Facilitating the Development of Knowledge Based Systems

A Critical Review of Acquisition Tools and Techniques

Hyacinth S. Nwana, Ray C. Paton, Trevor J. M. Bench-Capon& Michael J. R. Shave

The MEKAS Project, Department of Computer Science, University of Liverpool Liverpool, L69 3BX, U.K. Tel: +44 51 794 3781 Email: nwanahs@and.cs.liv.ac.uk

Certain problems associated with knowledge acquisition are identified and examined in this paper. We review a variety of methodologies and tools designed to address these problems and then argue that there is a strong case for a preliminary knowledge analysis or domain phase of KBS development. This phase facilitates subsequent design, development and maintenance phases. The details of our domain characterisation are not expounded upon in this paper. The paper concludes with a suggestion of a re-examination of some of the central metaphore acquisition.

Introduction

Knowledge acquisition is a key stage in any methodology for constructing Knowledge Based Systems (KBSs); deciding what knowledge should be brought to bear on a problem, how the knowledge can be used in a program, how to elicit, interpret, organise, model (i.e. represent) and encode it in a KBS are all aspects of knowledge acquisition. This process is complex and fraught with difficulties, some of which are discussed later; it is also often a lengthy and painful process. For these reasons, it has long been identified as the main bottleneck in the development of KBSs. This metaphor has its origin in the most popular principle of KBSs which states that the performance of a KBS critically depends on the amount of knowledge embedded in the system (Feigenbaum, 1977). Indeed, for more than 15 years now, since Davis's (1976) landmark TEIRESIAS program, researchers in Artificial Intelligence (AI) have viewed knowledge acquisition as a problem of 'expertise transfer'. Henceforth, the traditional knowledge engineer is the intermediary who must interview the expert and transfer the expert's knowledge to some computer representation. Since, by definition, the knowledge engineer is not very knowledgeable in the problem area and because the expert is unlikely to be able to encode his knowledge in a computer readable directly form him/herself, failures in communication are inevitable.

Ongoing research at Liverpool University over the last seven years has confirmed certain major problems associated with knowledge acquisition; these issues that will have to be addressed if the development of KBSs is to be facilitated. In this paper, we examine some of these problems. There are, so far, numerous methodologies that have emerged supposedly to address them; we review some and consider how successful they are. Furthermore, there is an

This article has been awarded the DIGITAL European AI Prize at the Avignon'91 Conference. Further information on page 66.

abundance of tools to support knowledge acquisition (see Boose, 1989). We provide a selective critical review of these tools. Following on from these analyses, we suggest the need for a preliminary characterisation stage of knowledge acquisition which places a primary focus on the nature of the domain. This preliminary phase is crucial to the success of second generation expert systems (Steels, 1984).

Major Problems Associated with Knowledge Acquisition

Major problems associated with knowledge acquisition include:

1. It can be hard for the domain expert and knowledge engineer to harmonize their mental models (Recogzei & Plantinga, 1987). Harmonization cannot be achieved directly as we cannot just merge the mental models together; the engineer and the expert are constantly revising their mental models using natural language (Motta *et al.*, 1990).

2. There often exist mismatches between elicitation techniques and the structure of the problem domain. Knowledge Elicitation is the process of extracting knowledge from an expert, to produce what may be called the 'raw data'. Such techniques range from structured and unstructured interviews with the expert to psychologically based methods such as laddered grid or card sorting; they are reviewed in Welbank (1990). Their efficacy varies considerably depending on the following (Burton *et al.*, 1987):
   - the type of knowledge the knowledge engineer is trying to obt;
   - the task in hand;
   - the psychological make-up of the expert.

It is suggested that many of these techniques can be criticised for imposing their structure upon the elicitation process and hence the domain. For example, multi-dimensional techniques including repertory grids and card sorting mainly elicit declarative knowledge while think aloud protocols primarily aim at eliciting procedural knowledge. Hence, these techniques, as suggested by the above three factors of Burton *et al.* have limited ranges of applicability.

A primary reason for this mismatch problem lies in the fact that literature does not make clear *when* and *where* to use these techniques; e.g. even when advice is offered by commentators concerning interviewing approaches for example (Grover, 1983; Welbank, 1989), this still fails to overcome difficulties. Current elicitation techniques can be classified into formal *vs* informal approaches, direct *vs* indirect methods and weak (domain independent) *vs* strong (domain dependent) (Motta *et al.*, 1990). Such a classification should be associated with, say, Burton *et al.*'s factors: it must be made clear what the presuppositions of these techniques are and when and where to use them in order to avoid this mismatch between techniques and problem domain structure (and sometimes the expert). Most crucially, and as we argue elsewhere (Paton *et al,* 1990a) and later in this paper, it is the nature of the domain that should guide the elicitation process (see also Woodward, 1989).

3. It is hard for the knowledge engineer to navigate and make sense of the sheer mass of information obtained. A recent comprehensive UK survey reported that KBS developers did not consider knowledge elicitation from the expert in itself a problem, rather, it was making sense and imposing a coherent organisation upon the elicited data and representing it (O'Neill & Morris, 1989). This is clearly seen when the knowledge exists in written form, as in legislation based applications (e.g. Bench-Capon, 1990).

4. Despite the numerous current tools (see Boose, 1989), a major obstacle is that little guidance is available to the domain expert or knowledge engineer to help with the following tasks (Kitto & Boose, 1989):
   - classifying the application task and identifying a problem solving method;
   - given the application task characteristics, select ing knowledge acquisition tools and strategies to be applied in creating and refining the knowledge base.

This is not a comprehensive list; others are mentioned later on in the paper. Our goal is to, at least, address these problems in our research.

KBS Methodologies for Knowledge Acquisition

The goal of this section is to overview KBS methodologies to date and examine their contributions to addressing the root causes noted in the previous section. Methodologies are necessary to facilitate the routine development of KBSs using standard methods (taking the art out of the process), to improve their quality and to train novice knowledge engineers.

