Cognitive plausibility of a conceptual framework
*(for modeling qualitative prediction of behaviour)**

Bert Bredeweg and **Cis Schut**
Department of Social Science Informatics (S.W.I.)
University of Amsterdam, Roetersstraat 15
1018 WB Amsterdam (The Netherlands)
Telephone: +31-20-5256788
Telefax: +31-20-5256896
E-mail: bert@swi.psy.uva.nl

Abstract

In this paper we investigate the cognitive plausibility of an integrated framework for qualitative prediction of behaviour. The framework is based on the KADS expertise modeling approach and integrates different approaches to qualitative reasoning. The framework is implemented in a program called GARP. To test the cognitive plausibility a physics problem involving qualitative prediction of behaviour was constructed. The behaviour prediction of this problem generated by GARP was compared to think-aloud protocols of human subjects performing the same problem solving task.

Introduction

Developing conceptual frameworks for modeling problem solving expertise is considered to be an important aspect of research on Knowledge Based Systems (KBS) [15; 8; 12]. It is a generally accepted hypothesis that such frameworks can usefully support the knowledge acquisition process. However, little research has been published that investigates this hypothesis. In particular little attention has been given to test the cognitive validity of these frameworks. After all they are used to interpret verbal data of human problem solving behaviour.

In this paper we investigate the cognitive plausibility of the conceptual framework proposed by the KADS methodology. Within this framework we focus on the reasoning task *qualitative prediction of behaviour* as the specific object of our research. We compare think-aloud protocols of human subjects predicting the behaviour of a complex configuration of balances with an implemented computer model of the same reasoning task.

The contents of this paper are structured as follows. The next two sections describe the KADS conceptual framework and the reasoning task qualitative prediction of behaviour. The paper then moves on to a protocol analysis of this problem solving task carried out by human subjects and describes to what extent the framework for qualitative prediction of behaviour fits the think-aloud protocol data. Finally, the discussion summarises and further discusses the main results.

Modeling problem solving expertise

KADS proposes a methodology for building KBS's [6; 7] and is based on the premise that knowledge used by people during reasoning processes can be distinguished according to several types, corresponding to the different roles the knowledge plays in the reasoning process (see figure 1). First, the *domain layer*

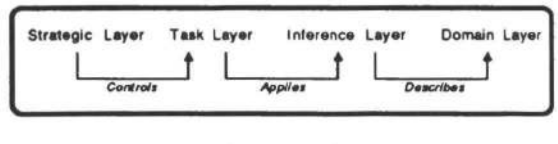

Figure 1: The KADS four layers

describes the domain knowledge in terms of

*The research reported here was partially funded by the Esprit Basic Research Programme of the Commission of the European Communities as project number 3178 (Reflect). The partners in this project are the University of Amsterdam (Amsterdam, The Netherlands), the Netherlands Energy Research Foundation ECN (Petten, The Netherlands), the National German Research Centre for Computer Science GMD (St. Augustin West-Germany) and BSR Consulting (Munich, West-Germany).

domain concepts, relations and complex structures such as models of processes or devices. This knowledge is static in nature, that is, it is not specified what control regime is needed in order to use the knowledge effectively

The second type of knowledge, the *inference layer*, concerns the canonical problem solving actions (*knowledge sources*) that are the basis of reasoning. Knowledge sources are elementary in the sense that other parts of the problem solver can not influence their internal control. They represent the way in which a domain relation can be used to make inferences. For example: a ***specification*** and an ***abstraction*** inference might both use a ***subsume*** relation at the domain layer to make their inference.

The different *roles* the domain knowledge plays in the reasoning process is described by *meta-classes*. For example, a domain concept like *faulty-transistor* may play the role of a *finding*, but may also play the role of a *hypothesis*. In this case finding and hypothesis represent domain independent use of domain concepts for diagnostic reasoning.

The third type of knowledge, the ***task layer***, describes what goals are involved in a particular problem solving task and how the available knowledge sources can be ordered and applied to satisfy those goal. For example, the problem solving task *monitoring* can be carried out *data-driven* or *model-driven*, both representing different task structures that can be used to do monitoring.

