Linking Knowledge Acquisition with CommonKADS Knowledge Representation

John Kingston

AIAI-TR-156

July 1994

This paper was presented at the BCS SGES Expert Systems '94 conference, St John's College, Cambridge, 12-14 December 1994.

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

Abstract

This paper describes approaches to the linking of knowledge acquisition techniques (such as transcript analysis, card sorting, the repertory grid, and the laddered grid) to the CommonKADS framework for knowledge representation and analysis. These links allow semi-automatic mapping from acquired knowledge to CommonKADS domain knowledge. The key to the success of the linking is that each knowledge acquisition technique produces knowledge within a structure, whether that structure is a taxonomic hierarchy (as produced by the laddered grid) or simply the grammatical structure of English. The structure produced is exploited to determine appropriate classification of knowledge into the CommonKADS domain ontology.

The techniques described in this paper have been used for commercial training at AIAI, and have also been implemented in a knowledge engineer's support toolkit, known as TOPKAT (The Open Practical Knowledge Acquisition Toolkit). This toolkit has been used to obtain experience of the practicality of the techniques recommended. Examples throughout this paper are drawn from TOPKAT.

Keywords: knowledge acquisition, knowledge representation, CommonKADS, natural language, ontological classification

1 INTRODUCTION

The major difference between knowledge engineering – the science of constructing knowledge-based software systems – and 'conventional' software engineering is the requirement for knowledge engineers to capture, represent, analyse and exploit knowledge in order to produce a successful system. Experience has shown that none of these tasks are simple; taking knowledge capture as an example, knowledge is typically only available within the head of an expert, or implicitly within written procedures or case records, and cannot be extracted from these sources without considerable effort. These difficulties have provided an incentive for the development of a variety of techniques to overcome the problems; techniques for knowledge capture, for example, are known as *knowledge acquisition* techniques. There is considerable literature proposing, analysing and advising on the use of knowledge acquisition techniques (e.g. [McGraw & Harb.-Briggs, 1989]; [Kidd, 1987]).

The task of *representing* the acquired knowledge in a format suitable for analysis is equally important for successful knowledge engineering; yet it has had a comparatively low profile. A number of different approaches have been suggested and used (see [Fox *et al*, 1987], [Kuipers & Kassirer, 1987] and [Johnson & Johnson, 1987] for examples). It is sometimes recommended that more than one representation is used, which suggests that no single representation is entirely adequate to represent acquired knowledge. It seems that there are different types of knowledge, which are better suited to different representations.

The KADS methodology for the development of knowledge-based systems has attempted to resolve the problem of adequate representation of acquired knowledge by suggesting that knowledge should be represented and analysed on several different levels simultaneously [Tansley & Hayball, 1993]. CommonKADS, the recent successor to KADS, has extended and refined the recommended representations; in particular, CommonKADS has introduced an ontological classification for domain knowledge [Wielinga, 1993]. These recommendations, coupled with libraries of generic templates for representing certain types of knowledge, have provided a workable and useful solution to the problem of representing acquired knowledge.

However, there are no knowledge acquisition techniques which generate output in a form suitable for direct input into CommonKADS models. Instead, knowledge acquisition techniques typically produce textual transcripts, or categorisations of domain terms on many different dimensions. This means that the knowledge engineer is required to use his expertise to identify important knowledge items and to classify these items into CommonKADS' domain ontology. This is often an onerous task; the main difficulty lies in the fact that CommonKADS provides little guidance on how to identify relevant knowledge, or to classify acquired knowledge into its ontology.

It has been observed, however, that the output generated by most knowledge acquisition techniques is not an unsorted jumble of items of knowledge; instead, the acquired knowledge is usually structured in one way or another. Even the transcript of an interview is structured according to the rules of grammar. The thesis of this paper is that it is possible to automate much of the identification and classification of domain knowledge by identifying and exploiting the structure of acquired knowledge. Some previous work has been done in this area, including the generation of production rules from a repertory grid (e.g. [Shaw & Gaines, 1987]), and the production of a common logical framework for communicating between different knowledge acquisition techniques (cf. [Reichgelt & Shadbolt, 1991]). However, no one has yet attempted to make use of the structure of acquired knowledge to perform the classifications required for the CommonKADS domain ontology.

This paper describe how such classification can be performed. The techniques presented in this paper have been implemented in a hypertext-based knowledge engineering toolkit, known as TOPKAT (The Open Practical Knowledge Acquisition Toolkit); Figure 1 illustrates the user interface of TOPKAT, and many of the diagrams in this paper have been drawn from applications developed using TOPKAT. The techniques have also been presented and used within a commercial training course; this course was part of an ongoing project which assists selected companies in introducing and applying CommonKADS. The integration of CommonKADS with other knowledge engineering techniques is a key element of the project, which is funded by the CEC's ESSI programme (project no. 10327, CATALYST) to support the commercial uptake of ideas from CommonKADS, which was also funded by the CEC.