First generation KBS methodologies are mainly of two types: 'Stage Based' approaches and prototyping approaches. (Strictly speaking, prototyping is not a methodology but a key technique in constructing KBSs; some reasons for this are well explained by Breuker *et al.* (1985)). The stage based approaches present life cycle definitions. For examples, Buchanan *et al.* (1983) propose a life cycle definition including the following stages: identification, conceptualisation, formalisation, implementation and testing/revision; Guida & Tasso's (1989) proposal includes plausibility study, demonstration of prototype, development of full prototype, development of target system, operation and maintenance/revision. Other examples are found in Grover (1983) and Wielinga & Breuker (1983). These methodologies bear similarities to conventional life cycle data analysis methodologies such as SSADM (Downs *et al.*). Clearly, these methodologies are very general as they attempt to address the entire complex knowledge engineering endeavour. In so doing, they fail to acknowledge knowledge acquisition as a separate stage. Indeed, prototyping, which has been much used to date, may not fully utilise the knowledge acquisition process because the underlying assumption is that one can uncover the structure of the expertise of the domain at a very early stage with little or no analysis.

A second generation of KBS methodologies is emerging which has started to acknowledge the complexity of the knowledge acquisition process and sees a solution in the construction, creation or modelling of expertise. These methodologies have suggested languages for conceptual modelling. They include: ontological analysis (Alexander *et al.*, 1986), Sowa's conceptual graphs (Clancey, 1985), approaches based on generic tasks (e.g. Chandrasekaran, 1985) and conceptual modelling (e.g. the KADS methodology (Breuker et al., 1987)).

Ontological Analysis is a methodology designed for knowledge-level analysis (Alexander *et al.*, 1987). Knowledge-level modelling refers to the modelling of an intelligent system's behaviour independent of whatever symbols might ultimately be used to implement those behaviours in a computer, e.g. frames, semantic nets or rules. Hence, at the knowledge level, one does not talk at the level of *how* the rule interpreter works, rather, one describes *what* it does. SUPESPOONS is the representational language which (Alexander *et al.*, 1987) have proposed to carry out such an analysis. Using domain equations of denotational semantics and algebraic specification, SUPESPOONS specifies domain objects in terms of their relationships and transformations; the main objective being to identify and construct an adequate knowledge representation for any given problem.

KADS methodology (Wielinga *et al.*, 1988) is a methodology in which the authors suggest a four layer architecture describing the domain, the type of inferences, the tasks, and strategic structures. They also describe a number of domain-independent conceptual primitives used for representing the inference and task layers. These primitives, called interpretation models in KADS, are quite similar to Clancey's (1986) and Chandrasekaran's (1985) six generic types which depict the levels where knowledge-level analysis is carried out (e.g. heuristic classification); these are further elaborated upon later. KADS's view of the knowledge acquisition process is one of 'interpretation': verbal data (e.g. from interviews with experts or textbooks) is interpreted or mapped onto other representations and structures (Hayward et al. , 1987); classical KBS development map directly from verbal data to, say, production rules with the obvious dangers of misinterpretation and underinterpretation. Although, they can not yet claim that such a set of primitives is complete, their proposal is already robust and detailed enough to be used in a number of practical problems.

Johnson (1989) has proposed modelling knowledge via systemic grammar networks (SGNs) which she espouses as a suitable "mediating representation" which mediates between verbal data and standardised knowledge representation schemes found in AI environments. Her underlying ethos lies in the thesis that premature encoding of knowledge in a KBS-driven representational form is often a hindrance to analysis. Johnson claims that SGNs are a well defined system based on the idea of choice and with no standard rules of application. Its proposed advantages include conceptual modelling without having to translate concepts to a knowledge representation language, allowing for changes through further acquisition as well as acting as a source of documentation.

Sowa's use of conceptual graphs (Clancey, 1985) is an obvious choice of representation language to some researchers (e.g. Recogzei & Plantinga (1987)); they stay close to the structure of the natural language used by both the knowledge engineer and the expert and they provide a clear notation for modelling. In addition, the notation is also directly machine representable. However, all it really provides is a good diagrammatic tool which allows the modelling process to be repeated until both the knowledge engineer and the expert converge on a set of diagrams which they believe adequately represents the domain. It does not guide the modelling as such.

Other AI representations, say logic or production systems (PSs), can be directly used for modelling knowledge (PSs have already been much used in prototyping approaches) but with the obvious disadvantage of the knowledge engineer prematurely concerning him/herself with issues of symbol-level modelling as Johnson warns, without any rigorous prior analysis of the domain concerned.

Discussion on Methodologies

Second generation approaches with their modelling view of the knowledge acquisition process better address the problems of Section 2 than the first generation ones; in fact, the latter hardly address the problem at all as earlier mentioned. Nevertheless, these second generation methodologies still suffer from many shortcomings. The guidelines they provide for conceptual modelling are still fuzzy. These approaches still do not fully address the issue of harmonising the expert's and the knowledge engineer's mental models (Recogzei & Plantinga, 1987); neither do they suggest where and when to use the various elicitation techniques nor what tools to use. KADS is an exception. Despite its limitations (e.g. separating domain and control knowledge is still a controversial one), KADS has advantages over the others: it is principled, it has reusable models and a good toolkit, and it is emerging as the *de facto* standard for knowledge acquisition, at least in Europe. In addition, formal models are being developed for it (Wetter, 1990).

These modelling approaches can be said to prescribe some domain conceptualisation phase, though they themselves mainly provide means/languages of knowledge modelling.

A Selective Critical Review of Knowledge Acquisition Tools

When developing a KBS, a requirements analysis phase is necessary to identify the task that the KBS will perform. Traditionally, knowledge engineers construct a model of the system's proposed behaviour which corresponds to the engineers' theory of how the expert solves problems. This modelling activity is what Newell (1982) refers to as *knowledge-level* modelling. Newell's work has heavily influenced most of the knowledge acquisition tools developed to date. Other investigators have built on his work to characterise generic tasks/methods of problem-solving that are independent of a specific inference engine and of specific application areas (Clancey, 1986, 1985). Hence, a knowledge-level description of a system identifies its abstract data and inference types and its generic control structure. Clancey (1986) identifies heuristic classification and heuristic construction as two fundamental examples of such generic problem-solving methods. The former is a common method in which concepts are heuristically related using a process of data abstraction, heuristic matching and solution refinement (see Figure 1 (adapted from Clancey (1986)); this method is suitable for problems in diagnosis, repair, catalogue selection or skeletal planning.

