Tailored Solutions to Problems Resolution – An Experimental Validation of a Cognitive Computational Knowledge Representation Model

Mehdi Najjar
University of Sherbrooke, Canada

Abstract— **In spite of an increasing development of virtual and distant applications which use the advantages of multimedia and the Internet for distance education, learning by means of such tutorial tools would be more effective if they were specifically adapted to each user needs. This paper proposes a knowledge representation model which judiciously serves the remediation process of an intelligent learning environment in interaction with students during virtual learning activities. By means of experimental results obtained thanks to practical tests, we show that our knowledge representation model facilitates the planning of a tailored sequence of feedbacks that significantly help the learner.**

Index Terms—**Technology in education, Intelligent learning environments, Knowledge representation models, Cognitive approaches of knowledge acquisition, Student tutoring, Tailored feedbacks.**

I. INTRODUCTION

Nowadays, having recourse to intelligent learning environments (ILE) in teaching is being more and more considered by an increasing number of universities and colleges [9,10,12,16]. However, if one has the ambition to build ILE which are able to (i) interact with learners that have various levels of intelligence and different capacities of knowledge acquisition and to (ii) provide tailored aid to students according to their cognitive states, then understanding the human learning processes and the manners of structuring and handling knowledge during those processes is a fundamental task [7]. Especially, if one wishes to exploit the information collected during the learning activities (and which translate the student behaviour) in order to conceive personalised suggestions, tailored examples and helpful exercises which are well adapted to the learner and are built from specific and quite detailed cognitive elements discovered from its behaviour.

In this paper, we propose a knowledge representation model which judiciously serves the remediation process of an ILE's tutoring system in interaction with students during virtual learning activities. The reminder of the article is organised as follows. First, we describe the knowledge representation theory. Our approach is inspired by the artificial intelligence research on the computational modelling of the knowledge and by cognitive theories which offer a fine modelling of the human learning processes. Second, we describe our principle of errors' personalised remediation and we present its experimental validation. Finally, by way of conclusion, we mention our current work.

II. THE KNOWLEDGE REPRESENTATION MODEL

Different approaches in cognitive psychology propose various sets of knowledge representation structures which are inspired from the human memory. Basically, it has been argued that knowledge is encoded in various memory subsystems not according to their contents but according to the way in which these contents are handled and used [5]. These subsystems are mainly divided in three main sections presenting – each one – a particular type of knowledge: (i) semantic knowledge [11], (ii) procedural knowledge [3] and (iii) episodic knowledge [15]. Although there is neither consensus on the number of the subsystems nor on their organisation, the majority of the authors in psychology mentions – in some form or in another – these three types of knowledge.

A The Semantic Knowledge Representation

Semantic knowledge is located in a particular memory subsystem. This latter is the memory of facts and symbols, of their relations, their functions and their genesis [8]. Our knowledge representation approach regards semantic knowledge as concepts taken in a broad sense. Thus, they can be any category of objects. Moreover, we subdivide concepts in two categories: primitive concepts and described concepts. The first is defined as a syntactically non-split representation; i.e., primitive concept representation can not be divided into parts. On the other hand, we define described concept as a syntactically decomposable representation. Thus, the semantic of a described concept is given by the semantics of its components and their relations (which take those components as arguments to create the described concept). In this way, it would be possible to combine primitive or described concepts to represent any other described concept.

B *The Procedural Knowledge Representation*

In opposition to semantic knowledge, which can be expressed explicitly, procedural knowledge becomes apparent by a succession of actions achieved automatically – following internal and/or external stimuli perception – to reach desirable states [3]. In other words, a procedure is a mean of satisfying needs without using the attention resources. For example, procedural knowledge enables us to recognise automatically words in a text, to write by means of the keyboard or to drive a car. This automation – via the use of procedures – reduces the cognitive complexity of problems solving [13]. In our model, we subdivide procedures in two main categories: primitive procedures and complex procedures. Executions of the first are seen as atomic actions. Those of the last can be done by sequences of actions, which satisfy scripts of goals. Each one of those actions results from a primitive or complex procedure execution; and each one of those goals is perceived as an intention of the student cognitive system.

