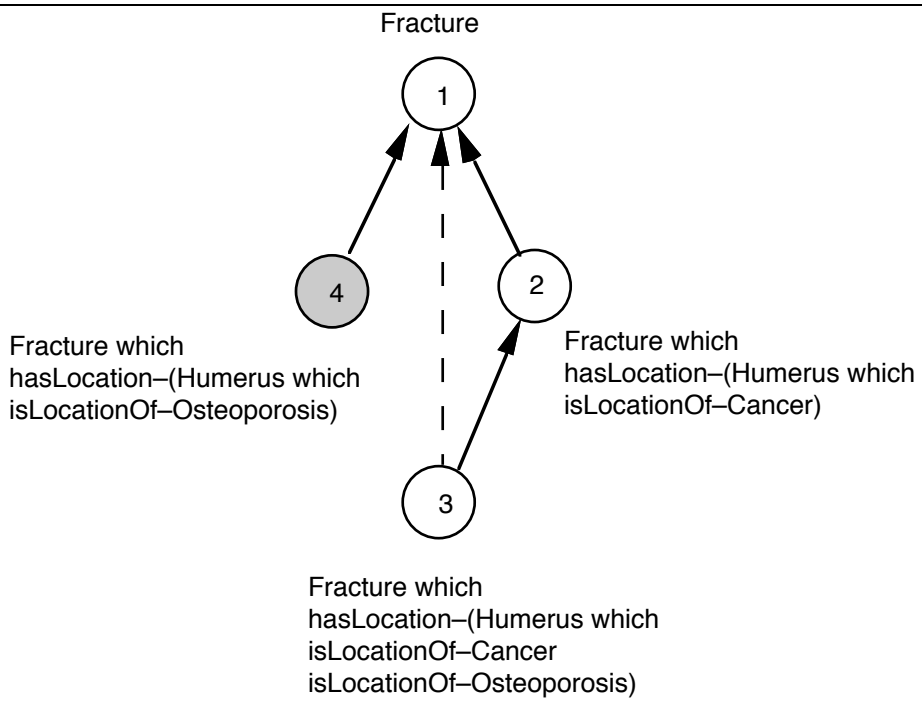

Figure 7.5: Failed subsumption because of multiple criteria

In figure 7.5 the entities 1, 2, and 3 are taken as already being reified in the network with the hierarchy as shown by the solid arrows. The classifier is attempting to place the definition of entity 4. It is trivially subsumed by entity 1. It then proceeds to entity 2 with which it is disjoint and abandons the search unaware of the existence of entity 3.

The problem is that entity 3 has a complex criterion whose value has two aspects, the *Cancer* and the *Osteoporosis*. It correctly relates to entity 2 because of the *Cancer* aspect, but the *Osteoporosis* aspect is lost. The dotted arrow shows that it bears a relationship to *Fracture* because of the *Osteoporosis* aspect but the classifier linearises such a triangular pointer arrangement. The dotted arrow is considered redundant. Unfortunately it is this dotted arrow which would allow the search algorithm to find entity 3. This suggests the need for some form of annotation of subsumption links as to why they are present, in order to avoid degeneracy.

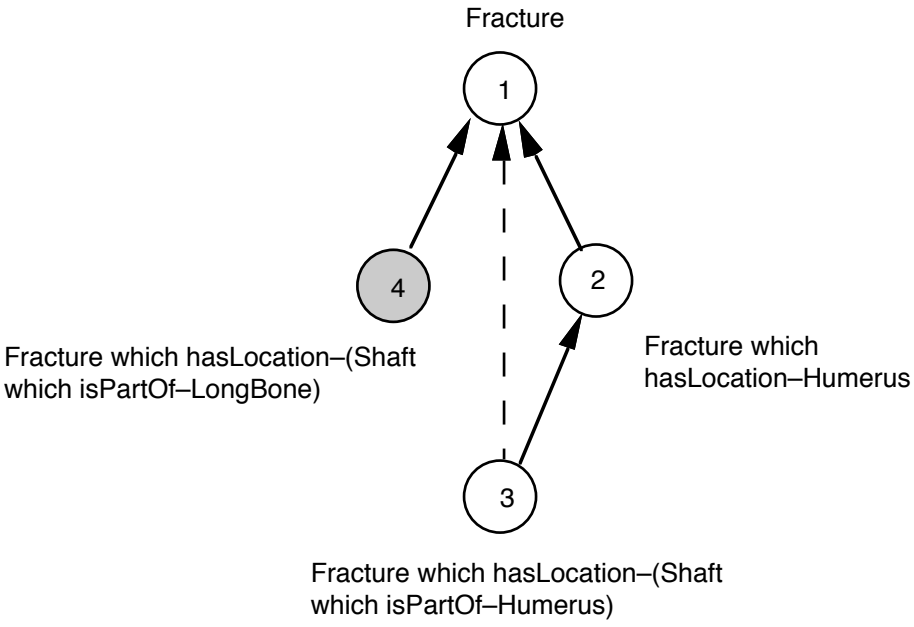

Figure 7.6 Failed subsumption because of refinement of *hasLocation* across *isPartOf*

The second example is similar but relates to the refinement of one attribute across another described in chapter 5. In figure 7.6 there is a similar arrangement to the previous example. This time however the degeneracy is due to refinement acting between attributes. Entity 3 is subsumed by entity 2 because of refinement of *hasLocation* across *isPartOf*. The subsumption is thus a specialised formal subsumption. However entity 3 has a relationship to entity 1 of a straight forward kind without invoking refinement. Unfortunately the triangle is again linearised and the new entity cannot get passed entity 2 to find entity 3. This again suggests the need for annotation which distinguishes simple and specialised subsumptions.

Both examples are aggravated by being dependent on the order in which the entities are reified. In both cases if entity 4 is reified before entity 3 then the resulting network will be correct. It is possible that this sort of dependency and apparent need for annotation to distinguish amongst subsumption links to avoid degeneracy and ensure completeness of searching is similar to the need for a complete lattice in Sowa's conceptual graph algorithms [Sowa 1984].

There is of course the fundamental question of whether tractable algorithms can exist for dealing with subsumption and the process of classification. A formal analysis of the properties of SMK in this respect has not been carried out. It is known however that tractable algorithms exist for related formalisms [Brachman 1985, Schmolze 1984, Rector 1986]. This work on the classifier has succeeded in defining requirements and producing an implementation which has functioned well, with some identified problems. Further work is required to address the outstanding issues.

7.5 The implementation map and SMK typologies

The SMK taxonomies and the Smalltalk classes are coordinated through the notion of typologies. This is managed by the implementation map associated with the terminology manager. SMK typologies are based on an analysis of those fundamental characteristics of entities which are independent of their medical meaning. Four primary characteristics form an orthogonal set of axes for characterising any entity:

Concept: whether the entity is a symbolic concept or one of the primitive value types which in this implementation are limited to string, number, and date entities.

Structure: whether the entity is elementary or complex

Polarity: a slightly unusual choice of name rooted in history but it denotes whether the SMKobject is an entity (node) or a relationship (arc)

SMK space: indicates to which of the three SMK spaces, (category, individual, or occurrence), the entity belongs, plus the slightly odd non space where the *NonEntity* resides. SMK spaces are discussed in chapter 9 on the representation of entities within the medical record.

The details of the typology axes are shown in figure 7.7

concept
- symbolic
- primitive
- string
- number
  - integer
  - float
- date

structure
- elementary
- complex

polarity
- entity
- relationship

SMK space
- category
- individual
- occurrence
- non space

Figure 7.7 The primary SMK typology axes and possible values

The two axes structure and polarity define the four basic SMK typologies

	entity	relationship
elementary	ElementaryEntity	Attribute
complex	Prototype	Triple

7.5.1 Use of typologies and reification

Typologies are a useful abstraction and provide an *inter lingua* between the SMK taxonomy and the Smalltalk class hierarchy. The reification of an entity as an object is dependent upon the following mechanisms.

- As part of the initialisation of the whole terminology engine, specific Smalltalk implementation classes are assigned to the various typologies within the implementation map

- When a new network is created the initial fundamental primitives of the network, such as *TopThing*,, have their typologies defined. All entities have a typology.

- All entity creation is performed through the use of the relevant typology

- One typology can be derived from another by a simple manipulation of the relevant axes. For example to create a prototype from an elementary entity simply requires switching the concept axis from *elementary* to *complex*.

- Given a typology for a new entity the relevant implementation class can be determined and the entity reified

This approach has proved very successful in co-ordinating the SMK and Smalltalk taxonomies. In the wider work it also extended to accommodate the external database and thereby coordinated entities, classes, and database types.

7.6 Additional information

The SMK Terminology Engine is implemented in Parc Place Objectworks\Smalltalk™ release 4 or higher. It is supported on Unix™ based, 386/486 Windows™ based, and Apple Macintosh™ platforms. It requires of the order of 8 to 16 MB of real memory.

Appendix 2 gives the full table of SMK objects and a definition of their structure.

Appendix 1 describes and summarises each of the main SMK operations, the compiler syntax and compiler operations, and some notes on the implementation details of the SMK operations interface and error handling.

8. Further Modelling in SMK and Its Relationship to Coding Schemes
This chapter demonstrates the use of SMK for modelling medical terminology based on an example from the domain of tumour pathology. The resulting model will then be used to represent part of the Read Clinical Classification, and to examine the relationship between an SMK model and traditional coding and classification schemes. The chapter concludes with a discussion of problems encountered in the use of the formalism and the possible need for extensions to the formalism.

8.1 An example model of tumour pathology

The process of modelling is always iterative. It starts with a relatively unstructured view of the domain, and works towards further detail as relationships are identified. The example discussed here has not been the subject of detailed revision and is not presented as a sufficient model of the domain. However it clearly reflects the material on which it is based and thus illustrates the first steps in modelling using SMK.

8.1.1 Source of the terminology

The content of the SMK model is derived from the commentary on the classification of tumours in the International Classification of Diseases [WHO 78]. This commentary identifies four important aspects or characteristics of a tumour; whether the tumour is benign or malignant, the degree of spread, the normal tissue type from which it is derived, and its cellular morphology (appearance). The other aspect considered is the anatomical location of the tumour. With a single exception this has been omitted from the SMK model. To deal adequately with this requires a detailed model of anatomical concepts that is beyond the scope of this exercise. The complete SMK source text for the model is shown in appendix 4.

8.1.2 Main aspects of the model

There are five primary components to the model.

