N. Aussenac G. Boy B. Gaines M. Linster
J.-G. Ganascia Y. Kodratoff (Eds.)

Knowledge Acquisition for Knowledge-Based Systems

7th European Workshop, EKAW '93
Toulouse and Caylus, France
September 6-10, 1993
Proceedings

Caylus
Toulouse
93
EKAW93, 6-10 Sept.93

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editor

Jörg Siekmann
University of Saarland
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3,
D-66123 Saarbrücken, Germany

Volume Editors

Nathalie Aussenac
CNRS-IRIT, University Paul Sabatier
118 route de Narbonne, F-31062 Toulouse Cedex, France

Guy Boy
European Institute of Cognitive Sciences and Engineering (EURISCO)
Prologue I, B.P.27-25, F-31312 Labège Cedex, France

Brian Gaines
Knowledge Science Institute, University of Calgary
2500 University Drive N.W., Calgary, Alberta, T2N 1N4 Canada

Jean-Gabriel Ganascia
LAFORIA, University Paris VI, Tour 46-00 2° étage
4, place Jussieu, B.P. 169, F-75525 Paris Cedex 05, France

Yves Kodratoff
University Paris Sud
LRI - Bât 490, F-91405 Orsay Cedex, France

Marc Linster
DEC LM02-K11
111 Locke Drive, Marlborough, MA, 01752 USA

CR Subject Classification (1991): I.2.4-6, I.2.8

ISBN 3-540-57253-8 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-57253-8 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Typesetting: Camera ready by author
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

As the term is most generally used, *knowledge acquisition* (KA) refers to the interdisciplinary study of problem solving models as well as life cycle and methodologies for knowledge-based systems. Knowledge acquisition is now recognized as an important research field that includes topics such as: elicitation; apprenticeship and learning systems; issues in cognition and expertise; knowledge acquisition from various media; context-dependent, dynamic knowledge; ontologies. This workshop focused on methodological guidelines for advanced system design.

Knowledge acquisition remains a crucial problem in artificial intelligence as well as in computer science and engineering in general. Each time a software system has to be developed, experience shows that the first step is always to state the problem that we want to solve! It seems that this common sense statement is not always a rule. Problem statement involves task analysis and end-user requirements definition. Knowledge acquisition enters into play when human know-how and heuristic knowledge need to be considered. This human factors viewpoint is becoming an issue in the knowledge acquisition community.

The Seventh European Knowledge Acquisition Workshop (EKAW '93) gathered a large variety of papers in this areas. Brian Gaines' introductory paper provides a very interesting scope of the previous Knowledge Acquisition Workshops and the emerging researches in the field. EKAW usually combines an open day meeting and a four-day closed workshop with a limited number of participants. In 1993, EKAW was held in Toulouse and Caylus, France. This volume reports the best papers presented during the workshop. The variety of these papers shows the diversity and maturity of the field.

Knowledge acquisition is often acknowledged as a modelling process. Brian Gaines explains how knowledge acquisition research came to this conclusion. He develops the current trends in this direction. As a complementary point of view, Guy Boy suggests a new direction of investigation for knowledge acquisition: the design of dynamic systems. His paper proposes a defintion of such systems and stresses their specificities and related knowledge acquisition issues.

Problem solving models

Characterizing knowledge acquisition as modelling defines a number of concepts and identified difficulties. Among them, problem solving models are essential. Building adequate models from specific expertise can be improved by the definition of guidelines and steps. Two papers are concerned with this objective. In *Steps in Constructing Problem Solving Methods* Akkermans[1] proposes a rational top-down

[1] Contributions are indicated only by first author's name for the sake of readability .

support for problem-solving methods construction, including conceptual refinement and operationalization. In *Modelling Artificial Legal Reasoning*, Breuker suggests a way of modelling legal reasoning that can be considered as an assesment task. He presents assesment models of problem-solving as well as an architecure for legal reasoning systems.

Another related research field is interested in the definition of support tools for knowledge modelling. The three following papers develop such works. In *A Machine Learning Tool Designed for a Model-Based Knowledge Acquisition Approach*, Thomas presents The ENIGME system a Machine Learning system that learns operative domain knowledge by exploiting a model of expertise as defined in the KADS methodology. *Systematic Building of Conceptual Classification Systems with C-KAT*, by Zacklad: C-KAT is an acquisition support method and tool dedicated to the design of a feature-oriented classification system. It uses a specialised problem-solving model: classification by structural shift. *Making Role-Limiting Shells More Flexible*, by Poek: Role-limiting methods shells are acknowledged as hardly wired. The authors analyse and decompose them into smaller mechanisms in order to enable new configurations of role-limiting methods and shells. This flexibility increases the applicability of methods and also reduces the cost of their development.

Several papers compare existing modelling approaches and environments. Such comparisons are the starting-point to better specify and define guidelines or modelling structures that should facilitate knowledge acquisition and knowledge-based system design. *Heuristic Control Knowledge*: From the study of control roles in problem solving methods in KADS and COMMET aproaches, Causse proposes an additional level of description for these models: the heuristic control level, where heuristic control knowledge is described. In *Generic Tasks in KEW*, Allemang relates an experiment in which generic tasks are cast in the KEW framework and formal language for model description. Its results not only prove the possibility of connecting the generic tasks and KADS-KEW approaches but it also leads to improvements in both of them. Linster's paper *A Review of Sisyphus 91 & 92: Models of Problem-Solving Knowledge* synthesizes the various contributions to the Sisyphus project in 1991 and 1992. A three-dimensional framework is presented to situate and to compare the approaches, highlighting the building blocks used to model and later implement a knowledge-based system.

Life cycle and methodologies

The second part of this volume gathers papers concerned with knowledge acquisition life-cycle and methodologies. This central part of knowledge acquisition research covers a wide set of dimensions: the specification of a refinement cycle during which knowledge is increasingly modelled, the definition of methodologies and workbenches as well as the study of dedicated elicitation techniques to be integrated as specific tools in these methodological frameworks.

Three papers propose to consider knowledge acquisition as an incremental process. They present methods and tools to support such a cycle. *Model Construction in MIKE*

(Model Based and Incremental Knowledge Engineering): The key dimension studied by Neubert in order to facilitate the incremental design of a knowledge model is knowledge representation. As an answer, the author promotes the combination of informal and semi-formal representations within an hypermedia environement, MIKE. *EXPECT: Intelligent Support for Knowledge Base Refinement*: As a response to the need of making knowledge acquisition tools easier to use for domain experts, Paris proposes to integrate explanations and new communication means in such systems. *CERISE: A Cyclic Approach for Knowledge Acquisition*, by Vicat: The CERISE workbench promotes a cyclic knowledge acquisition, firstly by refining a KADS model and secondly by validating and improving this model once it is made operational.

The following three papers provide different views on what a knowledge acquisition methodology should be, refering to psychological foundations, questionning knowledge analysis and modelling, or addressing the problem for specific kinds of knowledge. In *Personal Construct Psychology Foundations for Knowledge Acquisition and Respresentation*, Shaw gives an overview of personal construct psychology and its expression as an intensional logic describing the cognitive processes of anticipatory agents. These results are presented as a theory for knowledge acquisition and representation, as psychology offers the advantage of promoting a constructivist view when modelling human knowledge. In *Knowledge Acquisition Without Analysis*, Compton differentiates several kinds of KA methods. Some methods support knowledge analysis, based on a classification of ways of solving problems and providing adequate tools. Other methods focus on the addition of validated knowledge as long as mistakes are discovered by a system. Ripple down rules are presented as an illustration of this second kind of approach, which avoids analysing knowledge. In *Acquisition and Modelling of Uncertain, Incomplete and Time-Varying Knowledge*, Mengshoel proposes a methodology adapted to the acquisition of imperfect and temporal knowledge. A study of existing methodologies proves that this problem is not actually considered. Several propositions to extend them are presented as a solution.