C *The Representation of Goals*

Much, if not most, of our responses to the environment in the form of judgements, decisions and behaviour are determined not solely by the information available in that environment, but rather how it relates to whatever goal – corresponding to a need – we are currently pursuing [2]. In our approach, a goal can be described using a relation as follows: (R X, A1, A2, .. An). This relation (R) allows to specify a goal "X" according to the primitive or described concepts "A1, A2, .. An" which characterise an initial state. Nevertheless, in practice, the stress is often laid on methods to achieve the goal rather than the goal itself; since these methods are, in general, the object of practising. Consequently, the term "goal" is used to refer to an intention to achieve the goal rather than meaning the goal itself. Thus, procedural knowledge becomes the way carrying out this intention and a goal can be seen – computationally – as a generic function where procedures play the role of methods. To underline the intention idea, the expression representing "R" is an action verb. For example, in Boolean algebra, the goal "reduce (F & T)" means the intention to simplify the conjunction of the truth constant "False" with the truth constant "True". Although they are treated by means of procedures, our model considers goals as a special case of semantic knowledge which describes a state to be reached [14] and that represents the intentions behind actions of the cognitive system.

D *The Episodic Knowledge Representation*

The episodic memory retains details about our experiences and preserves temporal relations allowing reconstruction of previously experienced events as well as the time and context in which they took place [15]. In our approach, the episode representation is based on instantiation of goals. Each episode specifies a goal that translates an intention giving a sense to the underlying events and actions. If the goal realisation requires the execution of a complex procedure, formed by a set of "n" actions, then the goal will be composed of "n" subgoals whose realisation will be stored in "n" sub-episodes. Thus, executions of procedures are encoded in a simulated episodic memory of the learner and each goal realisation is encoded in an episode. In this way – and for each student – all facts during a learning activity are stored in her/his episodic memory.

III. THE ERRORS REMEDIATION PRINCIPLE

When interacting with an ILE during the problem solving activities, and when a learner makes an error, satisfying the goal that s/he wished to accomplish was realised by means of an erroneous procedure. This error results from bad interpretation of the situation, causing a choice of procedure which (i) can be correct but whose application cannot be done in the current context or (ii) is invented and completely false. The procedure is regarded as erroneous if the final result obtained by the learner is different from that of the tutor. In this case, the procedure will be labelled (within an episode in which the erroneous result is stored) as a "procedure-error" which has a unique identifier and which will lead to formulate a set of valid procedures that the learner should have used to achieve the goal. At this stage, learning and mastering these correct procedures will be one of the immediate objectives of the tutorial strategy. More precisely, as the episode containing the "procedure-error" comprises an instance of the goal, a set of valid procedures which satisfy it will be deduced starting from the goal prototype. The valid procedures contain the didactic resources necessary to teach their usage. In the case that those procedures are complex, each procedure specifies a set of subgoals whose each one contains its own set of valid procedures. In this recursive way, the tutor easily conceives an ordered sequence of valid procedures allowing the correct accomplishment of any goal. Particularly, those for which the learner has failed.

IV. THE EXPERIMENTAL VALIDATION

"Red-Bool" is an ILE which presents a problem solving milieu related to the simplification of Boolean expressions by using algebraic reduction rules. These are generally taught to undergraduate students. The goal of the ILE is to help students to learn Boolean reduction techniques. Preliminary notions, definitions and explanations (in the "Theory" section) constitute a necessary knowledge background to approach the Boolean reduction problem. This knowledge is organised into sub-sections and is available through exploration via clicking buttons. In the examples section, examples are given. Those are generated randomly with variable degree of difficulty chosen by the learner. Students can also enter, by means of a visual keyboard, any Boolean expression they want and ask the system to solve it. The problem solving steps and the applied rules are shown on a blackboard. Examples show optimal solutions to simplify expressions

and are provided to guide learner during the problem solving, which begins by clicking on the exercise button, allowing to access to the corresponding section. In this latter – and via the visual keyboard – students reduce a randomly generated or a specifically shaped (by the tutor) Boolean expression by choosing suitable simplification rules to apply in the order they want. Figure 1 shows the resolution steps made by a student (Marie) to reduce an expression. Although various tutorial strategies are to be considered, we use actually the "Cognitive Tutor" strategy [4], implemented within several intelligent tutoring systems and which its effectiveness has been largely proven [1,6].
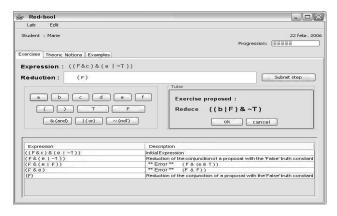

