Modeling the Knowledge-Based Components of a Learning Environment within the Task/Method Paradigm

Christophe CHOQUET, Frédéric DANNA,
Pierre TCHOUNIKINE, and Francky TRICHET

IRIN – Université de Nantes & École Centrale de Nantes
2, rue de la Houssinière – BP 92208 – F-44322 Nantes Cedex 03, France
{choquet|danna|tchou|trichet}@irin.univ-nantes.fr

Abstract. We show that the Task/Method paradigm is adapted for modeling the problem-solving method that the teacher wants to communicate to the student as well as the pedagogical strategies that have to be put into practice for managing the interaction with the student.

1 Introduction

Constructing an educational system that focuses on teaching a problem-solving method requires the construction of different knowledge-bases (KBs). First, as the system must be able to perform on its own the problem-solving method, this method must be modeled and implemented. Second, interacting with the student on the basis of a particular solving context can also be viewed as a specific problem-solving task.

In this paper, we argue in favour of modeling each of the KBs of the ITS with the Task/Method (T&M) paradigm [3], and of operationalising within a reflective architecture [4] both the KB that solves the domain exercises and the KB that manages the system-student interaction. From the point of view of problem-solving exercises, a T&M model allows an abstract modeling of the teacher's problem-solving method, representing knowledge at different levels of abstraction (e.g. [1]), presenting and accepting variants of the "ideal" problem-solving. From the point of view of the management of the system-student interaction, a T&M modeling allows an opportunistic behaviour of the system. The operationalisation of these different kinds of knowledge within an architecture that proposes reflective capabilities enables the construction of modules that analyse the problem-solving state, and provides the pedagogical module with the information that is needed to define the most pertinent interaction.

In the remainder of this paper, we develop these different points with examples from the Emma system. Emma is an educational system (under construction) that aims at training students in the practice of linear programming as a technique to solve concrete problems. Emma is a support tool; this characteristic has been set by the teacher. For this reason, Emma's KBs are constructed on the hypotheses that the system and the student share the same vocabulary and that the student is already familiarised with the problem-solving method proposed by the system.

2 Modeling the Domain Knowledge

In Emma, a typical problem describes a company that must face a command with constraints that it cannot manage. We will consider a company which has to produce and deliver two kinds of steel manufactured pieces in such quantities and delay that it cannot achieve it. The company needs to optimise the production in order to minimise the loss. The domain expert has to (1) identify the type of problem, (2) formalise the mathematical situation, (3) solve the problem (select and apply the method that enables the solving of the considered linear programming problem), and (4) analyse the coherence of the results.

2.1 Modeling the "Ideal" Problem-Solving Method

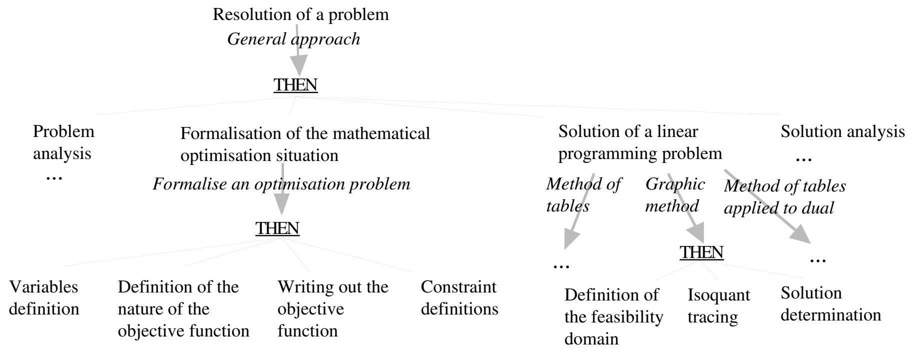
While elaborating a model within the T&M paradigm, one has to explicit what activities have to be achieved for solving the problem, as well as how the activities can be achieved, from the analysis of the problem up to the submission of a solution. This leads to define how tasks and methods should be described to model the expected behaviour, and to construct the T&M knowledge-base in a process guided by the adopted definitions.

