

A CASE STUDY IN STRUCTURED KNOWLEDGE ACQUISITION

Paul de Greef & Joost Breuker

Department of Social Science Informatics
University of Amsterdam, Herengracht 196
1016 BS Amsterdam

ABSTRACT

Building an expert system usually comprises an entangled mixture of knowledge acquisition and implementation efforts. An emerging methodology based on cognitive psychology and software development guides and supports knowledge acquisition while implementation is deferred. This allows for a more deliberate choice of architecture. This paper presents a case study to test the methodology.

1 INTRODUCTION

A major problem in the construction of expert systems is the method for knowledge acquisition. This may be one of the reasons that despite a widespread need there is little advance in building intelligent consultants for statistical problems. There are two distinct methods. The first one is the methodology of rapid prototyping in which knowledge acquisition and implementation are mixed. The knowledge engineer uses interview data from human experts and immediately starts building a prototype in an implementation formalism. How to make decisions regarding implementation and architecture remains unclear (cf. Hayes-Roth et al. 1983). The second methodology, which we call structured knowledge acquisition, is outlined by Wielinga & Breuker (1984). This case study is meant as a test of this KADS methodology (Knowledge Acquisition Documentation and Structuring). The methodology is implemented in a knowledge acquisition support system, KADS, written in Prolog, using a Prolog KLONE implementation as a structuring device.

A major characteristic of the methodology is a separation between knowledge acquisition and implementation. The task for the knowledge engineer is to bridge the gap between the verbal data from experts and the actual implementation of a system. The methodology provides a theoretically founded step in between. It guides the knowledge engineer in the mapping of verbal data onto an intermediate level provided by an interpretation model, which is an implementation independent description of the domain knowledge on an epistemological level (Brachman 1979, Clancey 1983). It consists of a typology of basic elements and structuring relationships for a certain class of problem solving tasks.

The basic elements are objects, knowledge sources, models and strategies. On the implementation level, knowledge sources can be algorithms or sets of production rules. On the epistemological level a knowledge source is a piece of knowledge that derives new information from existing data. It is equivalent to an elementary subtask, which cannot be decomposed further. A knowledge source that occurs in almost any problem solving task is for instance the classification of objects into categories. Knowledge acquisition consists of repeated cycles of elicitation and analysis of verbal data

aimed at refining (and if necessary rejecting) an interpretation model.

Crucial to the methodology is the use of thinking aloud data. They provide the most informative window to expertise in action. However, in knowledge engineering, these data are hardly ever used. They are assumed to be difficult to interpret (Welbank, 1984) and their use is only recommended as a check on the adequacy of a prototype. However, as psychology of problem solving shows, the analysis of thinking aloud data is feasible when an initial model of the task is used as an interpretative framework (Ericsson & Simon, 1984). In KADS, a classification of such models is available. The knowledge engineer selects one or more interpretation models, describing the expert tasks at a global level. There are interpretation models for specification, diagnosis, planning, design, etc. The advantage of interpretation models is that the knowledge engineer is equipped with a tool that is much closer to the verbal data than an implementation formalism. There are more practical advantages: repair and refinement of the model does not require throwing away some prototype gone stuck into a meddle of ad hoc solutions in an inappropriate formalism.

II Knowledge Acquisition

The KADS methodology prescribes a series of knowledge engineering tasks which can be classified into three types:

1. An analysis of the functions, the environment and the users of the expertise to arrive at a definition of the operational characteristics of the prospective system. The functional analysis defines the modality of the expertise. A knowledge based system contains two types of tasks: problem-solving tasks representing the expertise and communication tasks. These communication tasks are by no means trivial; they form the interface between the operational environment and the expertise. Modality may involve negotiating, explanation, coaching, documentation, etc.

2. An analysis of the static domain knowledge, starting with the collection of a lexicon, ending with concepts structured in (KLONE) concept hierarchies.

3. Analysis of expertise in action, i.e. the way problems are solved. This starts with a task-analysis: selection of one or more interpretation models that appear to represent the structure of the problem solving process. By matching the verbal data from interviews and in particular thinking aloud protocols, this initial model gets refined and modified into a detailed structure of knowledge objects, knowledge sources and strategies; much in the same way as Bennet's (1984) conceptual structures. The final conceptual structure of expert reasoning represents the basic architecture of the prospective system. In the conceptual structure

the static knowledge and the actions performed on them become integrated.

In this section we discuss these knowledge engineering tasks within the domain of statistical analysis of experiments. From now on "task" means task for a statistical consultant. The data used in this case study consisted of textbooks on statistics and twelve thinking aloud protocols, obtained from four experts working in the social science department at our university. One of us is an intermediate expert himself.

