A Cognitive Student Model – An Ontological Approach

L.Y., Tu, W. L., Hsu, S. H., Wu
{masia, hsu, shwu}@iis.sinica.edu.tw
IIS, Academia Sinica, Taiwan

Abstract

We present an ontological approach to the design of the student model for a tutorial agent system (TAS). Our model emphasizes the classification and detection of error types. If the student has any systematic and predictable misconceptions, the system attempts to determine the underlying reasons for such errors. We adopt the "Identification, Simulation, Interaction, and Mapping" (ISIM) strategy to achieve this goal. The tutorial agent system first identifies which problem solving method a student is using. It then simulates the procedure in a step-by-step fashion. If there is any ambiguity in the diagnosis of error types during the simulation, the system will interact with the student to resolve it. Finally, the interaction will lead to appropriate error types. The related knowledge is constructed in an ontological framework, InfoMap. In this paper, we shall focus on how to construct the knowledge and how the simulation works.

Key words: Student Model, ontological approach, primary mathematics

A Cognitive Student Model – An Ontological Approach

Abstract

We present an ontological approach to the design of the student model for a tutorial agent system (TAS). Our model emphasizes the classification and detection of error types. If the student has any systematic and predictable misconceptions, the system attempts to determine the underlying reasons for such errors. We adopt the "Identification, Simulation, Interaction, and Mapping" (ISIM) strategy to achieve this goal. The tutorial agent system first identifies which problem solving method a student is using. It then simulates the procedure in a step-by-step fashion. If there is any ambiguity in the diagnosis of error types during the simulation, the system will interact with the student to resolve it. Finally, the interaction will lead to appropriate error types. The related knowledge is constructed in an ontological framework, InfoMap. In this paper, we shall focus on how to construct the knowledge and how the simulation works.

1. Introduction

An ideal intelligent tutoring system (ITS) should be able to diagnose the student's errors and provide personalized instruction. In this respect, a human teacher usually does not have the time and patience to render such an instruction. We believe the main benefit of ITS is to help students to have more practice after class.

In general, ITS consists of four parts: expert system, student model, pedagogical module and interface. An expert system models the ideal student's behavior [1,11]. Similar to [13], we regard the student model as the system's belief about the student's knowledge. An ideal student model is just one of the methods that could be adopted in problem solving. Each method the student uses, reveals some important information to us, such as the student's ability and possible misconceptions. Whether the student solves a problem efficiently (as an expert) is not a major criterion for evaluating the student's method of problem solving. Thus, in designing our TAS, we emphasize the student model rather than the expert system.

Due to pedagogic aim, our student model not only detects a student's incorrect answer, but also the underlying cognitive reasons for such errors. Those errors due to systematic and predictable misconceptions by the student should be identified and corrected in the tutoring process. We propose a process called Identification, Simulation, Interaction and Mapping schema (ISIM) to determine the errors.

When the tutorial agent detects an incorrect answer, it will switch to the diagnosis mode. The system first identifies the student's problem solving method. It then simulates the procedure of the method step by step. If there is any ambiguity with regard to possible misconceptions during simulation, the system will interact with the student to resolve it. Finally, the student's mistakes will be mapped to appropriate error types.

2. The Scope of Student Model

When constructing a student model, there are three main questions to be asked: Who is the learner? What is the subject to be taught? And, how could the (subject) knowledge be constructed?[1]

In this paper, the learners are pupils in primary schools. Our subject is mathematical word problems in elementary schools, and the word problems are confined to additive or subtractive problems. The knowledge is collected from the errors that pupils made.

Mathematical word problems involve procedural knowledge. In the process of solving a problem, the reasons for student errors can be divided into three groups:

i. Carelessness;
ii. Systematic and predictable misconceptions;
iii. Random errors.

The systematic misconception is our main focus because we believe that "there exists a procedure in student's mind that produces the erroneous answers [4]." Diagnosing students' misconceptions is the goal of our student model. Therefore, it is important to collect the knowledge of error types.[2] Once the

[1] According to [13], the "who", the "what", the "how" and the "why" are used in defining a student model.