The definition of workbenches is also a means of providing support for knowledge acquisition. Steps in using the workbench are often defined by a related methodology. The three following papers focus on particular aspects of different workbenches: the combination of tools, the status of the end-user and the design of a knowledge-based system as the result of using a workbench. In *The Emerging VITAL Workbench*, Domingue discusses the general framework of the VITAL workbench, focusing on the user interface and the control integration. The author also describes the tools supporting the tool management, the knowledge-level modelling as well as the model implementation. *Multis II: Enabling End-Users to Design Problem-Solving Engines via Two-Level Task Ontologies*, by Tijerino: The Multis II environment is an acquisition system that interacts with domain experts that want to make a model of their knowledge and generate a customized knowledge-based system. In *The Participatory Design of a Computer Assisted Knowledge Engineering Methodology and Tool: The ALADIN+ Project*, Muzard presents ALADIN+, a computer assisted knowledge engineering method and tool. It promotes participatory design in accordance with a cybernetic approach of the organisation and of the design process.

New elicitation techniques still need to be defined in order to acquire specific kinds of knowledge such as graphical representations, gradual knowledge, knowledge in texts. The last three papers in this volume propose answers to such needs. *Knowledge Acquisition With Visual Functional Programming*, by Addis: The CLARITY environment combines two approaches for knowledge acquisition: visual functional programming based on a functional database language and a graphical representation. *Acquisition of Gradual Knowledge*, by Dieng: Topoï are gradual inference rules. This paper proposes to use them as a knowledge representation for gradual and qualitative knowledge, both at the symbol and at the knowledge level defined by Newell. In *Acquisition and Validation: From Text to Semantic Network*, Biébow considers semantic networks as a convenient knowledge representation that facilitates domain knowledge acquisition from texts and its validation. Knowledge based engineering and natural language processing also form the kernel of DASERT, a tool to support knowledge acquisition from texts.

The editors wish to extend their gratitude to the members of the International Program Committee listed below who have contributed essentially to the high quality of these proceedings. Their prompt and thorough reviews were extremely helpful to both individual authors and workshop organizers.

Toyoaki NISHIDA, Kyoto University, Kyoto (Jap)
Bruce PORTER, Univ. Of Texas, Austin (USA)
Ross QUINLAN, Sydney University, Sidney (Australia)
Alain RAPPAPORT, Neuron Data (USA)
Thomas ROTHENFLUH, Zurich (CH)
Franck SCHMALHOFER, DFKI, Kaiserslautern (G)
Guus SCHREIBER, Univ. of Amsterdam, Amsterdam (NL)
Nigel SHADBOLT, Univ. of Nottingham (GB)
Mildred SHAW, Univ. of Calgary (CA)
Ingeborg SOLVBERG, Trondheim (N)
Maarten Van SOMEREN, Univ. of Amsterdam (NL)
Hirokazu TAKI, Inst. for New Gen. Computer Tech. (Jap)
Masanobu WATANABE, NEC Corporation
Bob WIELINGA, Univ. of Amsterdam (NL)
Thomas WETTER, IBM, Heidelberg (G)
Brian WOODWARD, Univ. of Calgary (CA)

Many institutions and organizations have contributed to the success of EKAW93. Among them, the editors wish to thank AFCET, AFIA, ARC, and ECCAI as well as the Conseil Régional Midi-Pyrénées, the Conseil Général Haute-Garonne, the Conseil Général Tarn-Garonne and CNRS.

We would like to thank Léopold Viguier, Conseiller Général du Tarn-et-Garonne, for his continuous help in the organization of the closed workshop in Caylus.

Toulouse
July 1993

Nathalie Aussenac and Guy Boy
on behalf of the editors

Table of Contents

1 Introductory Papers

2 Problem Solving Models

2.1 Building Steps

2.2 Support Tools

2.3 Comparison of Approaches

3 Life Cycle and Methodologies

3.1 Refinement

3.2. Methodologies

3.3 Workbenches

3.4 Elicitation Techniques

Modeling and Extending Expertise

Brian R Gaines

Knowledge Science Institute, University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca

Abstract. This paper surveys the state-of-the-art in knowledge acquisition for knowledge-based systems. It gives an overview of three major areas of advance in recent years: in conceptual and theoretical terms, the characterization of knowledge acquisition as a process of *modeling expertise* with a view to emulating and extending it; in methodological terms, the provision of detailed *formal modeling methodologies* supporting such processes; and, in technological terms, the development of *computer-based tools* for knowledge acquisition supporting such modeling methodologies. The paper also presents the state-of-the-art in the context of its relation to other fields of activity such as developments in software engineering, system-theoretic aspects of modeling in general, and the variety of technologies that have been applied in knowledge acquisition such as those of hypermedia and machine learning.

1 Introduction

Knowledge acquisition emerged as a distinct area of research and development in the early 1980s as a response to the need to provide scientific and engineering methodologies for the construction of expert and knowledge-based systems. This workshop is the seventeenth in a series of international workshops on knowledge acquisition for knowledge-based systems that commenced in 1986 and have continued with two regular workshops a year in North America and Europe, and a third occasional workshop in Japan or Australia. The workshops in Europe have circulated between the UK, Germany, France and Holland, and this is the second workshop in France. These workshops have supported the growth of an international community concerned with research, development and application of knowledge acquisition theories, methodologies and tools. They have also supported the exchange of scientific and engineering knowledge in this community, and its wider dissemination associated publications in book and journals associated with the workshops.

It is fitting at each workshop to review the state-of-the-art in knowledge acquisition for knowledge-based systems, both in terms of what has been achieved within the workshop community, and in terms of its relationship to wider developments in information systems. It is important to evaluate progress and recognize changing perspectives within our primary domain, and it is also important to place that progress and those perspectives within the context of related activities. The field of knowledge acquisition has been characterized by significant advances in concepts, technologies and applications since its inception, and it is difficult to keep pace with all relevant developments. However, the excitement of developments within the field should not blind one to significant advances in information systems in general that may be very relevant to knowledge acquisition and knowledge-based systems. For example, that advances in the conceptual modeling and object-oriented architectures of databases may be providing more appropriate technologies for knowledge-based systems than those of specialist expert system shells, and that concepts of requirements engineering substantially overlap those of knowledge engineering.

This paper surveys the current state-of-the-art in knowledge acquisition for knowledge-based systems, noting that there have been three major areas of advance in recent years:

1 At the conceptual and theoretical level, to view knowledge acquisition as a process of *modeling expertise* with a view to emulating and extending it.
2 At a methodological level, to provide detailed *formal modeling methodologies* supporting such processes.
3 At a technological level, to provide *computer-based tools* for knowledge acquisition supporting such modeling methodologies.

It presents the state-of-the-art in the context of its relation to other fields of activity such as developments in software engineering, system-theoretic aspects of modeling in general, and the variety of technologies that have been applied in knowledge acquisition such as those of hypermedia and machine learning.

2 Knowledge Engineering and Software Engineering

The past decade has seen an explosive growth in theories, methodologies, tools, and applications experience relating to the development of knowledge-based systems, and it has become important in recent years to attempt to consolidate and structure the products of that growth. In particular, as the scope of knowledge-based systems has grown, implemented systems have come to incorporate many standard information technologies such as data processing, data bases, simulation and graphic user interfaces. Conversely, standard information system development has come to incorporate many aspects of knowledge-based systems. This raises questions as to the differentiation of knowledge-based systems from other systems, and to the differences between knowledge engineering and software engineering. Are these distinctions any longer relevant, and if so how may they be formulated in such a way as to theoretically sound and practically useful?

2.1 Knowledge-based systems as reflective information systems

It is both possible and reasonable to argue for either position: that knowledge-based systems have become subsumed within modern information system engineering, and the knowledge engineering versus software engineering distinction is increasingly irrelevant; or that there are essential, definable and useful differences. Any distinction is a human construct invented to serve some purpose, and the more significant that purpose the greater the controversy possible about the nature and status of the distinction. The synthesis is to view knowledge engineering as an aspect of information system engineering—one that places particular emphasis on the epistemological status of information that is classified as *knowledge*, that is as "justified, true belief" [46] and "more than opinion, less than truth" [48].