In Emma, an *activity*[1] is an aspect of the problem-solving method that teachers can identify when they present their reasoning process. Examples are "Formalisation of the mathematical situation" or "Resolution of a linear programming system". An activity is defined by a name, pre-conditions (when it is pertinent from the point of view of the problem-solving method), an activation context (when it is pertinent from the point of view of mathematical constraints), resources (what knowledge is required to achieve the activity), post-conditions (what strategic aspect has been obtained afterwards) and methods (the methods that can reach the goal underlying the activity).

A *method* is a possible means for achieving an activity. A *decomposition method* splits an activity into multiple sub-activities, whereas an *operational method* defines what process effectively achieves an activity. An activity can often be achieved by different methods. Figure 1 presents an extract of Emma's activities and methods.

Activities and methods are manipulated by mechanisms that perform the dynamic selection of the most pertinent tasks and methods according to the current state of the solving process. Such a selection is done by comparing the selection criteria of a task and/or a method with a set of valuated facts denoting the current state of the solving context. Theses facts and their relations (e.g. "Is a", "Imply" or "Exclude") are modeled within a *domain graph* [2]. With each

[1] In Emma's expert model, a *task* has been renamed an *activity* in order to respect the teacher's vocabulary.


Resolution of a problem
General approach
THEN
Problem analysis
...
Formalisation of the mathematical optimisation situation
Formalise an optimisation problem
THEN
Variables definition
Definition of the nature of the objective function
Writing out the objective function
Constraint definitions
Solution of a linear programming problem
Method of tables
...
Graphic method
THEN
Definition of the feasibility domain
Isoquant tracing
Solution determination
Method of tables applied to dual
...
Solution analysis
...

Text in plain denotes activities. Text in *italics* denotes methods.

Fig. 1. Extract of the T&M knowledge-base that models the expert knowledge

activity is associated knowledge that is used to interpret the results produced by the activity. This *interpretation knowledge* is composed of relations from the domain graph.

2.2 From an "Ideal" Solving Model to a Model of the Correct Solving Processes

As Emma is a support system, we have to use the problem-solving method taught in class as a reference. Nevertheless, some slightly different variants of the global solving process can be viewed as correct, even if they are not exactly the "ideal" one. In order to allow the system to follow slightly different versions from the expert problem-solving process, we have defined three possible status for an interpretation. An interpretation is *necessary* if the relation must be considered when the activity is finished (the resolution cannot go on if it is not considered). An interpretation is *advised* if the teacher usually considers the relation when the activity is finished (it is not necessary at this state of the solving process, but it is more efficient to consider it here). An interpretation is *possible* if the relation can be considered when the activity is over (it is mathematically possible to consider the interpretation at this state of the solving, but it is not usually done here).

For example, let us suppose that the terms of the problem have already been analysed (it is certain that it is an optimisation problem), and that the variables of the problem have been defined. In this context, the activity to be achieved next is to define the nature of the objective function (cf. fig. 1).

Figure 2 shows the representation of the activities "Definition of the nature of the objective function" and "Writing out the objective function". In order to explain how the system can decide whether a solving step is correct, we will take four examples of student's behaviour at the same step, the "definition of the nature of the objective function".

```
Name:                  Definition-nature-objective-function
Pre-conditions:        ((variables,defined))
Activation-context:    ((optimisation-problem,certain))
Resources:             (list-of-variables)
Post-conditions:       ((objective-function-nature,defined))
Methods:               (Define-nature-objective-function)
Interpretations:
     necessary-interpretations  (status,(variable-cost,(true,false))
                                (status,(variable-recipes,(true,false))
                                ((variable-cost,true) and (variable-recipes,true)) imply (maximise-the-margins,true) ...
     advised-interpretations    ((maximise-the-margins,true) or (maximise-the-recipes,true)) imply (maximisation,true))  ...
     possible-interpretations   (status,(objective-function,(defined,undefined)))
                                (status,(objective-function-linear,(true,false)))

Name:                  Writing-out-objective-function
Pre-conditions:        ((variables,defined), (objective-function-nature,defined))
Activation-context:    ((optimisation-problem,certain))
Resources:             (list-of-variables)
Post-conditions:       (objective-function,defined)
Methods:               (Write-objective-function)
Interpretations:
     necessary-interpretations  (status,(objective-function-linear,(true,false)))
```

