

# Two-level learner modelling in the tutoring of declarative knowledge based problem solving

*Lianjing Huang*

*Pierre Bonzon*

*University of Lausanne  
Switzerland*

## ABSTRACT

This paper discusses two different kinds of problem solving: procedural skill based problem solving and declarative knowledge based problem solving. An intelligent learning environment has been implemented for tutoring declarative knowledge based problem solving. The environment is built around two levels of learner modelling: shallow modelling and deep modelling. The first relates an error of the learner with a misconception. The second verifies the learner's acquisition of various factual knowledge or concepts needed, and helps the learner to solve a concrete problem. A learning environment for Accountancy which serves as a case study, has been implemented in Smalltalk V for OS2.

*Main conference themes:* artificial intelligence, software, tutoring

*Educational areas:* higher education

*Study topics:* business/commerce

*Secondary keywords:* cognition, modelling, problem solving

## INTRODUCTION

One of the most successful problem solving models in Intelligent Tutoring Systems (ITS) is that of ACT\* tutors developed by Anderson and his coworkers at Carnegie-Mellon University [1]. For most existing intelligent tutoring systems the philosophy behind the design is the same as that of ACT\* tutors which is decomposition of the over-all problem solving skill into subskills (or subgoals, or subplans, or production rules or steps according to the different contexts). However, as questioned indirectly by Glaser and Bassok [2, p. 639], this approach is only appropriate for tutoring elementary skills in well structured tasks such as the syntax and semantics of a new program language or proof finding in elementary geometry. It may not be appropriate for problem solving which needs declarative knowledge. Furthermore, in the view of Ohlsson we have reached a point where we know almost all there is to know on the acquisition of procedural skill, but very little on the learning of “conceptual, declarative, theoretical, abstract knowledge” [3].

This paper defines a kind of problem solving which needs the consideration of declarative knowledge instead of procedural skills, and proposes an intelligent learning environment which engages the learner in acquiring the concepts and declarative knowledge through exercises.

## PROCEDURAL VERSUS DECLARATIVE PROBLEM SOLVING

Greeno and Patel [4] identified three general kinds of problems in mathematics:

- problems which are solved by applying procedures;
- problems which are solved primarily through search or planning;
- problems which depend primarily on representation.

Unfortunately no further analysis was given in terms of the differences and special characteristics of these problems. For the first two types of problems procedural skills are needed in solving them, but solving of the third type of problem depends on the understanding and representation of declarative knowledge.

### **Procedural skill based problem solving**

The following are characteristic elements in the tutoring of procedural skill based problem solving:

*Decomposition:* The problem can be decomposed into subproblems. Or in ACT terms, a goal of the problem solving can be decomposed into subgoals.

*Intermediate units of problem solving:* These subproblems are the independent units which apart from order of preference (which we will discuss in the following point) can be perceived by the learner as independent and can become compiled individual units of knowledge.

*Order of preference:* These subproblems follow a (or several) order(s) of preference, that is one must solve the problem step by step. Or, in ACT terms one must apply a (or several) rule(s) before one can apply some other rules.

*Importance of order of preference:* The order of preference is important both for problem solving and for the pedagogical objective. These order makes manifest corresponding plans which the learner should follow to correctly solve the problem. Follow the order of preference is therefore also a procedural skill which the learner should acquire. In a functional model of problem solving [5] the learner has more freedom in following a certain order instead of the others. Errors result from the learner confusing the order of preference (the learner is either pursuing a nonapplicable goal or a goal which is applicable, but is not the best goal according to the control knowledge, a kind of metalevel knowledge). We see that preference orders are problem solving skills necessary for the learner. These subproblems or subgoals are perceptible units or primitive subskills [6], that is the system is able to diagnose the learner at the granularity of each sub-goal.

### **Declarative knowledge based problem solving**

The following are the characteristic elements of tutoring declarative knowledge based problem solving:

*Decomposition:* The problem must be perceived from different aspects or decomposed into different aspects. These different aspects represent factual knowledge, concepts and principles which are essential for solving the problem. The success of solving the problem depends on one's understanding (or beliefs) of these different aspects. The procedural skills or subskills as well as their preference order are not or are less important in this problem solving.