1 The fundamental representation of neoplasia – 'new growth':

NeoplasticProliferation	–	any abnormal growth of cells
Neoplasm	–	a solid neoplastic proliferation (tumour) represented here as an elementary entity

2 Four attributes corresponding to the four characteristics used to describe neoplasia:

hasNeoplasticBehaviour	–	whether a tumour is benign or malignant
hasMetastaticState	–	the degree of spread (metastasis) of a malignant tumour – primary or secondary
hasCellTissueType	–	the normal cell tissue type from which the tumour is derived eg. epithelial cell
hasCellMorphology	–	the particular appearance of the tumour cells eg. spindle cell

3 A submodel of entities that are suitable values for each attribute. The possible values for the first two attributes are straightforward. The cell tissue type and morphology are more complex. The relevant subsection of the source for the SMK model is shown in figure 8.1. No attempt was made to identify defining characteristics for these entities and this part of the model is built entirely from elementary entities and conventional subsumption. This is typical of small subsections of a model, particular in the early stages of development.

```
"Cell types and morphologies"
MedicalThing newSub CellType.
CellType newSub [CellTissueType CellMorphology].
CellMorphology newSub [ SmallCell LargeCell FusiformCell Anaplastic Pleomorphic
                        SpindleCell PolygonalCell SpheroidalCell Verrucous].
SmallCell newSub OatCell.
LargeCell newSub GiantCell.
CellTissueType newSub [ EpithelialCell PigmentCell NeuralTissueCell
                        ConnectiveTissueCell].
EpithelialCell newSub [ SquamousCell GlandularCell BasalCell TransitionalCell].
SquamousCell newSub PapillaryCell.
```

Figure 8.1 Subsection of neoplasia model covering the declaration of cell tissue types and morphologies

4 A pair of grammatical and possible statements for each of the four attributes, relating the appropriate neoplastic concept to appropriate values. For example any *NeoplasticProliferation* can be benign or malignant, hence

NeoplasticProliferation **triple** hasNeoplasticBehaviour NeoplasticBehaviour [grammatical possible].

However only malignant neoplasm can be spread, hence the statement

(Neoplasm **which** hasNeoplasticBehaviour malignant)
triple hasMetastaticState MetastaticState [grammatical possible].

5 Names for concepts to make subsequent expressions more compact and readable:

(Neoplasm **which** hasNeoplasticBehaviour malignant) **name** Cancer.

Note that naming does not add substantive knowledge to the model. No additional concepts can be expressed as a result of naming.

8.1.3 Consequences of the model

Figure 8.2 shows the hierarchy of neoplastic concepts that results from the model.

```
NeoplasticProliferation
------MalignantNeoplasticProliferation
------ ---- Cancer        (malignant neoplasm)
------ ---- ---- PrimaryCancer
------ ---- ---- Carcinoma        (malignant epithelial cell neoplasm)
------ ---- ---- -----Adenocarcinoma
------ ---- ---- -----{Neoplasm which
                          hasCellTissueType-SquamousCell
                          hasNeoplasticBehaviour-malignant }
------ ---- ---- -----{Neoplasm which
                          hasCellMorphology-Keratinising
                          hasCellTissueType-EpithelialCell
                          hasNeoplasticBehaviour-malignant }
------ ---- ---- -----{Neoplasm which
                          hasCellMorphology-NonKeratinising
                          hasCellTissueType-EpithelialCell
                          hasNeoplasticBehaviour-malignant }
------ ---- ---- SecondaryCancer
------ ---- DiffuseNeoplasticProliferation
------BenignNeoplasticProliferation
------ ---- BenignNeoplasm
------ ---- ---- Epithelioma
------ ---- ---- -----Adenoma
------Neoplasm
------ ---- Cancer ....^2
------ ---- BenignNeoplasm ....^2
------ ---- EpithelialNeoplasm
------ ---- ---- Carcinoma ....^2
------ ---- ---- Epithelioma ....^2
```

Figure 8.2 The formal subsumption hierarchy for neoplasms generated from the model of tumour pathology

Indentation indicates subsumption. When a concept is repeated the hierarchy is truncated, and this is indicated by the use of .....^n where n is the number of times the concept has occurred. The hierarchy has been printed using names when a name is defined for an entity. For example *Cancer* is the name for *Neoplasm* which *hasNeoplasticBehaviour–malignant*.

There are several important points to note:

1 all the entities in the hierarchy with the exceptions of *NeoplasticProliferation* and *Neoplasm* are **prototypes**. Most have been given names but these correspond to complex expressions.

2 The hierarchy is formed entirely by formal subsumption based on the definitions of the entities. For example *Cancer* subsumes *Carcinoma* because the former a 'malignant neoplasm' and the latter is a 'malignant epithelial cell neoplasm'.

3 The hierarchy is multiaxial with concepts appearing several times. For example *Carcinoma* is both a *Cancer* and an *EpithelialCellNeoplasm*

8.1.4 The use of elementary concepts and the limit on what is modelled

The use of elementary concepts reflects the limitations on what it is appropriate or possible to model. For example the entity *EpithelialCell* is elementary. However it could be possible to extend the model and construct a definition for this cell tissue type, based on the principles of histology and cell–biology. However such definitions are not immediately relevant in a model concentrating on clinical descriptions. Intracellular organelles and the electron microscopy of cell membranes do not form part of everyday clinical discourse. That is not to say a basic biological component to the model would be wrong, it is simply not essential. However it may be developed later if the scope of the model is widened and a new dimension added.

The addition of a new dimension to the model could be used to 'open up' elementary entities and provide definitions If this new dimension is orthogonal to the current ones then it will not cause a major disruption to the existing model. Furthermore the ability to name concepts would allow the name *EpithelialCell* to persist even if it became a complex concept. This localisation or decomposition of terminological knowledge in a model is an important feature of SMK.

8.2 Relationship between SMK and traditional coding and classification schemes

We shall now use the above model of tumour pathology to examine the relationship between SMK and traditional coding schemes.

8.2.1 Representational transformations between coding schemes and SMK

The terms of a coding scheme and a model in SMK are alternative representations of medical concepts. If the scheme is a simple classification then the terms comprise a code and rubric with classificatory relationships between them

<code1>–<rubric1>
----------<code2>–<rubric2>
---------- ----------<code3>–<rubric3>

For example in the Read Clinical Classification (4-digit version)

BB00–Neoplasm, benign
----------BB10–Epithelial tumour, benign
---------- ----------BB11 etc .....

The coding scheme contains two sorts of knowledge:

1 the meaning of the terms as inferred by reading the rubrics;

2 the codes and classificatory relationships between the terms.

It is the codes which are semi–formal but the rubrics which contain the vast bulk of the medical knowledge.

Consider a small SMK model, TM, intended to cover the concepts in a section of a coding scheme CS. For each code (term), c_i, in CS we will try to write an expression, e_i, consistent with TM. This creates a simple mapping

```
(Neoplasm which
        <hasNeoplasticBehaviour malignant
        hasCellTissueType EpithelialCell
        hasCellMorphology SmallCell>)
                public 'BB1J Small cell carcinoma NOS'.

    (Neoplasm which
        <hasNeoplasticBehaviour malignant
        hasCellTissueType EpithelialCell
        hasCellMorphology OatCell>)
                public 'BB1K Oat cell carcinoma'.

    (Neoplasm which
        <hasNeoplasticBehaviour malignant
        hasCellTissueType EpithelialCell
        hasCellMorphology SmallCell
        hasCellMorphology FusiformCell>)
                public 'BB1L Small cell ca.,fusiform'.

    (Neoplasm which
        hasCellTissueType SquamousCell)
                public 'BB2 Papill./squamous cell neop.'.
```

Figure 8.3 Example mapping of SMK expressions to Read Codes

CS TM

$c_i \Leftrightarrow e_i$

For example

BB00–Neoplasm, benign ⇔ *Neoplasm* which *hasBehaviour– benign*

If this is done for all terms in CS then the left to right mapping is complete. However it is unlikely that the right to left mapping will thereby be complete. There will be concepts consistent with TM which have no direct mapping to a term in CS. TM will be **sufficient** to cover CS but it need not **only** cover CS. In fact it would be disappointing if it that was all it could do.

The mapping does not include the classificatory relationships in CS and their correspondence with subsumption or other relationships in TM. No attempt will be made to explicitly transfer the classification from CS to TM. In fact the formal hierarchy within TM will be used to critique and extend the classification of CS. This is potentially a powerful use of formal models and a major motivation for performing the mapping.

8.2.2 Representation in SMK of a section of the Read Clinical Classification

We shall use the SMK model of tumour pathology to cover a small part of the Read Clinical Classification (4–digit version) dealing with neoplasms. The mapping is illustrative and is represented naively by using the Read Code and rubric as the public name of the corresponding entity:

<expression> **public** 'string of code and rubric'

This is not the basis of a practical transformation system, but it serves to show the principles and the main results. A small section of the mapping is shown in figure 8.3 and the full mapping in appendix 3.

The most important point about the set of statements in figure 8.3 is that they only involve naming and do not add any new knowledge to the model in SMK. All the concepts used in the mapping are implied by the model of tumour pathology described earlier. The statements only identify those entities which correspond to a Read Code.

The mapping is does not cover all of the terms in the relevant section of the Read Clinical Classification. The main reasons for this are:

- terms using NOS (not otherwise specified) were difficult to interpret because they imply exclusivity and depend on which terms are present in the rest of the scheme;
- terms involving 'Other' appear to have been used to distribute codes evenly amongst the levels of the coding scheme to aid navigation, and do not constitute useful concepts in the SMK model;
- the focus of the model is on the morphology and behaviour of tumours. Hence terms which require detailed anatomical modelling have been omitted.

8.2.3 The mapping of codes to entities

The mapping covers 58 Read codes, while the SMK model comprises 49 items of knowledge of which there are

35 elementary types
4 attributes
10 triples

All the attributes were required for the mapping but several of the elementary entities were not used. The count of entities also includes abstract concepts such as *NeoplasticProliferation* that have no counterpart in the coding scheme. The SMK model can represent far more than 58 concepts[8], but these have not been examined to see if all are sensible. It is typical however for the initial effort on a model to be relatively large. The pay–off comes as more and more implied concepts are dealt with.

8.2.4 Hierarchical relationships between 'codes' derived from the SMK model

The number of terms mapped to entities is only part of the comparison. The relationships between the entities in the SMK model are important. Figure 8.4 shows part of the hierarchy for those entities in the model that have a corresponding Read Code. The full hierarchy is in appendix 3. Figure 8.4 concentrates on the code 'BB2E Squamous ca.,small,non-ker'[9].

---

8 Based on the four attributes, their possible values, the dependency of *hasMetastaticState* on *hasNeoplasticBehaviour*, and assuming only a single value for each attribute, an estimate is (1 + 1 x (1 + 5)) x 10 x 15 = 1050

9 This is the original RCC abbreviation for squamous cell carcinoma, small–cell, non–keratinising

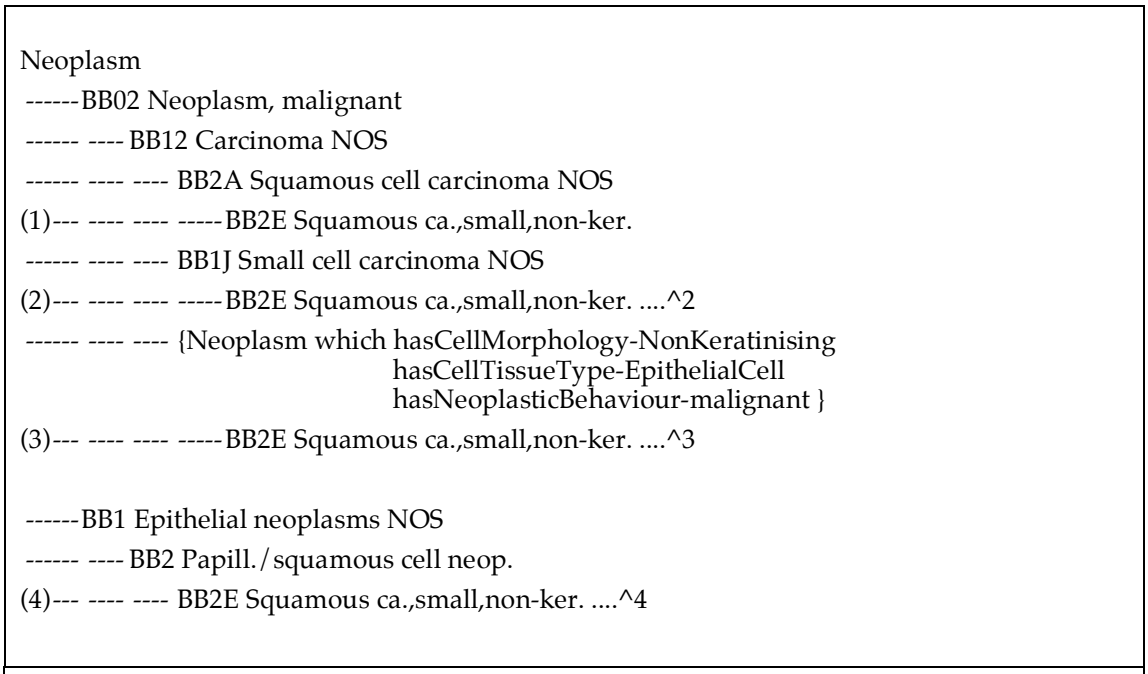

Figure 8.4 Part of the subsumption hierarchy of entities corresponding to Read Codes, derived from the SMK model of tumour pathology, concentrating on the code BB2E.