Fig. 2. Example of encoding of activities

The "not enough" student: The student has not considered some necessary interpretations. For instance, he states that both the cost and the recipes are variable but does not say that the objective function is to maximise the margins. The solving process cannot be pursued because defining the nature of the objective function is a *sine qua none* condition for writing out the objective function.

The "just enough" student: The student considers all the necessary interpretations, but considers neither the advised nor the possible interpretations. The solving process can be pursued, but the student will have to consider these possible and advised interpretations further during the solving.

The "system-like" student: The student considers both necessary and advised interpretations. The solving process can be pursued. Moreover, the student has achieved the activity in the way close to the problem-solving method that is taught in class.

The "exhaustive" student: The student considers, in addition to all the necessary and advised interpretations, possible interpretations that allow him to cut across some activities. For instance, let us assume that the student has considered the possible interpretations of the activity "Definition of the nature of the objective function". In this context, he has written out the objective function (`objective-function,defined`), and he has stated that the function is linear (`objective-function-linear,true`). The post-conditions of the activity "Write out the objective function" are satisfied, and all the necessary interpretations have been considered. Therefore, this activity has no longer to be achieved.

3 Management of the Interaction with the Student

In Emma, the student has to repeat the following steps until a solution is found: (1) select the most relevant activity to be performed next according to the current state of the solving; (2) select the most pertinent method to carry out the selected activity; (3) produce a set of results required to reach the objective of the selected activity (by the way of the selected method). After any of these three steps, the system can make comments on the student's actions.

Figure 3 presents three kinds of interaction we want our system to be able to manage. Note that we do not advocate that this is an exhaustive list.

Excerpt 1

L: The production cost is variable. The recipes are fixed. This is a maximisation kind of problem.

S: It's true that the production cost is variable, and you're right to say that it is a maximisation kind of problem. Now, why do you think it's a maximisation kind of problem?

L: It's a maximisation kind of problem because we have to maximise the margins.

S: It is true that we have to maximise the margins. Moreover, your justification is correct. But why do you say that we have to maximise the margins?

L: because the production cost is variable

S: You're right but this is not sufficient to state that the objective is to maximise the margins. You must add that the recipes are variable, which is true (and not false, as you said) for this exercise.

Excerpt 2

L: This is a maximisation kind of problem.

S: It's true that it is a maximisation kind of problem. Now, why do you think it's a maximisation kind of problem?

L: It's a maximisation kind of problem because we have to maximise the margins.

S: The production cost and the recipes are variable. So, you are right to say that the objective is to maximise the margins.

Excerpt 3

L: I am certain that this is an optimisation problem. The production cost is variable. The recipes are fixed.

S: It's true that this is an optimisation problem. But why do you say that?

L: because it's a maximisation kind of problem

S: Do you think that the objective is to maximise the margins or to minimise the production cost?

L: I think the objective is to minimise the production cost.

S: You're wrong. When one has to minimise the production cost, this is not a maximisation kind of problem. You were right to say that when the production cost is variable and the recipes are fixed, the objective is to minimise their cost. However, according to the terms of the exercise, the recipes are not fixed (as you have said) but variable. These two mistakes together lead you to fortunately give a correct answer.

L stands for "Learner", S for "System".

Fig. 3. Examples of interactions

3.1 Diagnosis of the Student's Answers

In this paper, we do not consider the student's choices of the activities and methods. The diagnosis is simplified to a classification of the results that the student produces for an activity. Three types of results are distinguished: correct, incorrect, and missing results. A student's result is *correct* (resp. *incorrect*) when its value is in accordance (resp. in contradiction) with the one expected in the domain graph. A result is *missing* when expected by the system but not produced by the student. A correct result can be *anticipated* when the student has not explicited some of the inference steps related to necessary interpretations.