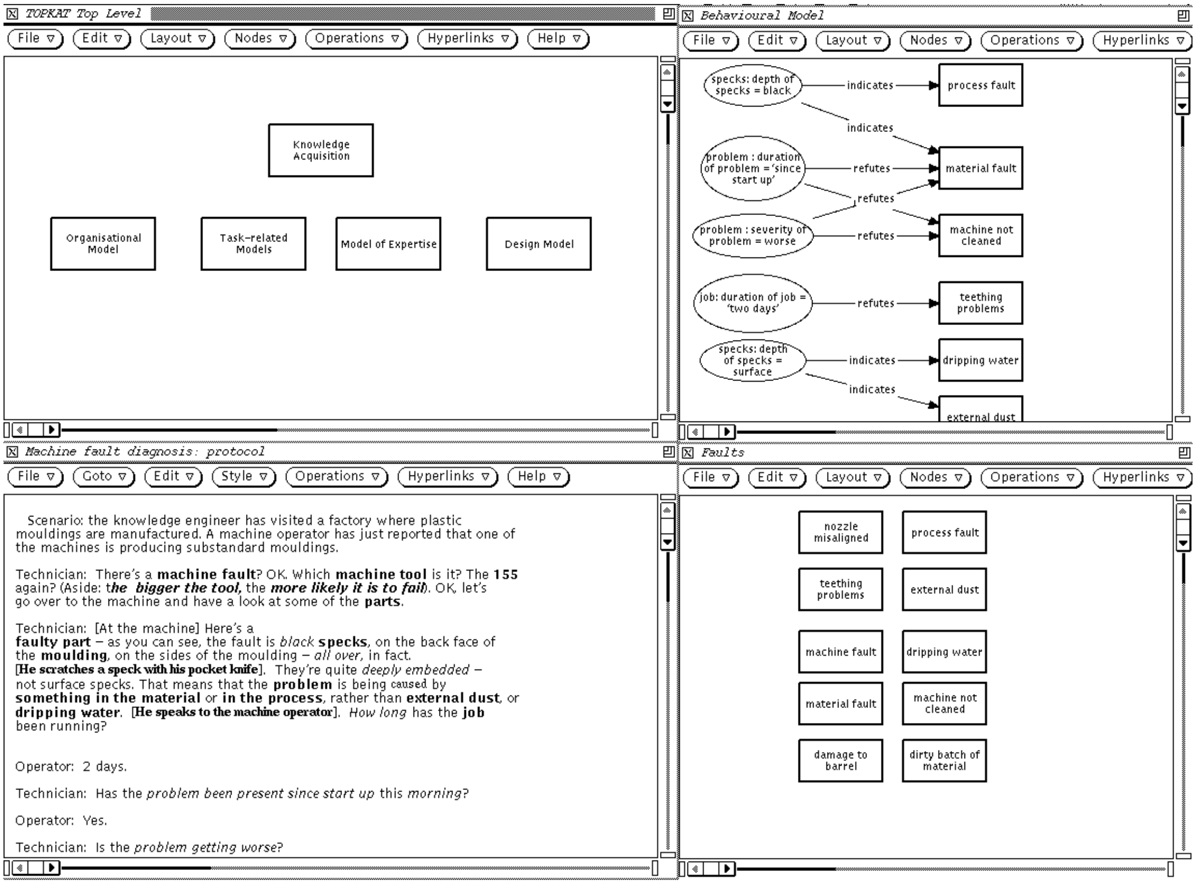

Figure 1: A selection of the hypercards provided by TOPKAT.

The format of the paper is:

- A description of the knowledge acquisition techniques which are implemented in TOPKAT;
- A brief description of the CommonKADS methodology (with particular emphasis on the domain knowledge in the expertise model);
- A description of the links between each knowledge acquisition technique and the CommonKADS classification system.

2 TECHNIQUES FOR KNOWLEDGE ACQUISITION

The most widely used method for knowledge acquisition has been the *interview* which, as the name implies, requires a knowledge engineer to interview an expert.

The normal approach is to record the entire conversation, transcribe the interview, and analyse the transcript in order to identify and extract relevant items of knowledge. Transcript analysis does provide useful knowledge, and the transcript forms a good record of the source of that knowledge; however, transcript analysis is time-consuming, prone to generate much irrelevant information, and provides no guarantees about the completeness of the knowledge acquired [Wells, 1994]. Alternative methods for obtaining a transcript, such as performing carefully structured interviews, or asking the expert to talk through a case history (*protocol analysis*), have been developed to provide more structured transcripts; such transcripts alleviate the problems associated with transcript analysis, but do not remove them.

In order to overcome some of these problems, knowledge engineers have drawn on the field of psychology to produce techniques such as the *card sort*, the *repertory grid* and the *laddered grid*. The benefits of these techniques are that they provide output in the form of categorisations and relationships; that they ensure complete coverage of knowledge, by continual prompting or by requiring all items to be categorised; and that they are relatively simple to administer. The repertory grid has been particularly well used, with over 150 applications to date having used this technique successfully [Boose & Gaines, 1988].

2.1 Laddered Grid

The laddered grid uses pre-defined questions to persuade an expert to expand a taxonomic hierarchy to its fullest extent [Burton *et al*, 1988]. Starting with a single domain term, the questions can elicit superclasses, subclasses or members of classes, which are linked to the existing object in a hierarchical "grid". Typical questions include "What is *term* and example of?", or "What other examples of *term-1* are there apart from *term-2*?". By repeatedly applying the same procedure to newly elicited objects, an extensive taxonomy can be built up.