The entities have formed a dense network of formal subsumption relationships based on the use of four attributes. The code 'BB2E' occurs four times, once for each of the facets of its definition, numbered 1 to 4 in figure 8.4. It is a malignant, epithelial, small cell, keratinising neoplasm. Note also the relationships of the Read Code numbers to each other. In general the 'BB1' codes have subsumed the 'BB2' codes showing that the original Read hierarchy is misaligned.

8.2.5 Potential benefits from the use of formal relationships

The comparison of the relationships in the SMK model and those in the original coding scheme have only been briefly inspected. A more principled approach would treat the classificatory relationships in the coding scheme as subsumption, and do a comparison of those relationships derived from the coding scheme and those inferred by the formal model. This would not only test the concurrence but indicate possible mis–classifications in the coding scheme and additional relationships between terms in the coding scheme.

The logical extension of this technique is to use the formal SMK model to define the relationships to be used in the coding scheme. In this approach the coding scheme is used as a nomenclature, and the relationships between its terms are derived via its representation in a formal model. If a uni–axial scheme is required then a ordering could be imposed on the attributes. For example tumours could be refined first by malignancy, then by tissue type, and then by cell morphology. The result would be a coding scheme that is a subset of the original model in a 'precompiled form'. In this form it may be more convenient for use by some applications, but has the benefit of being derived from, and consistent with, the larger formal model.

8.3 Problems with the model and limitations on the formalism

The problems of representing some of the constructs found in the coding scheme have been mentioned above. A discussion of a specific problem with part of the model will highlight several deficiencies and suggest the need for further developments.

The attribute *hasCellMorphology* is defined with a cardinality of **many**. This was done because some tumours are described as having several morphologies, for example 'mixed giant and spindle cell carcinoma'. A problem then arises when two or more morphologies are required to

be mutually exclusive. Keritanising and non–keritanising carcinoma is an example in the model. This has been handled by:

1 defining the two morphologies as kinds of the cell morphology of squamous cell carcinoma:

(CellMorphology **which** isCellMorphologyOf (Carcinoma **which** hasCellTissueType SquamousCell))
newSub [Keratinising NonKeratinising].

2 using two null statements to assert that keratinising carcinomas cannot be non–keratinising and *vice versa*:

(Carcinoma **which** hasCellMorphology Keratinising)
triple hasCellMorphology NonKeratinising null.

(Carcinoma **which** hasCellMorphology NonKeratinising)
triple hasCellMorphology Keratinising null.

The null qualifier has not been discussed before. It defeats a previously made possibility statement. It is now specifically not possible to describe a keratinising carcinoma as non–keratinising and *vice versa*. Note however that defeating a *possibility statement* is not the same as defeating part of the *definition of an entity* (criterion), which is always forbidden.

This solution is awkward and four possible other options are discussed below.

Option 1: define a new relationship

It is possible to define a new attribute specifically for keritanisation and give it a cardinality of one, for example *hasKeritanisation*. This has the drawback of making significant additions to the model for a relatively small conceptual extension. It also requires keritanisation to be something other than a morphology and results in fragmentation of the model. This may be the correct view, indicating the need for more detail and structure in the submodel of morphologies.

Option 2: restrict the cardinality and define special conjunctions

In this solution the cardinality of *hasCellMorphology* is reduced to one and special conjunctions are defined, such as *GiantAndSpindleCell*, to handle the mixed morphologies. This implies a composite giant–spindle cell which is different to a mixture of giant and spindle cells. Furthermore this solution requires changes to the model remote from the site of the original problem with keratinisation.

Option 3: permit some forms of negation

Keritanising and non–keritanising are complements, but negation is specifically excluded from SMK. However a solution does not require unrestricted negation. For example

Carcinoma AND (NOT *KeritanisingCarcinoma*)

means a kind of carcinoma, but not one that is keratinising, that is a 'non–keratinising carcinoma'. This type of relative complement is believed to be tractable. It suggests the need to extend definitions of entities to allow for criteria which must be excluded. This is close to the spirit of the current solution but a principled version is necessary.

Option 4: allow a more precise specification of cardinality

Cardinality could be defined more locally, in relation to possibility triples or even criteria within a specific context. This approach has general applicability to more than pairs of complementary entities. At present a set of values can be either independent or mutually exclusive. However, it would be useful if for a set of independent values, it is possible to define a subset of those values that are mutually exclusive.

Cases 1 and 2 extend the model but require some uncomfortable manœuvres that affect the model outside of the immediate area of interest. Cases 3 and 4 extend the formalism, but could provide more intuitive and localised solutions. Both of cases 3 and 4 are under consideration.

8.4 Summary of the experiment in modelling

SMK is useful for modelling medical terminology, as demonstrated by the example on tumour pathology. What is encouraging is that the major medical aspects of the informal model described in ICD–9 corresponded well to formal constructs in SMK. It was relatively easy to take the important first step in building an overall framework for the model which could then be examined and refined. As a result of constructing the model it was possible to derive explicit relationships between concepts that were not enumerated in the original source corpus.

The same SMK model was used to represent the terms from a section of the Read Clinical Classification. The model then automatically generated the complex hierarchical relationships between those terms implied by their formal definitions. This showed the original classification of those terms within the coding scheme to be both incomplete and incorrect. Furthermore the structure of the model and the definitions helped to explain why. This relationship between the formal model and the classification scheme is an important result. It suggests that formal representations can be used to improve the structure of existing classification schemes without having to replace them in end user applications in a single step. This result will be important in constructing practical development pathways for clinical systems.

Problems with the use of the formalism suggested the need for extensions to the formalism, in particular relative negation and a more detailed handling of cardinality.

Chapter 9 The Representation of Medical Records Using SMK

The principle purpose for developing SMK within the PEN&PAD programme of research was to support the representation of detailed, structured clinical records of individual patients. This chapter is not a full account of the model of the medical record used by PEN&PAD. The account is limited to the features of the SMK formalism that are important in supporting that model. This chapter begins by considering the relationship between terminology models and information models of the medical record. It then describes the use of SMK for representing the medical record as observations, and some of the consequences of the approach.

9.1 Terminology models and information models of the medical record

This section is a brief examination of the relationship between information models and terminology models and the trade–off between the two.

9.1.1 A basic information model of a medical record

The information model of the medical record describes how data items are to be used to form the record of an individual patient. For example a medical record system may be based on an information model that requires the date, clinician, patient and disease to be recorded

date	clinician	patient	disease
12.01.90	Dr Jones	Mrs Smith	pneumonia

The implication here is that the value 'pneumonia' for the field disease is to be found in a relevant terminology (domain). For example with a simple coding scheme the value will be a code. Note that there may be more than a single field whose domain of values is in the terminology. The interpretation of the complete relationship between date, clinician, patient, and disease is performed by an interpreter of the information model.

9.1.2 The trade–off between the terminology and information models

We shall now extend the requirement on the clinical content of the description to include the severity of the disease. To achieve this there are two options.

1 extend the information model and add a 'field' for severity

date	clinician	patient	disease	severity
12.01.90	Dr Jones	Mrs Smith	pneumonia	severe

2 extend the terminology to include the concept of severe–pneumonia

date	clinician	patient	disease

12.01.90	Dr Jones	Mrs Smith	severe–pneumonia

The first case increases the complexity of the information model but makes no demands on the terminology model. The second has the reverse effect. This demonstrates the trade–off between what is represented in the terminology model and what is in the information model and hence recorded in the associated database.

9.1.3 Consequences of the trade–off

The choice of whether to extend the information model or the terminology model is somewhat arbitrary, and depends on the uses for the overall system. However there are important considerations. If the terminology model is extended:

- a standard information model can accomodate 'new' medical descriptions. For example the same information model can handle the description 'severe–bilateral–basal–klebsiella–pneumonia' provided it can be defined in the terminology model.

- the description 'severe–pneumonia' can be interpreted with respect to the terminology model. For example it 'is a kind of' pneumonia.

If the information model is extended:

- no demands are made on the terminology model lessening the risks of the combinatorial explosion or excessive technical complexity

- the information model must respond to changes in clinical requirements, and these changes are not localisable. For example the 'field' severity is added to the entire model even if it is only required to describe a few diseases

- it is difficult to standardise the terminology. Different systems will adopt different information models and hence clinical descriptions. There will be no core terminology to assist in the integration of and communication between systems.

For the case of 'severe pneumonia' there is no clear choice as to which model is responsible for the modifier. If the description is 'myocardial infarction' the responsibility is likely to fall to the terminology model. In contrast it is unusual to find the individual patient suffering the pneumonia within the terminology model:

date	**clinician**	**patient–disease**
12.01.90	Dr Jones	Mrs Smith's–severe–pneumonia

The concept 'Mrs Smith's severe pneumonia' is a very special kind of pneumonia. However it is this approach that is adopted by PEN&PAD and determines the requirements on SMK.

9.2 The medical record in PEN&PAD

This section concentrates on the extension to SMK to permit descriptions of entities representing real things in the world such as a named patient. It is not a detailed account of the information model of the medical record adopted within PEN&PAD. The requirements for the medical record and its information model can be found elsewhere [Rector 1991 & 1992]. However two aspects of the approach to the medical record are relevant to this account of SMK:

1 the medical record is an account of what 'has been said' with the terminology model being a model of what 'can be said'. In this view the terminology model completely subsumes the information model. The information objects in the record are the 'instantiations' of the abstract concepts in the terminology model, and are subject to the same constraints as that model. This is an extreme view of the trade–off between the two models.

2 all observations of the patient are made by an observer, at a specific time and place

The medical record within SMK is a network of complex entities such as

johnSmith which
isSeenBy-(*drJones* which
isDoctorAt–(*theClinic* which
isOnDate–25January1989)

We shall now examine the formation and properties of such complex descriptions.

9.2.1 The spaces of SMK: categories, individuals, and occurrences

The information objects which comprise the record of an individual patient are also entities. Thus an SMK entity belongs to one of three SMK spaces.