The fourth type of knowledge concerns strategic knowledge. At the *strategic layer* strategies are described that control the overall reasoning process. It should, for instance, plan a particular task structure, monitor its execution, and, if needed, diagnose, repair or even substitute the current task structure with another task structure, until the desired problem solving goal is reached.

There is a tradeoff between the task and the strategic layer. In case of non-standard problems the problem solving process will use more strategic reasoning. This in contrast to more routine problems in which case the problem solver is more likely to execute an already known task structure.

Qualitative prediction of behaviour

The qualitative reasoning community has delivered a number of implemented AI programs along with theoretical insights of what constitutes qualitative reasoning [2; 13]. Qualitative prediction of behaviour is an analysis task during which new properties of a system, namely states of qualitatively distinct behaviour (=output), are derived from a structural description of the system (=input). We have developed a description of this prediction task, in terms of the conceptual framework provided by KADS [5; 3], that unifies the theoretical insights mentioned above. To test the validity of our framework we have implemented a program called GARP [4][1] that can simulate qualitative reasoning as described by the original approaches. GARP is therefore a qualitative reasoning shell that integrates the basic approaches to qualitative reasoning.

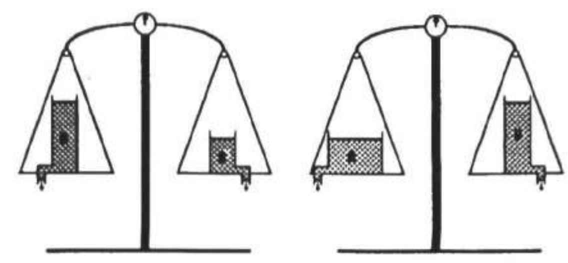

Figure 2: Two of the six balance problems

To investigate the cognitive validity of the theory underlying GARP, the qualitative reasoning task was operationalised with six balance problems (see for examples figure 2). The problem was to predict the behaviour of balances with containers filled with water on each balance-arm. Through outlets near the bottom of the containers the water gradually flowed out of the containers. In each version of the problem the two containers differed in shape and in the amount of water they contained.

In the next sections the 4-layer model for solving these balance problems is described.

Domain and inference layer

The original approaches to qualitative reasoning provide specific ontologies for structuring the knowledge at the domain layer (cf. [3]). At the inference layer the meta-classes describe how the domain knowledge is used and the knowledge sources describe the inferences that can be made. The framework distinguishes between the following meta-classes:

- *System model description*
  A description of the system during a period of time in which the behaviour of the

[1]GARP -A Generic Architecture for Reasoning about Physics- is implemented in SWI-Prolog [14]. Both GARP and SWI-Prolog are distributed without charge for non-commercial purposes. Please contact the first author for more details.

system does not change (=SMD). The notion of change is rather subtle because the actual (real-world) system may change whereas from a qualitative point of view its behaviour remains constant.

- *System elements*
  Entities (like physical objects) from the real-world explicitly represented in the problem solving process.
- *Parameters*
  The properties of system elements.
- *Parameter values*
  The values a parameter can have.
- *Quantity spaces*
  An ordered set of values a specific parameter can have.
- *Parameter relations*
  Dependencies between parameters.
- *Qualitative calculi*
  Define the semantics of a parameter relation.
- *Mathematical model*
  A set of relations that holds during a particular SMD.
- *Partial models*
  Partial models[2] are used to further specify an SMD. They represent the general knowledge of the domain, like for example the properties of a liquid contained by a container of some sort.
- *Transformation rules*
  Represent knowledge about how to find successive SMD's. Three types of rules have been identified: *termination*, *ordering* and *continuity* (they are explained in more detail below).
- *Behaviour descriptions*
  A set of SMD's ordered in time. It represents the possible behaviour of some real-world system.

There is not enough space in this paper to completely describe the domain knowledge needed to represent the balance problems. Some of the most important entities of the domain knowledge are therefore enumerated below, together with the meta-class that describes their role.