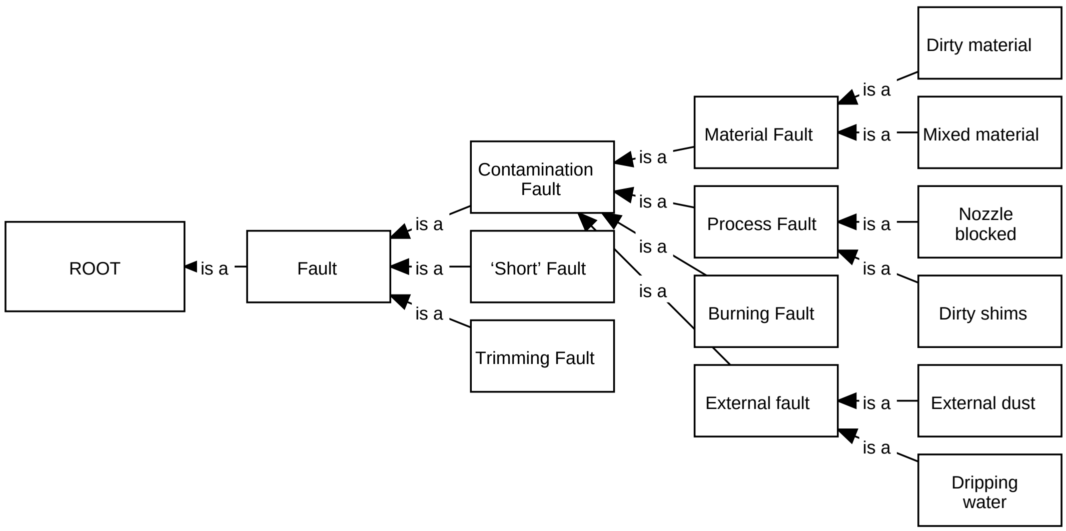

Figure 2: A laddered grid diagram

2.2 Card Sort

The card sort [Shadbolt & Burton, 1990] is a simple but surprisingly effective technique in which an expert categorises cards which represent terms from the knowledge domain. The names of various terms from the domain are written on individual index cards, and the expert is presented with the pile of cards and asked to sort them into piles in any way which seems sensible. When this has been accomplished, the classification of each card is noted, the cards are shuffled, and the expert is asked to repeat the procedure using a different criterion for sorting. This process is repeated until the expert cannot think of any more criteria on which to differentiate the cards. The output of the card sort is a set of classifications of domain terms into one or more categories on many different dimensions.

Figure 3 shows the result of a single categorisation of a set of 'cards' representing vehicles. In TOPKAT, the 'cards' are sorted into columns rather than piles.

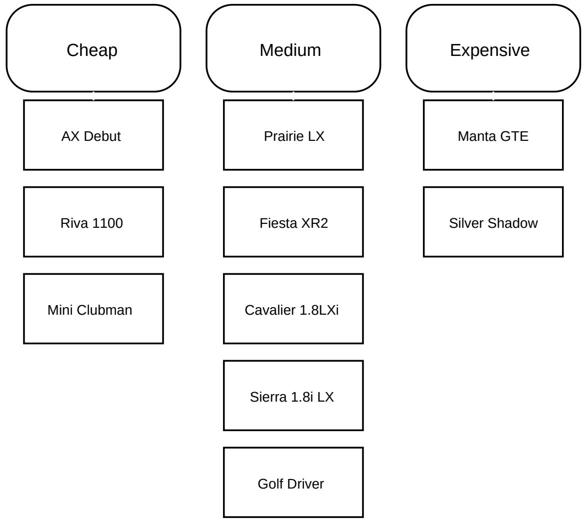

Figure 3: A set of cards representing vehicles, sorted according to price

2.3 Repertory Grid

The repertory grid is a technique in which an expert makes distinctions between terms in the domain on chosen dimensions [Boose, 1989]. The dimensions are similar to the categories generated by card sorting, except that they are all assumed to be continuous variables. Dimensions are usually generated by the 'triadic' technique – selecting three domain terms at random and asking the expert to name one way in which two of them differ from the third. All domain terms in the grid are then classified on each dimension (normally using a 1-5 or 1-7 scale), resulting in a grid in which every term is categorised on every dimension (see Figure 3).

One of the features of the repertory grid which sets it apart from other knowledge acquisition techniques is that the classifications in the grid can be analysed statistically, using cluster analysis, to see if the expert has implicitly categorised the terms in any way. The clustering of concepts produced by statistical analysis of the repertory grid is normally represented by a *dendogram* (literally a tree diagram), in which every domain term is a leaf node, and closeness in the 'tree' represents statistical similarity. TOPKAT represents the statistical clustering as a laddered grid (i.e. a taxonomy), in which the domain terms form the leaf nodes, and the "classes" indicate the level of similarity between domain terms using a percentage value (100% indicates the two objects are identical on all the dimensions, 0% indicates that they are at opposite ends of the spectrum on every dimension). The expert and/or the knowledge engineer is then allowed to rationalise this taxonomy by assigning meaningful names to some classes and deleting others. For example, Figure 5 shows statistical similarity of crimes (derived from the repertory grid shown in Figure 4), and Figure 6 shows a rationalised version of this hierarchy.

Scale	Crimes	Pilfering	Theft	Drug taking	Murder	Assault	Rape	Fraud	Speeding
1 = only men 5 = only women	sex-specificity	4	3	3	3	3	5	3	2
1 = Too long 5 = Too short	Severity of punishment	3	1	1	2	3	5	3	4
1 = Sensational 5 = Common	Frequency	3	5	1	1	4	5	4	5
1 = Premeditated 5 = Casual	Forethought	5	3	1	2	5	4	1	5
1 = Pleasureable 5 = Nasty	Pleasure for victim	2	2	1	5	5	5	2	4
1 = Nonpersonal 5 = Personal	Personal nature	2	2	1	5	4	5	2	1
1 = Petty 5 = Major	Seriousness	1	3	1	5	4	5	4	1
1 = Nonviolent 5 = Violent	Violence	1	1	2	5	4	5	1	1
1 = Full 5 = None	Possibility of restitution	1	1	2	5	4	5	1	5
1 = Very small 5 = Very large	Benefit to perpetrator	2	3	2	3	2	1	5	1