Category space

Entities in category space represent abstract concepts. All the SMK models considered so far have contained only categories. Categories are akin to classes in other representations. For example

Fracture

Fracture which *hasLocation–Humerus*

Individual space

An individual is an entity which represents a concrete thing in the world. Examples of individuals are *johnSmith*, *drJones*, and *theClinic*. Individuals are the instantiations or extensions of the categories. For example *johnSmith* is an elementary individuation of the category *Patient*.

johnSmith

A prototype may also be an individual if its base type or one of its criteria is an individual. For example

Fracture which *isHadBy–johnSmith*

This an individual fracture, 'John Smith's Fracture'.

Elementary individuals such as *johnSmith* are only permitted for entities that have a concrete extension, such as people and places. It is not possible to create an elementary individual of a medical concept such as diabetes.

Occurrence space

Occurrences are used to represent the observations in the record. An entity is an occurrence if as part of its formal definition there is reference to an entity representing a specific observer and date. For example

johnSmith which
isSeenBy-(*drJones* which
isDoctorAt–(*theClinic* which
isOnDate–25January1989)

is an occurrence of the individual johnSmith. It corresponds to 'John Smith when seen by Dr Jones at The Clinic on 25 January 1989'. The only elementary occurrences are dates.

9.2.2 The relationship between SMK spaces

The properties of individuals and ocurrences are no different to those of categories, with the same definitions of well–formedness and formal subsumption. Conventional subsumption is forbidden for individuals and occurrences, with the exception being the link between an elementary individual and its category. Thus only formal subsumption can occur and as a result the three spaces go to form three layers in the subsumption hierarchy (figure 10.1). SMK

thus has two levels of instantiation, category to individual, and individual to occurrence. This is in contrast to the more usual single level of class to instance found in other representations.

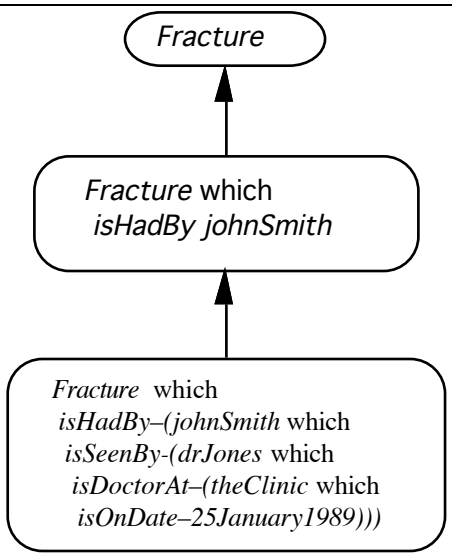

Figure 10.1 The three layers of SMK spaces and subsumption between category, individual, and occurrence

9.2.3 Occurrences as observations

The development of a uniform approach to the three spaces of SMK has been important in producing a principled relationship between the terminology and information models. There is however one further step required for occurrences to be interpreted as observations. A entity such as the occurrence of John Smith's fracture is an intensional definition. It is a very specialised kind of fracture. It is a sensible thing to say because fractures occur to patients. However there is nothing so far that states it was actually observed.

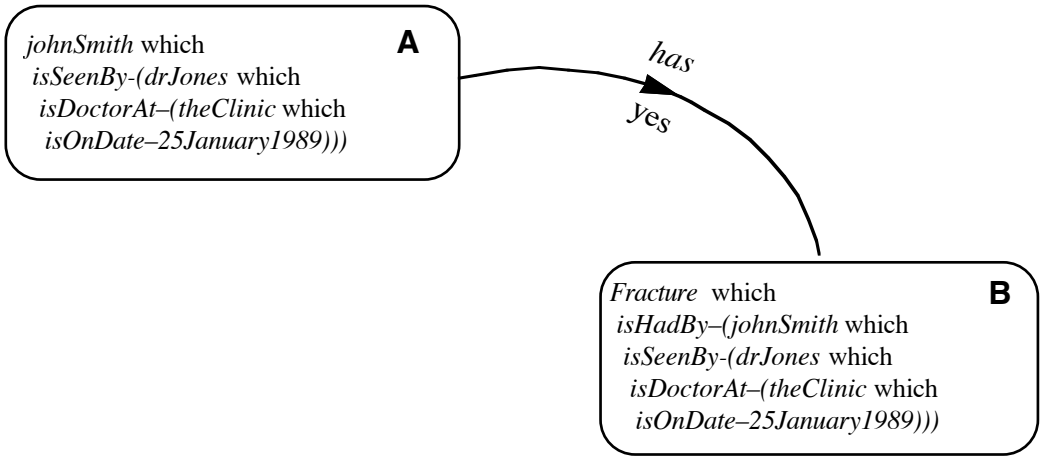

Figure 10.2An observation of John Smith's fracture

The fact of observation is represented by the use of a triple called a datum. For example in figure 10.2, the datum labelled by the attribute *has*, asserts that the occurrence of johnSmith (A) was observed to have the occurrence of the fracture (B). The qualifier on the datum is yes. This is a variant of the qualifier necessary reserved for use with a datum. This datum establishes the occurrence of the fracture as an observation The data form a network of observations and represent the information about the patient in the record. They record those things that *were* said out of all the things which *could* be said.

9.2.4 The effects of an observation

The use of the datum has a second effect. A datum is an assertion and thus adds to the necessary criteria of the entity it describes, in this example the occurrence of *johnSmith*. (A). The addition of a necessary criterion may change the classification of the occurrence of johnSmith (A). For example it would now be subsumed by the category Patient which has–Fracture. This re–classification may then have the effect of changing what it is possible to say following the observation. There may be things that according to the model can be said about patients who have a fracture that cannot be said about patients in general.

9.3 Summary

An extension to SMK allows for the represent of individual patient records. There are two levels of instantiation in SMK i) from categories to individuals and ii) from individuals to occurrences. Categories are similar to classes in other representations. Individuals represent concrete things such as people and places. Occurrences represent observations of those individuals by an observer at a particular time and place. A patients medical record is then represented as a network of occurrences. The information model of the record has all the expressive capabilities of the terminology model, and is thus capable of supporting complex clinical descriptions in a formal unified framework.

Chapter 10 Discussion and Issues Outstanding

This thesis concludes with a review of its aims and the extent to which they have been met. Problems and limitations are discussed and outstanding issues identified. The final thoughts are on the wider medical challenge.

10.1 Review of aims and outcomes

This thesis has presented the theory, design, and implementation of the Structured Meta Knowledge formalism for the representation of medical concepts, and demonstrated its utility for representing medical terminologies. The following sections review the aims presented in chapter 1.

10.1.1 Reasons for the inadequacies of current techniques for representing medical terminologies

Current techniques for the representation of medical concepts are based on coding and classification schemes that have their roots in the epidemiological and statistical traditions. Classification schemes are enumerative representations of medical concepts and the relationships between those concepts. Increasing the scope of a scheme to cover realistic clinical descriptions results in a 'combinatorial explosion' of terms. Furthermore the task of defining the relationships between those terms is unmanageable. The one existing compositional scheme lacks rules for forming medically sensible compositions and determining the relationships between compositions. It can thus generate medical nonsense. These are the main reasons why current classification schemes are an inadequate basis for the formal representation of detailed, structured clinical information in computer–based systems.

10.1.2 Requirements on a formalism for representing medical concepts

The inadequacies of traditional schemes indicated the need for a formalism that is *recursively compositional*, *constrained*, *generative*, and therefore capable of representing models of medical terminology that are *parsimonious*. These requirements are the basis of the Structured Meta Knowledge formalism for the representation of terminological knowledge. This formalism extends the definition of terminological knowledge to include limited forms of assertion for the creation of elementary entities, the assertion of subsumption, and statements about terminology. These statements about terminology represent what it is 'sensible to say' in medicine and are the basis for constraining the representation to 'sensible medical concepts'. This in turn is the basis for the generativity and hence parsimony. Theories were described that satisfied these requirements and defined the well–formedness of compositions and the relationships between those compositions, in particular that of subsumption.

10.1.3 The utility of the SMK formalism and its implementation

The implementation of SMK is consistent with its theories, with several recognised exceptions. Classification and the derivation of the canonical form of an entity are currently incomplete in specific situations involving complex embedded criteria. In practice the implementation of the terminology engine has proved robust and is now the basis of the PEN&PAD prototype workstation and the first version of the terminology module within the GALEN project[10] [GALEN 1992].

SMK is useful for modelling medical terminology, as demonstrated by the example on tumour pathology. What is encouraging is that the major medical aspects of the informal model

[10] GALEN – Generalised Architecture for Languages, Encyclopaedias, and Nomenclatures in Medicine. See section 10.4 for additional details of the GALEN project.

described in ICD–9 source terminology corresponded well to formal constructs in SMK. It was relatively easy to take the important first step in building an overall framework for the model which could then be examined and refined. As a result of constructing the model it was possible to derive medically useful relationships between concepts. These relationship were not explicitly given in the original source corpus.

The SMK model of tumour pathology was used to represent the terms from a section of the Read Clinical Classification. From this the terminology engine automatically generated the complex hierarchical relationships between those terms implied by their formal definitions. This showed the original classification of those terms within the coding scheme to be both incomplete and incorrect. Furthermore the structure of the model and the definitions helped to explain why. This relationship between the formal model and the classification scheme is an important result. It suggests that formal representations may be useful in improving the structure of existing classification schemes, without having to replace them in end user applications in a single step. This result will be important in devising practical development pathways for clinical systems.

10.1.4 Relationship to the information model of the medical record

An extension to SMK allows for the represention of individual patient records. There are two levels of instantiation in SMK i) from categories to individuals and ii) from individuals to occurrences. Categories are similar to classes in other representations. Individuals represent concrete things such as people and places. Occurrences represent observations of those individuals by an observer at a particular time and place. A medical record is represented as a network of occurrences. The model of the record has all the expressive capabilities of the terminology model, and is thus capable of supporting complex clinical descriptions in a formal unified framework.

10.1.5 Current status of SMK and its implementation

The aim of SMK is to make it easier to construct and maintain models of medical terminology that are useful in medical applications. Experience to date suggests that as a technique for representing medical terminologies SMK is a significant improvement on traditional coding and classification schemes. SMK is the basis of the medical record in the PEN&PAD clinical system, and several models have been developed by other workers to support that system. More recently the GALEN project is attempting to develop large scale and verified models of medical terminology based on a development of the theory and implementation of SMK. The terminology engine described in this thesis is the core of the first version of the GALEN software.

10.2 Limitations and problems with the formalism and its implementation

One of the most important outcomes of this work has been an understanding of the requirements on a medical terminology formalism and system. The work is not presented as a completed task. There are limitations and problems and some of these are discussed in the following section.

10.2.1 The need for extensions to accommodate common terminological constructs

Experience with the use of SMK for practical modelling is demonstrating the need for extensions to the formalism. Some of these were discussed in chapter 8. In particular there is a clear need for relative negation and a more detailed handling of cardinality. Constructs found in traditional classifications such as ‘other’ and ‘not otherwise specified’ cannot always be represented using SMK. There are however fundamental problems with the interpretation of these constructs within the source schemes and some are advocating their elimination.

