SEMI-AUTOMATED ANALYSIS OF PROTOCOLS FROM NOVICES AND EXPERTS SOLVING PHYSICS PROBLEMS

Konst, L.; Wielinga, B.J.; Elshout, J.J. and Jansweijer, W.N.H.
Institute for Cognitive studies
University of Amsterdam
Weesperplein 8, 1018 XA Amsterdam

ABSTRACT

PDP (Protocol Diaqnostic Program) is a proqram for analyzing think-aloud protocols of subjects solvinq elementary physics problems *. Two versions of PDP implement models of problem solving behaviour at novice and intermediate levels of expertise. In order to compare the behaviour of the models with think-aloud protocols of subjects, PDP is equiped with a think-aloud component, which communicates with a human coder. Any discrepancies between the behaviour of the model and the subject activate a diaqnostic component, which identifies the deviation and may redirect the behaviour of the model. The output of the system is an analysis of a protocol in terms of the computational model, annotated with possible discrepancies between model and subject. Results obtained by analyzing a number of protocols of experts and novices are presented.

1. Introduction

The goal of our project is to investigate differences between novice and expert problem solving in semantically rich domains in order to explicate what is learned by doing. A basic assumption is that the behaviour of a human problem solver is determined by his knowledge (or lack of knowledge) about the problem domain. Thus the main question to adress is what changes take place in the knowledqe base of a novice becominq an expert.

The domain chosen is that of elementary thermodynamics. The semantic richness of this domain does not allow the observation of subjects durinq the entire skill acquisition process from novice toward expert level. The methodoloqy adopted instead is to observe subjects at two different levels of performance: novice level and intermediate level, and to simulate this behaviour with a computational model for each of these levels.

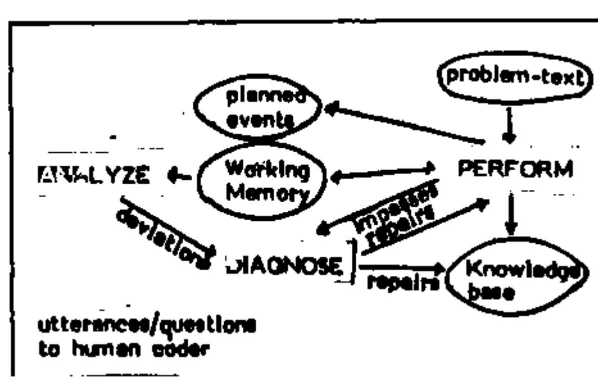
Since the problem solvinq behaviour of human subjects, novices in particular, is idiosyncratic and unpredictable we do not aim at predictinq all behaviour in detail, but adopt the strategy of semi-automatic simulation (Bhaskar and Simon, 1977): whenever the model fails to simulate the behaviour of the subject, it can be put back on the right track by a human coder. In this way protocols of subjects at different levels of expertise can be analyzed in terms of the computational model and annotated with discrepancies between model and subject.

* Netherlands Organization for the Advancement of Pure Research (ZWO).

2. Description of the system

The PDP system consist of three components. The problem solving component (PERFORM) is implemented using a locally developed (InterLisp) implementation of KL-ONE (Brachman, 1978). In contrast with Bhaskar and Simon (1977), the PERFORM component actually solves the problem. The KL-ONE network contains both the declarative knowledqe about the domain, and procedural knowledqe represented as concepts with attached demon-like Lisp-procedures. Procedural knowledqe is activated by a scheduler usinq a multi-level aqenda similar to KRL. All operations on the KL-ONE network (e.q. fillinq slots, individuatinq concepts), generate "events" which are sent to a second component of the system: ANALYZE. ANALYZE interprets the event with a production system and decides whether the event will be translated into natural language utterances. These utterances are presented to a human coder for comparison with a subject's protocol. If there is a match between an event and part of the protocol, the line numbers of that part are typed in by the coder and a signal is sent to PERFORM to continue. If the behaviour of the program and the subject do not match, a third component of PDP is activated: DIAGNOSE.

The DIAGNOSE component is equipped with rules representing knowledge about deviations from the normal line of behaviour of the model (e.g. errors and impasses). In interaction with the coder DIAGNOSE can identify the problem and redirect the behaviour of the model (e.g. by changing knowledge structures in the working memory or by reactivating certain procedures).