Figure 4: A repertory grid classifying crimes

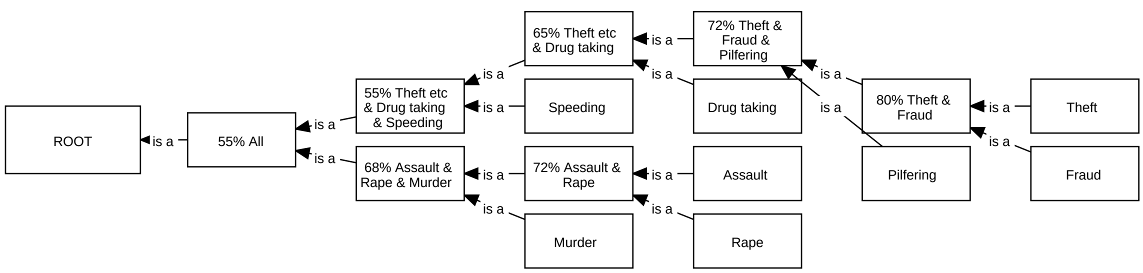

Figure 5: A statistical analysis showing an implicit categorisation of crimes

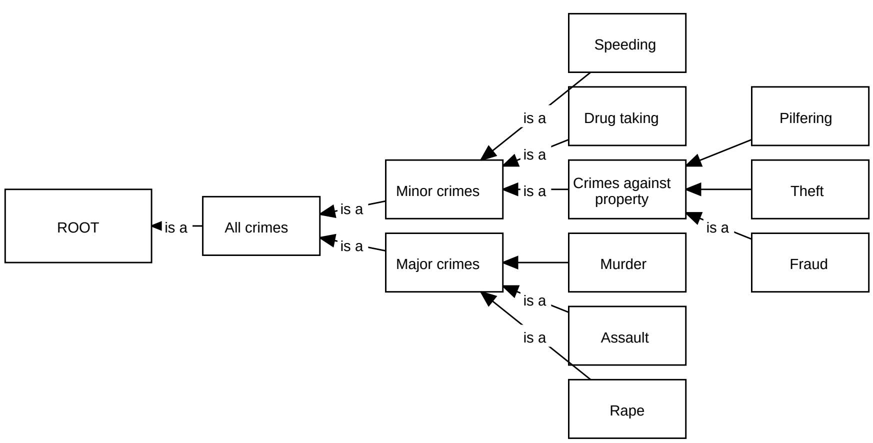

Figure 6: The hierarchy of crimes shown in Figure 5, after rationalisation

3 KNOWLEDGE REPRESENTATION IN COMMONKADS

CommonKADS is the name of the methodology developed by the KADS-II project, which was funded under the CEC ESPRIT programme [Wielinga, 1993]. CommonKADS views KBS development as a modelling process. Knowledge analysis is performed by creating a number of models which represent the knowledge from different viewpoints. CommonKADS recommends a number of different models, which start with the representation of various aspects of an organisation, and support the whole knowledge engineering process up to the point of producing a detailed design specification.

The key model – the *expertise model* – is divided into three "levels" representing different viewpoints on the expert knowledge:

- The **domain knowledge** which represents the declarative knowledge in the knowledge base;
- The **inference knowledge** which represents the knowledge-based inferences which are performed during problem solving;
- The **task knowledge** which defines a procedural ordering on the inferences.

The contents of these three levels can be defined graphically, or using CommonKADS' Conceptual Modelling Language. For a worked example of the development of each of these three levels, see [Kingston, 1993].

3.1 Modelling domain knowledge

The domain knowledge in the model of expertise represents the declarative knowledge which has been acquired. CommonKADS suggests that each item of declarative knowledge is classified into one of four ontological types. These types are:

- **Concepts**: classes of objects in the real or mental world of the domain studied, representing physical objects or states;
- **Properties**: attributes of concepts;
- **Expressions**: statements of the form "the *property* of *concept* is *value*";
- **Relations**: links between any two items of domain knowledge.

Once items of domain knowledge have been classified, they can be used in *domain models*, which show relations between different items of knowledge. For example, a domain model might show all acquired examples of one concept **causing** another; or it might show a taxonomic hierarchy of concepts, connected to each other by **is-a** relations. Figure 7 shows an example of a domain model.

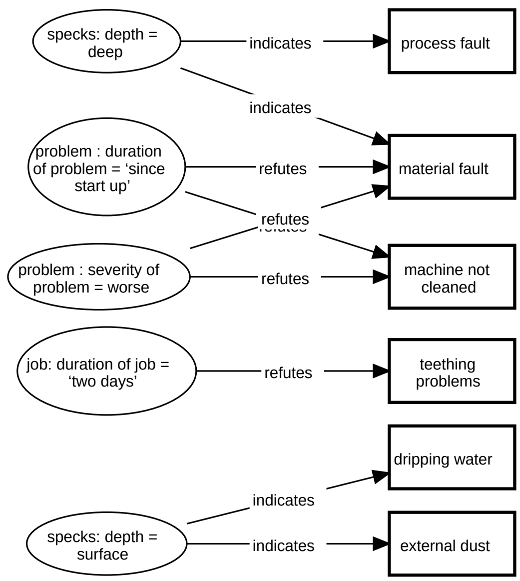

Figure 7: A domain model which links expressions with concepts

4 MAPPING ACQUIRED KNOWLEDGE TO THE COMMONKADS DOMAIN LEVEL

It can be seen from the previous two sections that the four knowledge acquisition techniques which are supported by TOPKAT produce output in differing formats, some of which are quite similar to CommonKADS' suggested representations of knowledge, and some of which are not. This section describes how TOPKAT uses the structure provided by each format to support identification and ontological classification of knowledge.