The use of defeasible assertions has not been discussed but work within PEN&PAD requires these. At present defeasible assertions are required to be consistent with the sanctions in the model, that is they must be possible, but they play no part in the formal theories, and their interpretation is outside the definition of the formalism. A definition of the use of defeasible statements within SMK is required.

10.2.2 Maintenance of a globally coherent model

The requirements on the global coherence of the model have not been explored in detail. The main source of concern is the use of conventional subsumption and necessary statements. Each of these changes the properties of existing entities and may result in contradictions or invalidate previously made inferences. For example the assertion that all cancers are severe denies the existence of the concept 'mild cancer', and subsequent attempts to form it would fail. However the concept of 'mild cancer' may have been previously reified and be the subject of specific knowledge. In this case the assertion that all cancers are severe clearly renders the model incoherent. The converse action of retracting knowledge raises similar issues of global coherence.

The classifier detects some of the problems as a side effect of inserting triples but the definition is incomplete. The current view is that ambiguity within the model should not be permitted. There is evidence that the problem is tractable [Rector 1986] and work is now underway.

10.2.3 Tractability

The computational tractability of SMK has not been proven in this thesis though there is strong evidence that similar representations are tractable (see chapter 3). However worse case behaviour has been questioned as the prime measure of utility, and the medical test will be how SMK works in practice, on realistic computers, manipulating large medical models. The best indication to date is that the present implementation succeeded in compiling and classifying 1,800 complex pharmaceutical preparations that formed a dense hierarchy of tablets, capsules, and injections.

10.3 Issues outstanding

The broader task of constructing large medical terminologies demands more than a formalism. These tasks have not formed part of this thesis but two of them, related to the task of modelling, are mentioned below.

10.3.1 The scaling properties of SMK models

The scaling properties of large models expressed using SMK have not been studied. Traditional schemes are known to suffer from the combinatorial explosion as they try to enumerate everything that can be said. In contrast SMK tries to capture generalities, enumerate only the exceptions, and depends on inference to do the rest. The proof of concept for this approach requires the development of large models with general utility. This is an empirical test and outside the scope of this thesis. However as an SMK model grows to the size required for practical clinical use there are at least two tests of its scaling properties:

1 does it become easier or more difficult for developers to understand the model, add knowledge, and put it to practical use? This will depend on the formalism, the domain, and whether the generalities win over the exceptions. The metric for this test is expert time and effort.

2 do the inferences remain tractable? This applies when the model is being developed and also when in use. The metric for this test is computing time and effort.

Success will depend on experience and compromise. This is no stranger to clinical medicine which is an extended exercise in compromise and the utilisation of the imperfect.

10.3.2 Methodologies and tools for modelling

For SMK to be useful it has to be usable. The tools described in this thesis were adequate for building and testing modest models. Other tools are required for the co-operative designing, building and maintaining of large models. A methodology is also required for modelling in SMK based on practical experiments. The goal should be a methodology which makes it easy to capture the essence of the semi–formal and informal medical models that will form the source corpora for large formal terminologies.

10.4 Future directions – PEN&PAD and GALEN

The PEN&PAD programme of research is continuing and has recently been extended to the shared hospital care of the elderly patients [Heathfield 1992]. This has introduced new medical terminologies, in particular that related to nursing care [Kirby 1992]. The main part of the work described in this thesis is continuing within the GALEN Project under the Advanced Informatics in Medicine initiative of the European Community [GALEN 1992]. GALEN is developing a *Master Notation* for the representation of medical terminologies based on the theories of SMK. It is also constructing a large and medically verified Coding Reference Model (CoRe Model) of medical terminology and developing associated methodologies for modelling terminologies. The products of GALEN are being developed and tested through the use of demonstrators and experiments in the areas of clinical systems, medical records, knowledge–based systems, and bibliographic thesaurii.

10.5 Medical challenges

SMK is now of proven utility in limited experiments and prototype clinical applications, and work is underway to extend that proof. However this thesis concludes with some brief thoughts on why the problems addressed in this thesis are medical problems.

It can be argued that clinical medicine is no longer doable. Advances in medical science have outstripped the ability of clinical practice to apply those advances reliably and repeatedly to the benefit of patients. As a result medicine is increasingly concerned with clinical audit, quality assurance, the setting of standards, and the promulgation of good practice. The question is whether or not information and information systems help restore the balance between what is possible and what happens in practice? If this answer is yes then the integration of clinical information systems into clinical care is not only useful but is essential. It represents a significant contribution to the process of medical care, comparable to any other medical intervention.

The computerisation of information is now a fact of clinical practice. Techniques, formalisms, and models can help in that process but the challenges and responsibilities are at heart medical. Information systems will influence how we think about, discuss, and practice medicine. Solutions to the problems described in this thesis and the many other problems facing the development of clinical information systems cannot be easy. The provision of health care is an important and costly endeavour, and if solutions were easy they would have been found by now. To make progress the medical professions must embrace the problem of understanding information and see it as their problem Above all else it is hoped that this thesis is a small contribution to that process.

References

Barr CE, Komorowski HJ, Pattison-Gordon E, Greenes RA (1988). Conceptual modeling for the Unified Medical Language System. In Greenes RA ed. Proceedings of the 12th annual symposium on computer applications in medical care. New York: IEEE, 1988;152-7

Brachman RJ (1979). On the epistemological status of semantic networks. In Findler NV (ed) Associative networks: representation and use of knowledge by computers, 3–50;New York, Academic Press, 1979

Brachman RJ (1983b) What ISA is and isn't: an analysis of taxonomic links in semantic networks. IEEE Computer 16(10); pp 30-36

Brachman RJ Fikes RE Levesque HJ (1983a) KRYPTON: A functional approach to knowledge representation IEEE Computer 16(10), 1983, 76-73

Brachman RJ Levesque HJ (1984). The tractability of subsumption in frame–based description languages. In Proceedings AAAI-84; 34–37.

Brachman RJ Levesque HJ (1985)A fundamental tradeoff in knowledge representation (revised version) in Brachman RJ Levesque HJ (eds) Readings in knowledge representation, 41-70, Morgan Kaufmann, California

Brachman RJ, Schmolze JG (1985). An overview of the KL–ONE knowledge representation system. Cognitive Science 9;171–216.

Brodie ML.(1984) On the development of data models. In Brodie ML, Myopoulos J, Schmidt JW (eds). On conceptual modelling. New York: Springer-Verlag; 1984

College of American Pathologists (1977). Systematized nomenclature of medicine.(first edition). Skokie, Illinois, USA: College of American Pathologists, 1977

College of American Pathologists (1982). Systematized nomenclature of medicine.(second edition). Skokie, Illinois, USA: College of American Pathologists, 1982

Cote RA, Rothwell DJ (1989) The classification-nomenclature issues in medicine: a return to natural language. Med Inform 1989 vol 14 (1);25-41

Doyle J, Patil RS (1989) Two dogmas of knowledge representation: language restrictions, taxonomic classification, and the utility of representation services. Massachusetts Institute of Technology, MIT Report MIT/LCS/TM–387.b

Evans DA, Miller RA (1987). Initial phase in developing representations for mapping medical knowledge: INTERNIST-I/QMR, HELP, and MeSH. UMLS Final task report (Task 2). University of Pittsburgh:1987

Evans DA, Rothwell DJ, Monarch IA, et al (1991). Towaards representations for medical concepts. Medical Decision Making 11(4) supplement; S102–S108.

Evans DA. Pragmatically-structured, lexical semantic knowledge bases for Unified Medical Language Systems (1988). In Greenes RA ed. Proceedings of the 12th annual symposium on computer applications in medical care. New York: IEEE, 1988;169-173

GALEN 1992. Project summary available from the Project Manager, Medical Informatics Group, Department of Computer Science, University of Manchester, UK.

GMSC-RCGP (1988). The classification of general practice data. Final report of the GMSC-RCGP Joint Computing Group Technical Working Party. BMA, London: August 1988

Haimowitz IJ, Patil SP, Szolovits P (1988). Representing medical knowledge in a terminological language is difficult. In Proceedings of the Twelth Annual Symposium on Computer Applications in Medical Care. Computer Society Press, Washington: pp 101-105

Heathfield HA, Kirby J, Nowlan WA and Rector AL (1992). PEN&PAD (Geriatrics): A Collaborative Patient Record System for the Shared Care of the Elderly. Published in ME Frisse (ed) Sixteenth Annual Symposium on Computer Applications in Medical Care. Proceedings of SCAMC 92, A Conference of the American Medical Informatics Association, McGraw-Hill, Inc, New York, 1992, pp 147-150.

Kirby J, Heathfield HA (1992). Meeting Nursing Requirements Using User-Centred Design and Predictive Data Entry in M Scholes and B Barber (eds) Informatics for the Nursing Professions: The what, why, when, how, where, and who of information use, Conference Proceedings of the Nursing Specialist Group of the British Computer Society, Eastbourne, 3-5 November 1992, BJHC Books, pp101-104.

Levesque HJ (1986) Making believers out of computers. Artificial Intelligence, 30(1):81–108.

Nebel B (1990). Terminological reasoning is inherently intractable. Artificial Intelligence 43 (1990) 235–249

Nowlan WA , Kay S, Rector AL, Horan B and Wilson A (1991). A Multi-Lingual Patient Care Workstation Based on a Unified Representation of the Medical Record and Medical Knowledgein KP Adlassnig, G Grabner, S Bengtsson, R Hansen (eds), Lecture Notes in Medical Informatics 45, proceedings of MIE 91, Springer-Verlag, Berlin.1991, pp.1043.

Nowlan WA et al (1990). PEN&PAD: a doctor's workstation with intelligent data entry and summaries. In Miller RA (ed) Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care (SCAMC 90), Washignton CD, 941–942

Nowlan WA, Rector AL (1991) Medical knowledge representation and predictive data entry. In Stefanelli S, Hasman A, Fieschi M, Talmon J (eds) AIME91, Lecture Notes in Medical Informatics no 44, Springer–Verlag, Berlin 1991, 105–116

Nowlan WA, Rector AL, Kay S, Horan B andWilson A (1991). A Patient Care Workstation Based on a User Centred Design and a Formal Theory of Medical Terminology: PEN&PAD and the SMK Formalism in PD Clayton (ed), Fifteenth Annual Symposium on Computer Applications in Medical Care. Proceedings of SCAMC 91, A Conference of the American Medical Informatics Association, McGraw-Hill, Inc, New York, 1991, pp 855-857.

Patel–Schneider (1989). Undecidability of subsumption in NIKL. Artificial Intelligence 39 (2) 263–272

RCGP (1982). Computers in primary care. RCGP Occasional paper 13 (second impression). London: Royal College of General Practioners, 1982

RCGP (1986). The classification and analysis of general practice data. RCGP Occasional paper 26. London: Royal College of General Practioners, 1986

Rector AL (1986). Defaults, exceptions, and ambiguity in a medical knowledge representation. Med Inform (1986) 4; 295–306.

Rector AL, Nowlan WA and Kay S (1990). Unifying Medical Information using an Architecture Based on Descriptions, in RA Miller (ed) Proceedings of the 14th Annual Symposium on Computer Applications in Medical Care , SCAMC 90, IEEE Computer Society Press, Los Alamitos, California ,1990, pp 190-194. (Internal Report 133)

Rector AL, Nowlan WA and Kay S,.(1991). Foundations for an Electronic Medical Record, in Methods of Information in Medicine, 1991; 30: pp 179-186, FK Schattaeur Verlagsgesellschaft mbH publications. Also reprinted in JH van Bemmel and AT McCray (eds), IMIA Yearbook of Medical Informatics 92, Advances in an Interdisciplinary Science, Schatteur Publications, 1992, pp 59-66.