For instance, let us consider that the student has already analysed the terms of the exercise, and is currently defining the type of the objective function (cf. fig. 1). Let us suppose that he produces the following results for the current activity: "the production cost is variable, the recipes are fixed and this is a maximisation kind of problem". At this point, the diagnosis is: "it is *correct* to state that the production cost is variable, and stating that this is maximisation kind of problem is *correct but anticipated* since the student has *omitted* to say that the objective is to maximise the margins; moreover, it is *incorrect* that the recipes are fixed".

3.2 Kinds of Interaction

The dynamic adaptation of the interaction is enabled by detecting special situations during the student's solving. With each kind of situation some knowledge is associated that will be used to build the interactions. When no special situation is detected, default interactions are used.

The *default interaction* consists in making comments on correct, then incorrect, and then omitted results. Other kinds of interactions can be thought of, which would be modeled and implemented in the same way. The one we have chosen seems to be of pedagogical interest since treating correct or incorrect results generally conducts to leaving out some omissions. The first excerpt of interactions provides an instance of the proposed default interaction. Asking the student why he thinks that it is a maximisation kind of problem leads him to explicit the reasoning process he has followed (following a backward chaining reasoning). In doing this, the origin of the mistake will finally be pointed out.

Up to now, we have modeled two kinds of *special situations*. These situations can be detected at different levels of interaction.

Fine-grained level interactions aim, for only one interaction step, at dynamically customising the default management of the session. The fine-grained interaction we describe in this paper consists in making comments upon some missing results before dealing with any other ones, when the student is considered as mastering these omitted results. For instance, within the second extract of interaction, the student first justifies the fact that it is a maximisation kind of problem because the objective is to maximise the margins. At this point, the default interaction would lead the system to ask the student why he thinks the objective is to maximise the margins. The fine-grained interaction we propose allows the student (if he masters the recognition of the nature of the production cost and recipes) not to justify his results. The pedagogical relevance of this fine-grained interaction relies on an optimistic analysis of the student's knowledge.

The identification of *remarkable configurations* aims at allowing a dynamic customisation of the default management of the session by treating several consecutive interaction steps as a unit. The remarkable configuration we consider in this paper is characterised by the following situation: the student infers some correct results from partially incorrect ones, and there exists more than one erroneous reasoning path underlying this mistake. The diagnosis has then to be refined in order to determine the erroneous paths the student has followed. The last excerpt of interactions shows an example of this remarkable situation. The dialogue starts with a default interaction which aims at diagnosing why the student has recognised an optimisation problem. Further, when the student states a maximisation kind of problem, the diagnoser identifies two mistakes which characterise the considered remarkable situation. The first mistake, "fixed recipes", is directly diagnosed because of the terms of the exercise. The second one is related to the reasoning path followed by the student. Two incorrect paths are possible: the first one consists in (step 1) inferring that "the objective is to maximise the margins" because "the production cost is variable" and "the recipes are fixed", and (step 2) concluding that "this is a maximisation kind of problem"; the second path consists in (step 1) inferring that "the objective is to minimise the production cost" because "this cost is variable" and "the recipes are fixed", and (step 2) concluding that "this is a maximisation kind of problem". In the former case, the first reasoning step is incorrect; in the latter, the second step is incorrect. Asking the student what he thinks about the objective (maximising the margins *vs* minimising the production cost) enables the diagnoser to state where precisely the mistake occurs. The pedagogical relevance of this remarkable configuration is to detect that the diagnosis has to be refined, and to put into evidence the contradiction in the student's reasoning (in the example, a minimisation instead of a maximisation kind of problem).