- *System elements*
  balance, container, water, balance-arm, outlet, contain-relation support-relation, connect-relation.
- *Parameters*
  pressure, amount, mass, volume, flow-rate, height, width.
- *Parameter values*
  zero, minus, plus, absent, present, low, high, equilibrated
- *Parameter relations*
  pressure $\rightarrow$ flow-rate, height $\rightarrow$ pressure, amount = volume, amount = mass.
- *Partial models*
  water-flow-from-container-to-world, balance-with-weights, contained-liquid

The partial models in the domain layer specify several possible viewpoints that subjects can use in solving the balance problems. In particular we expect the partial model 'water flow from container to world' to differ for subjects with respect to the factors that determine the speed of the water outflow:

A size of the outlet

B height of the water column

C volume of the water column

D width of the water column

Given the list of meta-classes, the canonical inferences (knowledge sources) used in qualitative prediction of behaviour can be described. Figure 3 gives an overview of these inferences.

The purpose of the *compound specify* inference is to develop a 'complete' description of a particular state of behaviour of a system (=augmented SMD), which means assigning a qualitative value to each parameter that is used to describe the system. Two inferences must be carried out in order to arrive at such a description, *assembling* a mathematical model and *computing* the corresponding qualitative values. Assembling a mathematical model (=partially augmented SMD) is done by finding partial models that apply to the input system (=partial SMD). Once the SMD is augmented with a mathematical model, the computation of qualitative values can be done. There are two outputs possible, (1) contradiction, which means there is no set of values consistent with the current set of relations, (2) solution, which means one, or more sets of values are consistent with the current set of relations.

The *compound transformation* inference is concerned with identifying successive states of behaviour. *Termination* rules specify the conditions under which a particular state of behaviour will terminate and are used to *select* the possible terminations. *Precedence* rules specify the order in which changes take place

[2] In previous work we referred to partial models as 'system structures'.

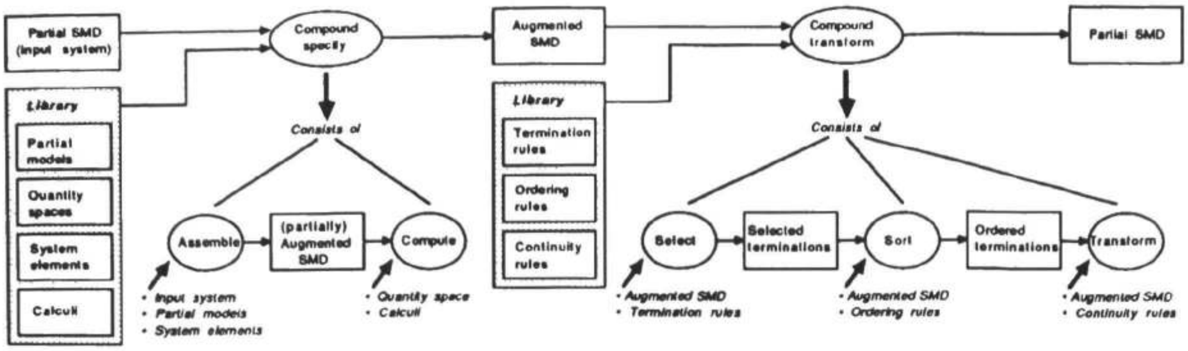

Figure 3: The inference structure for prediction of behaviour

and are used to *sort* the available terminations. Finally, ***transition*** rules specify the continuity conditions that need to be satisfied by the new state of behaviour in order to be a valid successor. They are, together with the ordered terminations, the input for *transforming* the current SMD into its successor.

Task and strategic layer

In the prevaling approaches to qualitative reasoning the task layer is only minimally filled. There are in fact only two tasks: 1) finding all states of behaviour and 2) finding one specific trace of behaviour. In case of the latter additional input parameter values are taken into account to limit the number of SMD's that can be found.

