# Sharing Learners' Behavior to Enhance a Metacognition-oriented Intelligent Tutoring System

Julien Broisin, Philippe Vidal

# Sharing Learners' Behavior to Enhance a Metacognition-oriented Intelligent Tutoring System

Julien Broisin, Philippe Vidal
Institut de Recherche en Informatique de Toulouse
Toulouse
France
{broisin,vidal}@irit.fr

**Abstract**: Literature shows that Intelligent Tutoring Systems (ITS) are growing in acceptance and popularity because they increase performances of students, leverage cognitive development, but also significantly reduce time to acquire knowledge and competencies. Moreover, monitoring metacognitive skills enables learners to assess performance and select appropriate fix-up: individuals unable to ensure self-monitoring cannot detect errors and as a consequence, they process information less efficiently than skilled monitors. Thus, we present an ITS offering the opportunity of evaluating various metacognitive indicators and able to share this information with others learning tools. Our online tutor is based on an existing ITS authoring tool that we extended to support metacognition and share learners' profiles and activities into a standardized, distributed and open tracking repository. This framework, validated by an experimentation, thus helps to correlate metadata experiences with real performance and help seeking, while promoting share and reuse of learners' specificities required to elaborate personalized ITS.

## Introduction

Intelligent Tutoring Systems (ITS) have proven their worth in multiple ways in education (Anderson, 1995). Namely, evaluations of these tutors showed significant achievement gain: students could achieve at least the same level of proficiency as conventional instruction in one third of time (Anderson, 1995). They have been developed for a number of domains including mathematics, computer science, languages or speech recognition. ITSs are growing in acceptance and popularity because (1) performances of students are increased, (2) cognitive development is encouraged, and (3) the time for the student to acquire skills and knowledge is commonly reduced (Graesser, 2001).

On the other hand, leading researchers state that metacognition ensures effective learning (Bransford, 2000). Its role is to monitor and control cognition so that one's goal can be achieved. Metacognitive experiences (Efklides, 2001) are important as they have a bearing on the efficiency or the control decisions in learning situations with respect to effort allocation, time investment, strategy use and teaching procedures which students follow. For example, lack of confidence or low confidence makes the learner to further pursue a learning goal (Schraw, 1995). High feeling of confidence, on the other hand, makes them more decisive; self-efficacy theory predicts that students work harder on a learning task when they judge themselves as capable (Mayer, 1992). And finally, if the feeling of difficulty is high and associated with negative affect, the learner quits the task (Panaoura, 2007).

Even if several researchers have developed systems supporting various metacognitive attributes such as self-explanation (Aleven, 2002), gaming the system (Baker, 2006), self-monitoring (Bull, 2010) or help-seeking (Aleven, 2006), a few of them dig into some of dimensions of metacognitive experiences such as self-representation, self-image, self-evaluation or feeling of confidence. Moreover, most of these learning tools are stand-alone systems: it is very difficult to benefit from data produced by learners during a learning session, even though this information may be crucial for third parties aiming at providing students with efficient intelligent tutoring systems.

To tackle the above issues, works presented here relate on an ITS providing a generic approach to support any metacognitive attribute. This ITS, based on an existing authoring tool called Cognitive Tutoring Authoring Tools (CTAT) and elaborated by Carnegie Mellon university, collects metacognitive experiences of learners before, during and after a task processing through a Likert scale. The system also integrates the model of desired help-seeking behavior (Aleven, 2006) by generating hints when learners go through a wrong learning path. All information translating interactions between learners and the system (including profiles, cognitive activities, metacognitive experiences) are collected in real time and then stored into an external component responsible for the management of these data. The trace-based system is characterized by an open (in terms of interoperability), extensible (in terms of information to supervise) and standardized (in terms of data representation and protocols) architecture that makes it possible to share and reuse tracking information between web-based learning tools.

The remainder of the paper is organized as follows. The next section reviews two intelligent tutoring paradigms and points out the limitations of CTAT about metacognition. Section 3 thus details how the tool can be extended to integrate metacognitive attributes, and validates our approach by applying the generic framework to a concrete learning activity. In section 4, we describe the trace-based system able to gather and store data produced by the ITS. Finally we conclude before exposing some future works.

## Paradigms of Intelligent Tutoring Systems

Numerous approaches dealing with student modeling emerged but two of them are mainly adopted: cognitive tutoring or Model Tracing Tutoring (MTT) and Constraint-Based Modeling (CBM). Our tutoring system is based on the MTT approach because it evaluates action instead of problem state, it provides an immediate feedback (McKendree, 1990), it includes hints (Walker, 2009), and it triggers bugs (Shute, 2008). However, an ITS and especially an MTT tends to be harder to implement. (Aleven, 2006) estimates that building an ITS needs 200-300 hours of development per hour of instruction. To tackle this issue, Cognitive Tutor Authoring Tools (CTAT) is intended to make tutor development both easier and faster for experienced modelers and possible for potential modelers who are not experts in cognitive psychology or artificial intelligence programming (Koedinger, 2003). CTAT is a rule-based system, that is a computer program that uses rules to reach conclusions from a set of premises.

However, even if this authoring tool brings numerous functionalities to easily build an ITS, it doesn't support metacognition: students are unable to reflect or monitor their knowledge from the tutoring activities they process. Therefore, the next section introduces a generic set of classes designed to support any metacognitive attribute within CTAT and shows how it can be used to gather Judgment of Learning (JOL) (Wang, 1993), Feeling of Confidence (FOC) and Feeling of Satisfaction (FOS) of learners.