These definitions provide us with a context in which the *credibility* and *derivation* of information are significant, and need to be taken into account as overt data in their own right. That is, the knowledge-based components of an information system will tend to be those in which *meta-information* and *meta-information-processing* is required—components that do not just process information but also process meta-information *about* that information and its processing. A knowledge-based system may be characterized succinctly as an essentially *reflective* information system [40].

In some applications where the final product is a totally automated system, knowledge engineering may be regarded as an aspect of software engineering, but, more generally, this is too narrow a viewpoint. The term 'software' is best applied only to the programmed control of the computational components of a system, whereas the term 'knowledge' properly encompasses certain information flows and storage in both the computational and human components of a system. Some of the most interesting systems are those in which computational processes are used to support the knowledge processes of people, to extend rather than replace their expertise. The loci of knowledge within such systems become distributed across a social network of computers and people, and much of the justification for applying the term knowledge to information in a computer is precisely because it becomes irrelevant to distinguish whether the relevant information is located in a computer or in a person.

2.2 From expertise transfer to expertise modeling

Early research on knowledge acquisition for knowledge-based systems emphasized the acquisition of the knowledge assumed to underlie human expertise in areas such as medical diagnosis, where conventional system analysis and software engineering had failed to provide computer emulation of the expertise. This led to the "expertise transfer" paradigm in which the primary function of knowledge engineering was seen to be the transfer of a human expert's knowledge to a computer system:

> "Knowledge acquisition is a bottleneck in the construction of expert systems. The knowledge engineer's job is to act as a go-between to help an expert build a system. Since the knowledge engineer has far less knowledge of the domain than the expert, however, communication problems impede the process of transferring expertise into a program. The vocabulary initially used by the expert to talk about the domain with a novice is often inadequate for problem-solving; thus the knowledge engineer and expert must work together to extend and refine it. One of the most difficult aspects of the knowledge engineer's task is helping the expert to structure the domain knowledge, to identify and formalize the domain concepts." [34]

This discussion is still valuable today in characterizing the problems of eliciting knowledge from human experts. However, as numbers of knowledge-based systems were developed of increasing scope it became apparent that human experts were only one source from which knowledge was being acquired. Knowledge engineers are pragmatic in developing systems based on every available source of relevant information. It also became apparent that the status of the 'knowledge' assumed to underlie human expertise was itself problematic:

> Knowledge can be represented, but it cannot be exhaustively inventoried by statements of belief or scripts for behaving. Knowledge is a capacity to behave adaptively within an environment; it cannot be reduced to representations of behavior or the environment." [11]

That is, the overt knowledge that we see in text, diagrams and computer data structures, and the invisible 'knowledge' that we impute to human experts to account for their skilled behaviors are two distinct entities. It is simplistic and misleading to assume that the process that leads to the emulation of human expertise in a computer program is one of transferring knowledge—'expertise transfer' is an attractive metaphor but it leaves open the questions of what is expertise and how it may be transferred. A better metaphor might be one of modeling, that the emulation involves building a model of the expertise, where a 'model' is according to Webster's dictionary:

> "a representation, generally in miniature, to show the construction or serve as a copy of something."

3 Knowledge Engineering and Modeling

The notion that what is being done in the development of an expert system is a modeling activity has become a major theme in the literature. The KADS methodology is presented as one concerned with "developing a knowledge-level model of expert reasoning" [1]. Clancey raises the question "How do expert systems differ from conventional programs?" and answers it by:

> "expert systems contain qualitative world models...Briefly put, qualitative models describe systems in the world in terms of causal, compositional, or subtypical relationships among objects and events....Knowledge engineering is not just a new kind of programming. It is a new methodology for modeling systems so that we can assemble, modify, and control them automatically. We are not so much programmers as engineers, scientists, and even philosophers." [11]

3.1 Conceptual models in knowledge engineering

Figure 1 shows the major conceptual models that may be developed in knowledge engineering, distinguished by their sources, and indicating some of the knowledge engineering processes and skills involved [30]. This figure attempts to be comprehensive, showing knowledge sources not only in association with the expert and his or her behavior, but also knowledge derived from others, the literature and through the application of laws and principles.

The complexity of the knowledge engineering process is very apparent in Figure 1. It is ironic that the expert may be able to function effectively with very little overt knowledge, whereas the knowledge engineer, reflecting on that expertise, becomes involved in synthesizing a model from a heterogeneous range of sources. The variety of skills demanded of the knowledge engineer seem unlikely to exist in all but a few people and suggest the need for a team approach, training programs and support tools.

As emphasized already, another significant aspect of the knowledge engineering task that is apparent in Figure 1 is that the final model developed may bear little resemblance to the mental models assumed to exist within the expert. There is no reason to require that such models should be independent of the knowledge engineering process, or even exist before it is commenced.

The outer boxes in Figure 1 emphasize that neither the expert nor the knowledge engineer are completely autonomous and self-contained systems. They are each situated in an organizational infrastructure that plays a major role in providing motivation, objectives and support systems for both the expertise and its modeling. The organizational infrastructure is, in its turn, situated in a wider socio-economic environment that itself plays a major role in providing motivation, objectives and support systems for the organization. Processes of expertise involve individuals termed 'experts' but they cannot be fully characterized by features of those experts alone. Human knowledge processes are socially situated, and their overt analysis involves modeling some aspects of the situation.

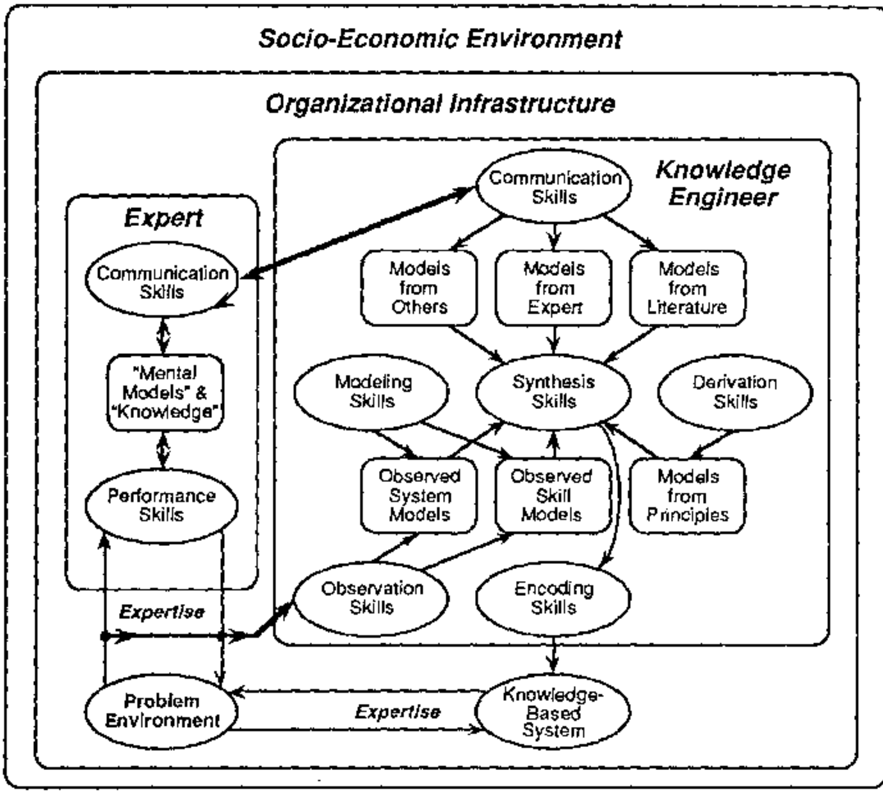

Figure 1 Knowledge sources and modeling processes in knowledge engineering

3.2 Levels of modeling

The modeling processes in Figure 1 are not all at the same level. There are fundamental differences between the observation and modeling of action, for example, and the verbal discussion of the intentions behind and logic underlying that action. It is useful to organize the modeling processes themselves within a framework that differentiates and classifies them in terms of the acquisition and analysis processes involved.