*Intermediate units of problem solving:* Intermediate units are various concrete aspects of the problem. For example, to find a pair of correct financial journal entries for an accounting transaction, one has to know the following aspects of the problem. The agent who exercises the transactions, the operations which the agent has executed, the object which has been used by the agent to execute an operation, the duration of operation and the value of the object. Note that here agent, operation, object and duration are not terms in Accounting. These are used as a means to represent the necessary aspects of the problem. The learner may use other means to achieve the same goal. The system can suggest

a means to the learner so as to help concentration on the essential aspects while solving a problem. More importantly, through explicit representation of the essential aspects of the problem the system is able to detect the errors which would be otherwise hidden in the learner's cognitive process of problem solving.

*There is no order of preference among intermediate units:* These aspects or intermediate units need not necessarily follow a certain order (although these could). The learner can 'attack' or perceive some aspects of the problem and neglect others. The learner normally attacks the problem by recalling previous problem solving experiences and available knowledge. It is very difficult for the machine to know exactly which aspect of the problem is the learner's current interest. We can nevertheless simply ask the learner, as suggested in a different context by McCalla [7].

These aspects are perceptible units in that the system instructs and models the learner at this level of granularity. Because the order is not important, the learner need not be confined to a certain problem solving path: the learner can attack the problem at will.

## SHALLOW MODELLING

We have argued that in order to solve a problem one needs to consider various declarative aspects of the problem. In the real world, however, learners (especially advanced learners) sometimes solve a problem by using a short cut or even pattern matching to a previous problem without giving explicit consideration to all the declarative aspects of the problem. This happens, we would suggest, because of the following two reasons.

First, the essential aspects of the problem have been internalized or **compiled** by the learner after a series of similar problem solving activities. These internalized aspects are **abstracted** into a single or a few shallow chunks or models. Here **shallow** means that the learner can retrieve the shallow models without **deep** cognitive process and without consideration of the details of the models (these are abstracted away). We do not intend to explain how these shallow models are formed (it is actually an open research issue in cognitive science). The task of shallow modelling in the intelligent tutoring system is to detect the incorrect use of correct shallow models and the correct use of incorrect shallow models (below we will give some examples in Accounting tutoring).

Second, the previous experience of solving a similar problem helps the learner to come up with a problem solving **pattern** which is easily recalled by

the learner without resorting to the concrete aspects of the problem. Again, we call this pattern a **shallow** model for the similar reasons as we have stated above.

---

<b>Bug 01</b>	
<i>Shallow model:</i>	Learner confuses the direction, i.e., considering debit account as credit account and vice versa.
<i>Bug recognizer:</i>	input pattern (D,C, _); output Pattern (C,D,_).
<i>Remedy:</i>	tell the learner that a direction mistake has been made and remind the learner of the double-entry book keeping conventions.

---

**Fig. 1** Direction bug

---

<b>Bug 02</b>	
<i>Shallow Model:</i>	learner thinks that whenever the company in question pays for or buys something, the account a <b>kind of Expenses</b> should be debited
<i>Bug recognizer:</i>	input pattern (a <b>kind of Fixed Assets</b> , _, _); output pattern (a <b>Kind of Expenses</b> , _, _).
<i>Remedy:</i>	tell the learner that paying for or buying something does not necessarily mean an expense to the company. When the company buys something like a car, equipment, or buildings etc., the company actually does investments for its future economical activities.

---

**Fig. 2** Investments as expenses bug

Now look at the above two shallow models which were implemented in our Accounting Learning Environment. Note that the shallow models are used to explain the bugs (or errors which the learner makes). These examples illustrate what we have said above. For example, Bug 01 represents the direction mistake. This mistake could be accounted for by many deep models. Here we only assume an **incorrect** shallow model—the learner just does not understand double-entry book keeping conventions.

Bug 02 assumes that the learner has a **correct** version of a shallow model which enables the learner to correctly find a journal entry for the accounting transactions such as paying for the salaries and electricity, and buying goods etc. Therefore, when the learner is asked to find the journal entries for the accounting transactions like buying a car, equipment or a building etc., he (or she) will inappropriately use this correct shallow model.

The declarative bugs discussed above are different from traditional procedural bugs which are usually procedural or localized errors rather than

deep, pervasive misconceptions [6, p. 153]. The declarative bugs are however somewhat declarative, global, pervasive and shallow misconceptions or models.

In conclusion, shallow models are used to detect abstract and compiled misconceptions during learner's problem solving endeavours. The deep models are used to detect learner's errors in concrete, detailed and **uncompiled** aspects or **attacking points** of the problem.