4.1 Transcript analysis: classification according to word class

Textual transcripts differ from the output of the other knowledge acquisition techniques supported by TOPKAT in that they rarely produce knowledge which is obviously structured in a taxonomic or relational manner. Of course, language does have a structure; words take on the role of different *word classes* (parts of speech), which may only appear in particular combinations permitted by the rules of grammar. Is it possible to make use of the grammatical structure of language to perform ontological classification?

The starting point for this discussion is Woods' linguistic test [Woods, 1975]. While discussing the nature of links in semantic networks, Woods asserts that, given an object O, it is possible to use a linguistic test to determine if A is an attribute of O. The test is that it must be possible to state that "V is the/an A of (some) O". If this test is passed, then in CommonKADS terminology, O is an concept, A is a property of that concept, and V is a value of that property. From a grammatical viewpoint, however, it can be seen that O must be a noun, A must be a singular noun, and V must be an adjective which modifies O. From this analysis, it seems that there is some connection between the CommonKADS ontology and word classes in English (and grammatically similar languages).

A second link between the CommonKADS ontology and word classes can be found in the definitions of some of the ontological types within CommonKADS (see section 3.1).

- Concepts are classes which represent objects or states. The Shorter Oxford Dictionary defines nouns as "names of persons or things"; if it is assumed that all objects or states are "persons or things" in the "real or mental world of the domain" (cf. [Wielinga *et al*, 1992]), then it can be seen that all concepts can be named using nouns.

- Properties are attributes of concepts. It is difficult to assign attributes to particular word classes, because while attributes can be represented as singular nouns (according to Woods' linguistic test), they may also be identified using plural nouns (e.g *instances*) or verbs (e.g. *has-part*). Nor is the preposition "of" a universal indicator of a property: other prepositions may sometimes be used instead (e.g. O1 is *connected to* O2), and the word "of" may appear in idioms such as "a matter of course".

- Relations form links between concepts in which one concept affects another. This is normally accomplished linguistically by a transitive verb, and so it seems that a verb which links two objects or states probably indicates a relation. The identification of transitive verbs with relations is further supported by the correspondance between adverbs and CommonKADS' facility which allows relations to have properties of their own; if relations correspond to verbs, then adverbs represent (values of) properties of relations. For example, in the sentence "Peter married Jane yesterday", the adverb (*yesterday*) would be classified as a value of the *date of marriage* property.

From these analyses, it seems that identification of nouns, adjectives, verbs, and the words which they modify (if any) can provide a great deal of information for ontological classification in CommonKADS. There is therefore considerable potential for automated classification if a textual transcript can be parsed (providing grammatical information), or at least lexically tagged, so that the word class of each word is known.

TOPKAT uses the analyses above to support partial classification of a transcript.[1]. A publicly available tagging package is used to add lexical tags to a transcript; once this has been performed, TOPKAT offers the user the options of identifying concepts and properties in the transcript. This is accomplished by:

- Collecting all nouns in the transcript into a list (classifying any instances of two adjacent nouns as a single compound noun);
- Sorting the nouns according to their frequency of occurrence in the transcript, compared with their expected frequency in everyday English. Nouns which appear much more frequently than expected are placed at the head of the list, on the basis that these nouns are more likely to represent domain-specific concepts.
- Presenting the list of nouns to the knowledge engineer, and asking which nouns represent concepts that are relevant to problem solving.
- Identifying any adjectives which immediately precede concepts in the transcript, and using a question based on Woods' linguistic test to define a name for the property associated with that adjective.

This approach was used to produce the classification shown in Figure 8.

TECHNICIAN: Here's a *faulty* **part** – as you can see, the **fault** is *black* **specks**, on the *back* **face** of the **moulding**, on the sides of the moulding – all over, in fact. [He scratches a speck with his **pocket knife**]. They're quite deeply embedded – not surface **specks**. That means that the problem is being caused by something in the **material** or in the **process**, rather than *external* **dust**, or *dripping* **water**. [He speaks to the machine operator]. How long has the **job** been running?

Key:

Concepts are in **bold font**

Properties are in *italics*

Figure 8: Transcript classified using TOPKAT's semi-automatic natural language analysis

[1]It is expected that future versions of TOPKAT will provide more comprehensive support; see section 5.

Two features of this approach to classification are immediately obvious: firstly, that it is highly interactive; and secondly, that it is based on a pragmatic but simple approach to natural language understanding, which means that it is vulnerable to errors in lexical tagging and in adjective/noun pairing. The key to the success of TOPKAT's approach is that these two features balance each other out. Much of the work which has been carried out on understanding natural language has attempted to analyse language with maximum accuracy and minimum human intervention; despite the high level of sophistication of some systems, it has proved difficult to comprehend language unambiguously without considerable use of general knowledge, which is difficult to encode. TOPKAT's natural language capabilities, however, are complemented by the domain knowledge and general knowledge of the knowledge engineer using the system, which enables an accurate and largely complete classification to be produced; while for the knowledge engineer, providing guidance to TOPKAT is much less effort than performing transcript analysis without assistance.

The usefulness of TOPKAT's approach was verified during a practical exercise on the CATALYST training course, in which the attendees were asked to perform transcript analysis and ontological classification. Despite the availability of hypertext-based software support, the task took more than an hour; but a knowledge engineer using TOPKAT identified the important concepts and properties in about 10 minutes.

4.2 Laddered grid: from one taxonomy to another

The output of the laddered grid technique is a taxonomic hierarchy of domain objects. It is taxonomic because the prompt questions which are used should only generate examples or subclasses of other domain terms[2]; the domain terms are assumed to be objects which are capable of possessing subclasses or examples.