The latter (i.e. heuristic construction method) constructs solutions by generating complete solutions or assembling them from components while satisfying constraints; this is most suited to synthesis application tasks (design, configuration, scheduling, etc). Chandrasekaran (1985) also describes six generic tasks in knowledge based reasoning which, to some

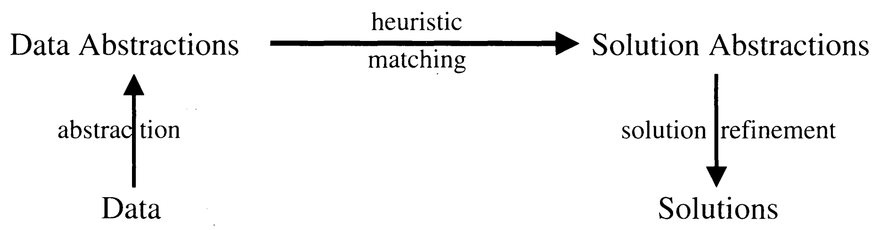

Figure 1. Heuristic Classification problem-solving method

degree, resemble the problem-solving methods proposed by Clancey. They include classification, state abstraction, knowledge-directed retrieval, object synthesis by plan selection and refinement, hypothesis matching, and assembly of compound hypotheses for abduction. Like Clancey, Chandrasekaran views these generic tasks as problem-solving methods which can be combined to perform knowledge-based reasoning for an application task. The influence of such work is evident in the overview of the tools that follow.

Numerous tools currently exist which were designed to support the knowledge acquisition process, even though most of them are just prototypes. They could be classified as:

- Task specific tools
- Problem-solving method specific tools
- Repertory-grid based elicitation tools
- Machine learning tools
- General tools.

Task specific tools (Model-Extension tools)

OPAL (Musen *et al.*, 1987) is an example in this category. It is an interactive program that acquires new cancer treatment plans for an expert system called ONCOCIN (Shortliffe *et al.*, 1981) - a program that provides therapy advice to physicians who take care of cancer patients. Expert oncologists use OPAL to describe new chemotherapy regimens for ONCOCIN by filling out graphical forms and by drawing flowchart diagrams on a workstation display. PROTÉGÉ (Musen, 1989) is the interactive graphics program that assists knowledge engineers to create general models of application tasks that can be solved with the problem-solving method of skeletal-plan refinement (Friedland & Iwaski, 1985), which the expert fills and refines. OPAL uses *task-based* conceptual models. (Note: this is not to be confused with the concept of 'generic tasks' mentioned earlier). OPAL's tasks model includes domain-specific concepts, e.g. chemotherapy, drug, toxic reaction, lab test, etc.) Expert physicians with little computing experience have successfully used these models to enter specific cancer treatment plans. The generic task (or problem-solving method) is skeletal-plan refinement.

Knowledge acquisition can be viewed as comprising of two interrelated stages: building a generic task model - i.e., creating an intention of the proposed system's behaviour; and filling in the specific content knowledge in the domain that is consistent with the general model - i.e., creating extensions (Addis, 1987). PROTÉGÉ falls in the former category (i.e. model building tool) while OPAL falls into the latter (i.e., model-extending tool).

Problem-solving method specific tools

TEIRESIAS (Davis, 1979; Davis & Lenat, 1982) is a classic example in this category. It is a system devised to help with the development and maintenance of a large knowledge base for a particular expert system shell - the EMYCIN shell (Van Melle, 1979). It is basically a failure-driven tool in that interactive transfer appears in the context of discovered shortcomings in a knowledge base. For example, when the expert system fails to function as expected, TEIRESIAS applies its "rule models" to detect missing information, helps place new knowledge into exist-ing rule structures, and assists in changing these rule structures. The problem-solving method it supports is a form of heuristic classification (Clancey, 1986) . TEIRESIAS operates (elicits knowledge) at Newell's (1982) symbol level; users must appreciate both the nature of the representations used to encode knowledge in the target expert system and the consequence of applying particular inference mechanisms to these symbols. Users here must understand how expert systems work. TEIRESIAS was never actually used by expert physicians for whom it was intended largely for this reason and motivated the development of OPAL. ROGET (Bennett, 1985) is another classic tool which conducts a dialogue with its user asking about the *problems* to be diagnosed, about the *causes* of those problems and about *data* to confirm/refute those causes and problems. The resulting knowledge-level specification was then translated and used to develop EMYCIN-based expert systems. ROGET's problem-solving method was thus some form of heuristic classification (see Figure 1).

Another example in this category is SALT (Marcus, 1988). It is based on the propose-and-revise method of heuristic construction: it constructs solutions to a problem by successive revisions. Other examples include MOLE (Eshelman *et al.*, 1987) and its predecessor MORE (Kahn *et al.*, 1985) which are both based on an instantiation of the heuristic classification method called cover-and-differentiate. TEIRESIAS is unique to all the others mentioned in this section in that it is the only one of them whose problem-solving method is quite implicit in its code; the rest have explicit conceptual models of problem-solving.

Repertory grid based tools

ETS (Expertise Transfer System) is an interactive grid-based technique for eliciting knowledge (Boose, 1985). It automates psychiatric interviewing techniques/theories that were originally devised by therapist George Kelly (1955) to learn how people make distinctions of the world; Kelly's theory is called the Personal Construct Theory. (A construct is defined as an internal bipolar scaled dimension which brings out the similarity of a set of elements and the difference of this set of elements from other elements). ETS uses a structured dialogue with an application expert to solicit the elements of the solution set (the possible classifications that may apply to a given case) and the features that may be relevant at arriving at a classification. Thus, it elicits conclusions with their similarities/differences in order to establish traits. By rating pairs of traits, it constructs a rating grid from which it infers an entailment graph which reveals the semantic distance between concepts and possible implicational relations. IF-THEN production rules are then generated (entailed and induced) from this graph. The biggest problem with ETS is that it elicits only simple classifications from the experts; it has difficulty expressing causal, procedural and strategic knowledge. The classification associations produced are also frequently spurious.