Figure 2 presents a modeling framework for knowledge acquisition methodologies, techniques and tools based on the distinctions already discussed and the incorporation of system analysis and knowledge engineering procedures [30]. In the leftmost column are the knowledge sources in terms of systems and modeling schema. In the column to the right of this are the processes giving access to these models.

In the next column are shown the knowledge acquisition procedures appropriate to each of the access processes. These generate data and knowledge bases as shown to their right, which are in one-to-one correspondence with the original systems and models in the leftmost column. In the rightmost column are shown analysis techniques that draw on these databases to generate the computational knowledge base, and also mediate between them generating one form of data or knowledge from another. These combine with synthesis techniques that integrate the results of analysis and of derivations from various knowledge sources to synthesize a computational knowledge base.

Thus the overall schema consists of five types of component:

1. *Systems and modeling schema:* the problem environment, performance skill to be emulated, expert's mental models, knowledge engineer's conceptual models, and, possibly, objective models.
2. *Access processes:* instrumentation of the target system, the expert's interaction with it, his or her introspection about the skill, communication about it, and its "precisification" in formal terms, possibly resulting in the kind of "explication" accepted as the basis for explanation and justification through objective knowledge [8].
3. *Knowledge acquisition procedures:* observation of the target system, observation of the expert's behavior, elicitation procedures, discourse procedures, formalization procedures, and implementation procedures.
4. *Data and knowledge bases:* database of system data; database of behavioral data; informal knowledge base; formal knowledge base; computational knowledge base; objective models.
5. *Analysis and synthesis procedures:* classical system identification can be used to build system models from observation data; empirical induction and case-based clustering can be used to build skill models from behavioral data; conceptual organization and linguistic analysis techniques can be used to build a formal, or structured, knowledge base from an informal, or intermediate, one; knowledge modeling techniques can be used to represent the formal knowledge base in computational form; and logical deduction from laws and principles may be used to provide some knowledge about a system and this, together with the results of data analyses from various sources needs to be integrated to form a computational knowledge base.

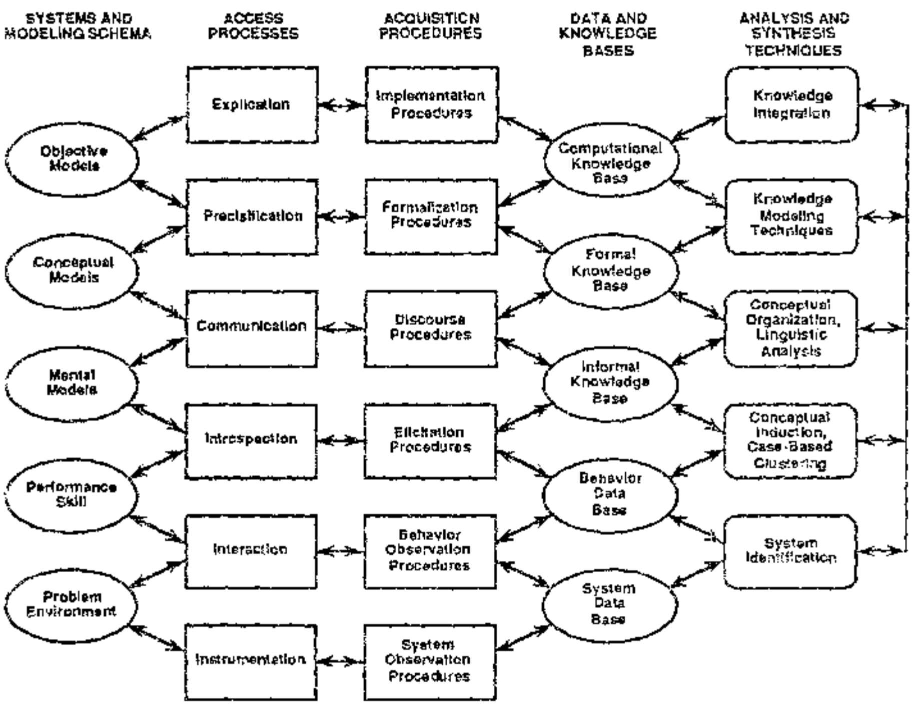

Figure 2 A hierarchical modeling framework for knowledge engineering

Figure 2 illustrates the way in which knowledge engineering as a system design methodology is sandwiched between two classical approaches to system engineering. At the bottom of the figure is the path to system design through instrumentation, data collection and system identification. At the top of the figure is the path to system design through existing objective knowledge of the physical world allowing explication of particular requirements to lead directly to implementation. The middle layers represent the enrichment of the design process when we draw on human skills as exemplars of the system to be designed. Such a process has been common informally in engineering design, and knowledge engineering may be seen as formalizing it now that computer technology makes it feasible to develop knowledge-based systems operationalizing human expertise.

4 Modeling Methodologies

There have been two major modeling methodologies developed in knowledge acquisition research: the KADS methodology [49] focusing on the derivation of the formal knowledge base in Figure 2, and its translation into a computational knowledge base; and the second the PCP methodology [28] focusing on the derivation of the informal knowledge base in Figure 2, and its translation into a formal knowledge base.

4.1 KADS: a principled approach to knowledge-based system development

The KADS methodology is the outcome of a number of ESPRIT project activities centering on the University of Amsterdam but involving researchers and practitioners from many institutions, countries and disciplines. KADS is intrinsically a modeling approach with seven types of model distinguished [49]:

4.1.1 The organizational model

An organizational model provides an analysis of the socio-organizational environment in which the knowledge-based system will have to function. It includes a description of the functions, tasks and bottlenecks in the organization. In addition it describes how the introduction of a knowledge-based system will influence the organization and the people working in it.

4.1.2 The application model

An application model defines what problem the system should solve in the organization and what the function of the system will he in this organization. In addition to the function of the knowledge-based system and the problem that it is supposed to solve, the application model specifies the external constraints that are relevant for the development of the application.

4.1.3 The task model

A task model specifies how the function of the system as specified in the application model is achieved through a number of tasks that the system will perform. Establishing this relation between function and task is not always as straightforward as it may seem. Given a goal that a system should achieve, there may be several alternative ways in which that goal can be achieved. Which alternative is appropriate in a given application depends on the characteristics of that application, on the availability of knowledge and data, and on the requirements imposed by the user or by external factors.

4.1.4 The model of cooperation

The model of cooperation contains a specification of the functionality of those sub-tasks in the task model that require a cooperative effort between the agents to whom the sub-tasks have been distributed. Some of the sub-tasks will be achieved by the system, others may be realized by the user. The result is a model of cooperative problem solving in which the user and the system together achieve a goal in a way that satisfies the various constraints posed by the task environment, the user and the state of the art of knowledge-based system technology.

4.1.5 The model of expertise

Building a model of expertise is a central activity in the process of knowledge-based system construction. It distinguishes knowledge-based system development from conventional system development. Its goal is to specify the problem solving expertise required to perform the problem-solving tasks assigned to the system. The KADS methodology focuses on expertise as the *behavior* that the system should display, and on the types of knowledge that are involved in generating such behavior, abstracting from the details of how the reasoning is actually realized in the implementation.

4.1.6 The conceptual model

Together, the model of expertise and the model of cooperation provide a specification of the behavior of the artifact to be built. The model that results from merging these two models is similar to what is called a *conceptual model* in database development. Conceptual models are abstract descriptions of the objects and operations that a system should know about, formulated in such a way that they capture the intuitions that humans have of this behavior. The language in which conceptual models are expressed is not the formal language of computational constructs and techniques, but is the language that relates real world phenomena to the cognitive framework of the observer. In this sense conceptual models are subjective, they are relative to the cognitive vocabulary and framework of the human observer.