Strategic knowledge is not present in the original approaches. It is not possible to monitor the inference process and modify or change the reasoning process automatically.[3] However, it is likely that the reasoning process will encounter difficulties, i.e. that the problem solving goal can not be reached. We expect two types of difficulties to emerge (cf. [1; 10]): first, problems with the available knowledge (what we refer to as 'knowledge conflicts'), and second, shortage of processing capacity. We will not discuss processing capacity further in this paper. For the knowledge conflicts concerning missing, ambiguous, or contradictory facts in the domain knowledge, we expect the following repairs:

- **Missing knowledge**
  Practical repairs:
  read the question again
  - ask for additional information
  Repairs by reasoning:
  continue after making an assumption
  try extreme values
  use analogy
  use different/other domain knowledge
- **Ambiguous knowledge**
  - try one (randomly or based on heuristic)
  try all
  - reason backwards from a known state
- **Contradictory knowledge**
  try again
  - check computations
  - use other domain knowledge

Once a repair is selected its plan is tuned to fit on the specific instance of the impasse. This means that strategic reasoning may involve a revision at the task layer.

Protocol analyses of the conceptual framework

To test the conceptual framework we compared problem solving activities as manifested by human subjects in think-aloud protocols with those predicted by the framework. The problem solving model that was built and implemented in GARP was therefore translated into a *coding template* with which the think-aloud protocols could be analysed. For every inference activity in the reasoning process that was distinguished in the model, a category was created and a code assigned. The subjects were ten psychology sophomores who had taken high school courses in physics. Three protocols were coded entirely, resulting in 673 coded expressions. All ten protocols were screened on the occurrence of following three phenomena:

- *The model is incomplete*
  There are expressions that can not be coded.
- *The model is too detailed*
  There are coding categories that are not

[3] There is some work going on as an extension of the constraint centered approach (cf. [11]) to filter (and thereby reduce) the number of generated states.

used.

- *The model is wrong*
  There are deviations of the expected order in which expressions appear.

Below the results of this analysis are described for each knowledge layer of the model.

The domain layer

In going through the sequence of problems subjects seemed to have a tendency to abstract from the details of the system. While solving the first problems specific *system elements*, like container and water, were referred to. In the later problems the behaviour of the system was described as if only one object was present, the balance, with a mass-loss on two sides, one losing mass faster than or with equal speed as the other. In a similar way many *parameters* were used while solving the first problems. For example, height (or amount etc.) was seen as determining the pressure, which in turn determines flow-rate which equals mass-loss. In abstracted form the height determined the mass-loss directly. Although each specific model of the balance problems can be represented in GARP, the abstraction process itself can not.

Subjects used different *partial models*. One subject used viewpoint A (size of outlet) but shifted on a later version to D (width of water column). One subject used viewpoint B (height of water column) (although three other subjects also considered it). Six subjects used viewpoint C (volume of the water column), three of whom after showing doubts about other viewpoints. Only one subject used viewpoint D from the outset. The remaining subject used an alternative viewpoint: the weight of the air over the water-surface determines the water-pressure.

The inference layer

Compound specify made up 37 percent of the protocols statements, with a ratio of 2:3 between *assemble* and *compute*. Assembling the mathematical model typically appeared in the protocols as expressions in which was stated that a parameter had a value or that a relation or inequality applied. Computing values made implicit use of a qualitative calculus. It appeared in remarks like: "because the pressure in A is higher the water flows out with more force". Below some examples of these inferences from the protocols are listed:

- *Specify*
  It just matters ...
  In a balance it just matters what the weights are
- *Compute*
  Well, I think first it goes down at the side of container A
  Because at the outset it is heavier off course
  Because here [A] is more water
- *Specify/Compute*
  Let me think if it flows here [A] faster than here [B]
  Here [A] the pressure is, let me think
  Yes, here [A] it is higher of course at the bottom of the container
  Does it influence the flow-rate, let me think
  Yes, the force of outflow will be bigger, undoubtedly

The sequence of assemble followed by compute was regularly violated and as such differed from what the model predicted. First for one part of the system, applicable partial models were found and computable parameter values were calculated. This was repeated for the next part of the system. It seemed as if founded partial models focussed the search for new ones (cf. [9]).