On the basis of this structure, each term in the laddered grid is classified as a *concept* in the CommonKADS domain ontology, and the entire laddered grid is mapped to a taxonomic *domain model* at the CommonKADS domain level.

4.3 Card sort: roles, qualities, parts and taxonomies

The card sort produces a number of domain *terms* which are classified into different *categories* on a number of *dimensions* (sorts). It can be seen that the categories supplied for each dimension form a range of possible values for that dimension; this correlates closely with the relationship between properties and values, and so

[2] The laddered grid technique can also be used with different sets of prompt questions, in which case the taxonomic assumption will not apply. TOPKAT currently only supports prompt questions which will generate a taxonomic laddered grid.

it seems likely that dimensions will map to properties in the CommonKADS domain ontology, and categories will map to values of those properties. Furthermore, the dimensions appear be properties of the domain terms, which implies that the domain terms should be mapped to concepts, as in the laddered grid.

However, it turns out that dimensions cannot be uniformly classified as properties. The reason for this is that the flexibility of the card sorting technique; the expert is simply asked to "sort the cards in any way which seems sensible". The resulting dimensions might differentiate the cards in several ways. For example, if knowledge acquisition was being performed to learn about the task of maintaining a zoo, then a card sort might be performed with the name of a zoo animal on each card. The resulting card sorts might include:

- A sort according to the animals' size, with categories such as "small", "medium" and "large". In this case, *size* can safely be assumed to be a property of each animal;
- A sort according to the genus of the animals (reptiles, mammals, etc). This is clearly a *taxonomic* classification of animals;
- A sort according to the zoo collection to which animals belong, which may include categories such as "monkey house" or "children's corner". In this case, the animals are considered as *part* of a particular collection, which in turn is part of the zoo's overall population. This constitutes a hierarchical (though non-taxonomic) classification of animals.

The approach taken in TOPKAT to classification of dimensions is to use questions to guide the knowledge engineer to an appropriate classification for each sort. TOPKAT's guidance starts by obtaining a name for the property. The name is obtained by asking a question based on Woods' linguistic test (see section 4.1). Using zoo animals as an example, the question would be:

```
Small is the/an WHAT of Hamster?
```

which is generated by selecting one card (**Hamster**) from the first category (**Small**) and instantiating a template question with these two values.

Once the prospective property has been named, it is necessary to determine whether it really is a property. This is achieved in TOPKAT by asking further questions of the knowledge engineer. The questions are derived from a semi-formal approach which classifies concepts into roles, qualities, parts and "natural concepts"[Guarino, 1992]; this approach is altered slightly so that it can be used to classify properties rather than concepts.[3] The classification is illustrated in Figure 9.

[3] Many properties can be considered to be concepts in their own right, if a different viewpoint is taken on the domain knowledge. The implications of this important insight for re-usability of domain knowledge are explored in [Robertson, 1993].

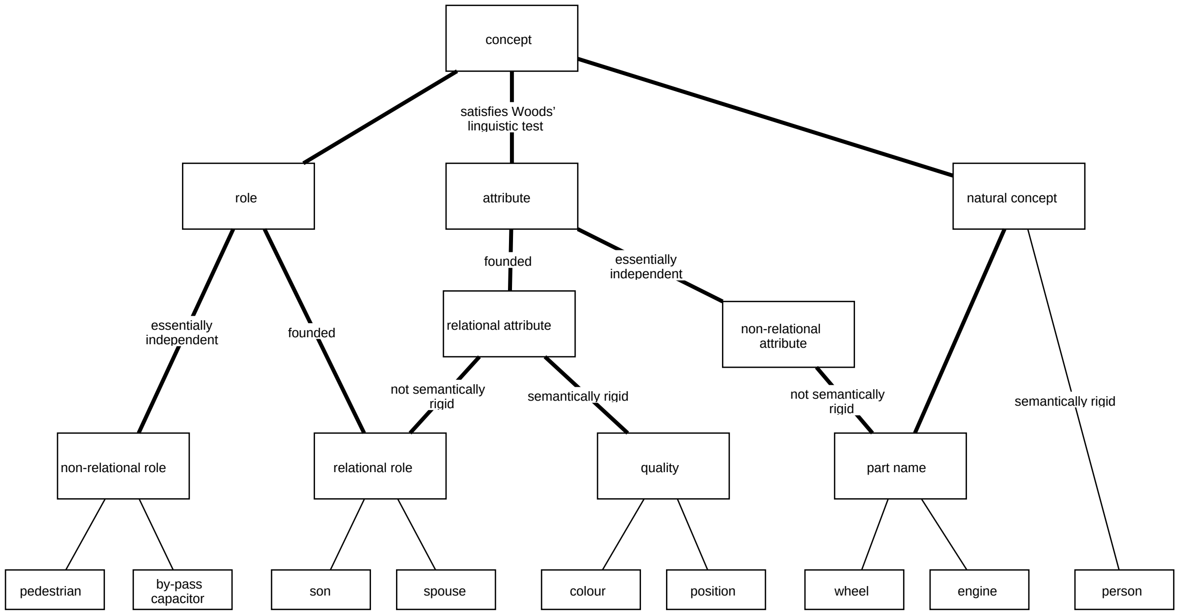

Figure 9: Ontology of attributes (from [Guarino, 1992])

It can be seen from Figure 9 that classification depends on determining :

- Whether the prospective property is *founded* or *essentially independent*. A property is considered to be founded if it can only exist if its accompanying concept also exists; for example, the *size* of an animal is founded, but the *zoo collection* of a animal is essentially independent, because the collection can exist even if the animal does not exist. The foundedness of a prospective property is determined by asking " Can the/an *property* of *concept* exist if (the/an) *concept* does not exist?"; e.g.

  ```
  Can the/an Zoo collection of Hamster exist even if (the/an) Hamster
  does not exist?
  ```

- Whether the property is *semantically rigid* or not. A property is semantically rigid if it is a necessary condition for the identity of its **value**; for example, *colour* is semantically rigid, because **red** must be a colour in order to exist, but *mate* is not semantically rigid, since an animal does not need a mate in order to exist.

