Searching for student intermediate mental steps

Vivien Robinet, Gilles Bisson, Mirta Gordon, Benoît Lemaire

Laboratory TIMC-IMAG (CNRS/UJF UMR 5525)
University of Grenoble, Faculté de Médecine
38706 La Tronche Cedex, FRANCE
{first name.last name@imag.fr}

Abstract. This paper presents a general method for identifying student intermediate mental steps from sequences of actions stored by problem solving-based learning environments, in order to provide feedback to teachers on knowledge that statistically seems to be used by a particular student. When many intermediate mental steps are possible, ambiguity is removed using what is already known about the student. The system uses a student model to search within a huge space of possible actions, and updates this student model consequently. The user model distinguishes between two different cognitive processes: (1) planning the action by focusing on a particular part of the environment and considering an action type and (2) performing the action.

1 Introduction

We are concerned with learning environments in which students are required to perform successive actions. In this paper, we are more specifically interested in the way we may automatically discover student *mental intermediate steps* from a set of observable actions recorded from the environment. This problem compares to the famous *assignment of credit* problem [**1**], in which the goal is to determine knowledge elements directly involved in the observable student behavior. In our case, these knowledge elements are only unitary mental operations. These are called knowledge events by VanLehn [2]. Although our approach is intended to be hooked up to various learning environments, we are currently focusing on algebra learning using the APLUSIX learning environment [3]. Given algebraic equations or inequations to be solved, students using APLUSIX proceed step by step as they would do on a notebook with the only imposed constraint that the expressions entered at any resolution step must be syntactically well formed. In this context, our goal is to discover *mental intermediate steps* of a student modifying an equation. For instance, if a student realizes a wrong transformation from "2x+9=8+6x" to "8x=17", we could assume that he probably performed these mental intermediate steps (Hyp 1), which could be correct or incorrect[1]:

2x+9=8+6x $\rightarrow_{\text{correct movement}}$ 2x-6x+9=8 $\rightarrow_{\text{incorrect calculation}}$ 8x+9=8 $\rightarrow_{\text{correct movement}}$ 8x=8-9 $\rightarrow_{\text{incorrect calculation}}$ **8x=17** (Hyp 1)

However, the previous student action could actually be explained in another way, (Hyp 2) involving correct algebraic calculations and incorrect movements:

2x+9=8+6x $\rightarrow_{\text{incorrect movement}}$ 2x+6x+9=8 $\rightarrow_{\text{correct calculation}}$ 8x+9=8 $\rightarrow_{\text{incorrect movement}}$ 8x=8+9 $\rightarrow_{\text{correct calculation}}$ **8x=17** (Hyp 2)

Without any additional information, it is not possible to select which path the student has most probably mentally followed. The usual way is to rely on statistical information from huge sets of

[1] Even if calculation precedes algebra in teaching, our students often make wrong calculations when asked for solving algebraic problems

student problem-solving data. Teachers have compiled this information from experience, but other approaches are possible. For instance, Tsiriga & Virvou [4] rely on machine learning techniques to initialize the student model. First, students are assigned a stereotype depending on their ability to perform a preliminary test. Student's degree of knowledge is then estimated using a distance weighted k-nearest neighbor algorithm by positioning student among others whose knowledge is already known.

The specificity of our approach is that we see the problem as a recursive problem: discovering this path is dependent on the student model which is in turn updated from these intermediate steps. In other words our approach is to take into account the information which is already known about the current student to adjust what we know from the general statistical information. Let us illustrate, this point: suppose we know the student had performed the following steps (in bold) just before:

a) **3+2x+9=5+4x-2x** ➔correct movement. **2x+9=5-3+4x-2x**

b) **2x+9=5-3+4x-2x** ➔incorrect calculation 2x+9=8+4x-2x ➔incorrect calculation **2x+9=8+6x**

From this data, our partial student model will be something like: "The student tends to perform correct movements and incorrect algebraic calculations". The first path (Hyp 1), which involves correct movements and incorrect algebraic calculations, will thus be considered more probable for this particular student, even if it is not the case for the majority of students.