Figure 1 – The problem solving steps made by Marie to simplify the given expression.

Consequently, in the case of erroneous rule choice (or application) on any of the sub-expressions forming the initial given expression, the system notifies the learner and shows her/him – in the "advices" window – (i) the selected sub-expression, (ii) the applied rule to reduce it, (iii) the resulted simplified sub-expression and (iv) the current state of the global expression. If there were mistakes, then at the end of each exercise, the tutor proposes to the student a related example or suggests to her/him to solve another exercise. In this last case, the Boolean expression suggested to reduce is considered as a personalised feedback with regards to the made errors.

TABLE 1. – MAIN PARAMETERS OF THE EXPERIMENT

Complexity	1	2	3	4	5
Number of exercises	4	4	5	6	6
Number of students	10	10	10	10	10

We asked students in mathematics who attend the courses "MAT-113" or "MAT-114" dedicated to logic calculus and discrete math, to practice the reduction of Boolean expressions using "Red-Bool". By this experiment, our interest was to record the resolution's traces of each learner during problems solving tasks (in the "exercises" section) in order to evaluate the aptitude of the feedbacks' model to enlighten the tutor when making tutorial decisions. Data and parameters of this experiment are reported in Table 1.

According to the proposed theoretical approach described above, each step in a learner's resolution process (during a solving task) corresponds to a transition realisable by means of primitive or complex procedure which was applied to satisfy a goal or a subgoal. This procedure handles primitive and/or described concepts such as rules, proposals, logical operators and truth constants. For each student and each exercise made, the system deduces (starting from the low-level observations sent by the graphical interface of the ILE) the procedures used as well as the instances of knowledge created and handled. Since a procedure is generally called to achieve a goal, the collected data allows deduction of goals (and their subgoals) formulated during the Boolean reduction process. At the end of the exercise, the system saves the trace of the resolution in an "episodic" XML file which serves for the errors' analysis. For example, let's consider the case of John who tried to reduce the expression "(a & ~T)" by (1) applying the simplification rule of the "True" truth constant negation which substitutes "(~T)" by "(F)" transforming "(a & ~T)" into "(a & F)" and (2)

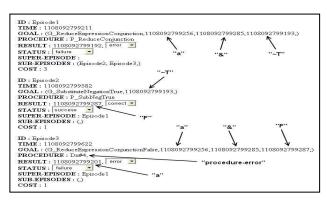Figure 2 – A part of the John's episodic history related to the reduction of the expression "(a & ~T)".

changing the resulted expression into "(a)". Here, John makes a mistake. Theoretically, the reduction of "(a & F)" is correctly made by applying the conjunction rule of a proposal with the "False" truth constant (p & F → F, where p is a proposal) which results in transforming "(a & F)" into "(F)". In this case, the main goal "reduce (a & ~T)" was achieved by a complex procedure giving rise to two subgoals : (1) "substitute (~T ; F)", which was achieved by the primitive procedure "P_SubNegTrue" calling the substitution rule of the "True" truth constant negation; and (2) "reduce (a & F)" which was achieved by a procedure calling an unknown erroneous rule (noted in figure 2 and figure 3 by Dx#4) unseated of the primitive procedure "P_ReduceConjunctionFalse" which call the conjunction rule of a proposal with the "F" truth constant. Figure 2 illustrates the episodic history related to this exercise. "Episode1" reflects the main goal realisation which was split into two sub-events: "Episode2" and "Episode3". The former represents the substitution of the "True" truth constant negation and the latter corresponds to the "(a & F)" erroneous simplification.