The **functional analysis** aims at identification of the objects, agents and functions involved in the expert task. An informal description of the functional analysis is the following: A statistics expert is consulted by a researcher who has the intention to investigate empirical relations among variables. The specification of these intentions is called a **CONCEPTUAL MODEL**. It contains conceptual variables and research questions about conceptual relations among them. This **CONCEPTUAL MODEL** is transformed into a **RESEARCH PLAN** for the collection of empirical DATA. This plan contains steps like **SAMPLING**, **MEASUREMENT**, etc. The transformation is usually done by the researcher, but this may lead to **BUGS** in the research plan. Abstract properties of the research plan -the underlying **DESIGN**- determine which research questions can be investigated and by which **ANALYSIS MODEL** the data should be analyzed. A **BUG** is a property of the research plan that prohibits a clear and correct answer to a specific research question. The expert produces an **ADVICE** which consists of an **ANALYSIS METHOD** and identified **BUGS** in the **RESEARCH PLAN**.

The **domain lexicon** is collected from verbal data and textbooks. It provides a vocabulary to communicate with experts, and to identify the domain specific concepts. These concepts need to be structured in hierarchies. One approach is to use experts or textbooks to identify the general concepts at the top. However, such an approach is naive, because these structures rather reflect the support knowledge that is used for a theoretical justification of statistical designs than the way the expert uses concepts while performing his task. Initial structuring is provided by the knowledge objects specified by the functional analysis and knowledge objects which are the interfaces between knowledge sources in the conceptual structure of the reasoning. During the analysis many new concepts were added, because statistics textbooks lack concepts relevant to application.

In the **task analysis** an initial template interpretation model was selected, derived from a problem solving system in the domain of thermodynamics (Jansweijer et al, 1982). These domains have in common that an informal problem is transformed into a formal structure to which formal principles can be applied. The template model describes a first stage **ORIENTATION** in which the process **SKETCH** obtains an overview of the informal problem situation which is transformed by **SCHEMATISE** into a formal problem statement. The second stage solves the problem and is followed by an evaluation stage.

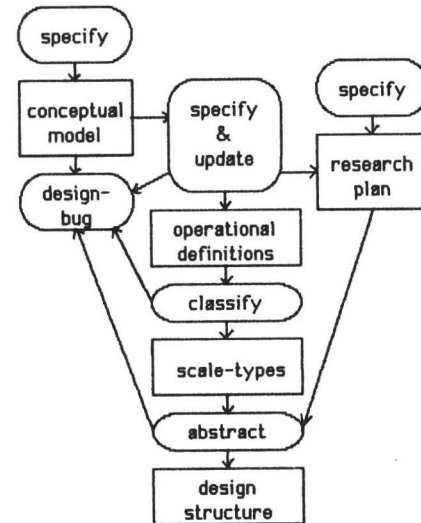
The thinking aloud protocols were especially useful for identification and refinement of the knowledge sources in the successive interpretation models.

III. CONCEPTUAL STRUCTURE OF TASKS AND SYSTEM DESIGN

An overview of the interpretation model for the **ORIENTATION** stage is presented in fig. 1 (more details can be found in the

Knowledge Acquisition Document (De Greef, 1984)).

Fig. 1: tasks in the **ORIENTATION** stage. (Ovals represent types of knowledge sources, rectangles represent types of knowledge objects.)



The control of the problem solving process is guided by a plan that can be expressed as a tree. A general plan for execution of these tasks is shown in table 1.

Table 1: Tasks in the orientation stage of statistical consultation.

ORIENTATION
SKETCH
SPECIFY CONCEPTUAL MODEL
SPECIFY OPERATIONAL DEFINITIONS
UPDATE RESEARCH PLAN
SCHEMATISE
CLASSIFY OPERATIONAL VARIABLES
ABSTRACT THE DESIGN-STRUCTURE

A knowledge source may modify the plan. For instance, **SPECIFY OPERATIONAL DEFINITION** may run into trouble, because the client cannot answer a question or because the current information leaves an ambiguity about the **RESEARCH PLAN**. Then, before finishing and returning control, it may create a new subplan, **SPECIFY RESEARCH PLAN**. Such change of strategy was frequently observed with human experts. Another instance: **CLASSIFY OPERATIONAL VARIABLES** may conclude that specific properties of certain **OPERATIONAL VARIABLES** prohibit the investigation of a specific **RELATION** in the **CONCEPTUAL MODEL**. Then a knowledge source of the **DESIGN-BUG** type is inserted as a new subplan. **DESIGN-BUG** will communicate the conclusion to the user and change or augment the **CONCEPTUAL MODEL**. Here human experts often try to improve the **RESEARCH PLAN**. Since this is plan is usually executed and done, this is of little practical value and need not be included in the system. The final product of the orientation stage is a formal problem statement. The next stage selects a model that defines which relations can be investigated and which statistical models should be used. There exist algorithms to do the actual assembly of statistical analysis models, provided that certain assumptions about the dependent variables can be made. If not so, a decision tree is

sufficient to find an assumption free analysis technique, if there exists an applicable one.