Rector AL, Nowlan WA and Kay S.(1991). Conceptual Knowledge: The Core of Medical Information Systems. in KC Lun, P Degoulet, TE Pierre, O Rienhoff (eds) MEDINFO 92, Proceedings of the Seventh World Congress on Medical Informatics, North-Holland Publishers 1992, pp 1420-1426.

Rector AL, Nowlan WA, Kay S, Goble CA and Howkins TJ (1992). A Framework for Modelling the Electronic Medical Record. To be published in Methods of Information in Medicine, FK Schattaeur Verlagsgesellschaft mbH publications

Rector AL, Nowlan WA, Kay S. Conceptual knowledge: the core of medical information systems. Proceedings of MEDINFO–92 1992, Springer.

Ringland GA, Duce DA (eds) (1988) Approaches to knowledge representation: an introduction. Research Studies Press, Taunton, England, 1988.

Schmolze JG, Lipkis TA (1984). Classification in the KL–ONE knowledge representation system. Proc AAAI–84.

SNOMED III. This is obtainable in computerised form with some associated release notes. See reference to SNOMED I & II [College of American Pathologists 1977 &

Sowa JF (1984) Conceptual structures: Information processing in minds and machines, Addison–Wesley: Reading, MA

Wingert F, Rothwell D, Cote R (1989). Automated indexing into SNOMED and ICD. In Scherrer JR, Cote RA, Mandil SH (eds) Computerised natural medical language processing for knowledge engineering. North Holland: Elsevier Science Publishers, 1989;5-17

Woods WA (1975) What's in a link: foundations for semantic networks. Reproduced in in Brachman RJ Levesque HJ (eds) Readings in knowledge representation, 41-70, Morgan Kaufmann, California, 1985

Appendix 1 SMK Operations

A1.1 SMK Operations and Compiler Syntax

This section describes the main SMK operations in the form of the compiler syntax. It is intended to be a guide to the use of the compiler and some of the main SMK operations both in terms of their SMK syntax and the Smalltalk method selector as implemented in the class Network.

The sub–sections cover:

SMK types

SMK operations

Compiler constructs

Compiler operations

A1.2 SMK types

SMK is very strongly typed and the type of an argument is specified in the definition of the operation. The allowed types at present are:

A1.2.1. Entities <entity>

These are either literals or expressions which evaluate to an entity

Literal entities

literal type	example
word	Cancer
number	
integer	23
float	12.756
date (dd/mm/yy)	23/9/87
string	'Arthur Wallace'

Expressions

These are SMK operations which evaluate to an entity.

A1.2.2. Qualifiers <qualifier>

There are currently five supported qualifiers

conceivable

grammar

possible

necessary

and the restricted qualifier

null

A qualifier is an object, but not at present an entity, which defines the level to which a triple belongs.

A1.2.3. Inheritance Patterns <inheritance>

There are the four modes of inheritance corresponding to the possible combinations of *nil* (not inherited) and *all* (inherited)

allAll
nilAll
allNil
nilNil

In the current use of SMK all attributes are inherited and hence the pattern is allAll

A1.2.4. Cardinality Patterns <cardinality>

There are four supported cardinalities corresponding to the possible combinations of *one* and *many*.

oneOne
oneMany
manyOne
manyMany

A1.2.5. Identifier <identifier>

These are words eg. Bone and are used to denote the identifier to be applied to a newly created elementary entity. The identifier must not have already been used by any existing entity. Identifiers are subsequently recognised by the compile as a literal entity. Note that in the current version of SMK identifies follow the same conventions as names, and when a new elementary entity is created its name is set by default to be the same as its identifier.

A1.2.6. Name <name>

These are words eg. Cancer and are used to give a name or 'nickname' to an existing entity. The name must not be in use by any existing entity. Names are subsequently recognised by the compiler as a literal words.

A1.2.7. Literals which are not entities

There are three types of literal which are interpreted as simple Smalltalk objects

type	example
<integer>	12
<float>	24.786
<string>	'heart attack'

It is important to note the distinction between literal entities and literals which are not interpreted as entities. A literal entity is a full SMK entity which is placed in the network. Literals which are not entities are simple Smalltalk objects and are generally used in the internal

structure of entities. The best example is the string currently used as the public name of an entity.

A1.3 SMK operations available via the SMK Compiler

This section describes the SMK Operations currently available via the SMK Compiler. For each operation the keyword, SMK syntax, and whether or not the operation adds knowledge to the network are given. These are followed by an explanation of the purpose of the operation, some notes on errors, and examples. Finally there is the Smalltalk method selector as currently implemented in the class Network which corresponds to the operation.

newSub

SMK Syntax

<entity> **newSub** <identifier>

Adds knowledge

Result

Returns the new entity

Purpose

Create a new elementary entity with identifier <identifier> and make it an explicit sub of the existing entity <entity>. The identifier must be unique.

Examples

RespiratorySymptom **newSub** Cough

Beta-blocker **newSub** Atenolol

SeverityValueType **newSub** [mild moderate severe]

SMK Errors

<identifier> is already used or is not a valid identifier

Network method selector

smk: *anEntity* newSubIdentifier: *aSymbol*

newAttribute

SMK Syntax

<attribute> **newAttribute** <identifier1> <identifier2> <inheritance> <cardinality>

Adds knowledge

Result

Returns the new attribute

Purpose

Create a new attribute with identifier <identifier1> and inverse identifier <identifier2>, and make it an explicit sub of the existing attribute <attribute>. Both identifiers must be unique. The inheritance and cardinality of the new attribute are specified.

Examples

SpatialAttribute **newAttribute** hasPart isPart allAll oneMany

Note

The possible values for inheritance and cardinality are discussed in the section on attributes.

SMK Errors

<identifier1> or <identifier2> is already used.or is not a valid identifier

Invalid inheritance or cardinality

Network method selector

smk: *anAttribute* newSubIdentifier: *aSymbol1* inverseIdentifier: *aSymbol2* inheritance: *anInheritancePair* cardinality: *aCardinalityPair*

which

SMK Syntax

<entity> **which** <criterion>

Does not add knowledge

Result

Returns the entity as specified by the expression

Purpose

This is the operation which requests an entity according to an expression. It does not conceptually create a new entity because its existence must at least have been implied by the existing state of the network ie. there must be a possibility triple of the correct form in place. It may be the case that prior to the request the entity may not 'exist' within the current network in terms of being represented by a Smalltalk object or appearing in any hierarchies etc. The operation does not however distinguish between 'existing' or 'implied' entities. It will find the existing entity or instantiate one if necessary, with the result of the operation being the same in both cases.

The operation will also accept multiple criteria.

Examples

Fracture **which** hasLocation Humerus

Cough **which** hasSeverity-severe

Cancer **which** <hasLocation-Lung hasCellType-EpithelialCell>

SMK Errors

The source triple is not acceptable - in this case it must be a possibility triple.

The expression is not well-formed

eg. contradictory criteria

Fracture **which** hasLocation-Humerus, hasLocation-Femur

Humerus **which** isLocationOf-(Fracture which hasLocation-Femur)

Redundancies and tautologies

In general these do not generate notified

eg. Fracture **which** hasLocation-Humerus hasLocation-LongBone

the criterion hasLocation-LongBone is redundant.

Network method selector

smk: *anEntity* which: *aCriterion*

Note the variant for direct use from Smalltalk

smk: *anEntity1* which: anAttribute s: *anEntity2*

This generates the criterion *anAttribute-anEntity2*

whichQ

SMK Syntax

<entity> **whichQ** <criterion> <qualifier>

Does not add knowledge

Result

Returns the entity specified by the expression

Purpose

This operation is essentially the same as the **which** operation but it allows independent specification of the required qualifier of the sanctioning source triple. The unqualified **which** operation requires the presence of a suitable possibility triple. The **whichQ** operation is intended to allow the use of a grammatical triple as the source. It is a restricted operation used to create abstract concepts such as Disease which hasLocation-Liver ie. liver disease in general.

The **whichQ** used with the qualifier possible is equivalent to the more usual unqualified **which**.

Examples

Disease **whichQ** hasLocation-Lung grammatical

BodyPart **whichQ** isPartOf-Head grammatical

SMK Errors

Essentially the same as the simpler **which** operation.

Network method selector

smk: *anEntity* which: *aCriterion* qualifierLevel: *aQualifierOrQualifierPair*

Note that if aQualifierPair is used the first of the pair is interpreted as the qualifier for the operation

addSub

SMK Syntax

<entity1> **addSub** <entity2>

Adds knowledge yes

Result

Returns <entity2>

Purpose

Makes <entity2> an explicit sub of <entity1>. Note that this differs from newSub in that no new entity is created, both <entity1> and <entity2> must already exist.

Examples

SeriousDisease **addSub** Cancer

SMK Errors

These will depend on the integrity checks within the system eg. circular subsumption of the form A addSub B then B addSub A etc.

Network method selector

smk: *anEntity1* addExplcitSub: *anEntity2*

addSuper

SMK Syntax

<entity1> **addSuper** <entity2>

Adds knowledge

Result

Returns <entity2>

Purpose

This is equivalent to <entity2> addSub <entity2>

Examples

See addSub

SMK Errors

See addSub

Network method selector

smk: *anEntity1* addExplicitSuper: *anEntity2*

triple

SMK Syntax

<entity1> **triple** <attribute> <entity2> <qualifier>

Adds knowledge

Result

Returns the newly created triple or the existing triple of exactly the same specification

Purpose

Create or find the existing triple of the form topic-attribute-value: qualifier or

where:

topic <entity1>

attribute <attribute>

value <entity2>

qualifier <qualifier>

This is the fundamental operation for creating new triples. If a triple of the precise form already exists in the network then it is return, if not then a new triple is created. This triple must be sanctioned by an existing triple or attribute. There is a precedence amongst the qualifiers

conceivable (attribute)

grammatical

possible

necessary

For a triple to be valid there must already exist within the network an appropriate source triple of the next higher precedence to the new triple, which relates the topic and value together eg. the creation of a possible triple requires the presence of a suitable grammatical triple. In most situations the source triple will be inherited from supers of the topic and value.

The operation also results in the creation of the inverse triple

value-inverseAttribute-topic: qualifier

Examples

Disease **triple** hasLocation BodyPart grammatical

Arm **triple** hasPart Hand possible

Leg **triple** hasPart Foot necessary

SMK Errors

A suitable source triple was not found eg.

Arm hasPart Hand: possible

could not be created because the source triple was

BodyPart hasPart BodyPart: grammatical

Network method selector

smk: *anEntity1* attribute: *anAttribute* value: *anEntity2* qualifiedBy: *aQualifierPair*

name

SMK Syntax

<entity> **name** <name>

Adds knowledge

Result

Returns <entity>

Purpose

Apply the knowledge name <name> to <entity>. The knowledge name is the 'nickname' of the entity. Both elementary and complex entities can have knowledge names. Knowledge names are unique.