[2] We cooperate with the psychologist Prof. C. W. Hu of National Taiwan University, who collects

misconception is diagnosed, a better tutoring becomes possible.

In general, there could be more than one error type. For instance[3], in making the error: *173 – 48 = 135*, it might be that the student

i. ignores position, i.e. the student always uses the bigger digit to subtract the smaller digit;
ii. forgets borrowing.

To identify the exact error type, it is important for a student model to simulate the problem solving process of the learner in a step-by-step fashion.

In solving a mathematical word problem, a student needs the following three kinds of knowledge to understand the problem, which must be constructed in the student model:

i. Domain knowledge: all the necessary and unnecessary steps which some students will follow in solving the problem
ii. Error classification knowledge: Error types that students may make.
iii. World knowledge: Common sense and lexical knowledge

3. Error Type Analysis – A Way to Personalization

ITS provides "a personal training assistant for each learner" [8]. This implies that individualization is an important criterion to evaluate an ITS. Personalization involves adaptive teaching after suitable student classification. In this section, we focus on the criteria for such classification.

To personalize TAS, the first step is to identify a student's need. Error type analysis provides valuable information about what the student is lacking. Our system will follow a student's problem solving procedure in order to know exactly in which step the student made a mistake.

Table 1 shows the correspondence between the procedure of column addition and the error types.[4] Based on this, we construct the domain knowledge in Figure 3.

Table 1. The correspondence between procedures and error types of addition.

	Procedure of column addition	**Observation point**	**Possible error types**	**Examples of incorrect answers**
1	Checking positions of digits	Finding $T_1 + B_1$, $T_2 + B_2$, etc.	*Adding each digit directly*	*256+127=23* *2+5+6+1+2+7=23*
			Catenating each digit	*256+127=256127*
2	Calculating $T1 + B1$	If $T_1 + B_1 < 10$	Multi-carrying, calculating ($T_2 + B_2$) + 1	*56+22=88* *where* *5+2+1=8 and 6+2=8*
		If $T_1 + B_1 \geq 10$	Ignoring carrying, calculating ($T_2 + B_2$)	*56+27=73* *where* *5+2=7 and 6+7=3*
			Multi-carrying, calculating $(T_2 + B_2 + 1) + 1$	*56+27=83* *where* *5+2+1=7 and 6+7=13*
			Carrying into a wrong place, calculating $(T_3 + B_3) + 1$	*256+127=473* *where* *2+1+1=4, 5+2=7 and 6+7=103*
3	Writing down the answer		Writing the answer in a wrong position	*25+10=350*
			Forgetting or ignoring to write down the answer	*125+211=blank*
4	Keeping on calculating		Stopping calculation	*125+211= 6*
The other procedures are similar and omitted here.				

We represent each digit by variables T_i and B_i and represent the 3-digit addition problem as "*T3T2T1+B3B2B1*".

By following the student's procedure, our system would recognize the student's problem solving

---

incorrect answers by students and analyzes the error types.

[3] This instance is cf. [3], p. 163.

[4] The data of table 1 comes from Prof. Hu, who covers all the error patterns that students made in his survey of more than 100 students.

method. Based on this information, the system can then evaluate the ability of the student. For instance, there are two students solving the problem, "*17+123*", by two different strategies: the first one starts from *17*, and adds *10* repeatedly, i.e., *17 + 10 = 27, 27 + 10 = 37*, etc., *12* times. Finally, he adds *3* to *137*. The other one uses a similar strategy, but starts from *123,* adds 10, and then adds 7. Hence, we know that the two students have different levels of understanding about the problem. Since different methods give rise to different errors, identifying them would greatly simplify the task of error type disambiguation.