  Developing a suitable question to determine semantic rigidity is not as simple as it first appears. The template "Is *value* necessarily a *property*?" can be used to distinguish parts from natural concepts; however, it is likely to obtain the wrong answer (`No`) when instantiated with **Small** and **Size**, because **Small** is a possible value of many properties. This problem can be circumvented by making use of another of Guarino's observations: that the values of qualities (which are semantically rigid) can be considered as predicates, whereas the values of roles (which are not semantically rigid) can be described as instances of the role. On the basis of this, the question which was devised for distinguishing between relational roles and qualities was "Is *value* an instance of *property*, or a predicate describing the value of *property*?". For example,

```
Is Red an instance of Colour, or a predicate describing the value
of Colour?
```

TOPKAT therefore asks if a dimension is founded, and then asks the appropriate question to determine semantic rigidity. On the basis of the answers to these two questions, a property can be classified into Guarino's suggested ontology. If a property is defined as a relational role or a quality, then it is added to CommonKADS' domain ontology as a property (along with its possible values); if the property is defined as a *part* relation, then each category is taken to be a concept, and an appropriate domain model is created or updated; and if the 'property' is a natural concept (e.g. the **Genus** of an animal), then a taxonomic hierarchy is created, in which each category is linked to the new concept by a *subclass* link. Non-relational roles (such as *pedestrian* or *by-pass capacitor*) should be filtered out by Woods' linguistic test.

4.4 Repertory grid: assigning names to numbers

The repertory grid technique produces two outputs. The first is the repertory grid itself; the second is the statistical clustering produced by comparing the elements using the numbers in the grid. It has already been seen (in section 2.3) that the statistical clustering can be represented as a laddered grid; in order to produce a meaningful hierarchy, the "classes" which represent statistical closeness must be interpreted, which TOPKAT currently handles by asking the knowledge engineer and/or the expert to perform interpretation and rationalisation.

Once the hierarchy has been rationalised, TOPKAT treats it as if it were a laddered grid. The domain terms are therefore mapped to concepts in the CommonKADS domain ontology, and the (rationalised) statistical clustering is converted into a taxonomic domain model.

As for the repertory grid itself, it is necessary to decide how the dimensions and the accompanying numeric values should be treated. The questions used to determine property classification for card sorts are used to classify dimensions in a repertory grid; however, before the property classification questions can be asked, the numeric values in the repertory grid must be translated into textual values. TOPKAT makes use of the assumption that dimensions in the repertory grid are continuous variables to generate text which corresponds to each value; this text is based on the name of the dimension, and the names of the poles (low and high values) assigned by the knowledge engineer. Using the repertory grid shown in Figure 3 as an example, TOPKAT will generate the following text for the *Frequency* dimension:

1. Sensational
2. Fairly Sensational
3. Average Frequency
4. Fairly Common
5. Common

The knowledge engineer is prompted to edit this text until satisfied with it. This text will then be used in the property classification questions (and in the domain ontology); so the knowledge engineer will be asked:

`Can the/an Frequency of Theft exist even if Theft does not exist?`

and

`Is Sensational an instance of Frequency, or a predicate describing the value of Frequency?`

The answers to these questions should be "No" and "Predicate" respectively, which classifies Frequency as a quality.

The repertory grid can also be used to generate a large number of *expressions* in the CommonKADS domain ontology – one expression for each numeric value in the grid. These expressions could be used as individual conditions of production rules, which is the principle used by tools such as KITTEN and NEXTRA to derive rules from repertory grids [Shaw & Gaines, 1987].

5 CONCLUSION

It can be seen that all the knowledge elicitation techniques supported by TOPKAT produce output which consists not only of knowledge, but of a structure within which knowledge is stored. The output of these knowledge elicitation techniques can be used to generate concepts, properties, expressions, relations, and even domain models directly, with only occasional assistance from the knowledge engineer. This semi-automatic generation of domain knowledge is not only useful as a labour-saving device, but also introduces a degree of consistency into the domain ontology: Woods' linguistic test in particular enforces naming discipline on properties, and helps to differentiate properties from concepts. It is even possible that the effort of conforming with a naming discipline will produce new insights about the conceptual structure of a domain [Guarino, 1992], thus leading to more accurate knowledge modelling.

There are many opportunities for future work on improving the linking of knowledge acquisition techniques to CommonKADS knowledge representation:

- For the card sort, the classification of properties into *part* relations could be extended by using a *mereology* (classification scheme for *part* relations) e.g. that suggested in [Gerstl & Pribbenow, 1994].
- For the card sort and the repertory grid, Woods' linguistic test could be used when dimensions are created. While this might restrict the breadth of the acquired knowledge, it should produce a more coherent set of dimensions, which is particularly important in the repertory grid where dimensions are compared against one another. The effort of finding a correct name would also be transferred from the knowledge engineer to the expert by this technique.
- For transcript analysis, there are many possible improvements:
  - Use a chart parser to obtain grammatical information about a transcript, permitting extensive automatic identification of properties, and perhaps of relations;
  - Feed back linguistic information obtained from a knowledge engineer to the lexical tagger or parser, to improve accuracy;
  - Define and apply a "coding schema" [Wells, 1993] – a set of phrases which are known to indicate the presence of certain ontological types;
  - Use questionnaires or structured interviews to obtain highly structured transcripts which are written in simple declarative sentences. It should be possible to parse these transcripts and classify the knowledge contained therein without any human intervention (cf. [Inder *et al*, 1993]).