## DEEP MODELLING

Deep modelling is a means to monitor learner's understanding or awareness of the declarative aspects of the problem. The distinction between declarative and procedural knowledge has long been a research issue in Artificial Intelligence and Cognitive Science [8, 9, 10]. Some fundamental questions still remain unresolved. Among these are questions like what is the nature and function of declarative knowledge, how is declarative knowledge acquired, what methods for task analysis and what methods for cognitive diagnosis [9]. This paper does not intend to answer all these questions. It is first of all our intention to identify a kind of problem solving which needs more declarative knowledge than procedural skills. Many problems found in Philosophy, History, and Accounting, etc. are examples of this kind.

Deep modelling is the process of getting a learner's representation of one or several intermediate units of problem solving by asking explanatory questions. A short critique on cognitive diagnosis by asking explanatory questions was given by Ohlsson [9, p.92]. From the point of view of the present paper there are three advantages of deep modelling through explanation questions:

- It helps the learner concentrate on the necessary aspects of the problem (in our Smalltalk implementation of Accounting learning environment, all declarative aspects of a current problem are listed on a list pane, and necessary details and explanations of every aspect are displayed in a text pane);
- It helps the learner make explicit what is otherwise implicit in problem solving;
- It provides a means for the system to monitor learner's understanding or awareness of the declarative aspects of the problem.

### **The differences between shallow modelling and deep modelling**

Shallow models are:

- internalized or compiled models(or existing models);
- abstract and declarative models;
- explicitly used by the learner;
- global, pervasive misconceptions.

Deep models are:

- not internalized or uncompiled models;
- concrete and declarative models;
- implicitly used by the learner (deep modelling helps the learner make explicit what is otherwise implicit or hidden by the learner);
- local to a problem.

### **IMPLEMENTATION OF AN EXAMPLE IN ACCOUNTING**

Below follow the design principles for the implementation of a learning environment in Accounting which engages the learner in a declarative knowledge based problem solving context:

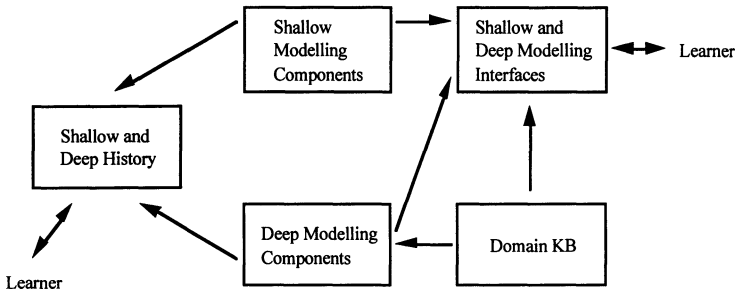
- When the learner makes a mistake, the system proposes first shallow modelling and then deep modelling (if shallow modelling fails).
- Shallow modelling uses a predefined bug catalogue.
- Deep modelling displays explicitly all declarative aspects of the problem and lets the learner choose one aspect to further the dialogue.
- The system provides an opportunistic (or mixed initiative) interface to the learner, which means that the learner is allowed to choose between shallow and deep modelling.

On these principles a system called IALE (Intelligent Accounting Learning Environment) has been implemented in Smalltalk V for OS/2.

### **Over-all architecture of IALE**

The following figure shows an overall architecture of IALE. There are five principal components. The domain knowledge base is an expert module in Accounting. Shallow modelling components are composed of three subcomponents for accomplishing shallow modelling task. The first subcomponent is called Bug Detector which matches learner's input with a predefined bug catalogue. The second subcomponent is called Remedy Component which provides the remedial information to the learner. The third

subcomponent is called Learner's Shallow Model which keeps the errors (at shallow level) which the learner made during all previous sessions.



**Fig. 3** The overall architecture of IALE

Deep modelling components are composed of six cognitive tools [11] and a structure for keeping learner's deep models. The rationale and the detailed functionality of the cognitive tools will be discussed in another paper. Here, we will give a brief description of each tool.

First, the **Ask Tool** asks the learner questions. Second, the **Verify Tool** verifies if the learner's answer is correct and then gives feedback. These two tools are in fact used by the system, not by the learner. Thus, these are not cognitive tools in the strict sense. The third is **Why Tool** which allows the learner to ask the why question on the current aspect of the problem. Fourth, the **Test Tool** allows learners to generate and test hypotheses in the context of problem solving. Fifth, the **View Tool** allows the learner to view his own performance. And sixth, the **Control Tool** allows the learner to control the system going back and forth during problem solving.

Shallow and deep modelling interfaces are the interfaces between the learner and the system. Shallow and deep history is the information which combines the results of deep and shallow modelling.