4. Information Map – An Ontological Approach

"Ontology is the way we carve up reality in order to understand and process it [6]." To us, "reality" is the data of a student's performance. In our ontological approach, we collect, from student data, the knowledge necessary for problem solving as the domain ontology. It includes various knowledge the student possesses such as: What the student knows about the world, what technique the student uses, and what error the student usually makes.

Our knowledge representation scheme, InfoMap, consists of a knowledge editor for the construction of domain ontology. The design of the InfoMap scheme is meant to facilitate both human browsing and computer processing of the knowledge structure in the computer system. Based on the InfoMap, various applications can be built such as: question answering, text-to-speech conversion, speech recognition and intelligent tutoring systems. For each application, based on the knowledge in the InfoMap, extracting concept structures such as the topic and context provides the understanding of a sentence.

We use the phrase "concept structure" to distinguish it from simple concepts such as a noun or a verb in a dictionary. A concept structure can refer to a pattern composed of several simple concepts together with certain relationships among these concepts. It can also refer to a complex procedural concept such as the concepts of "addition", "multiplication" (for the purpose of teaching), etc. that are beyond the descriptive power obtained from the combination of simple concepts. Examples of concept structures range from simple concepts such as a word, a phrase or an event to more complex ones such as a sentence, a paragraph, a script (a collection of related events), or a story, that can describe different subjects such as "the passive tense of English", "a mathematical theorem", "a theory in physics" etc.[15]

The following figures show the knowledge we have constructed. Figure 1 shows the three kinds of knowledge that we mentioned above. Error types are knowledge of students' possible incorrect answers, and they will correspond to misconceptions. Under the node "word problem" the steps of problem solving are described. Figure 2 displays an instance of world knowledge, which is common sense. It is unnecessary to collect all the common sense of mankind, but we have to collect that which an elementary student has to possess to deal with word problems. For example, a student learns that "cars", "trucks", and "buses" are instances of "transportation". Also, he knows that passengers get on a bus, or that a car may accommodate three or four people. Actually, world knowledge provides the system not only with common sense, but also with some key "noun-verb" concept pairs to understand a sentence. There is another type of important information (an attribute), namely quantifiers which can't be ignored, since some quantifiers provide hints at what possible objects might follow in a sentence in mandarin.

One advantage of the ontological approach is that knowledge in InfoMap is easy to maintain. It is convenient to insert new items, delete inappropriate items, and construct interconnections between items. Furthermore, a piece knowledge constructed in one place can be referred to in many other places and does not need to be duplicated. The built-in inference scheme suffices to produce effective conclusions given that sufficient knowledge is available.

Our TAS diagnoses errors through the ISIM schema, which is described below:

i. Identification: There is more than one strategy a student can choose to solve the problem, and each strategy has a different meaning to the system. There are some methods such as "counting" or "decomposing and combining". To identify the method the student is using the system needs to use the knowledge of simulation. Consider the problem "*15 + 28*". The student can use "counting" to solve as follows *15 + 10 = 25, 25 + 10 = 35, 35 + 8= 43*; if the student uses the "decomposing and combining" method, he will decompose the numbers first, namely 15 = 10 + 5, 28 = 20 + 8, and then combine them as follows: 10 + 20 = 30, 5 + 8 = 13, 30 + 13 = 43. There are distinct patterns to discriminate these two different methods and the pattern knowledge will be described in InfoMap.

ii. Simulation: In a step-by-step fashion, the system attempts to simulate the learner's knowledge of mathematical problem solving. In a previous example, the student gives the answer 135 to the problem "173-48". After determining that the student applies the "calculating in column" method, the system will simulate the procedure as follows: 3-8 = 5 (borrowing = 1), (7 - 1) – 4 = 2 and search for associated error patterns. It will then discover that the student's problem lies in ignoring the borrowing of 1 from the tens position.

iii. Interaction: The system will interact with the learner when there is an ambiguity in the system's analysis. For example, when there are more than one possible error types in a certain step, the system can ask another question to make sure which error type is appropriate. Suppose the system gives a second problem "146-38". If the student's error type is "using the bigger number minus the smaller one and ignoring the position", then the answer would be 112; if the student forgets he has borrowed a 1 in the last step, the answer would be 118. Also, the system can choose other interaction strategies as suggested by educational psychologists.

iv. Mapping: Following simulation and interaction, the system would determine a learner's misconceptions.