AQUINAS (Boose & Bradshaw, 1987) is the successor to ETS which improves on some of the latter's limitations, mainly in representation and reasoning. It allows elicitation from multiple experts and stores it in a hierarchical network of repertory grids. This knowledge can be examined and refined using tools that do clustering, similarity analysis, implicational analysis and consultation testing. These tools use techniques to analyse information in grids and suggest ways to refine the knowledge bases. Both ETS and AQUINAS generate operational knowledge bases for a number of expert system shells (e.g. KEE, OPS5, etc) from the common internal representation (Boose, 1985). In fact, more than 500 knowledge based system prototypes were generated by ETS during three years of its use at the Boeing Company; a typical prototype was constructed in less than 2 hours (Kitto & Boose, 1987). Other tools based on the repertory grid interviewing techniques include PLANET, KITTEN (Shaw & Gaines, 1987) and NEXTRA (Rappaport & Gaines, 1988). NEXTRA is a knowledge acquisition toolbox based on an extendible set of techniques developed for the knowledge support system KSS0 (Shaw & Gaines, 1988). The NEXTRA technology encompasses elicitation tools, visual analysis and display tools, group comparison tools, inductive tools, and knowledge base generative tools. NEXTRA creates a knowledge base for use by the performance system NEXPERT. The KSS0 system, which subsumes the NEXTRA knowledge acquisition toolbox, supports a multi-expert grid-based elicitation recognising consensus, correspondence, conflicts and contrasts (Shaw & Gaines, 1988). KRITON (Diederich *et al.*, 1987) combines repertory grid interviewing and protocol analysis to

build knowledge bases.
All such repertory grid based tools have implicit models of heuristic classification problem-solving and thus operate at the knowledge-level (Newell, 1982). In this way, they are similar to MOLE or MORE in that they implement model-based conceptual models. However, in these repertory grid-based tools, these models are implicit in contrast to the more transparent models in tools like ROGET and MOLE.

General tools

This refers to toolkits that attempt to provide a comprehensive range of tools to help the knowledge engineer bridge the gap between the expert's knowledge and some final runnable system. KEATS, developed by British Telecom and the Open University (Motta *et al.*, 1988) excellently falls into this category. It aims to provide complete life-cycle support for knowledge engineers beginning with knowledge elicitation, through problem conceptualisation, knowledge encoding and debugging. It provides useful enhancements to other modern shells, toolkits and environments for knowledge engineering (e.g. KEE, ART, etc.):

- Semi-automated transcript analysis facilities.
- Sketchpad on which the KE may draw a free hand representation of a domain, from which code is automatically generated. It also has a hybrid representation formalism that includes a frame-based language and rule interpreter.

KEATS is an impressive system which comprises several tools including ACQUIST (a hypertext-based domain conceptualisation tool). KEATS's theory is mentioned only in passing: it may be typical of AI systems where the theory and the system are one and the same (Anjewierden, 1987). Another example is SHELLEY (Anjewierden et al., 1990) which is the toolkit that supports the KADS methodology.

Machine Learning tools

Learning is without doubt one means of knowledge acquisition (Chandrasekaran, 1989). So learning programs are being developed/used by knowledge engineers to produce operational representations of knowledge from expert performance. Machine learning promises a way to get computers to behave as we wish without programming them (or, equivalently, without having to build and debug complex knowledge bases). It can facilitate knowledge acquisition by transforming knowledge that is in a readily available form, such as examples, into a more useful form, such as diagnostic rules. If experts could simply show a machine what to do, rather than program it, then the knowledge acquisition problem would be solved - at least, this is the theory.

ID3 (Quinlan, 1983) learns similarities from training sets by optimising global parameters. AQ11 (Michalski, 1983) induces rules from sets of positive and negative training examples. LEAP (Michel *et al.*, 1985) uses apprenticeship learning to learn steps in VLSI design by watching experts solve problems; it is, however, not a stand-alone system. On some level, other tools already mentioned could be viewed as learning tools (e.g. ETS/AQUINAS could be seen as learning via induction; admittedly, there is a difference in the way the latter functions compared to say AQ11). A more recent development of a machine-based knowledge acquisition system is BLIP (Morik, 1987). However, for any application of machine learning to knowledge acquisition, the knowledge engineer has to set up the learning problem for the induction algorithm, design a representation for examples and generalisations, define all the terms in the language(s), encode a set of training examples in the representation and provide background knowledge (Michalski, 1983) that guides the induction algorithm to choose the right generalisations from the potentially infinite set of possibilities. This can require significant knowledge engineering effort, especially if the task is more complicated than simple classification.

Discussion on Tools

A range of tools for knowledge acquisition have just been explored. As may have been recognised, quite a number of them are similar in terms of their functionality. It may be necessary to start off with a word of warning about tools in general. Most tools are very seductive, but users of them must realise that they carry along with them assumptions which are not normally made explicit. Users must be aware of *what* they are using and *when* to use it in addition to the usual *how* to use these tools reported in papers describing them. For example, ETS/AQUINAS really apply to declarative domains and are not of much good to expressing causal, procedural and strategic knowledge. Hence some pre-analysis needs to have been done to suggest the use of ETS/AQUINAS. OPAL requires the domain to have been analysed to realise that a skeletal-plan refinement problem-solving method would be applicable. Indeed, most of these tools implicitly implement knowledge-level problem solving methods.

The key to the success of many of these tools is that they use knowledge about the task (as in OPAL) and/or a particular problem-solving method (as in ROGET, MOLE, SALT, etc.) supported by the architecture of the expert system (McDermott, 1986). This has two primary benefits for knowledge acquisition: first, the interface between the human and machine can be strutured to acquire knowledge relevant to the task and only elicit knowledge in the form useful to the method (e.g. OPAL); second, since it is designed to work with a specific problem-solving method, and therefore knows how the method applies domain knowledge, the tool can analyse the user's input for completeness and consistency. In effect, the knowledge acquisition tool reduces the task of building a knowledge base to the task of instantiating pre-designed and well understood representations for a specific domain. Thus, these tools all require knowledge-level analysis to be performed in advance of their implementation. Furthermore, users must know *a priori* that the particular method of problem-solving built into the tool can be applied to the task they wish the target KBS to address.