### **Specific problem solving in Accounting**

Specific problem solving in Accounting (especially finding journal entries) is single step, i.e. there are no explicit intermediate steps. The solution is a pair of journal entries; declarative aspects of the problem are either considered explicitly or implicitly. Implicit cognitive process should be made explicit through deep modelling.

We present an example of deep modelling in IALE inspired by the previous accounting tutoring system developed at University of Lausanne [12].



**Example**

Enterprise John has sold in cash hot chestnuts for a total of \$ 302.

The deep model for this example includes the following aspects:

- The **type** of this transaction is called ‘buying & selling’.
- The enterprise (**agent**) under consideration (John) is seller.
- Two **operations** related to this transaction are **acquisition** of an asset and **production**.
- The agent (John) acquires the **object** (cash) through the operation acquisition
- The **duration** of the object (cash) is immediate and the **value** of the object is \$ 302
- The agent (John) realizes the object (final\_product) through the operation production
- The duration of the object (final\_product) is not important in accounting and the value of the object is \$ 302.

The following shows an example of how the IALE system interacts with the learner (the real scenario includes the windows, command buttons, list panes and text panes).

---

*IALE:* What is the type of the current transaction? (Ask Tool)  
*Learner:* buying & selling  
*IALE:* That is right. Now tell me what kind of operation is related to debit account. (Verify Tool)  
*Learner:* Production  
*IALE:* That is not right. The correct answer should be acquisition.  
*Learner:* Why? (Why Tool)  
*IALE:* Because for debit aspect of the current transaction, the Enterprise John has acquired the cash which is a kind of asset.  
*Learner:* IF I choose operation as an acquisition, object as a Bank note, what account should be credited or debited? (Test Tool)  
*IALE:* The account Bank\_Note should be debited.  
*Learner:* I want to view my own problem solving trace (View Tool)  
*IALE:* You have chosen ‘buying & selling’ as the type of current transaction, ‘production’ as the operation for the debit account.  
*Learner:* ....

---

**Fig. 4** A textual summary of an example showing how the IALE system works

## REFERENCES

1. Anderson, J. R., Boyle, C. F., Corbett, A.T., and Lewis, M.W. (1990) *Cognitive modelling and intelligent tutoring*. Artificial Intelligence. **42**, pp. 7-49.
2. Glaser, R. and Bassok, M. (1989) *Learning theory and the study of instruction*. Annual Review of Psychology, **40**, pp. 631-666.
3. Sandberg, J. and Barnard, Y. (1991) *Interviews on AI and Education: Allan Collins and Stellan Ohlsson*. AICOM Vol. **4** (4).
4. Greeno, G. J. and Patel, V. L. (1991) *A view from medicine*, in Smith, M. U. (ed.), *Toward a Unified Theory of Problem Solving: View from the content domains*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 35-44.
5. Johnson, K. A. (1993) *Exploiting a Functional Model of Problem Solving for Error Detection in Tutoring*. PhD, The Ohio State University.
6. Wenger, E. (1987) *Artificial Intelligence and Tutoring Systems*. Los Altos: Morgan Kaufmann.
7. McCalla, G. I. (1992) *The Central Importance of Student Modelling to Intelligent Tutoring*, in Costa, E. (ed.), *New Directions for Intelligent Tutoring Systems*, NATO ASI Series Vol. **91**, Springer-Verlag
8. Boley, H., Finin, T., Moss, C., Richter, M. M. and Voronkov, A. (1991) *Declarative and Procedural Paradigms—Do they Really Compete?* In Richter, M. M. and Boley, H. (eds). PDK91. Proceedings of the International Workshop on Processing Declarative Knowledge. Kaiserslautern, Germany.
9. Ohlsson, S. (1992) *Towards intelligent tutoring systems that teach knowledge rather than skills: five research questions*. In Scanlon, E. & O'shea, T. (Eds) *New directions in Educational Technology*, NATO ASI Series Vol. **96**, Springer-Verlag, pp. 72-96.
10. Winograd, T. (1975) *Frame representations and the declarative/procedural controversy*. In Daniel G. Bobrow and Allan Collins (eds.). *Representation and Understanding—Studies in Cognitive Science*. Academic Press, pp. 185-210.

11. Lajoie, S. P. (1990) *Computer environment as cognitive tools for enhancing mental models*. 1990 Annual meeting of the American Educational Research Association, Boston.
12. Bonzon, P., Apothéloz, B., Zucchinetti, A. and Forte, E. (1991) *A knowledge based system for teaching financial accounting*. PEG91, Genova, Italy.