Compound transform made up 25 percent of the protocols statements. *Selecting* terminations was expressed in ways a state of behaviour ends: "then the amount of water in container A becomes equal to that in B". The *sorting* of terminations was only present in the protocols when the reasoning process was disrupted, i.e. if ambiguity in a situation was detected. Below some examples of these inferences from the protocols are listed:

- *Select (terminations)*
  A goes down ...
  Until B is entirely empty ...
- *Sort (terminations)*
  Yes, they either are empty at the same time or ...
  B is already ...
  I can not imagine A being sooner empty, but why can't I ?
  I just, it's more a feeling, let me think
  A has got greater pressure but ...
  No, I wouldn't know if they'd become empty at the same moment

The limited occurrence of the sorting inference can be explained in two ways. First, sorting occurs only if more than one termination is found. If so, the subjects did not find impossible terminations and terminations that were aspects of the same change, they just found possible ones (sometimes comprising different aspects). Second, impossible terminations and different aspects *were* found but the sorting of terminations in unambiguous cases is an automated process and therefore not reportable. The fact that the knowledge used to make this inference is not domain specific, makes an automated process plausible. However, additional research is needed to resolve this issue.

Little evidence was found for the third inference that constitutes the ***compound transformation*** (note that in figure 3 this inference, which follows sort, is also called transformation). We must however assume that subjects make this inference in order to preserve the continuity between one state and the next: values and inequalities that do not change in the next state are propagated. The lack of expressions confirming this inference can be explained by regarding it as a default mechanism. It provides a good example of how subjects deal with the frame problem, that is, unless something explicitly changes, they consider things to be constant over states.

The task and strategic layer

A total of 40 percent of the protocol expressions was coded to involve strategic reasoning, of which 7 percent was devoted to the detection of impasses, 33 to the overcoming of these impasses. This includes 12 percent specification and 11 percent transformation tasks that were executed in, for example, reasoning backwards from the derived final state, starting all over again and re-checking the computations. The remaining 10 percent covered other repairs such as making assumptions, comparing competing knowledge, using analogies and reading the question again. Below follow examples of these repairs in the protocols:

- *Extremes*

  I wonder if in case of a narrow, very narrow and very high column ...

  If the pressure is higher than in a very wide container with a very thin layer of water

  Just to imagine if it matters, if it's volume that matters or just the height of the column

- *Assumption*

  I think it's volume that matters

  So they flow out with equal speeds and the balance remains equilibrated

- *Analogy*

  Maybe it's like, a bit strange maybe:

  Taller people have bigger feet, so the pressure is more spread

- *Back from end*

  So, after a while it will ...

  Yes, in the end it will come back to equilibrium, no doubt, but ...

  Let me see

The expressions that could not be coded, were all of one type: they referred to an activity that related different problems of the sequence. Apparently the subjects were able to see similarities in two versions of the problem. This also explained the occurrence of the immediate production of an answer without any apparent specification and transformation in a few places. When the model, built for one version of the problem, was applied to another in the sequence, the complete behaviour description was applicable so the answer could be produced immediately. This type of inferencing could not be accommodated for by the GARP model.

Discussion

The model presented in this paper extends previous descriptions of qualitative prediction of behaviour by distinguishing between domain, inference, task, and strategic knowledge. The conceptual framework implemented in GARP, which is based on these knowledge types, appears to be very useful for describing and interpreting the reasoning processes involved in this problem solving task. Both the different viewpoints subjects have on the domain knowledge as well as their reasoning process can easily be modeled by the framework. The canonical inferences and the metaclasses defined in the model, provide strong means to interpret the steps of the reasoning process in the protocols. The notion of strategic reasoning explains disruptions and changes of order in which new states are determined.

However, from the presented experiment it also becomes clear that the model of the dynamics of the qualitative reasoning process of human subjects needs to be refined. In particular it is unclear what determines the order of the assemble and the compute inference during the specification of an SMD (=state of behaviour).