3.3 Modeling the Management of the Interactions within the T&M Paradigm

While managing the interaction with the student, the system can either act independently of the student's behaviour, observe the student's behaviour, or react according to the student's behaviour. Let us suppose (1) that the system has already initialised the interaction by proposing the first exercise to the student, and (2) that the student has given a first set of results for which the system has elaborated a diagnosis. At this point, the aim is to influence the student.

In order to influence the student, one can make comments on a diagnosis, explain the current situation of solving in order to deal with the student's misunderstanding, or present the student with a more adapted exercise. These different possibilities are modeled by the way of methods associated with the task "Influence the student". One of these methods is dynamically selected according to the problem-solving context.

In order to make comments on a diagnosis, a strategy for interacting (i.e., defining an order for dealing with all the diagnosed results of the student) is first selected and then applied. This is modeled as presented in fig. 4 by way of the method "Make comments on a diagnosis" which describes a sequential control over a sub-task decomposition ("Select a strategy for interacting" and then "Comment on the results"). The method associated with the first task ("Select a strategy for interacting") explicits the standard way for interacting. In practical terms, it sorts the diagnosed results according to their respective nature. The only method associated with the second task ("Comment on the results") defines an iterative control. It states that the current objective is to make comments on a set of results while a result that is not yet commented exists. At this point, one can make comments on a set of results related to a remarkable configuration and/or make comments on all the results not yet commented according to the adopted strategy.

When a remarkable situation is detected (cf. the third excerpt), a predefined interaction is performed and all the results related to the detected remarkable

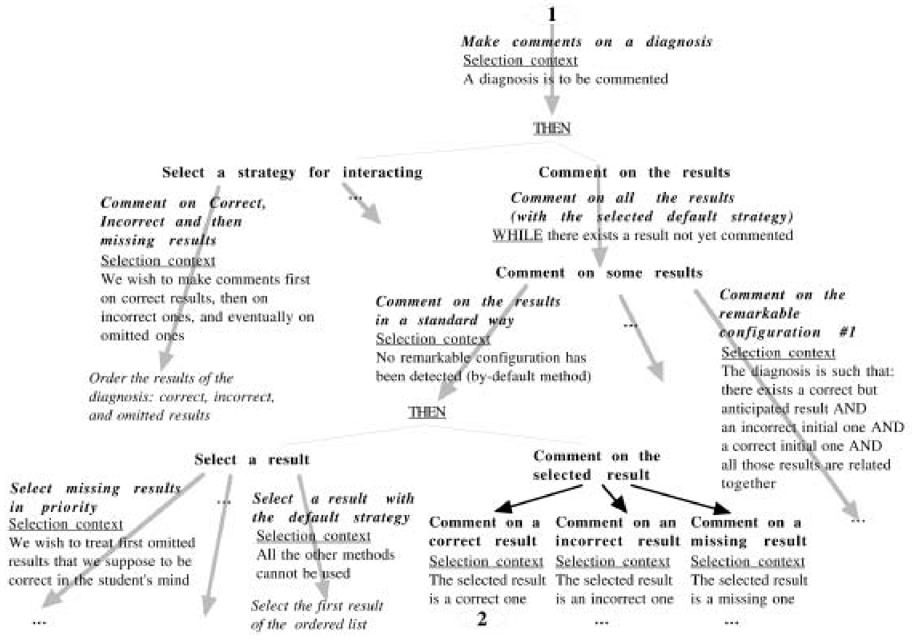

1
Make comments on a diagnosis
Selection context
A diagnosis is to be commented
THEN
Select a strategy for interacting
Comment on Correct, Incorrect and then missing results
Selection context
We wish to make comments first on correct results, then on incorrect ones, and eventually on omitted ones
Order the results of the diagnosis: correct, incorrect, and omitted results
Comment on the results
Comment on all the results (with the selected default strategy)
WHILE there exists a result not yet commented
Comment on some results
Comment on the results in a standard way
Selection context
No remarkable configuration has been detected (by-default method)
Comment on the remarkable configuration #1
Selection context
The diagnosis is such that: there exists a correct but anticipated result AND an incorrect initial one AND a correct initial one AND all those results are related together
THEN
Select a result
Select missing results in priority
Selection context
We wish to treat first omitted results that we suppose to be correct in the student's mind
Select a result with the default strategy
Selection context
All the other methods cannot be used
Select the first result of the ordered list
Comment on the selected result
Comment on a correct result
Selection context
The selected result is a correct one
2
Comment on an incorrect result
Selection context
The selected result is an incorrect one
Comment on a missing result
Selection context
The selected result is a missing one