Figure 3 shows the slots' content of the goal *"P_ReduceExpressionConjunctionFalse"* that John attempts erroneously to achieve.

The analysis of errors consists in (1) scanning the content of the XML file to research the errors occurred during the reduction of the expression and, for each detected error, (2) identifying a valid procedure (which we note "*P_valid*") allowing to achieve the student goal and which could have been used instead of the erroneous procedure (which we note "*P_error*"). The identification of a correct procedure – which makes use of Boolean reduction rules – is made thanks to a second XML file that contains the domain knowledge. In that case, the tutor proposes to the learner a new Boolean expression (which we note "*Expr_FBack*") that the simplification will (in theory) make use of "*P_valid*". In this sense, "*Expr_FBack*" can be seen as a personalised remediation following the occurrence of "*P_error*".

The individualised feedback generation process

The slot "exercises" defined in the structure of the valid procedure includes a script containing

Figure 3 – Some slots of the goal
"G_ReduceExpressionCunjonctionFalse".

dynamic (not predefined) didactic resources. i.e., a generic model of exercises. In order to propose an exercise to resolve, the generic model specifies a sequence of goals which are of the type "G_build". The type "G_build" enables to create (1) a primitive object (concept) starting from its class or (2) a complex object starting from the classes of its components. Arguments of each goal of the type "G_build" are formulated starting from clues discovered in the episodic XML file. In other words, the structure of the episodic memory permits to the tutor to find, thanks to the erroneous procedures, the episodes in which errors have occurred. These episodes contain indices which are taken as parameters by the goal of the type "G_build"; and thus, which are useful to scaffold an exercise with regards to the generic model. For example, and as shown in Figure 1 which illustrates the steps made by Marie to reduce the expression "((F &

c) & (e | ~T)), the student deals firstly with the sub-expression "(F & c)" and applies the conjunction rule of a proposal with the "False" truth constant to obtain "(F)". At step 2, she simplifies the sub-expression "(~T)" to "(T)". Here, Marie makes a mistake. Theoretically, the reduction of "(~T)" is correctly made by applying the negation rule of the "True" truth constant. At step 3, another error was made when Marie simplifies the sub-expression "(e | F)" to "(F)". The reduction of "(e | F)" is correctly made by applying the disjunction rule of a proposal with the "F" truth constant (p | F → p, where p is a proposal), which results in transforming the sub-expression into "(e)", not into "(F)". At the last step, Marie applies the conjunction rule of a proposal with the "False" truth constant to reduce "(F & e)" into "(F)".

At the end of the exercise, and in consequence with the two made errors, the objective of the tutorial strategy is to teach Marie (1) the use of the simplification rules of the negation of a truth constant and (2) the application of the reduction rule of the disjunction of a proposal with the "False" truth constant. To this end, the generic model of the didactic resources of each valid procedure which allows achieving a failed goal (i.e., the intention to simplify the negation of the "True" truth constant or that to reduce the disjunction of a proposal with the
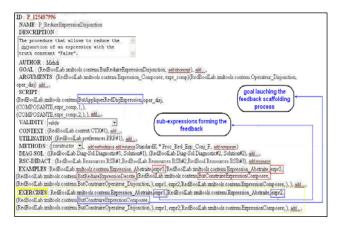

Figure 4 – Some slots of the procedure
"P_ReduceExpressionDisjonction".

"False" truth constant) is requested to scaffold an exercise that will be proposed – to the learner – as a tailored feedback. To remedy her two gaps, the tutor proposes to Marie to practice the simplification of the expression "((b | F) & ~T)". This one is formulated starting from the scripts of the slots "exercises" of the procedures "P_Apply_ReductionNegation_True" and "P_ReduceExpressionDisjonction". Figure 4 shows some slots of the latter which simplifies the disjunction of a proposal with the "False" truth constant. For example, Table 2 comprises feedbacks generated following the resolution of the expression "(((F | c) & (E & ~V)) & (~a | ~F))" which was given as exercise to six (6) students. Because of the difference of the made errors, feedbacks (provided in terms of suggested exercises) are dissimilar.