4.1.7 The design model

The description of the computational and representational techniques that the artifact should use to realize the specified behavior is not part of the conceptual model. These techniques are specified as separate *design decisions* in a design model. In building a design model, the knowledge engineer takes external requirements such as speed, hardware and software into account. Although there are dependencies between conceptual model specifications on the one hand and design decisions on the other hand, building a conceptual model without having to worry about system requirements makes life easier for the knowledge engineer.

The overall KADS methodology, and its supporting literature, provides a rich set of material relating to the detailed design of such models, their integration in systems, sub-methodologies and tools supporting development, and applications experience. The development of KADS and the availability of this material are major landmarks in research on knowledge acquisition for knowledge-based systems. They also represent a major area of ongoing research continuing to involve many people in many countries.

4.2 PCP: a personal construct psychology approach to knowledge acquisition

A number of methodologies and tools that have been highly influential in knowledge acquisition research have been based on personal construct psychology (PCP), Kelly's [36] formal, constructivist model of the epistemological processes whereby people acquire expertise. In 1980, Gaines and Shaw suggested that the tools developed by Kelly for eliciting personal models, would provide a useful development technique for expert systems [25], and performed a validation study of the elicitation of the BIAIT methodology from accountants and accounting students using computer-based repertory grid elicitation [50]. Boose in an independent parallel study reported success in a wide range of industrial expert system developments using computer elicitation of repertory grids [2], and since then many knowledge acquisition systems have incorporated repertory grids as a major elicitation technique [5, 14, 16, 31, 52].

The repertory grid is, however, only one technique for knowledge acquisition that may be derived from personal construct psychology. The formal model proposed by Kelly is highly general because of its system-theoretic derivation from the single primitive process of making dichotomous distinctions. Consideration of the recursive processes of making distinctions between distinctions leads to hierarchies of distinctions having both the generality and the complexity to encompass any model from the informality of human cognitive processes to the formality of mathematical, axiomatic systems [26]. Kelly presented his work as the foundations of a psychological system and emphasized the intensional basis of distinctions as *personal constructs* that could differ widely between individuals leading to very different personal models of the world. However, the same system is applicable to shared social *constructs* and impersonal *formal constructs* based on intensional definitions of distinctions in communal terms, or extensional definitions in concrete terms. Thus, the notion underlying personal construct psychology provide a universal foundation for modeling methodologies.

A companion paper at this meeting gives details personal construct psychology, its origins, its foundations and their application to the formal derivation of KL-ONE-like knowledge representation schema [54]. Other papers give details of knowledge acquisition and tools based on personal construct psychology, including later developments of the repertory grid and visual languages for semantic networks [7, 28]. Some examples of tools based on this approach are given in Section 5.

4.3 General modeling methodologies in relation to KADS and PCP

Clearly knowledge acquisition for knowledge-based systems is not uniquely characterized by its emphasis on modeling techniques. There exists in the scientific, mathematical and engineering literatures very rich frameworks encompassing the nature, function, formation and evaluation of models, including a very wide variety of techniques and tools for the development of models which have become operationalized using computers. In information technology, the notions and techniques of modeling are central to areas such as operations research and simulation—so central, in fact, that many of the key textbooks in these areas do not use 'model' as an index term since it would have so little selectivity.

If one investigates books on classical system analysis the term 'model' is far less used, often absent in both main text and index. The reason for this is significant, and can be seen best in the context of a definition of system analysis, such as that in Couger's survey of the *Evolution of Business System Analysis Techniques* [13]. He defines system analysis as concerned with two initial phases of the system development cycle:

"Phase I—Documentation of the existing system.

Phase II—Analysis of the system to establish requirements for an improved system (the logical design)"

These two phases, which come ahead of design and implementation, clearly satisfy the definition of a model above, in providing a representation that serves a well-defined purpose in relation to the system that is represented. Indeed that purpose may be viewed as supporting the design of an improved operational model, and it is this that probably inhibits the use of the term model in the systems analysis literature for two reasons:

- There is a connotation in operations research that 'models' are operational, that is provide the basis for computer simulation. Hence the results of system analysis were not seen as a 'model.'
- The implementation of a system during the later stages of design and coding involves the construction of sub-systems involving structures that have little resemblance to the system being modeled. Hence the results of system implementation were not seen as a 'model.'

In recent years, as formal specification techniques have been developed, the term 'model' has come into use as a significant methodological concept:

"In the model-based approach, specifications are explicit system models constructed out of either abstract or concrete primitives...This contrasts with the axiomatic approach where specifications were given in terms of axioms which define the relationships to each other, and thus no explicit model was formulated." [12]

The development of formal requirements specification has been part of a three-fold move towards: proof of correctness of implementations as satisfying requirements; simulation of requirements to support system specification; automatic generation of efficient implementations directly from requirements. All of these involve introducing the operationality into system analysis that was missing in its initial history, that is a move from human interpretation of the results of system analysis to computer interpretation of those same results.

Thus, returning to the theme of Section 1 relating to the similarities and differences between knowledge engineering and software engineering, it is not the general notion of modeling that characterizes knowledge acquisition research but rather the types of model developed. Both KADS and PCP place most emphasis on the meta-modeling aspects of their methodologies, on the reflective information structures that characterize knowledge-based systems. There are general modeling formulations that also stress the recursive and reflective nature of the modeling processes, and it is appropriate to review briefly two very powerful general approaches that have found application in knowledge acquisition.

4.3.1 Checkland's soft systems methodology

Checkland's soft systems methodology [10] is a framework for system analysis that provides very powerful techniques for the analysis of systems with human and social components, and has been widely applied to difficult problem areas [55]. There are seven stages of system analysis in soft systems methodology as shown in Figure 3. The initial stages are concerned with system analysis and the later stages with system design. The CATWOE methodology of stage 4 is particularly interesting in its identification of the roles and expertise involved in the system definition.

Stage 1: The problem situation—unstructured

Stage 2: The problem situation—expressed

Stage 3: Root definition of relevant systems—CATWOE methodology

Stage 4: Making and testing conceptual models

Stage 5: Comparing conceptual models with reality

Stage 6: Determining feasible, desirable changes

Stage 7: Action to improve the problem situation

Figure 3 Seven stages of soft systems methodology

Checkland's methodology prescribes six essential components of a system that must be identified at the conceptual modeling stage. The CATWOE mnemonic is a reminder to search for each of these components in the system situation and make them overt in modeling. A system is defined through a *transformation* carried out by people who are the *actors* within it. The system affects beneficially or adversely other people who are its *customers* and there is some agency with power of existence over it who is its *owner*. The system has to exist within a outside constraints forming its *environment* and the whole activity of system definition takes place within an ethos or *weltanschauung* that affects our views of it. The methodology is essentially pluralistic in emphasizing that there will generally be multiple choices for most or all of these components, and the particular choices made will result in different system models.

There are natural links between personal construct psychology and soft systems analysis, and repertory grid techniques have been applied to the computerization of the CATWOE conceptual modeling process [51]. Soft systems analysis is applicable to each of the seven KADS modeling areas, and provides a set of general concepts linking across areas. For example, do those who play the role of owners in one model also play the role of customers in another? Its emphasis on the context in which a system is being designed corresponds to emphasis on meta-information in knowledge-based systems.

4.3.2 Klir's architecture of systems problem solving

Klir architecture of systems problem solving analyses the processes involved in any modeling system in terms of an infrastructure for them that can be instantiated in different ways to encompass many different modeling schema [38, 39]. His basic constructs form a hierarchy of systems: a *source system* providing a descriptive terms, a *data system* providing descriptions in these terms, a *generative system* providing a regeneration of these descriptions in terms of a *structure system* providing theoretical terms, itself described through *meta-systems*, *meta-meta-systems*, etc. Figure 4 shows this modeling hierarchy expressed in terms of a primitive process of forming a construct by making a distinction [26]. Thus, a general modeling system may itself be modeled as a process that makes distinctions in the world, gathers data in terms of those distinctions, selects from a repertoire of representations those which best generate the data, analyzes relations between the structures of such representations, and recursively repeats such analysis to generate higher levels of the hierarchy. The term *anticipation* is used for the output to capture both prediction and action. It is not necessary in general to distinguish whether the system anticipates correctly by passive prediction, or by actively changing the world to be predictable.