Fig. 4. T&M decomposition for making comments on a diagnosis

situation are considered as being treated. When no remarkable situation is detected, the comments on the student's results are given in a standard way (cf. fig. 4). In this case, a result that has not yet been commented has to be chosen, and comments on it have to be given to the student. The fine-grained adaptive capabilities of our system are made possible through the task "Select a result": at this point, some results can be punctually preferred to other ones according to some pedagogical criteria (for example, the one we have used in the second excerpt of interaction).

Once a result is chosen, comments have to be made according to its nature: correct, incorrect or omitted. To model this aspect, we have used the notion of *prototypic task*. A prototypic task is dynamically instantiated, at run-time, by an effective task. Let us assume that the system is currently treating a correct result; that is, the prototypic task "Comment on the selected result" has been instantiated by the effective task "Comment on a correct result" (cf. fig. 4). At this point, if the result that is currently being commented on has been introduced by the student in order to justify another one given previously, we consider that the current interaction should take into account this history. Taking such a history into account allows linking two (or more) interactions (in other words, also making comments on the inference steps that underly the produced results). To do so, each method possesses an "Interaction context" slot (cf. fig. 5).

Within this approach, the interaction we obtain for the first excerpt is the following one: "It is true that the objective is to maximise the margins. Moreover, it is true that when the objective is to maximise the margins then this is a maximisation kind of problem.". This list of sentences is computed by selecting the method "Confirm the answer and the underlying reasoning step" (because there exists a context of interaction related to the justification of the previous result "this is a maximisation kind of problem"). This method splits into three sub-tasks. The first one aims at communicating to the student that he is right for the current result. The second task aims at recalling the context of interaction. In our case, the method that is applied for doing so is to "Highlight a correct reasoning step" because (1) the inference step that he has used to justify his result is correct (satisfaction of the Selection context criterion), and (2) the result that the student is justifying is correct but anticipated (satisfaction of the Interaction context criterion). Finally, the third task aims at refining the diagnosis since we have decided that the student has to explicit each of his reasoning steps, and that he has omitted some of them (the current result, "the objective is to maximise the margins", is anticipated).