TABLE 2. GENERATED FEEDBACKS FOLLOWING THE
RESOLUTION OF THE SAME EXPRESSION

Student	Feedback
1	$((T \& d) \& (\sim T \& (T \mid a)))$
2	$(\sim F \& (c \& F))$
3	$(\sim F \& ((T \mid e) \mid (F \& F)))$
4	$((c \mid T) \& \sim T)$
5	$((\sim F) \& (\sim T))$
6	$((F \& \sim a) \& (\sim T \& (b \mid F)))$

V.   CONCLUSION

We have presented a knowledge representation model which is inspired by the artificial intelligence research on the computational modelling of the knowledge and by cognitive theories that offers a fine modelling of the human learning processes. We have introduced an original principle of personalised remediation to students' errors. By means of experimental results obtained thanks to practical tests, we have show that our knowledge representation model facilitates the planning of a tailored sequence of feedbacks that significantly help the learner. We are actually refining the knowledge representation structures – by taking into account pedagogical and didactic knowledge – and setting about new experiments with others teaching domains; such as, teaching heuristic techniques in operational research and teaching the resolution by refutation in the predicate calculus.

ACKNOWLEDGMENT

REFERENCES

[1] Aleven, V. & Koedinger, K. (2002). An Effective Metacognitive Strategy: Learning by doing and explaining with computer-based Cognitive Tutors. Cognitive Science. 26 (2): 147-179.

[2] Altmann, E. & Trafton, J. (2002). Memory for goals: An Activation-Based Model. Cognitive Science, 26, 39-83.

[3] Anderson, J. R. (1993). Rules of the mind. Lawrence Erlbaum Eds.

[4] Anderson, J. R., Corbett, A.T., Koedinger, K.R. & Pelletier, R. (1995). Cognitive Tutors: Lessons learned. The Journal of Learning Sciences. 4 (2):167-207.

[5] Baddeley, A. (1990). The Human Memory : theory and practice. Hove (UK): Lawrence Erlbaum.

[6] Corbett, A., Mclaughlin, M. & Scarpinatto, K. C. (2000). Modeling Student Knowledge: Cognitive Tutors in High School and College. Journal of User Modeling and User-Adapted Interaction. (10): 81-108.

[7] de Rosis, F. (2001). Towards adaptation of interaction to affective factors. Journal of User Modeling and User-Adapted Interaction. 11(4).

[8] Gagné, R.M, Briggs, L. & Wager, W. (1992). Principles of Instructional Design. (4th edition), NY: Holt & Winston.

[9] Heermann, D. & Fuhrmann, T. (2000). Teaching physics in the virtual university: the Mechanics toolkit, in Computer Physics Communications, 127(1), 11-15

[10] Lintermann, B. & Deussen, O. (1999). Interactive modelling of plants, IEEE Computer Graphics & Applications. 19 (1).

[11] Neely, J. H. (1989). Experimental dissociations and the episodic/semantic memory distinction. Experimental Psychology: Human Learning and Memory. (6): 441-466.

[12] Rzepa, H & Tonge, A. (1998). VChemlab: A virtual chemistry laboratory. Journal of Chemical Information and Computer Science, Vol 38, pp1048-1053.

[13] Sweller, J. (1988) Cognitive load during problem solving: effects on learning. Cognitive Science, 12, 257-285.

[14] Trafton, J. G., Altmann, E. M., Brock, D. P. & Mintz, F. E. (2003). Preparing to resume an interrupted task: effects of prospective goal in encoding and retrospective rehearsal. International Journal of Human-Computer Studies. 58, 583-603.

[15] Tulving, E. (1993). Elements of Episodic Memory. Oxford University Press, New York.

[16] Wells, L.K. and Jeffrey, T. (1996). LabVIEW for Everyone: Graphical Programming Made Even Easier, Prentice Hall Eds., NJ.

AUTHOR

Mehdi Najjar is with the Dept. of Computer Science, University of Sherbrooke, Canada (e-mail: mohamed.mehdi.najjar@usherbrooke.ca).