2 Our user model

The foundation of our model is to consider that in many learning problems, when students are faced with a new state of the environment on which they have to perform an action, they would engage in two kinds of cognitive processes:

1) *planning the action* which reflects the *intention* of the student, consist of focusing on a particular part of the environment in view of a planned type of action. Let us take some Air Traffic Control (ATC) examples, for illustrative purpose only. For instance, an air traffic controller would select a plane with the idea of asking him to wait a bit more before landing, similarly a student faced with "2x+9=8+6x" and asked to solve for x, would select "+6x" with the idea of moving it on the other side of the equation, etc.
2) *performing this action*. For instance, in ATC, the controller would ask the plane to wait a bit more by entering in a well-defined communication procedure. In algebra the student would change "+6x" into "-6x" while moving it to the other side of the equation, etc. Here is an example:

2x+9=8+6x	***----intention--->*** ***(planning phase)***	2x+9=8[+6x]	***----action--->*** ***(performing phase)***	2x-6x+9=8
	ActionType="movement" Focus area="polynomial focus" (+6x)		Argument="polynomial" Side="right" ChangeSign="true" IsCorrect="true"	

Fig. 1. Illustration of the two cognitive processes, (intention, action), leading from one state to the next one.

It is crucial to distinguish among these two steps since a student can be good at identifying useful actions, but fails to perform them, whereas another one may select inappropriate actions but perform them correctly.

We will now present how this model can be implemented in a probabilistic framework. This kind of approach has been already used in the literature, for instance by means of bayesian networks [5].

2.1 Modeling the planning phase

In the student model, this phase is represented as a twofold object containing the *focus area* where an action could be performed and the *type of this action*. In our algebra domain, we identified 61 such pairs: <explicit factorization, polynomial focus>, <explicit factorization, negative number>, <reduction, positive number>, <direct calculation, positive number>, <movement, polynomial focus>, etc.

At a step t, to each pair is attached a probability which depends on the prior probability at time t-1, the number of action types that may be applied and the focus area chosen by the student. If several pairs are candidates, the one which is actually applied by the student (or which we guess has been mentally applied) will have its probability increased while probabilities of other possible focus will be decreased (Fig.2).

	Probability	Possible focus	Actual focus		Probability	Possible focus	Actual focus		Probability
<calculation, polynomial focus>	0.2	0	0		0.2	1	0		0.1
<movement, positive number>	0.2	1	0		0.15	0	0		0.15
<movement, polynomial focus>	0.2	1	1	→	0.3	1	1	→	0.4
<implicit factorization, positive focus>	0.2	1	0		0.15	0	0		0.15
<fraction addition, polynomial focus>	0.2	0	0		0.2	0	0		0.2

Fig. 2. This example corresponds to the intention presented in Fig.1. Three intentional pairs were possible, but the polynomial movement was used by the student. The latter got its probability increased whereas the other two were decreased according to the actual number of possible focus. In the next step, two pairs were candidates and the polynomial movement action was applied again. Probabilities were updated accordingly.

This part of the user model, which is continuously updated, therefore contains probability values for each kind of action the user is likely to consider. Thus, at a given moment, probability values reflect the student's beliefs.

2.2 Modeling the performing phase

This part of the user model describes at a high-level of generalization the user behavior when he does an *action*. We can compare this approach to the one presented by Freyberger, Heffernan and Ruiz [6] in which they construct a transfer model to provide information about what skills are required by the student to solve a particular problem. Similarly, our process will be able to find relevant cross-interactions between attributes and will generalize attributes' values that correspond to similar student behaviors.

An *action* is a generalized vector of *context* and *transformation* attributes that are domain-dependent; their goal is to describe the environment and the student operations. For instance, in the ATC domain, *context* attributes could be "number of planes", "local weather", "fuel level" for each plane, etc. whereas *transformation* attributes could be "ask plane to wait" "ask plane for landing" "ask plane for changing altitude", etc.