Note:

At present the identifiers and knowledge names must be unique with respect to each other.

When an elementary entity is created its knowledge name is set by default to be the same as its identifier.

Knowledge names are recognised by the compiler.

Examples

MyocardialInfarction **name** CoronaryThrombosis

(Neoplasm which hasNeoplasticBehaviour malignant) **name** Cancer

SMK Errors

<name> is already used or is not a valid identifier

Network method selector

smk: *anEntity* knowledgeName: *aSymbol*

public

SMK Syntax

<entity> **public** <string>

Adds knowledge

Result

Returns <entity>

Purpose

Apply the public name <string> to <entity>. The public name is at present only of use in relation to the user interface when displaying or printing the entity.

Note

Public names are not unique

They are not recognised by the compiler

Examples

Myocardialnfarction **public** 'heart attack'

SMK Errors

<string> is not a valid string

Network method selector

smk: *anEntity* publicName: *aString*

specialisedBy

SMK Syntax

<attribute1> **specialisedBy** <attribute>

Adds knowledge

Result

Returns <attribute1>

Purpose

States that <attribute1> is specialised by <attribute2> ie. is transitive across it

Examples

hasLocation **specialisedBy** isPartOf

As a consequence of this hasLocation-(Shaft which isPartOf-Humerus) is subsumed by hasLocation-Humerus

SMK Errors

Invalid argument

Network method selector

smk: *anAttribute1* specialisedBy: *anAttribute2*

of

SMK Syntax

<attribute> **of** <entity>

Does not add knowledge

Result

Returns a set of entities or Non

Purpose

Return the values of all the criteria with attribution <attribute> which apply to <entity>

Examples

hasLocation **of** (Inflammation which hasLocation-Liver)

will return the value Liver

hasPart **of** Arm

will return the set {Hand, Elbow, Wrist etc} assuming that necessary statements for each of these have been created

SMK Errors

Invalid argument

Network method selector

smk: *anAttribute* of: *anEntity*

descriptions

SMK Syntax

<entity> **descriptions** <qualifier>

Does not add knowledge

Result

Returns a set of triples which describe <entity> and whose qualifier is <qualifier>

Purpose

Returns a set of triples which describe <entity> either directly or are inherited by <entity>. The set excludes triples which are supers of another triple in the set eg. a possibility triple will exclude the relevant grammatical triple. The qualifier restricts the set to those triples whose qualifier is <qualifier>

Examples

Arm **descriptions** necessary

may return the set {Arm-hasPart-Hand: necessary, Arm-hasPart-Wrist: necessary, etc}

SMK Errors

Invalid argument

Network method selector

smkAllDescriptionsOf: *anEntity* qualifiedBy: *aQualifierOrQualifierPair*

Note that if aQualifierPair is used then only the first of the pair is relevant

subs

SMK Syntax

<entity> **subs**

Does not add knowledge

Result

Returns a set of entities or Non

Purpose

Return all the immediate subs of <entity> both explicit and formal as a set. If there are none then the non-entity Non is returned

Examples

Disease **subs**

may return the set {CardiovascularDisease, RespiratoryDisease etc}

SMK Errors

Invalid argument

Network method selector

smkSubsOf: *anEntity*

supers

SMK Syntax

<entity> **supers**

Does not add knowledge

Result

Returns a set of entities or Non

Purpose

Return all the immediate supers of <entity> both explicit and formal as a set. If there are none then the non-entity Non is returned

Examples

CardiovascularDisease **supers**

may return the set {Disease}

SMK Errors

Invalid argument

Network method selector

smkSupersOf: *anEntity*

A1.4 Compiler Constructs

There is a limited range of constructs to facilitate the writing of SMK text files.

1. Parentheses ()

These are used to delimit expressions eg.

(Fracture **which** hasLocation Humerus) **name** FractureOfTheHumerus

2. Expandable lists []

These delimit a list or arguments which are to be **expanded** by the compiler prior to the performance of any operations eg..

Drug **newSub** [Aspirin Penicillin Morphine]

The list of drug names will be expanded and the **newSub** operation performed three times.

Expandable lists may be used for any type of argument and may contain complex expressions. Multiple expandable list may be used for a single operation.

3. Non-expandable lists <>

These delimit a list of arguments which will not be expanded by the compiler but are passed in order to the the relevant operation eg..

Cough **which** <hasSeverity-severe, hasProgress-worse>

4. Period .

This is used to denote the end of an expression

A1.5 Compiler Operations

There are several operations which are directives to the compiler itself to perform operations

These are denoted by {}.

---

include **<string>**
Compile the SMK source text in the file whose name is <string> eg..

{**include** 'CancerMorphology.smk'}

---

reset
Reset the SMK Network ie. create a new network completely destroying any existing network. A confirmer box will be generated prior to performing this operation.

---

resetNoCheck
Same as reset but no confirmation is required

---

resetAndNameNetwork <string>
Same as reset but the name of the network to be created is given as <string>

---

smkError <string1> <string>
This is a special operation which traps smkErrors which have been recorded in the change log of a network. When an error occurs in an operation the error is noted on the change log of the network in which it occurred. The change log is suitable for compiling and this operation notes any errors which may have been recorded. The argument <string1> is the text form of the operation which generated the error and <string2> is the error message. In the current implementation of the compiler this operation has no effect.

Appendix 2 Summary of objects within SMK

<object> : <entity>
| <criterion>
| <cardinality>
| <qualifier>

SMK objects (entities)

<SMKobject> : <entity>
| <relationship>

<entity> : <elementary type>
| <prototype>

<elementary entity> : <identifier>

<prototype> : p(<elementary entity>, {<criterion}$_C$)

<relationship> : <attribute>
| <triple>

<attribute> : <identifier>:<cardinality>

<triple> : t(<entity>–<criterion>:<qualifier>)

Criterion

<criterion> : <attribute>–<entity>

Conventional subsumption

<subsumption> : s(<entity>←<entity>)
| s(<attribute>←<attribute>)

Cardinality

<cardinality> : one
| many

Qualifier

<qualifier>	:	conceivable
	|	grammatical
	|	possible
	|	necessary

Appendix 3 Example model of tumour pathology and mapping of Read Clinical Classification

A3.1 SMK model of tumour pathology

"A simple model of terminology for describing neoplasia, both benign and malignant"

"Create the basic concepts of NeoplasticProliferation and Neoplasm"

Disease newSub NeoplasticProliferation.

NeoplasticProliferation newSub Neoplasm.

"Create the concept and attribute for neoplastic behaviour"

SymbolicValueType newSub NeoplasticBehaviour.

NeoplasticBehaviour newSub [benign malignant].

DescriptiveAttribute newAttribute hasNeoplasticBehaviour isNeoplasticBehaviourOf allAll manyOne.

"State that NeoplasticProliferations can be benign or malignant, generate two prototypes and give them names"

NeoplasticProliferation triple hasNeoplasticBehaviour NeoplasticBehaviour [grammatical possible].

(NeoplasticProliferation which hasNeoplasticBehaviour malignant) name MalignantNeoplasticProliferation.

(NeoplasticProliferation which hasNeoplasticBehaviour benign) name BenignNeoplasticProliferation.

"Create a diffuse proliferation which must be malignant"

MalignantNeoplasticProliferation newSub DiffuseNeoplasticProliferation.

"Protoypes for benign and malignant neoplasms"

(Neoplasm which hasNeoplasticBehaviour malignant) name Cancer.

(Neoplasm which hasNeoplasticBehaviour benign) name BenignNeoplasm.

"Go on to introduce the concept of metastasis"

SymbolicValueType newSub MetastaticState.

MetastaticState newSub [primary secondary].

primary newSub inSitu.

secondary newSub generalisedDissemination.

DescriptiveAttribute newAttribute hasMetastaticState isMetastaticStateOf allAll manyOne.

“Only malignant tumours can metastasise”

Cancer triple hasMetastaticState MetastaticState [grammatical possible].

(Cancer which hasMetastaticState primary) name PrimaryCancer.

(Cancer which hasMetastaticState secondary) name SecondaryCancer.

Tissue types and morphologies of tumours"

MedicalThing newSub CellType.

CellType newSub [CellTissueType CellMorphology].

"Cell morphology"

DescriptiveAttribute newAttribute hasCellMorphology isCellMorphologyOf allAll manyMany.

Neoplasm triple hasCellMorphology CellMorphology [grammatical possible].

CellMorphology newSub [SmallCell LargeCell FusiformCell Anaplastic Pleomorphic SpindleCell PolygonalCell SpheroidalCell Verrucous].

SmallCell newSub OatCell.

LargeCell newSub GiantCell.

"Tissue types"

DescriptiveAttribute newAttribute hasCellTissueType isCellTissueTypeOf allAll manyOne.

Neoplasm triple hasCellTissueType CellTissueType [grammatical possible].

CellTissueType newSub [EpithelialCell PigmentCell NeuralTissueCell ConnectiveTissueCell].

EpithelialCell newSub [SquamousCell GlandularCell BasalCell TransitionalCell].

SquamousCell newSub PapillaryCell.

"Naming of various types of tumours for convenience"

(Neoplasm which hasCellTissueType EpithelialCell) name EpithelialNeoplasm.

(Cancer which hasCellTissueType EpithelialCell) name Carcinoma.

(BenignNeoplasm which hasCellTissueType EpithelialCell) name Epithelioma.

(Cancer which hasCellTissueType GlandularCell) name Adenocarcinoma.

(BenignNeoplasm which hasCellTissueType GlandularCell) name Adenoma.

"Keratinisation is akward – the values are mutually exclusive"

(CellMorphology which isCellMorphologyOf (Carcinoma which hasCellTissueType SquamousCell)) newSub [Keratinising NonKeratinising].

(Carcinoma which hasCellMorphology Keratinising) triple hasCellMorphology NonKeratinising null.

(Carcinoma which hasCellMorphology NonKeratinising) triple hasCellMorphology Keratinising null.

"Knock together a bit of the gastrointestinal tract for the purpose of the example"

(TopographicalSegment newSub GastrointestinalTract) name GIT.

(TopographicalSegment whichQ isPartOf GIT grammatical) newSub [Pancreas Stomach HepatoBiliaryTract].

(TopographicalSegment whichQ isPartOf HepatoBiliaryTract grammatical) newSub BileDuct .

Neoplasm triple hasLocation TopographicalSegment possible.

A3.2 Hierarchy generated by the tumour pathology model

NeoplasticProliferation
---- MalignantNeoplasticProliferation
---- ---- Cancer
---- ---- -----PrimaryCancer
---- ---- -----Carcinoma
---- ---- ----- ---- Adenocarcinoma
---- ---- ----- ---- {Neoplasm which
hasCellTissueType-SquamousCell
hasNeoplasticBehaviour-malignant }
---- ---- ----- ---- {Neoplasm which
hasCellMorphology-Keratinising
hasCellTissueType-EpithelialCell
hasNeoplasticBehaviour-malignant }
---- ---- ----- ---- {Neoplasm which
hasCellMorphology-NonKeratinising
hasCellTissueType-EpithelialCell
hasNeoplasticBehaviour-malignant }
---- ---- -----SecondaryCancer
---- ---- DiffuseNeoplasticProliferation
---- BenignNeoplasticProliferation
---- ---- BenignNeoplasm
---- ---- -----Epithelioma
---- ---- ----- ---- Adenoma
---- Neoplasm
---- ---- Cancer ....^2
---- ---- BenignNeoplasm ....^2
---- ---- EpithelialNeoplasm
---- ---- -----Carcinoma ....^2
---- ---- -----Epithelioma ....^2

A3.3 Experiment in mapping Read Codes to SMK expressions

“Note that the main uncertainty is over the interpretation of NOS"

“This model depends upon the file describing the basic model of tumour pathology.