The moral of all this is that without judicious use, these tools may compound the knowledge acquisition problem because the wrong tool, problem-solving method, task choice or elicitation technique could be used on the wrong domain with disastrous results. Tools are powerful in what they are good at and designed for and their success, if rightly used, can not be overstated. However, they do not still tackle the problems identified in Section 2 even though, admittedly, they could contribute to its solution. For example, they do not address the problem of harmonising the expert's and the knowledge engineer's mental models. Also, there is little guidance available to the domain expert or knowledge engineer to help with: classifying the application task and identifying a problem-solving method; given the application task characteristics, selecting knowledge acquisition tools and strategies to be applied in creating and refining the knowledge base. Kitto & Boose (1989) have gone some way in addressing the latter problem with their on-going work on the AQUINAS Dialog Manager. Admittedly also, toolkits like SHELLEY and KEATS provide the knowledge engineer with tools to make sense of a domain.

Domain Characterisation: Towards easing the Knowledge Acquisition Problem

We contend that the problems associated with knowledge acquisition will not be answered by developing more knowledge modelling languages/-methodologies or by developing more tools; even the current trend of grouping some of these tools into toolkits and workbenches in an unprincipled manner, is unlikely to do the trick. It will only be overcome by deliberately attempting to tackle the root causes identified in Section 2. We are currently involved in a project called MEKAS, an acronym for <u>ME</u>thodology for <u>K</u>nowledge <u>A</u>nalysi<u>S</u>. The need for this kind of research has emerged from a variety of industry-sponsored knowledge acquisition projects based at the University of Liverpool (Finch, 1989; Hughes, 1986; Plant; 1987; Watson *et al.*, 1989) and is further supported by the requirements of two major industrial collaborators. In our MEKAS project we seek to address the following problems:

1. Developing techniques to enable the expert(s) and the knowledge engineer(s) to harmonise their mental models as suggested by Recogzei & Plantinga (1987). (The recognition that this is even desirable is important).

2. Providing techniques of analysing a problem domain so as to reveal *what* elicitation techniques/tools to use *when* so as to address the representation mismatch problem of Section 2.

3. Develop techniques to navigate and make sense of the sheer mass of information involved in knowledge acquisition. For this, we draw some knowledge from the literature of what has been achieved thus far with methodologies such as KADS and Ontological Analysis.

In addition to these, we consider some fundamental problems with the state-of-the-art in knowledge acquisition which recent workshops have identified. Those from IJCAI-89 include:

4. Knowledge acquisition tools are often based on the intended final form of the knowledge base. This backwards approach puts "the cart before the horse". It is the nature of the *domain* that should guide the knowledge acquisition process (Paton & Nwana, 1990); the final form of the knowledge base should emerge from a characterisation of the domain.

5. An unfortunate result of current unstructured approaches to knowledge acquisition is that Knowledge Engineers often move too quickly through some of the cognitive definition and organisation work to enter the later phases of acquisition and implementation without an adequate *specification* of the domain. This is partly due to the fact that the designs and functions of many available knowledge acquisition tools were driven by implementation rather than cognitive concerns (Shaw, 1989). This creates problems for the neglected area of knowledge base maintenance which is a real issue for practical systems.

One key issue highlighted at the Banff Knowledge Acquisition Workshop (KAW-1989) was:

6. There is as yet no theory of how to acquire human knowledge. The epistemological, cognitive and conceptual foundations of knowledge acquisition leave much to be desired (Bradshaw & Woodward, 1989).

The approach to knowledge analysis we are developing attempts to address these issues which we consider to all contribute to the knowledge acquisition problem. We contend that part of the solution to these problems is revealed as we make distinctions between knowledge analysis (as we see it) and knowledge-level analysis. A key achievement of our work would be to highlight this distinction while, at the same time revealing their complementary natures. It is important here to clearly disambiguate these two. The approach to knowledge analysis which we are evolving addresses the six issues raised above. Knowledge-level analysis emphasises three aspects: the function of the model, the representation and organisation of the knowledge, and the control strategy. These constitute the problem-solving method or generic task. Analysis in this case reduces to mapping a task model to a problem-solving method. This is normally called the *design* problem (Schreiber et al., 1988). The important issue of note here is that knowledge-level analysis really concerns design of the KBS which necessitates that some domain analysis/characterisation should have earlier been done to reveal the tasks of the domain. Woodward (1990) acknowledges this fact when he notes that "they (i.e. problem-solving methods/generic tasks) are extremely rigorous and useful frameworks but are not required at the domain organisation level of analysis". Building a KBS is not yet engineering the design more fully.

Hence, knowledge analysis in our view, really concerns the characterisation of a domain (Paton & Nwana, 1990). As shown in Figure 2, it is really the phase 1 activi ty in the development of KBSs. Our contention is that the root causes of knowledge acquisition problems reside in the inadequate attention (some may go as far as say, neglect) given to this phase by researchers as testified by the literature. Many authors suggest a feasibility analysis where issues of domain boundary, experts availability, stability of procedures in domain, recognising the commitment in time and funding and consultation of users of the system (e.g. Parsaye & Chigwell (1988); Martin & Oxmann, 1988) should be investigated.

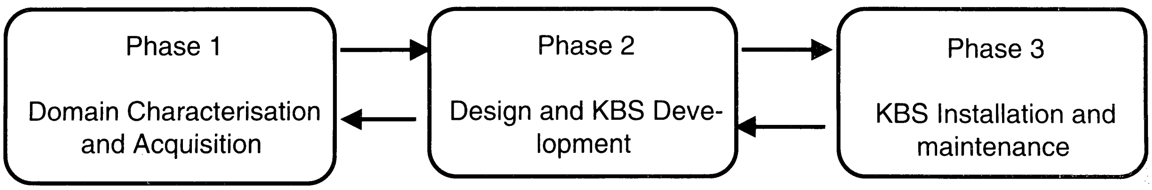

Figure 2. Phases of KBS Development

However, these are only general guidelines and do not address domain characterisation; specific techniques which address critical issues like knowledge content, organisation and structure are needed to make critical decisions before concentrating on the expensive effort of the second phase of knowledge engineering.