A most critical task for a consultation system is obtaining the correct problem statement from the client. Many misinterpretations may occur, as is shown in a number of protocols. A misinterpretation is usually discovered by the client after hearing a solution from the expert. This causes a very complex process of debugging the problem statement (CONCEPTUAL MODEL and/or RESEARCH PLAN) and adaptation of the solution (the ADVICE). The system can be made less complex if it lets the user evaluate the correctness of the problem statement before attempting to solve the problem. For evaluation by the user and subsequent refinement and debugging of the problem statement we propose the system displays graphical structures which can be edited by the user. For the CONCEPTUAL MODEL, a labeled graph can be used. For the RESEARCH PLAN it can use flow diagrams as proposed by O'Keefe (1981).

The system design discussed so far, is suitable for users with little but sufficient experience in statistics. A user with insufficient knowledge will not be able to answer all questions. More sophisticated users may be bored or irritated, not being able to use any shortcuts. Although a system like this will be useful as a front end to a set of statistical software packages, there is still something to be desired: adaptation to users with different levels of statistical knowledge, explanation of bugs in the research plan, use of the system as a helper in selecting a design and making a research plan (instead of being only a critic), etc. These properties have nothing to do with the statistical problem-solver but with the modality of consultancy. It is desirable that a consultant has qualities of an Intelligent Tutoring System and is also a capable research designer and planner.

IV IMPLEMENTATION

We have implemented a small prototype. It can solve correlational and experimental problems following the plan outlined in table 1 and two other plans. The concepts are part of a KLONE structured inheritance network (Brachman 1979), implemented in Prolog. The interpreter language which can be translated to Prolog is borrowed from the PDP problem solver (Jansweijer et al. 1982). After six man months of knowledge acquisition, it took four weeks each for two computer science students to implement a shell for the prototype using the PDP tools and the knowledge acquisition document in De Greef (1984). The prototype does not yet contain user editable graphical displays.

V CONCLUSIONS

The case study shows that the methodology is viable and that a separation between analysis and implementation is possible. The result of analysis is a knowledge acquisition document which provides a sufficient basis to enable two students unfamiliar with the statistical domain to implement a prototype in short time.

The methodology is an alternative to the method of rapid prototyping, in which a knowledge engineer very quickly makes architectural decisions and builds a prototype. The prototype is shown to the expert who is expected to provide useful criticism. Then the prototype is changed or a new system is built. This case study used a methodology in which major commitments to architecture and implementation efforts are postponed until it is clear what has to be implemented.

A conceptual structure of a task can be evaluated by an expert in much the same way as a prototype. The difference is that the evaluation is focussed on the the problem solving process; not on the performance of a system. Moreover, changing a conceptual structure takes far less effort, than debugging a prototype system. We have experienced that this method of obtaining feed back on the interpretation of expertise in action is less time consuming and probably as effective as evaluating a prototype, whose inner workings may appear rather obscure to the expert.

ACKNOWLEDGMENTS

We want to thank the domain experts for sharing experience and thoughts, Kees Buisman and Dick v.d. Vlugt for their work on the prototype, and Robert de Hoog and Bob Wielinga for valuable advice.

REFERENCES

- Bennett, J. S. "ROGET: a Knowledge-Based Consultant for Acquiring the Conceptual Structure of an Expert System." Memo HPP-83-24, Stanford University, 1984
- Brachman, R. J. "On the Epistemological Status of Semantic Networks." in: Findler, N. V. (ed) "Associative Networks, Representation and Use of Knowledge by Computers." New York, Academic Press, 1979, 3-50
- Clancey, W. J. "The Epistemology of a Rule-Based Expert System, a Framework for Explanation." *Artificial Intelligence* 20, 1983, 215-251
- De Greef, H. P. Report 1.7. ESPRIT Project 12 Department of Social Science Informatics, University of Amsterdam, Amsterdam, 1984
- Ericsson, K. A. & Simon, H. A. *Protocol Analysis: Verbal Reports as Data*. Cambridge Mass., The MIT Press, 1984
- Hayes-Roth, F., Waterman, D. A., Lenat, D. B. (eds.), *Building Expert Systems*. Addison Wesley, Reading, MA, 1983
- Jansweijer, W., Konst, L., Elshout, J. J., Wielinga, B. J. "PDP: a Protocol Diagnostic Program for Problem Solving in Physics." In *Proc. ECAL-82*, Orsay, 1982
- O'Keefe, R. A. "Automated Statistical Analysis", DAI working paper 104, Department of Artificial Intelligence, University of Edinburgh, Edinburgh, 1981
- Welbank, M. "A Review of Knowledge Acquisition Techniques for Expert Systems", Martelsham Consultancy Services, Ipswich 1983
- Wielinga, B. J. & Breuker, J. A. "Interpretation of Verbal Data for Knowledge Acquisition." In *Proc. ECAL-84*, Pisa, Italy, 1984, pp. 41-50

* This work was supported in part by the ESPRIT programme of the European Community