In our algebra domain restricted to actionType="movement", we are using 27 *context* attributes such as "sign of focus area", "side of focus area" or "polynomial focus area" and 13 *transformation* attributes such as "change sign of focus area" or "correctness of the transformation".
Each time our system predicts a mental action, a new *context-transformation* vector is generated. Moreover, in order to identify some general student behaviors, these transformation vectors are aggregated using a hierarchical clustering method based on a Manhattan distance between actions. During the aggregation process, context and transformation attributes are generalized inside each cluster to produce generalized vectors of *actions*, as presented below:

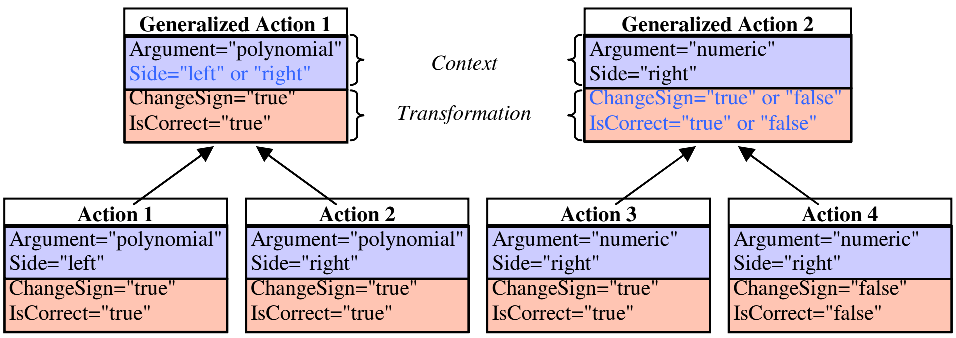

Fig. 3. Two examples of generalized-actions. Left: aggregation of two similar actions leading to the generalization of the "left" and "right" values of the side *context* attribute. Right: aggregation of both ChangeSign and IsCorrect *transformation* attributes, to give Generalized Action 2.

Clustering stops at a predefined threshold depending on a generalization level which was experimentally set. The result is a set of generalized *actions* that the student is likely to perform. To each generalized *action* is assigned a probability value that depends on the number of aggregated *actions* in the cluster (i.e. relative frequency). This information is used in the process of detecting mental intermediate steps as we will now describe.

3 Predicting student intermediate mental steps

Given two student states produced within the learning environment, the goal is to identify intermediate mental steps in-between, that is a sequence of alternating steps of *intention* (I) and *action* (A). The chain between two consecutive explicit states (initial and final state) may involve N mental steps as follows:

initial state ➔ I_1 ➔ A_1 ➔ mental state ➔ .... ➔ mental state$_{N-1}$ ➔ I_N ➔ A_N ➔ **final state**

Since very many pairs (I, A) could have been performed mentally by the student at each stage, we are faced with a huge search space in which we are looking for the most probable path according to what we know about this student.

The user model gives a probability value to each *intention* (I) and *action* (A) candidate. This value will be used to select the next node in the search space. Searching in this space is done by a best-first search algorithm. This kind of algorithm expands the most promising node, according to a heuristic function. In our case, this function takes into account first the probability of the operations as defined in the student model and second, the distance to the goal, which is the distance between the current state and the final state. In our algebra domain, defining such a distance is tricky because algebraic expressions can be very close while having very different surface forms. For instance, "2-4x=11" appears quite different from "11=-4x+2" at the surface level, although it is the same. Expressions are therefore transformed into trees before computing this distance, and the algorithm recursively tries to match nodes in order to minimize the distance between sub-nodes. Fig.4 presents the searching process.