Domain characterisation is usually done via interviews between experts and the knowledge engineer and involves organising the knowledge (raw data) gained from human experts, the relevant literature, manuals, journals and other sources (e.g. examples, case histories) into a coherent, unambiguous structure for the domain. It is the stage which Woodward (1990) refers to as domain definition and we concur with him when he states that it is a crucial stage in knowledge engineering which "structures the domain to isolate and organise key areas of content, then presents the knowledge in a manner that feeds more specific knowledge acquisition tools". In the approach to characterisation which we are developing, we are seeking to provide a comprehensive description of the static and dynamic nature of a domain. Furthermore, it also makes better software engineering sense to produce a detailed characterisation at an early stage so that problems that may arise later on, concerned with maintenance or extensibility, are minimised (see e.g., Bench-Capon, 1990).

Current Tools/Methodologies and the Necessity for Domain Characterisation

We further concur with Woodward (1990) when he notes that all knowledge acquisition procedures assume domain characterisation in a manner which is conductive to starting KBS activity. It also assumes that decisions on boundaries of the domain have been identified; indeed, it is normally assumed in these papers that the domain organisation has been completed (Waterman, 1986).

Furthermore, powerful state-of-the-art tools like AQUINAS (Boose & Bradshaw, 1987) and KSS0 (Gaines, 1987) require primitive elements and constructs. They also assume a certain level of specificity and granularity which presumes *a priori* domain characterisation or content analysis. Using any particular elicitation technique/tool, as has been earlier noted, would normally require some *a priori* domain analysis or characterisation. It has also been established in Section 4.6 that other tools, e.g. SALT, MOLE, ROGET, etc., also require the users to know *a priori* that the particular problem-solving method built into the tool can be applied to the task they wish the target KBS to address. For example, SALT requires the expert to slot in a variety of types of information in the form of events, objects, names, formulae to suit its propose-and-revise problem-solving method. Also, NEXTRA requires specifications of entities, attributes and constructs within a domain. Clearly, these tools require some prior content organisation/analysis of the domain before using them. Section 3.1 also noted that current methodologies remain fuzzy when it comes to conceptual modelling. For example in KADS, there are no clear guidelines as to the choice of some interpretation model. Indeed, KADS's main strengths are in its provision of possibly the most comprehensive set of interpretation models. These models really constitute design primitives; hence, they address more of phase 2 proces of figure 2 (i.e. design of the artefact) than of phase 1 since some *a priori* analysis needs to be done to make use of them. This same argument also applies to other generic task based approaches (e.g. of Chandrasekaran (1985) or Clancey (1986)). The importance of domain characterisation can not therefore be overstated.

Re-examining the Metaphors used in Knowledge Acquisition

In this paper we have deliberately avoided using the idea of a knowledge acquisition bottleneck. Some researchers have started questioning the use of this metaphor, suggesting it is wrong and misleading (Clancey, 1989). It suggests the problem is that of squeezing a large amount of already-formed concepts and relations through a narrow communication channel. Metaphors play a key role in the language and ontologies we use (Paton et al, 1990b). So far, the bottleneck metaphor has encouraged the domination of the 'transfer' (expertise transfer) and 'performance' (the more knowledge, the better the KBS) views to knowledge acquisition. It is evident that the metaphor chosen shapes the research approach of many researchers. Hence, the shortcomings which have resulted in the persistence of the knowledge acquisition problem must necessitate a re-examination of some of the central metaphors.

The suggestion of the transfer view that knowledge is in already formed concepts or relations in an expert's head waiting to be transferred, surely does not stand up to close scrutiny (Clancey, 1989). Likewise, is the performance view. The 'mining' view is not as popular as the afore-mentioned two. It suggests 'mining' into an expert's head to 'dig' out the relevant knowledge; this is more of a problem for psychology. Clancey (1989) also notes that even the knowledge-level modelling view is problematic. Hence, the conclusion is that the bottleneck metaphor, which is still the most popular, should be replaced as it is misleading and the research agenda should be amended accordingly.

Conclusions

In this paper, we have examined certain problems associated with knowledge acquisition. We reviewed how successfully the numerous methodologies and tools designed have addressed these problems. We then argued the case for a preliminary knowledge analysis or domain characterisation phase of KBS development. This facilitates subsequent design, development and maintenance phases. It should be noted that the details of our domain characterisation approach are not expounded upon in this paper; they are reported elsewhere (e.g. see Paton & Nwana (1990). The paper concludes with a suggestion of a re-examination of some of the central metaphore acquisition.

Acknowledgements

The authors sincerely acknowledge the comments of Drs Marc Linster (GMD, St. Augustin) and Anjo Anjewierden (University of Amsterdam) on an earlier draft of this paper. Discussions with Enrico Motta (Open University, UK) have also proved very helpful. This research is sponsored by Shell Research UK (Thornton) and Unilever Research (Port Sunlight).

References

Alexander, J. H., Freiling, M. J., Shulman, S. J., Staley, J. L., Rehfus, S. & Messick, S. L. (1986), "Knowledge level Engineering: Ontological Analysis", *Proceedings of AAA 5*, Philadelphia, 963-968.

Alexander, J. H., Freiling, M. J., Shulman, S. J., Rehfus, S. & Messick, S. L. (1987), "Ontological analysis: an on going experiment", *International Journal of Man-Machine Studies 26*, 473-485.

Anjewierden, A. (1987), "Knowledge Acquisition Tools", *AICOM 0(1)*, 29-38.

Bench-Capon, T. J. M. (1990), *Legal Applications of Knowledge Base Systems*, London: Academic Press.

Bennett, J. S. (1985), "A knowledge-based reasoning system for acquiring the conceptual structure of a diagnostic expert system", *Journal of Automated Reasoning 1*, 49-74.

Boose, J. H. (1985), "A knowledge acquisition program for experts systems based on personal construct psychology", *International Journal of Man-Machine Studies 23*, 495-525,

Boose, J.H. (1989), "A survey of knowledge acquisition techniques and tools", *Knowledge Acquisition 1*, 3-37.