Overview of PDP

The Output of the model is an analysis of the protocol in terms of the procedural and declarative knowledge used by the model. Discrepancies between model and protocol are also indicated.

3. Intermediate level model

As a starting point for the intermediate level model, an instructional method developed for an undergraduate level physics course was used, which prescribes the actions and methods needed to solve Droblems in thermodynamics (Mettes et al.,1981). The procedural knowledge of the model consists of three main procedures: ORIENTATE, SOLVE and EVALUATE. The task of ORIENTATE is to read the problem and to analyze the problem in terms of concepts relevant to the physics domain. The output of ORIENTATE is a transformed representation of the problem suitable for use in the solving phase.

SOLVE implements a backwards search for a seguence of eguations which will solve the problem. When such a seguence is found, numeric values are filled in the eguations and the answer is computed.
EVALUATE currently only implemented in a rudimentary form- checks the consistency and plausibility of the answer against gualitative expectations etc. More details of the intermediate level model are given in Jansweijer et al. (1982).

With the intermediate level version of POP we analysed 17 protocols of a second year physics student. More than 80% of the protocol statements could be matched with corresponding activities of the model. In general, the model and the behaviour of the subject were in concordance with respect to the order in wich different procedures were activated. The table below shows the relative number of activities performed in the three main procedures of the model, compared to the relative number of activities actually identified in the protocols.

	percentage of activities in: ORIEN-TATE	SOLVE	EVAL-UATE	Abs. total number of activities
Model	52%	42%	6%	298
Subject	46%	44%	10%	169

The data indicate that the relative amount of processing performed in different phases of the problem solving proces, is accurately predicted. This is the more remarkable since the model puts much more emphasis on the orientation phase than most previous models of physics problem solving. The MECHO program (Bundy et al. 1979; Luger, 1981) for instance starts to select a physical principle and makes most of the necessary inferences and transformations when particular pieces of information are needed. The ABLE model (Larkin 1980) also concentrates to a large extent on the ways in which principles are selected. Our model is more similar to the ISAAC program (Novak, 1977) in its emphasis on transformation of the problem representation before actual principles are selected.

As can be seen from the above table the model performs almost twice as many activities as are present in the protocols. One reason for this is that the subject does not verbalize all of his activities in the same detail as PDP does. Another, more important, reason for this phenomenon is that, during the course of the experimental session, the subject learns from his own problem solving efforts: the type of problem is recognized, certain operations become automatic, analysis of problem features becomes more goal oriented, seemingly guided by problem specific schemata. This learning results in a smaller number of (verbalized) activities, in particular during the orientation process. The model however is not eguipped with any learning capabilities. Quantitativily this can be illustrated by comparing the results for a series of problems occuring early in the sessions and an isomorphic series presented at a later stage. In the late problems we saw only half as much ORIENTATION activity as in the early isomorphic problems. For SOLVE the difference is less: 85 % of the activities remain present.

Some activities of the subject could not be coded in terms of the model. These are: recognition of problems as similar tn previously solved ones and checkinq of dimensions.

4. Novice level model

Analysis of novice protocols, with the intermediate version of PDP showed us that novices behave quite differently: the order in which novices perform the various activities deviates strongly from the model. In particular the Orientation process is only present in a rudimentary form, resultinq in an incomplete analysis of the problem. This causes problems durinq later staqes of the problem solvinq proces. The subject qets into "impasses" and needs to perform problem solvinq activities to overcome the impasse.

On the basis of these qeneral observations a first model at novice level was constructed. It differs from the model at intermediate level in two ways. First, a number of procedures are deleted from the intermediate model, in particular those related to transformation of the problem representation. Second, a number of repair procedures were added to the DIAGNOSE component of the model (c.f. Brown & vanLehn, 1980). When the problem solving proces gets into an impasse, DIAGNOSE tries to classify the cause of the impasse and suggests one or more repairs. The human coder selects the applicable repair by inspecting the novice protocol.