Figure1. Necessary knowledge that the system need

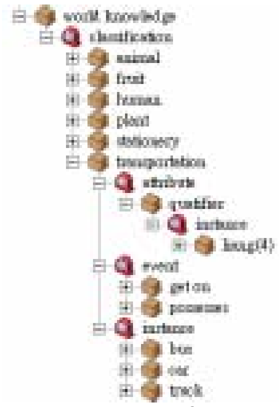

Figure 2. An instance of world knowledge

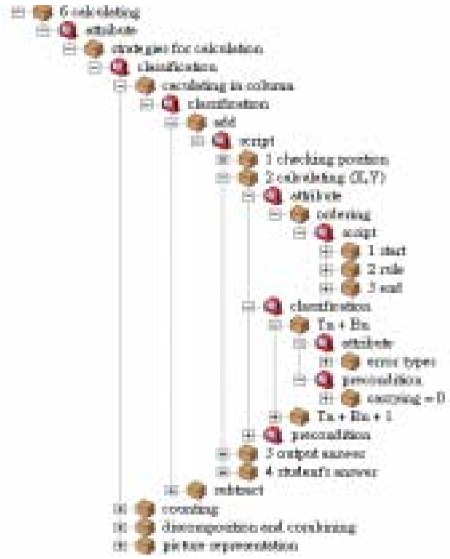

Figure 3. An instance of knowledge of problem solving

The ISIM schema suggests a flow chart for evaluating learners' mathematical abilities and prepares tutoring instructions through the pedagogical module. The necessary knowledge and inference mechanisms are all represented in InfoMap. In the next section, we shall illustrate how InfoMap uses related knowledge to simulate the student's problem solving methods.

格式化: 項目符號及編號

5. An Anatomy of Column-wise Addition Using InfoMap

In the following example, we illustrate how the system simulates the "counting (addition) in column" method. In a top-down fashion, the simulation can be decomposed into three parts: checking the position of digits, calculating (which can be further divided into subparts), and producing the answer as in Figure 3.

Our decomposition is based on the following:

i. The order of solving a problem by counting in column.

ii. Within these three parts, each has its own error types. Consider the problem "*215 + 136*". If the student ignores the position of *5*; he might add *5* to *3*. (See table 1 for more examples)

We shall use the following notations: the number 215 is represented by *T3=2, T2=1, T1=5* and the number 136 is represented by B3 =1, B2=3, B1=6 where the addition at each column is represented by "*Tn + Bn*" in which *Tn* stands for top digits, and *Bn* for bottom digits. The most important part is calculating. It can be further divided into two cases: first, there is no carrying, which corresponds to the classification node, "Tn + Bn" in InfoMap. Second, there is a carrying, which corresponds to the classification node, "*Tn + Bn + 1*". The addition at different positions is a repetitive action.

The node "Ordering" in Figure 3 provides instructions on how to do a repetitive action. It involves three steps, "start", "rule", and "end". "Start" decides where the system sets out, then the system should follow the "rule". Finally, it will stop where the "end" tells it to. In this particular case, the system will start with $T1 + B1$. Then it follows the rule $Tn + Bn$, where the subscript n increases progressively. At the end, it stops when the position of Tn or Bn equals the maximum position of the top number and the bottom number.

Since the system has the ability to work out the problem step by step itself, it can follow a student's problem solving procedure. This helps the system to detect at which step (say $T1 + B1$, $T2 + B2$, or otherwise) the student makes a mistake, and then to determine the reason that the student made the mistake. The latter would give clues on how to provide personalized instructions.