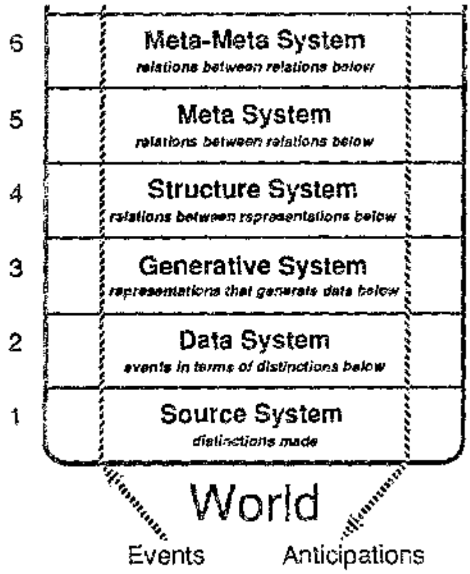

Figure 4 Klir's modeling hierarchy

The instantiation of this hierarchy with particular data description languages, model classes and measures of model complexity and model-data approximation has proven in practice to account for the wide range of mathematical modeling techniques [18, 39]. It is interesting as an account of generalized adaptive processes since it makes clear the presuppositions necessary for modeling to take place—a *tabula rasa* cannot begin to model and some degree of 'innateness' is required. The trade-off between the amount of data needed in learning and the innately 'assumed' constraints upon the world can be investigated [17], as can that with the socio-cultural filtering of data to improve its support of learning [20].

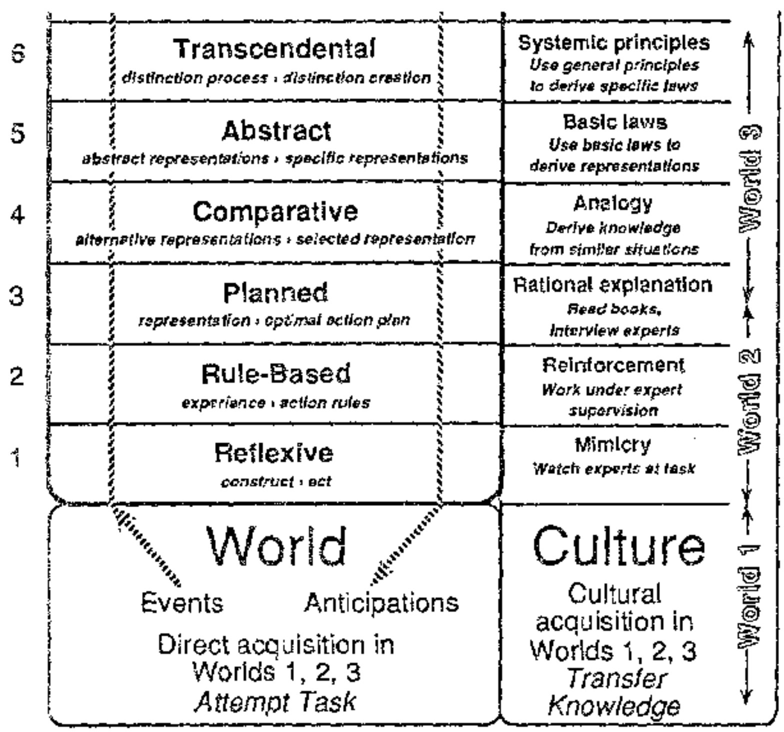

Figure 5 Knowledge acquisition and transfer in the modeling hierarchy

Klir's modeling architecture can be viewed as a general expression of the recursive processes in distinction making at the heart of personal construct psychology, and leads to a systemic model of psychological processes [20].This in turn leads to a general analysis of knowledge transfer processes at different levels in the modeling hierarchy as shown in Figure 5. There is a close mapping between Figures 2 and 5 which throws light on the systemic principles underlying the pragmatic derivation of the layers in Figure 2. Again, the analysis in these general terms is applicable to all of the different modeling areas of KADS, and, in general, it is apparent that there is fruitful convergence between the methodologies developed specifically for knowledge-based systems and the more general modeling methodologies of the general systems literature. This may be expected to result in increasing synergy between knowledge acquisition research and general systems studies in the future.

5 Modeling Tools

The complexity of knowledge engineering as illustrated in Figures 1 and 2, and the multi-faceted demands of methodologies such as KADS suggest that computer support of the knowledge engineering process is essential. Computer-aided software engineering (CASE) tools for knowledge-based system development have been a major theme at these workshops over the years [4, 6], and may be expected to continue to be so. This has given rise to the problem that there are now very large numbers of tools from a wide variety of sources with many different names, and it is becoming very difficult to keep track of them, the techniques involved, and their relevance to particular knowledge engineering tasks. This section gives a brief classification of the methodologies underlying the majority of current tools with examples [27].

5.1 Semi-formal elicitation and structuring through hypertext and hypermedia

Much knowledge is informal yet still valuable in an knowledge-based system. Text and pictures can encode expertise, supplementing computational knowledge. Thus, the parallel development of hypertext and hypermedia is having a substantial impact on expert system architectures and knowledge acquisition tools. Figure 6 shows some of the features of modern document processing systems that impinge on knowledge acquisition. Documents may be acquired from many sources, displayed, re-used in other documents, and linked for hypertext navigation. The text in documents may also be analyzed for associative clusters and these clusters may be grouped to indicate significant concepts.

Hypertext-based knowledge acquisition tools have been developed for use by domain experts to enter relevant case histories directly [35, 47]. They have also been used to support the knowledge engineer in structured analyses of interview material. For example, Woodward's Cognosys [56] supports the analysis of protocols in terms of Graesser and Clark's [33] linguistically derived "general knowledge structures". Other knowledge acquisition tools such as KEATS [44] have been built around a hypertext environment specifically designed for knowledge acquisition. There is also knowledge acquisition and linguistics research targeted on the direct transfer of knowledge expressed in text to structures of frames and rules [32]. Since so much knowledge is already overtly encoded in text and diagrams, in the long term this will become an essential knowledge acquisition technology. Hypertext systems have been coupled to knowledge acquisition tools to provide annotation of the distinctions made and cases described which can then be used to provide explanation facilities in the final performance system [24].

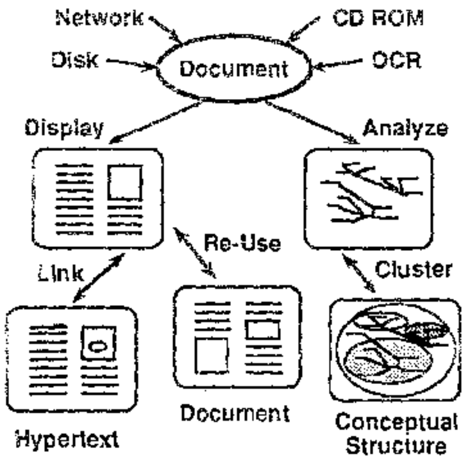

Figure 6 Hypermedia and text processing

5.2 Direct editing of knowledge in a semantic network, frame, rule, representation

Once some informal perspective on a domain has been developed and domain experts have been identified, in some domains where knowledge is already overt it may be possible to move directly to knowledge modeling. Graphic editors providing direct access to semantic network representations allowing knowledge to be encoded in frames and rules provide the most common development environment for knowledge-based systems. They are part of the application programming support environment of most expert system shells, and the widespread availability of modern graphic workstations has made it possible to provide excellent knowledge visualization environments. A wide range of knowledge acquisition tools have been developed that structure and improve the graphic editing environment, often taking advantage of domain knowledge to provide a more specific, meaningful and familiar knowledge framework to the expert. Examples are MOLE [15], KNACK [37], SALT [41], KEATS [44] and KRS [21].