For example, when a principle is selected for which the preconditions are not satisfied, the system gets into an impasse. Two possible repairs are then suggested: "reject principle" and "relaxation". When the coder selects the first repair, the principle is rejected and another principle is selected. When the coder selects the second repair, continuation of the proces with the selected and possibly wrong principle is forced. Another typical example of an impasse is when the system ends up with a filled eguation in which there is no unknown left to be computed. In such a case the repairs "start anew" and "reject principle" are suggested.

A summarized example of an output-trace of PDP for a novice protocol is shown below. The uppercase words denote the procedures of the model.

```
RESEARCH-DIRECTOR
  ORIENTATE
    READ-PROBLEM
      ABSTRACT-CUES
  SOLVE
    SELECT-PRINCIPLE
      IMPASSE
      (precondition not satisfied)
        REPAIR
          RELAXATION
  COMPUTE
    FILLING-EQUATION
      TRANSFORM-UNITS
        IMPASSE
        (transformation of units failed)
          REPAIR
            IGNORE-UNITS
    CALCULATE
  EVALUATE
```

The first activity of the model is to read the problem and abstract relevant cues from the problem text. On the basis of these cues, a principle is selected. Because the precondition of this principle is not met, the system gets into an impasse. Since the subject iqnores this precondition the impasse is repaired by relaxation of the precondition. The values of the variables are now filled in the equation. The procedure which transforms units fails, due to lack of knowledge and the resultinq impasse is repaired by iqnoring the units altogether. Finally the asked is computed and the solution is evaluated. The figure above represents problem-solving behaviour typical for a novice, in which the orientation process is only present in a rudimentary form.

5. Discussion

The intermediate level model fits the behaviour of experienced subjects quite well, indicating that subjects with good knowledge of physics and experience in solving this type of problem, are able to work systematically and do perform an extensive analysis of the problem before selecting eguations.

The model for novice behaviour, based on the assumption that lack of procedural and declarative knowledge causes impasses which are resolved by local repairs produces behaviour which is typical for novice protocols. A substantial number of protocols have been analyzed using this model. Although the model accounts for a large percentage of protocol utterances, a number of repairs had to be introduced in a rather ad hoc way. Also, non-local repairs were observed in the protocols (e.g. reactivation of the orientation proces when the solve proces gets stuck), which are not easy to model in this way. In contrast with Brown & vanLehn (1980), we have not been able to identify and implement a parsimoneous repair strategy.

Currently a new version of the novice model is under construction, which is controlled by goals embodied in a global plan. In this system the DIAGNOSE component will be able to generate subgoals, whenever an impasse *occurs.* In this way, we expect to gain flexibility in control over repairs.

REFERENCES

Bhaskar, R. and Simon, H.A. "Problem solving in semantically rich domains: an example from engeneering thermodynamics." Cognitive Science, 1 (1977) 216.

Brachman, R.J. "A structural paradigm for representing knowledge." BBN report 3605 (1978).

Brown, J.S. and vanLehn, K. "Repair Theory: A Generative Theory of Bugs in Procedural Skills." Cognitive Science, 2 (1980) 155- 192.

Bundy, A.; Byrd, L.; Luger,G.; Mellish, C; Milne, R. and Palmer, M. "MECHO: a program to solve mechanics problems." Working paper, Department of Artificial Intelligence, University of Edinburgh U.K. (1979).

Jansweijer, W.N.H.; Konst, L.: Elshout, J. and Wielinga, B. "PDP: a protocol diagnostic program for problem solvinq in physics." In Proc. ECAI-82. Orsay, France, July, 1982, pp. 278-280.

Larkin, J.H.; McDermott, J.; Simon, D.P. and Simon, "Models of Competence in solving physics problems." Cognitive Science, 4 (1980) 317-345.

Luger, G.F. "Mathematical model building in the solution of mechanics problems." Cognitive Science, 5 (1981) 55-77.

Mettes, C.T.C.W.; Pilot, A. and Roossink, H.J. "Linking factual and procedural knowledge in solving science problems: A case study in a thermodynamics course." Instructional Science, 10 (1981) 333-361.

Novak, G.S. "Computer understanding of physics problems stated in natural language." In Proc. IJCAI-77. Intelligence, 286-291, Cambridge, USA, 1977, pp. 286-291.