Simulation is not only conducive to diagnosis, but also to teaching strategies. Sometimes, students would like to see how the system solves a problem using a specific method. With the ability to simulate, the system has enough knowledge to accomplish this. It can either demonstrate all problem solving steps or ask the student to fill in some missing steps in an interactive fashion.

6. Conclusion

Personalization, knowledge construction, and the ISIM strategies are three important issues discussed in this paper. Personalization is implemented through the ISIM schema. The latter is supported by the knowledge in our ontology, InfoMap. An important ingredient of InfoMap is that it contains a plethora of basic program modules so that domain experts (or non-programmers) can manipulate these modules by following the top-down knowledge construction as illustrated in Figures 1, 2 and 3. Furthermore, the constructed knowledge is easy to maintain.

In the future, we will extend the knowledge in InfoMap to cover mathematical word problems. For these problems, the first difficulty we face is natural language understanding. The system would need to identify the important roles and the scenario from the problem in order to translate the natural language description into mathematical formula.

7. References:

1. Beck, J., Stern, M., & Haugsjaa, E. (1996). Applications of AI in Education. In ACM Crossroads (http://www.acm.org/crossroads/xrds3-1/aied.html, accessed on 2002/3/8).
2. Bork, A. (1997). The Future of Computer and Learning. In T.H.E. Journal Online (http://www.thejournal.com/magazine/vault/A1682.cfm, accessed on 2002/3/7).
3. Brown, J.S., & Burton, R.R. (1978). Diagnostic models for procedural bugs in basic mathematical skills. In Cognitive Science, 2, 155-192.
4. Brown, J.S., & Vanlehn, K. (1980). Repair Theory: A Generative Theory of Bugs in Procedural Skills. In Cognitive Science 4, 379-426.
5. Burton, R.R. & Brown, J.S. (1979). An Investigation of Computer Coaching for Informal Learning Activities. In Int. J. Man-Machine Studies 11, 5-24.
6. Castel, F. (2002). Ontological Computing. In Communications of the ACM, Feb 2002; Vol.45, Iss.2, PP. 29-30.
7. Evens, M.W. et al. (2001). CIRCSIM-Tutor: An Intelligent Tutoring System Using Natural Language Dialogue. In 12th Midwest AI and Cognitive Science Conference, Oxford OH, pp. 16-23.
8. Jim, O. & Ramachandran, S. (2000). Intelligent Tutoring Systems: The What and the How. In Learning Circuits (http://www.learningcircuits.org/feb2000/ong.html, accessed on 2002/3/8).
9. K. Joseph, N. SuNil, & V. John (2000). A user-centered design approach to personalization. In Communications of the ACM, Aug 2000; Vol.43, Iss.8, PP. 44-48.
10. Mark Urban-Lurain (1996). Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology. In http://aral.cse.msu.edu/Publications/ITS/its.htm (accessed on 2001/11/15).
11. McArthur, D., Lewis, M., & Bishay, M. (1993). The Roles of Artificial Intelligence in Education: Current Progress and Future Prospects. In http://www.rand.org/education/mcarthur/Papers/role.html (accessed on 2002/3/8).
12. Molnar, A. (1997). Computers in Education: A Brief History. In T.H.E. Journal Online (http://www.thejournal.com/magazine/vault/A1681.cfm, accessed on 2002/3/7).
13. Stauffer, K. (1996a). Student Modeling and Web-based Learning Systems. In http://ccism.pc.athabascau.ca/html/students/stupage/Project/initsm.htm (accessed on 2002/3/6).
14. Stauffer, K. (1996b). Applications of Student Modeling. In http://ccism.pc.athabascau.ca/html/students/stupage/Project/sm_app.htm (accessed on 2002/3/6).
15. W.L., Hsu, S.H., Wu, and Y.S., Chen (2001). Event Identification Based On The Information Map - INFOMAP, in symposium Natural Language Processing and Knowledge Engineering of the IEEE Systems, Man, and Cybernetics Conference.