Figure 7 characterizes the major features of these direct editing systems. The expert interacts through a graphic interface with a semantic network knowledge representation schema which may already contain pre-encoded domain knowledge. What is elicited are:

- The distinctions that the expert makes about domain entities (attributes and relations).
- The way in which these distinctions are grouped and constrained to form concepts.
- The entailments between concepts that constitute decision-making rules in the domain.

Older systems have less well-structured knowledge representations but Figure 9 captures the essence of recent developments in knowledge representation that are moving towards very clean, and theoretically well-founded schema.

The knowledge base of frames and rules developed in this way is then usually exported to a performance tool and validated against test case data. This generates the application loop shown on the right of Figure 7 in which the expert's distinctions lead to a description of the problem which is structured through the concepts leading to the application of the inference rules that link them. Many acquisition tools also incorporate or link to some form of performance tool so that this validation can be made part of the elicitation process.

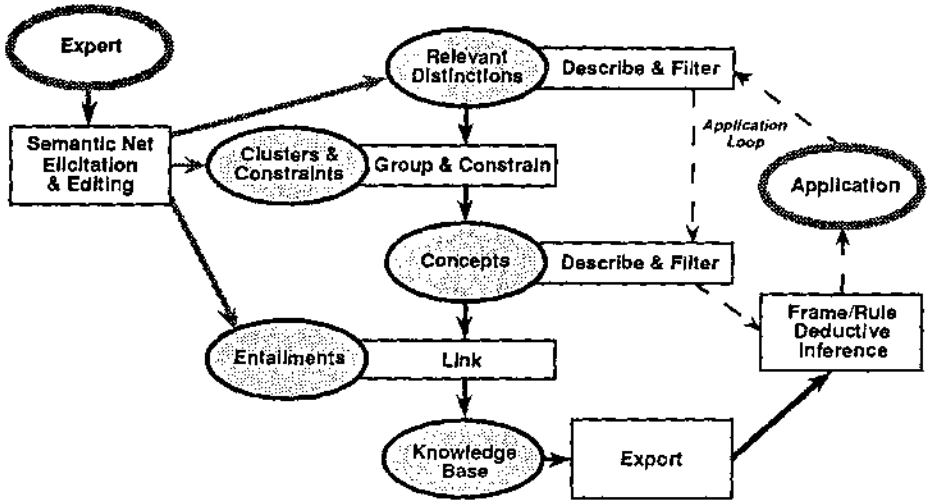

Figure 7 Semantic net architecture

Visual representation of knowledge structures with the potential for editing and enhancement is an attractive way of dealing with the results of other forms of elicitation, and hence semantic network editors are not so much competitors to other approaches but rather important complements to them. Thus, integration with knowledge acquisition sources as well as the capability to export to performance systems are important capabilities of any knowledge editing tool. Many indirect knowledge acquisition tools leave the knowledge presentation and editing to the associated performance tools since these often have excellent facilities. However, in an integrated architecture it is important to incorporate editors in the knowledge acquisition tool that interact effectively with all the different forms of knowledge captured. One of the major problems to be overcome is that once the knowledge has been exported and edited in the performance system it has lost its relation to the acquisition system. Many current knowledge acquisition tools do not support long-term development and knowledge base maintenance largely because of this lack of integration.

5.3 Indirect elicitation through critical cases described in relevant attributes

Repertory grids provide a technique useful for knowledge elicitation when experts cannot directly enter a knowledge structure. They prompt the expert for distinctions relevant to the problem domain and for critical cases that exhibit significant phenomena in the domain. The prompting is done through online analysis of the data being entered leading to feed back to the expert suggesting missing distinctions and cases. This highly focused feedback aids the expert in developing his or her mental model of the domain. It also reduces the inefficiencies of duplication and the mental blocks of psychological set, supporting rapid prototyping. A wide range of knowledge acquisition tools have been developed that incorporate repertory grid elicitation and analysis as their major interface to the expert. Examples are PLANET [50], ETS [2, 3], AQUINAS [5], KRITON [14], KITTEN [52], and KSS0 [29].

Figure 8 characterizes the major features of repertory grid elicitation systems. The expert interacts through a graphic interface to enter individuals in the domain (elements) and bipolar distinctions (constructs). Conceptual clustering techniques are used to feedback the elicited domain structure in an easily assimilated form for validation. Rule induction is used to generate entailments, or, more recently, conceptual induction as discussed in the next section to generate a default rule structure. What is elicited are:

• The distinctions that the expert makes about domain entities.
• Critical cases exhibiting the major phenomena affecting decision-making in the domain.
• The way in which distinctions are grouped and constrained to form concepts.
• The entailments between concepts as induced from the critical cases.

Older systems did not have explicit conceptual induction but left grouping into concepts or frames as a task for the export module.

The clustering and induction modules in Figure 8 are extensions of the basic repertory grid technique incorporated in PLANET and KSS0, and other major extensions have been incorporated in other tools. In particular, AQUINAS makes provision for a wider range of data types than the rating scales of the basic grid, and also allows hierarchies of cases and attributes to be specified that are related to those of semantic nets. Both AQUINAS and KSS0 also incorporate tools supporting multiple sources of expertise and analyzing the relationships between different sources [53].

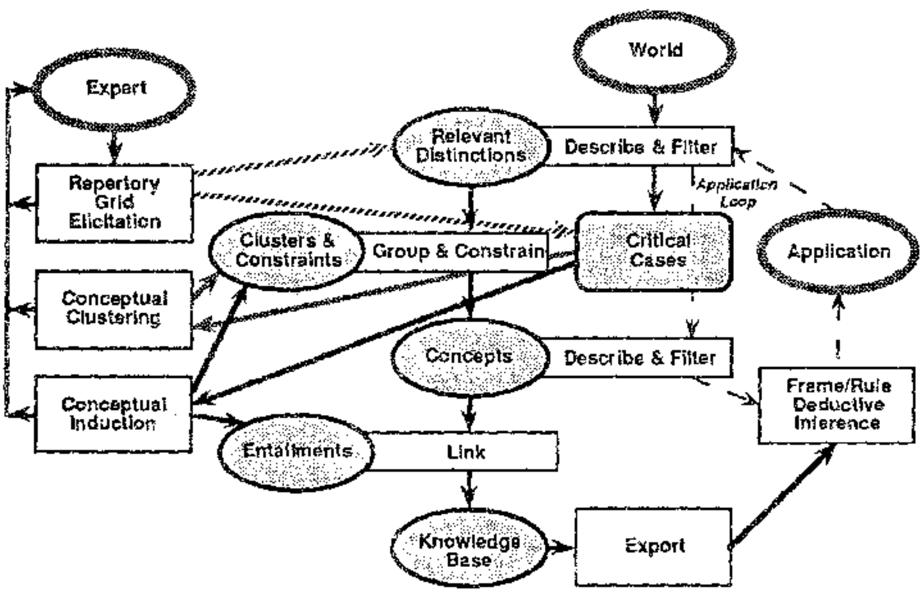

Figure 8 Repertory grid architecture

5.4 Inductive derivation of knowledge from data sets of varying quality

When experts can neither directly enter a knowledge structure emulating their expertise nor enter critical cases stereotyping that expertise, they may still be able to point the knowledge engineer towards case histories that incorporate that expertise and are described in terms of largely relevant attributes and largely correct decisions. Empirical induction techniques may then be used to derive knowledge structures underlying the decisions made in these cases [42].

The best known empirical induction methodology is that of Quinlan's ID3 which has been refined in many ways, particularly to tolerate noise (incorrect decisions), resulting in the current implementation, C4.5 [45]. The original decision tree structure of ID3 based on a subsumption hierarchy of concepts with rules at the leaf nodes only is unnecessarily large in many situations, and extensions to ID3 have been developed that generate modular production rules directly, such as Cendrowska's [9] Prism. This also can be extended to deal with noisy data as in Induct [19]. The extensions that deal with noisy data also make it possible to combine the decision tree and modular rule methodologies to generate default reasoning in which rules are placed at non-leaf nodes in the subsumption structure, and more specialized rules override more generalized ones [22]. Such default rule structures are more compact than either decision trees or modular rules—they are generated by both C4.5 and Induct. Inductive methodologies have also been combined with direct knowledge editing tools, for example in BLIP [43].