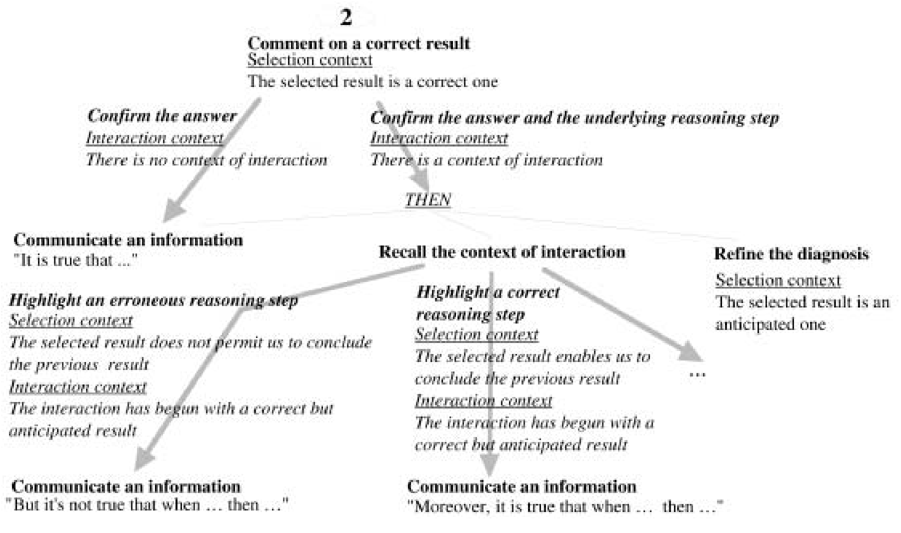

2
Comment on a correct result
Selection context
The selected result is a correct one
Confirm the answer
Interaction context
There is no context of interaction
Confirm the answer and the underlying reasoning step
Interaction context
There is a context of interaction
THEN
Communicate an information
"It is true that ..."
Recall the context of interaction
Refine the diagnosis
Selection context
The selected result is an anticipated one
Highlight an erroneous reasoning step
Selection context
The selected result does not permit us to conclude the previous result
Interaction context
The interaction has begun with a correct but anticipated result
Highlight a correct reasoning step
Selection context
The selected result enables us to conclude the previous result
Interaction context
The interaction has begun with a correct but anticipated result
...
Communicate an information
"But it's not true that when ... then ..."
Communicate an information
"Moreover, it is true that when ... then ..."

Fig. 5. The T&M decomposition for making comments on a correct result

4 Discussion

We have used the same Task/Method paradigm for modeling the pedagogical module, expert model, and diagnoser. In doing this, the whole system is endowed with reflective capabilities that allow it to dynamically analyse parts of its knowledge-bases in order to manage the interaction. This is the case when the system has to trigger the domain expert model in order to be able to select between the possible methods. That is, the system can detect that some information is required in order to select between a task or a method, and can dynamically launch the reflective modules that will retrieve this information. An example of this reflective process may occur when the task "Comment on some results" (cf. fig. 4) is being performed: for a remarkable configuration to be detected, one must determine whether the correct but anticipated result is a logical conclusion that can follow from the correct and incorrect initial results.

The problem-solving model has been implemented, and the interaction model is currently under implementation. In both cases we use the DSTM framework [5]. An important advantage of the T&M paradigm is that the designers explicitly state the pedagogical knowledge. As DSTM flexibility allows a strict structural correspondence between the implementation and the model, this results in a KB in which the pedagogical expertise is explicitly represented. This greatly simplifies any subsequent modification of the KB and leads to a KB that makes sense both for the designers and the teachers.

The pedagogical relevance of the interactions provided by our system has been confirmed from a theoretical point of view by the pedagogical expert of our team. In order to carry out an experimentation with students in real situation, we are currently refining the implementation of the current version of the tasks and methods that manage the interactions (the selection mechanisms being directly provided by the DSTM framework). We believe that such an experimentation will bring to light new remarkable situations and/or new fine-grained level interactions.

References

[1] E. Aïmeur and C. Frasson. Eliciting the learning context in cooperative tutoring systems. In *Working notes of the IJICAI-95 workshop on modelling context in knowledge representation and reasoning*, LAFORIA report 95/11, France, pages 1–11, 1995. 56

[2] C. Choquet, P. Tchounikine, and F. Trichet. Training strategies and knowledge acquisition: using the same reflective tools for different purposes. In *8th International Portuguese Conference on Artificial Intelligence (EPIA'97)*, LNAI, Coimbra, Portugal, 1997. Springer-Verlag. 57

[3] J.M. David, J.P. Krivine, and R. Simmons. *Second Generation Expert Systems.* Springer-Verlag, 1993. 56

[4] M. Reinders, E. Vinkhuyzen, A. Voss, H. Akkermans, J. Balder, B. Bartsch-Sporl, B. Bredeweg, U. Drouven, F. van Harmelen, W. Karbach, Z. Karsen, G. Schreiber, and B. Wielinga. A conceptual modelling framework for Knowledge-Level Reflection. *AI Communications*, 4(2-3):74–87, 1991. 56

[5] F. Trichet and P. Tchounikine. Reusing a flexible task-method framework to prototype a knowledge based system. In *9th International Conference on Software Engineering and Knowledge Engineering (SEKE'97)*, pages 192–199, Madrid, Spain, 1997. 65