All the statements are of the form:

<smk expression> public ‘read code–’rubric’

It is intended to illustrate the realtionship between codes and smk expressions.”

(Neoplasm which hasNeoplasticBehaviour benign)
public 'BB00 Neoplasm, benign'.

(Neoplasm which hasNeoplasticBehaviour malignant)
public 'BB02 Neoplasm, malignant'.

((Neoplasm which hasNeoplasticBehaviour malignant) which hasMetastaticState secondary)
public 'BB03 Neoplasm, metastatic'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellMorphology SmallCell>)
public 'BB08 Malig.tumour, small cell'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellMorphology GiantCell>)
public 'BB09 Malig.tumour, giant cell'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellMorphology FusiformCell>)
public 'BB0A Malig.tumour, fusiform cell'.

(Neoplasm which hasCellTissueType EpithelialCell)
public 'BB1 Epithelial neoplasms NOS'.

(Neoplasm which <hasNeoplasticBehaviour benign hasCellTissueType EpithelialCell>)
public 'BB10 Epithelial tumour, benign'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState inSitu hasCellTissueType EpithelialCell>)
public 'BB11 Carcinoma in situ NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell>)
public 'BB12 Carcinoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState secondary hasCellTissueType EpithelialCell>)
public 'BB13 Carcinoma, metastatic NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState generalisedDissemination hasCellTissueType EpithelialCell>)
public 'BB14 Carcinomatosis'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology LargeCell>)
public 'BB17 Large cell carcinoma NOS'.

"BB18()
public 'Carcinoma, undiff. type NOS'."

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology Anaplastic>)
public 'BB19 Carcinoma, anaplastic NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology Pleomorphic>)
public 'BB1A Pleomorphic carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology GiantCell hasCellMorphology SpindleCell>)
public 'BB1B Giant cell+spindle cell ca.'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology GiantCell>)
public 'BB1C Giant cell carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology SpindleCell>)
public 'BB1D Spindle cell carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology PolygonalCell>)
public 'BB1F Polygonal cell carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology SpheroidalCell>)
public 'BB1G Spheroidal cell carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology SmallCell>)
public 'BB1J Small cell carcinoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology OatCell>)
public 'BB1K Oat cell carcinoma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType EpithelialCell hasCellMorphology SmallCell hasCellMorphology FusiformCell>)
public 'BB1L Small cell ca.,fusiform'.

(Neoplasm which hasCellTissueType SquamousCell)
public 'BB2 Papill./squamous cell neop.'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState inSitu hasCellTissueType PapillaryCell>)
public 'BB21 Papillary carcinoma in situ'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType PapillaryCell>)
public 'BB22 Papillary carcinoma NOS'.

(Neoplasm which <hasCellTissueType PapillaryCell hasCellMorphology Verrucous>)
public 'BB23 Verrucous papilloma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType PapillaryCell hasCellMorphology Verrucous>)
public 'BB24 Verrucous carcinoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour benign hasCellTissueType PapillaryCell>)
public 'BB25 Squamous cell papilloma'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState inSitu hasCellTissueType SquamousCell>)
public 'BB29 Squam cell ca-in-situ NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType SquamousCell>)
public 'BB2A Squamous cell carcinoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState secondary hasCellTissueType SquamousCell>)
public 'BB2B Squamous cell ca.,metastat.'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType SquamousCell hasCellMorphology NonKeratinising hasCellMorphology LargeCell>)
public 'BB2D Squamous ca.,large,non-ker.'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType SquamousCell hasCellMorphology NonKeratinising hasCellMorphology SmallCell>)
public 'BB2E Squamous ca.,small,non-ker.'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType SquamousCell hasCellMorphology SpindleCell>)
public 'BB2F Squamous ca.,spindle cell'.

(Neoplasm which hasCellTissueType BasalCell)
public 'Basal cell neoplasms'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType BasalCell>)
public 'BB31 Basal cell carcinoma NOS'.

(Neoplasm which hasCellTissueType GlandularCell)
public 'BB5 Adenomas/adenocarcinomas'.

(Neoplasm which <hasNeoplasticBehaviour benign hasCellTissueType GlandularCell>)
public 'BB50 Adenoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState inSitu hasCellTissueType GlandularCell>)
public 'BB51 Adenocarcinoma in situ'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell>)
public 'BB52 Adenocarcinoma NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasMetastaticState secondary hasCellTissueType GlandularCell>)
public 'BB53 Adenocarc., metastatic NOS'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell hasLocation GastrointestinalTract>)
public 'BB57 Adenocarcinoma, intestinal'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell hasLocation Pancreas>)
public 'BB5B Pancreatic adenoma/ca.'.

(Neoplasm which <hasCellTissueType GlandularCell hasLocation Stomach>)
public 'BB5C Gastrinoma and carcinomas'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell hasLocation Stomach>)
public 'BB5C1 Gastrinoma, malignant'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell hasLocation HepatoBiliaryTract>)
public 'BB5D Hepatobil. tract adenoma/ca'.

(Neoplasm which <hasNeoplasticBehaviour malignant hasCellTissueType GlandularCell hasLocation BileDuct>)
public 'BB5D0 Bile duct adenoma'.

A3.4 Hierarchy of entities that correspond to a Read Code

Neoplasm
---- BB02 Neoplasm, malignant
---- ---- PrimaryCancer
---- ---- -----BB11 Carcinoma in situ NOS
---- ---- ----- ---- BB29 Squam cell ca-in-situ NOS
---- ---- ----- ---- ---- BB21 Papillary carcinoma in situ
---- ---- ----- ---- BB51 Adenocarcinoma in situ
---- ---- BB12 Carcinoma NOS
---- ---- -----BB1D Spindle cell carcinoma
---- ---- ----- ---- BB2F Squamous ca.,spindle cell
---- ---- -----BB11 Carcinoma in situ NOS ....^2
---- ---- -----BB1F Polygonal cell carcinoma
---- ---- -----BB31 Basal cell carcinoma NOS
---- ---- -----BB2A Squamous cell carcinoma NOS
---- ---- ----- ---- BB22 Papillary carcinoma NOS
---- ---- ----- ---- ---- BB24 Verrucous carcinoma NOS
---- ---- ----- ---- ---- BB21 Papillary carcinoma in situ ....^2
---- ---- ----- ---- BB2D Squamous ca.,large,non-ker.
---- ---- ----- ---- BB2B Squamous cell ca.,metastat.
---- ---- ----- ---- BB2F Squamous ca.,spindle cell ....^2
---- ---- ----- ---- BB29 Squam cell ca-in-situ NOS ....^2
---- ---- ----- ---- BB2E Squamous ca.,small,non-ker.
---- ---- -----BB1J Small cell carcinoma NOS
---- ---- ----- ---- BB2E Squamous ca.,small,non-ker. ....^2
---- ---- ----- ---- BB1K Oat cell carcinoma
---- ---- ----- ---- BB1L Small cell ca.,fusiform
---- ---- -----BB17 Large cell carcinoma NOS
---- ---- ----- ---- BB1C Giant cell carcinoma
---- ---- ----- ---- ---- BB1B Giant cell+spindle cell ca.
---- ---- ----- ---- BB2D Squamous ca.,large,non-ker. ....^2
---- ---- -----{Neoplasm which hasCellMorphology-NonKeratinising
hasCellTissueType-EpithelialCell
hasNeoplasticBehaviour-malignant }
---- ---- ----- ---- BB2E Squamous ca.,small,non-ker. ....^3
---- ---- ----- ---- BB2D Squamous ca.,large,non-ker. ....^3
---- ---- -----BB1A Pleomorphic carcinoma
---- ---- -----BB19 Carcinoma, anaplastic NOS
---- ---- -----BB13 Carcinoma, metastatic NOS
---- ---- ----- ---- BB53 Adenocarc., metastatic NOS
---- ---- ----- ---- BB2B Squamous cell ca.,metastat. ....^2
---- ---- ----- ---- BB14 Carcinomatosis
---- ---- -----BB1G Spheroidal cell carcinoma
---- ---- -----{Neoplasm which hasCellMorphology-Keratinising
hasCellTissueType-EpithelialCell
hasNeoplasticBehaviour-malignant }
---- ---- -----BB52 Adenocarcinoma NOS

---- ---- ----- ---- BB51 Adenocarcinoma in situ ....^2
---- ---- ----- ---- BB53 Adenocarc., metastatic NOS ....^2
---- ---- ----- ---- BB57 Adenocarcinoma, intestinal
---- ---- ----- ---- ---- BB5B Pancreatic adenoma/ca.
---- ---- ----- ---- ---- BB5D Hepatobil. tract adenoma/ca
---- ---- ----- ---- ---- -----BB5D0 Bile duct adenoma
---- ---- ----- ---- ---- BB5C1 Gastrinoma, malignant
---- ---- BB03 Neoplasm, metastatic
---- ---- -----BB13 Carcinoma, metastatic NOS ....^2
---- ---- BB08 Malig.tumour, small cell
---- ---- -----BB1J Small cell carcinoma NOS ....^2
---- ---- BB0A Malig.tumour, fusiform cell
---- ---- -----BB1L Small cell ca.,fusiform ....^2
---- ---- BB09 Malig.tumour, giant cell
---- ---- -----BB1C Giant cell carcinoma ....^2
---- BB00 Neoplasm, benign
---- ---- BB10 Epithelial tumour, benign
---- ---- -----BB50 Adenoma NOS
---- ---- -----BB25 Squamous cell papilloma
---- BB1 Epithelial neoplasms NOS
---- ---- BB10 Epithelial tumour, benign ....^2
---- ---- BB12 Carcinoma NOS ....^2
---- ---- Basal cell neoplasms
---- ---- -----BB31 Basal cell carcinoma NOS ....^2
---- ---- BB2 Papill./squamous cell neop.
---- ---- -----BB22 Papillary carcinoma NOS ....^2
---- ---- -----BB2D Squamous ca.,large,non-ker. ....^4
---- ---- -----BB25 Squamous cell papilloma ....^2
---- ---- -----BB2B Squamous cell ca.,metastat. ....^3
---- ---- -----BB2F Squamous ca.,spindle cell ....^3
---- ---- -----BB29 Squam cell ca-in-situ NOS ....^3
---- ---- -----BB2E Squamous ca.,small,non-ker. ....^4
---- ---- -----BB23 Verrucous papilloma
---- ---- ----- ---- BB24 Verrucous carcinoma NOS ....^2
---- ---- BB5 Adenomas/adenocarcinomas
---- ---- -----BB51 Adenocarcinoma in situ ....^3
---- ---- -----BB53 Adenocarc., metastatic NOS ....^3
---- ---- -----BB57 Adenocarcinoma, intestinal ....^2
---- ---- -----BB5C Gastrinoma and carcinomas
---- ---- ----- ---- BB5C1 Gastrinoma, malignant ....^2