Figure 9 characterizes the major features of conceptual induction systems. The expert indicates a database whose designer has supplied distinctions to categorize the world and cases described in terms of these distinctions to represent it. What is derived are:

- That subset of the distinctions that are relevant to the decisions.
- The way in which these distinctions are grouped and constrained to form concepts.
- The entailments between concepts that regenerate the decisions in the database.

Classic empirical induction tools do not generate the conceptual structure but this is a fairly simple extension.

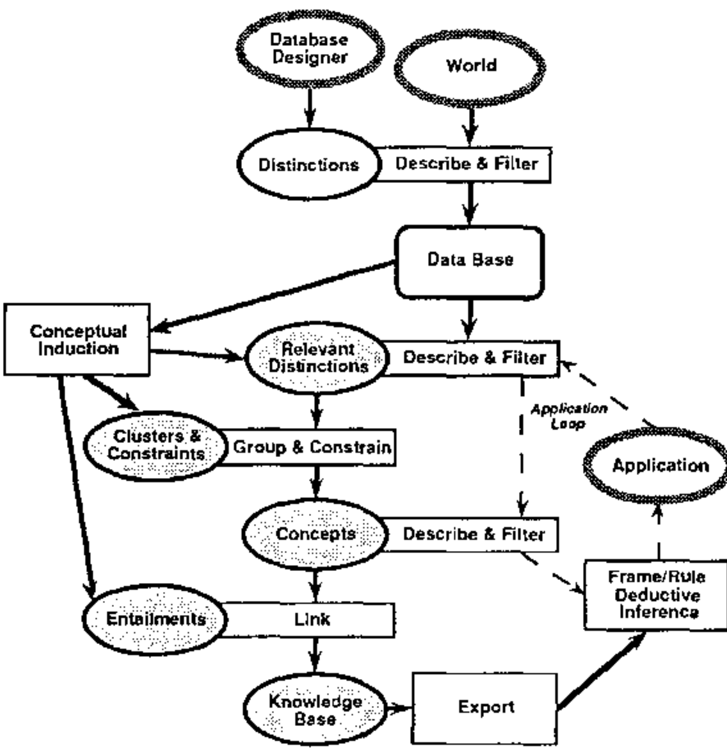

Figure 9 Conceptual induction architecture

5.5 Comparing knowledge modeling techniques

Figures 7, 8 and 9 indicate strong similarities between the outcomes of direct knowledge elicitation, repertory grid elicitation and conceptual induction. This is as it should be since all three techniques are building a complete knowledge base. However, what is not apparent is the relative efficiencies of the different methodologies—that is, how large a database is required to generate the knowledge required to solve a problem? Figure 10 shows the results of one study to investigate the relationship between empirical induction and expertise transfer as knowledge acquisition methodologies [23]. Cendrowska's contact lens data was subjected to random distortion with known statistics to generate large datasets with a certain number of irrelevant binary attributes and a certain percentage of incorrect decisions. Induct was then run on the dataset with 5,000 items, 4,999 items, and so on, until the dataset failed to generate rules giving correct performance. This was done ten times for different datasets of the same type to give estimates of the mean and standard deviation of the size of dataset required to generate correct performance for different forms and levels of distortion.

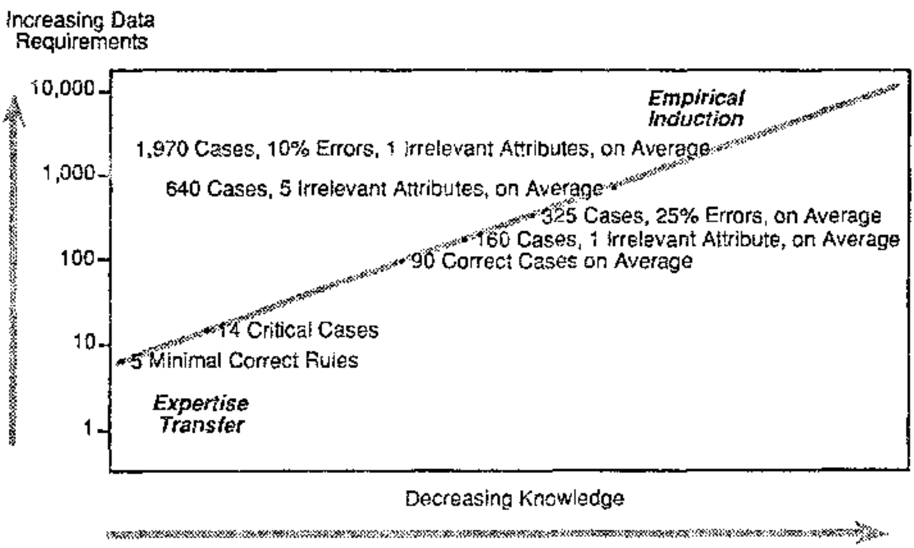

Figure 10 Data/knowledge tradeoff—expertise transfer and empirical induction

The results shown in Figure 10 indicate the very wide range of the tradeoff between data and knowledge: from direct entry of the minimal knowledge structure of 5 default rules; through entry of 14 critical cases; through an average of 90 randomly selected correct cases; to 325 cases with 25% errors; 640 with 5 irrelevant binary noisy attributes; to 1,970 when a single irrelevant attribute interacts with a 10% error rate.

The moral from Figure 10 is not that expertise transfer is better than empirical induction, although the direct entry of overt knowledge is clearly highly ergonomic if it is available. It is rather that all three techniques described above are capable of producing equivalent quality knowledge, and there is a continuum between them in which knowledge is traded for data.

6 Conclusions

Much of the research and practice in knowledge acquisition for knowledge-based systems during the past decade may be brought within a unified framework as the development of theories, methodologies, tools and application experience in the *modeling of expertise*. This is fortunate in enabling a very wide diversity and volume of activities to be encompassed within one conceptual framework. In particular, it captures the essence of major methodologies for knowledge acquisition such as the KADS approach to knowledge modeling and the PCP approach to human modeling processes.

The modeling perspective raises obvious questions as to how modeling for knowledge-based systems differs from system modeling in general, and these seem best answered in terms of fundamental definitions of knowledge itself. In any information system, if the credibility, derivation and context of information are significant, and need to be taken into account as overt data in their own right, then it is probably appropriate to take a knowledge-based approach. That is, the knowledge-based components of an information system will tend to be those in which meta-information and meta-information-processing is required. The major methodologies for knowledge acquisition all emphasize the role of meta-information, and support its acquisition and processing.

The diversity of tools developed to support the knowledge engineering process may also be brought within this unified framework and characterized in terms of their sources of data and the forms of model developed from them.

This year has been one of major advance and consolidation for the knowledge acquisition community with the publication of a number of books and papers presenting integrative accounts of much past research in terms of this modeling perspective. These provide very solid foundations for the next phase of research, development and applications. The systematization of notions of knowledge acquisition, representation and processing, the integration of these notions with advances in information systems engineering, and the incorporation of powerful modeling frameworks and technologies generated within other disciplines, will be the major directions for research in knowledge acquisition for knowledge-based systems during the next few years.

Acknowledgments

Financial assistance for this work has been made available by the Natural Sciences and Engineering Research Council of Canada. Much of the material presented has been drawn from joint works with my colleagues Mildred Shaw and Brian Woodward, and I am grateful for their permission to include it.

References

[1] H. Akkermans, F.v. Harmelen, G. Shreiber, and B. Wielinga, "A formalisation of knowledge-level models for knowledge acquisition," *International Journal of Intelligent Systems*, vol. 8, no. 2 pp. 169-208, 1993.

[2] J.H. Boose, "Personal construct theory and the transfer of human expertise," in *Proceedings AAAI-84*. American Association for Artificial Intelligence: California. p. 27-33, 1984.

[3] J.H. Boose, *Expertise Transfer for Expert Systems*, Amsterdam: Elsevier. 1986.