Boose, J. H. & Bradshaw, J. M. (1987), "Expertise transfer and complex problems: using AQUINAS as a knowledge acquisition workbench for expert systems", *International Journal of Man-Machine Studies 26*, 3-28.

Breuker, J. A., Wielinga, B. J. & Hayward, S. (1985), "Structuring of Knowledge Based Systems Development", *ESPRIT Project Report*, University of Amsterdam.

Breuker, J. A., Wielinga, B. J., van Someren, M., de Hoog, R., Schreiber, G., de Greef, P., Bredeweg, B., B., Wielemaker, J., Billault, J. P., Davoodi, M. & Hayward, S. (1987), Model-Driven Knowledge Acquisition Interpretation Models, Deliverable Task A1, ESPRIT Project 1098, Commission of the European Community.

Buchanan, B. G., Barstow, D., Bechtal, R., Bennett, J., Clancey, W. J., Kulukowski, C., Mitchell, T. & Waterman, D. A. (1983), "Constructing an Expert System", In Hayes-Roth, F., Waterman, D.

A. & Lenat, D. B (eds), *Building Expert Systems*, Reading, Mass.: Addison- Wesley.

Burton, A. M., Shadbolt, N. R., Hedgecock, A. P., & Rugg, G. (1987) "A formal evaluation of elicitation techniques for expert systems: domain 1", In Moralee, D. S. (ed), *Research and Developments in Expert Systems*, Cambridge University Press.

Chandrasekaran, B. (1985), "Generic tasks in Knowledge Based Reasoning: Characterizing and designing Expert Systems at the 'Right' level of Abstraction", *Proceedings of the IEEE 2nd Conference on Artificial Intelligence Applications*, 294-300.

Chandrasekaran, B. (1989), "Task-Structures, Knowledge Acquisition and Learning", *Machine Learning 4*, 339-345.

Clancey, W. J. (1985), "Review of J. F. Sowa's Conceptual Structures", *Artificial Intelligence 27*, 113-128.

Clancey, W. J. (1986), "Heuristic Classification", In Kowalik, J. S. (ed), *Knowledge based Problem Solving*, Englewood Cliffs, N. J.: Prentice Hall, 1-67.

Clancey, W. J (1989), "The Knowledge Level Reinterpreted: Modeling How Systems Interact", *Machine Learning 4*, 285-291.

Davis, R. (1979), "Interactive transfer of expertise: acquisition of new inference rules", *Artificial Intelligence 12*, 121-157.

Davis, R. (1976), Applications of Meta Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases, Ph.D. thesis, Technical Report STAN-CS-76-564, Stanford University, Stanford, CA.

Davis, R. & Lenat, D. B. (1982), *Knowledge-Based Systems in Artificial Intelligence*, New York: McGraw-Hill.

Diederich, J., Ruhmann, I. & May, M. (1987), "KRITON: a knowledge acquisition tool for expert systems", *International Journal of Man-Machine Studies* 26, 29-40.

Downs, E., Clare, P. & Coe, I. (1988), *Structured Systems Analysis and Design Method: Application and Context*, Hemel Hempstead, Herts: Prentice Hall.

Eshelman, L., Ehret, D., McDermott, J. & Tan, M. (1987), "MOLE: a tenacious knowledge acquisition tool", *International Journal of Man-Machine Studies 26*, 41-54.

Feigenbaum, E. A. (1977), "The art of artificial intelligence: themes and case studies of knowledge engineering, *Proceedings of IJCAI-77*, Cambridge MA: Morgan Kauffmann.

Finch, I. (1989), "Explanation systems in engineering domains", *Proceedings of the International conference on systems in engineering*, Wuhan, China.

Friedland, P. E. & Iwasaki, Y. (1985), "The concept and implementation of skeletal plans", *Journal of Automated Reasoning 1*, 161-208.

Gaines, B. R. (1987), "An overview of knowledge acquisition and transfer", *International Journal of Man-Machine Studies 26*, 183-202.

Grover, M. D. (1983), "A Pragmatic Knowledge Acquisition Methodology", *Proceedings of IJCAI 8*, 436-438.

Guida, G. & Tasso, C. (1989), "Building expert systems: From Life Cycle to development methodology", In Guida, G. & Tasso, C. (eds), *Topics in Expert Systems Design, Methodologies and Tools*, Amsterdam: North Holland, 3-24.

Hayward, S. A., Wielinga, B. J. & Breuker, J. A. (1987), "Structured analysis of Knowledge", *International Journal of Man-Machine Studies 26(4)*, 453-472.

Hughes, S. (1986), "Question classification in rule-based systems", In Bramer, M. A. (ed), *Research and Development in Expert Systems III*, Cambridge: University Press, 123-131.

Johnson, N. E. (1989), "Mediating representations in knowledge elicitation", In Diaper, D. (ed), *Knowledge Elicitation: Principles, Techniques and Applications*, Chichester: Ellis Horwood, 179-194.

Kahn, G., Nowland, S. & McDermott, J. (1985), "MORE: An intelligent knowledge acquisition tool", In *Proceedings of the International Joint Conference on Artificial Intelligence*, Los Angeles, CA, 581-584.

Kelly, G. A. (1955), *The psychology of personal constructs*, New York: Norton.

Kitto, C. M. & Boose, J. H. (1987), "Heuristics for expertise transfer: an implementation of a dialog manager for knowledge acquisition", *International Journal of Man-Machine Studies 26*, 183-202.

Kitto, C. M. & Boose, J. H. (1989), "Selecting knowledge acquisition tools and strategies based on application characteristics", *International Journal of Man-Machine Studies 31*, 149-160.

Marcus, S. (1988), "SALT: A knowledge acquisition tool for propose-and-refine systems", In Marcus, S. (ed), *Automating knowledge acquisition for expert systems*, Boston: Kluwer Academic Publishers.

Martin, , J. & Oxmann, S. (1988), *Building Expert Systems: A Tutorial,* Englewood Cliffs, NJ: Prentice Hall.

McDermott, J. (1986), "Making expert systems explicit", *Proceedings of IFIP World Congress*, Dublin, Ireland.

Michalski, R. S. (1983), "Theory and methodology of inductive learning", in Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. (eds), *Machine learning: An artificial intelligence approach*, New York: Tioga.