It is hoped that some of these refinements will be introduced into TOPKAT in the near future.

References

[Boose & Gaines, 1988] Boose, J. and Gaines, B. (1988). *Knowledge Based Systems.* Academic Press, Vol 1: Knowledge Acquisition for Knowledge-based Systems

Vol 2: Knowledge Acquisition Tools for Expert Systems.

[Boose, 1989] Boose, J.H. (1989). A survey of knowledge acquisition techniques and tools. *Knowledge Acquisition*, 1(1).

[Burton *et al*, 1988] Burton, A.M., Shadbolt, N.R., Rugg, G. and Hedgecock, A.P. (1988). Knowledge Elicitation Techniques in Classification Domains. In *Proceedings of ECAI-88: The 8th European Conference on Artificial Intelligence.*

[Fox *et al*, 1987] Fox, J., Myers, C.D., Greaves, M.F. and Pegram, S. (1987). A systematic study of knowledge base refinement in the diagnosis of leukemia. In Kidd, A.L., (ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook*, chapter 4, pages 73–90. Plenum Press.

[Gerstl & Pribbenow, 1994] Gerstl, P. and Pribbenow, S. (1994). Midwinters, End Games, and Bodyparts. In Guarino, N. and Poli, R., (eds.), *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Dordrecht. Kluwer. Forthcoming.

[Guarino, 1992] Guarino, N. (1992). Concepts, attributes and arbitrary relations. *Data & Knowledge Engineering*, 8:249–261. North-Holland.

[Inder *et al*, 1993] Inder, R., Goodfellow, E. and Uschold, M. (June 1-4 1993). Knowledge Engineering without Knowledge Elicitation. In P. Chung, G. Lovegrove and M. Ali, (ed.), *Proceedings of the Sixth International Conference on Industrial and Engineering Applications of AI and Expert Systems*, City Chambers, Edinburgh. Gordon and Breach. Also available as AIAI-TR-126.

[Johnson & Johnson, 1987] Johnson, L. and Johnson, N.E. (1987). Knowledge elicitation involving teachback interviewing. In Kidd, A.L., (ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook*, chapter 5, pages 91–108. Plenum Press.

[Kidd, 1987] Kidd, A., (ed.). (1987). *Knowledge Acquisition for Expert Systems: A Practical Handbook*. Plenum Press.

[Kingston, 1993] Kingston, J.K.C. (1993). Re-engineering IMPRESS and X-MATE using CommonKADS. In *Expert Systems 93*. British Computer Society, Cambridge University Press. Also available as AIAI-TR-130.

[Kuipers & Kassirer, 1987] Kuipers, B. and Kassirer, J.P. (1987). Knowledge acquisition by analysis of verbatim protocols. In Kidd, A.L., (ed.), *Knowledge Acquisition for Expert Systems: A Practical Handbook*, chapter 3, pages 45–71. Plenum Press.

[McGraw & Harb.-Briggs, 1989] McGraw, K.L. and Harbinson-Briggs, K. (1989). *Knowledge Acquisition: Principles and Guidelines*. Prentice-Hall International.

[Reichgelt & Shadbolt, 1991] Reichgelt, Han and Shadbolt, Nigel. (1991). ProtoKEW: A knowledge based system for knowledge acquisition. In Sleeman, D. and Bernsen, O., (eds.), *Recent Advances in Cognitive Science:*. Lawrence Earlbaum Associates.

[Robertson, 1993] Robertson, S. (Sept 1993). A KBS to advise on selection of KBS tools. Unpublished M.Sc. thesis, Dept of Artificial Intelligence, University of Edinburgh.

[Shadbolt & Burton, 1990] Shadbolt, N. and Burton, A.M. (1990). Knowledge elicitation. In J. Wilson and N. Corlett, (ed.), *Evaluation of Human Work: A Practical Ergonomics Methodology*, pages 321–346. Taylor and Francis.

[Shaw & Gaines, 1987] Shaw, M.L.G. and Gaines, B.R. (1987). KITTEN: Knowledge initiation and transfer tools for experts and novices. *International Journal of Man-Machine Studies*, 27:251–280.

[Tansley & Hayball, 1993] Tansley, D.S.W. and Hayball, C.C. (1993). *Knowledge-Based Systems Analysis and Design: A KADS Developers Handbook*. Prentice Hall.

[Wells, 1993] Wells, S. (1993). Configuration Control. KADS-II Project Report KADS-II/M2.2/TN/LR/0099/0.2, Lloyds Register.

[Wells, 1994] Wells, S. (1994). Data-Driven Modelling. In *Expertise Modelling in CommonKADS*.

[Wielinga, 1993] Wielinga, B. (October 1993). Expertise Model: Model Definition Document. CommonKADS Project Report, University of Amsterdam, KADS-II/M2/UvA/026/2.0.

[Wielinga *et al*, 1992] Wielinga, B., Van de Velde, W., Schreiber, G. and Akkermans, H. (1992). The KADS Knowledge Modelling Approach. In *Proceedings of the Japanese Knowledge Acquisition Workshop (JKAW'92)*.

[Woods, 1975] Woods, W.A. (1975). What's in a link: Foundations for semantic networks. In Bobrow, D.G. and Collins, A.M., (eds.), *Representation and Understanding: Studies in Cognitive Science*, New York. Academic Press. Also in R. Brachman and H. Levesque, eds., *Readings in Knowledge Representation* (Morgan Kaufmann, Los Altos, CA, 1985).