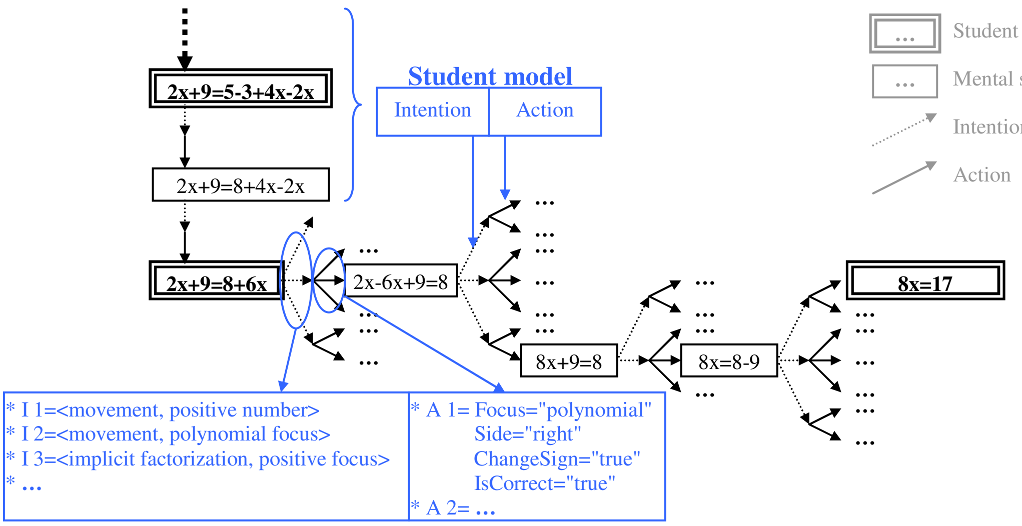

Fig. 4. Example of searching process from student initial equation "2x+9=8+6x" to "8x=17" using intention (I) and action (A) given by the partial student model.

4 Conclusion

This method has been applied to data produced by 40 French secondary school students. Each student performed about 50 movement steps, from which we discovered about 100 mental steps. Computing takes about two minutes per student, leading to about five generalized actions

We have created a model that is able to adapt to various levels of granularity in the student's production. To reach this goal, it is necessary to make hypotheses about intermediate steps students could have performed mentally. But several interpretations (paths) are possible for a same pairs of initial / final states. Our idea is to supplement the classical approach which tends to choose the most probable actions among a large set of students, by introducing what is already known about the particular student.

To do that, we dynamically use probabilities given by our partial student model at each step of our research tree. It is therefore possible to have an idea of how a student will prepare his/her action, i.e. on which terms he/she will focus on, and which type of action he/she will choose. It is also possible to characterize the way the student will probably perform the chosen action, i.e. what transformation s/he will accomplish given a particular focus.

Given a sequence of equations, we are able to find intermediate steps that are probable for a particular student. We believe this method is quite general because the representation formalism is based on attributes, which are appropriate for most domains.

Most of this work has been implemented: the student intention model is operational and guides the search of intermediate mental steps between student equations. Probabilities evolve over time while the model is built. The only thing which remains to be done is to update our equiprobabilized initial model with a priori statistical knowledge about students. The action phase works independently but it is not yet connected to the detection of mental steps. Consequently, these probabilities do not evolve over time.

References

1. Beck, J.E., Woolf, B.P. : High-level Student Modeling with Machine Learning. Proceedings of the Intelligent Tutoring Systems Conference (2000) 584-593.
2. VanLehn, K.: The Behavior of Tutoring Systems. International Journal of Artificial Intelligence in Education 16 (2006) 227-265.
3. Nicaud, J.-F., Bouhineau, D., Huguet, T.: The Aplusix-Editor: A New Kind of Software for the Learning of Algebra. Proceedings of the 6th International Conference on Intelligent Tutoring Systems. Lecture Notes In Computer Science, Springer Vol. 2363. Biarritz, France and San Sebastian, Spain (2002) 178-187.
4. Tsiriga, V., Virvou, M.: Initializing the Student Model using Stereotypes and Machine Learning. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (2002).
5. Anthony J.: Numerical Uncertainty Management in User and Student Modeling: An Overview of Systems and Issues (1995) 193-251.
6. Freyberger, J., Heffernan, N., Ruiz, C.: Using Association Rules to Guide a Search for Best Fitting Transfer Models of Student Learning. Proceedings of 7th Annual Intelligent Tutoring Systems Conference, Maceio, Brazil. (2004).