Mitchell, T. M., Mahadevan, S. & Steinberg, L. I. (1985), "LEAP: a learning apprentice for VLSI design", *Proceedings of the International Conference on Artificial Intelligence.* Los Angeles.

Morik, K. (1987), "Acquiring domain models", *International Journal of Man-Machine Studies 26*, 93-104.

Motta, E., Eisenstadt, M., Pitman, K. & West, M. (1988), "Knowledge acquisition in KEATS: The knowledge engineer's assistant", *Expert Systems 5(2).*

Motta, E., Rajan, T. & Eisenstadt, M. (1990), "Knowledge Acquisition as a process of model refinement", *Knowledge Acquisition 2*, 21-49.

Musen, M. A. (1989), "Conceptual models of interactive knowledge acquisition tools", *Knowledge Acquisition 1*, 73-88.

Musen, M. A., Fagan, L. M. Combs, D. M. & Shortliffe, E. H. (1987), "Use of a domain model to drive an interactive knowledge-editing tool", *International Journal of Man-Machine Studies 26*, 105-121.

Newell, A. (1982), "The knowledge level", *Artificial Intelligence 18*, 87-127.

Parsaye, K. & Chignell, M. (1988), *Expert Systems for Experts*, Toronto: Wiley.

Paton, R. C. & Nwana, H. S. (1990), "Domain Characterisation through Knowledge Analysis", *Proceedings of AAAI-90 Workshop on Knowledge Acquisition: Practical Tools and Techniques*, Boston, Mass., July.

Paton, R. C., Nwana, H. S., Shave, MJR , Bench-Capon T.J.M. & Hughes, S, (1990a), "Foundations of a Structured Approach to Characterising Domain Knowledge", Paper presented at the Eighth Workshop of the European Society for the Study of Cognitive Systems (ESSCS), Oxford, September, 1990.

Paton, R. C., Nwana, H. S., Shave, MJR , Bench-Capon T.J.M. & Hughes, S, (1990b), " Transfer of Natural Metaphors to Parallel Problem Solving Applications", *Proceedings of the First International Workshop on Parallel Problem Solving from Nature*, Dortmund, October, 1990.

Plant, R. (1987), A Methodology for Knowledge Acquisition in the Development of Expert Systems, Unpublished Ph.D. thesis, University of Liverpool.

O'Neill, M. & Morris, A. (1989), "Expert systems in the United Kingdom: an evaluation of development methodologies", *Expert Systems 6(2)*, 90-98.

Quinlan, J. R. (1983), "Learning efficient classification procedures and their applications to chess and games", In Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. (eds), *Machine learning: An artificial intelligence approach*, New York: Tioga.

Rappaport, A. T. & Gaines, B. R. (1988), "Integrating knowledge acquisition and performance systems", *Proceedings of AAAI-88 Workshop on Integration of Knowledge Acquisition and Performance Systems*, St Paul, Minnesota.

Schreiber, G., Breuker, J., Bredeweg, B. & Wielinga, B. (1988), "Modelling in KBS Development", *Proceedings of third European Workshop on Knowledge Acquisition for Knowledge Based Systems*, EKAW '88 (June), Bonn.

Shaw, M. L. G. & Gaines, B. R. (1987), "Techniques for knowledge acquisition and transfer", *Interna-*

tional Journal of Man-Machine Studies 27, 251-280.

Shaw, M. L. G. & Gaines, B. R. (1988), "A methodology for recognising consensus, correspondence, conflict and contest in a knowledge acquisition system", *Proceedings of workshop on knowledge acquisition for knowledge based systems*, November, Banff, Canada.

Shaw, M. L. G. (1989), "A Grid-Based Tool for Knowledge Acquisition", *Proceedings of IJCAI-1989 Workshop on Knowledge Acquisition*, Detroit, Michigan, 19-22.

Shortliffe, E. H., Scott, A. C., Bischoff, M. B., van Melle, W. & Jacobs, C. D. (1981), "ONCOCIN: an expert system for oncology protocol management", In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Vancouver B. C., Canada, 876-881.

Steels, L. (1984), "Second Generation Expert Systems", Paper presented at the International Symposium on Fifth Generation and Supercomputers, Rotterdam, The Netherlands, December.

Van Melle, W. (1979), "A domain-independent production rule system for consultation programs", *Proceedings of the International Joint Conference on Artificial Intelligence.*

Waterman, D. (1986), *A Guide to Expert Systems*, Reading, MA: Addison Wesley.

Watson, I. D., Shave, M. J. & Moralee, D. S. (1989), "A knowledge analysis methodology using an intermediate knowledge representation based on conceptual graphs", *Proceedings of the 9th Int. Workshop on Expert Systems and their Applications 1*, Avignon, France, 183-198.

Wetter, T. H. (1990), "First order logic foundations of the KADS conceptual model", In Wielinga, B., Boose, J., Gaines, B., Schreiber, G. & Van Someren, M. (eds), *Current Trends Knowledge Acquisition*, Amsterdam: IOS, 356-375.

Welbank, M. (1990), "An overview of knowledge acquisition methods", *Interacting with Computers 2(1)*, 83-91.

Wielinga, B. J., Bredeweg, B. & Breuker, J. A. (1988), "Knowledge Acquisition for Expert Systems", *Proceedings of ACAI-88.*

Woodward, B. (1989), "General Structures as Domain Models to Guide Knowledge Acquisition, *Proceedings of IJCAI-1989 Workshop on Knowledge Acquisition*, Detroit, Michigan, 10-14.

Woodward, B. (1990), "Knowledge acquisition at the front end: defining the domain", *Knowledge Acquisition 2*, 73-94.

This paper submitted to the Avignon'91 Conference on Expert Systems and their Applications was awarded with the 1991 Digital Equipment Corporation European AI Prize. The paper was selected by DIGITAL from a short list provided by the conference program committee.

Republication of this prize was granted by EC2, the publisher and organizer of the Avignon Conference Series. Proceedings of the Avignon Conferences can be ordered from:

EC2, 269 Rue de la Garenne,
92024 Nanterre Cedex, France.
Tel: +33 1 47 80 70 00, fax: ..80 66 29.