The modeling framework also does not account for learning over a number of problem solving sessions. As the subjects moved from one problem to another they abstracted from irrelevant details in the description and the analysis of the problem. The framework does not support this abstraction process. It seems however that this lack of support is part of a larger problem, namely the lack of structuring principles at the domain layer. The modeling framework provides no normative support to choose a level of abstraction at which the knowledge in the domain layer should be modeled.

The notion of a task structure was less useful in analysing the protocols. Experts performing a particular problem solving task may develop, through the repeated execution of the same sequence of inferences, a trace of this sequence and as such learn a task structure. That is, an instance of a strategic reasoning process. Non-experts, like the subjects in the experiment reported on here, just use general strategic reasoning.

In conclusion we believe that the model presented in this paper provides a strong basis for further research on the way humans perform qualitative prediction of behaviour. Such research should in particular focus on the learning aspects and the knowledge structuring principles that people use to develop their domain knowledge.

Acknowledgement

Bob Wielinga and Maarten van Someren provided helpful comments on earlier drafts of this paper. Also we would like to thank the referees who reviewed our paper for their detailed and accurate comments.

References

[1] B. Bartsch-Sporl, M. Reinders, H. Akkermans, B. Bredeweg, T. Christaller, U. Drouven, F. van Harmelen, W. Karbach, G. Schreiber, A. Voß, and B. Wielinga. A tentative framework for knowledge-level reflection. ESPRIT Basic Research Action P3178 REFLECT, Deliverable IR.2 RFL/BSR-ECN/I.3/1, BSR Consulting and Netherlands Energy Research Foundation ECN, August 1990.

[2] D.G. Bobrow, editor. *Qualitative Reasoning about Physical Systems.* Elseviers Science Publishers B.V., Amsterdam, 1984.

[3] B. Bredeweg. Introducing meta-levels to qualitative reasoning. *Applied Artificial Intelligence*, 3-2:85–100, 1989.

[4] B. Bredeweg, M. Reinders, and B.J. Wielinga. Garp: A unifying approach to qualitative reasoning. Technical report, University of Amsterdam, January 1990.

[5] B. Bredeweg and B.J. Wielinga. Integrating qualitative reasoning approaches. In *Proceedings of ECAI-88, Munich*, pages 195–201, 1988.

[6] J. Breuker, B. Wielinga, M. van Someren, R. de Hoog, G. Schreiber, P. de Greef, B. Bredeweg, J. Wielemaker, J-P Billault, M. Davoodi, and S. Hayward. Model Driven Knowledge Acquisition: Interpretation Models. ESPRIT Project P1098 Deliverable D1 (task A1), University of Amsterdam and STL Ltd,, 1987.

[7] J. A. Breuker and B. J. Wielinga. Model Driven Knowledge Acquisition. In P. Guida and G. Tasso, editors, *Topics in the Design of Expert Systems*, pages 265–296, Amsterdam, 1989. North Holland.

[8] B. Chandrasekaran. Towards a functional architecture for intelligence based on generic information processing tasks. In *Proceedings of the 10th IJCAI*, pages 1183–1192, Milano, 1987.

[9] A.A. DiSessa. Phenomenology and the evolution of intuition. In D. Gentner A.L. Stevens, editor, *Mental Models.* Lawrence Erlbaum, Hillsdale, 1983.

[10] W. Jansweijer, J. Elshout, and B. Wielinga. The expertise of novice problem solvers. In *Proceedings ECAI'86, Brigthon*, 1986.

[11] B. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29:289–388, 1986.

[12] L. Steels. Components of expertise. *AI Magazine*, (summer issue), 1990. Also as: AI Memo 88-16, AI Lab, Free University of Brussels.

[13] D.S. Weld and J. De Kleer. *Readings in Qualitative Reasoning about Physical Systems.* Morgan Kaufmann Publishers Inc., 1990.

[14] J. Wielemaker. SWI-Prolog 1.4 Reference Manual. Technical report, SWI, University of Amsterdam, Amsterdam, The Netherlands, January 1990.

[15] B. Wielinga and J. Breuker. Models of expertise. In *Proceedings ECAI'86*, pages 306–318, 1986.