## Extending an ITS Authoring Tool

### Some Generic Classes to Support Metacognition

Since CTAT is able to work with any resources composing a tutoring tool, we designed a set of classes presenting a high level of abstraction and detailed on Fig. 1. The abstract class IsMeta is the root element to model a metacognitive attribute, and indicates whether a message box should appear on the graphical interface so that students are able to self-monitor their cognitive activities. The other abstract class called Resource represents all resources to be manipulated; the Widgets property refers to the computing element that will host the question (text field, chooser, etc.). We finally defined an aggregation association between the above abstracts classes so that one resource may deliver an undefined number of metacognitive indicators.

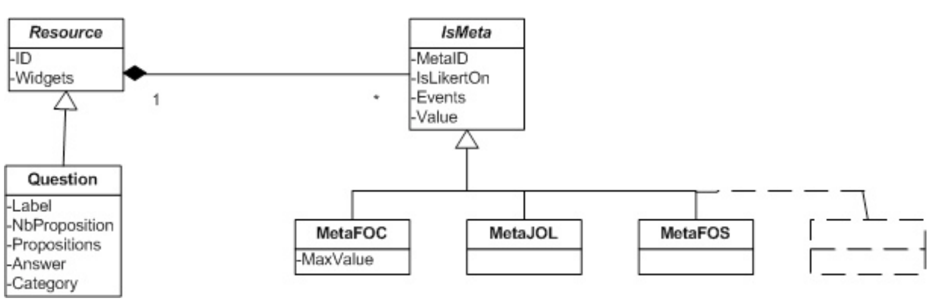


**Figure 1**: A set of classes for metacognition.

**Gathering JOL, FOC and FOS**

The generic approach presented in the previous section has been specialized to elaborate a tool dedicated to the curriculum Certificat Informatique et Internet (C2I). This certification ensures that students have basic competencies about computer science and Internet services. Our ITS named MetaCTAT and illustrated by Fig.2 consists in 20 questions endorsed by psychologists. Thus, we designed the class Question characterized by some properties such as the label of the question, the number of proposition, the label of proposition or the answer.

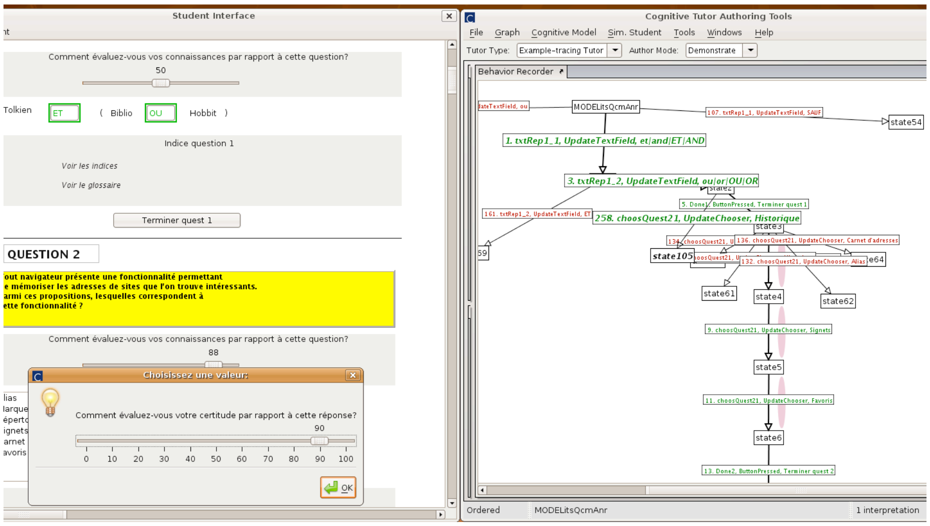

**Figure 2**: MetaCTAT and the behavior recorder.

This application focuses on metacognitive experiences because they are connected to the cognitive regulatory loop which concerns an ITS. MetaCTAT takes into account 3 indicators: Judgment of Learning (JOL) to recognize how students predict their performance, Feeling of Confidence (FOC) to determine the level of confidence that students express, and Feeling of Satisfaction (FOS) to reveal if the system could improve learner performance.

Once logged in MetaCTAT, a student completes repeatedly one question from another according to the following process: first, he reads the question and immediately reports his prediction (JOL) through a Likert scale (a Likert scale is the most widely used scale in survey research and refers to a rating scale). Then, the student reads the propositions and finally answers to the question. As soon as the last event occurs, the learner has to indicate his level of confidence (FOC) using another Likert scale. Even if the graphical component hosting the answer is automatically colored in green or red color (respectively for correct and incorrect answers), the learner can’t access this information because the message box showing the FOC indicator is programmed and designed to hide the behavior of the main graphical interface (see Fig.2). Moreover, the learner can’t go forward into the quiz if he does not report one of the metacognitive attributes.

Since the Behavior Recorder mentioned in section 2 and comprised within MetaCTAT allows handling events according to an answer, the tutor suggests hints related to the question when learners give a wrong answer: it explains why the answer is incorrect and provides definitions or concepts helping him to guess the solution.

This concrete use case demonstrates how various metacognitive experiences can be easily supported by our generic approach. To exploit this information, there is a need for a mechanism able to collect and store these data for further analysis and reuse. However, the format of CTAT log files recording data related to learners is specific to this ITS, and lacks a lot of precisions: it is not possible to know if learners used or ignored hints, it does not give the number of answers for the same question nor the time spent on a specific question, etc. We then introduced an additional functionality responsible for collecting and externalizing metacognitive experiences and learning activities into a dedicated system presented in the following section.

## The trace-based system

The Web-Based Enterprise Management (WBEM) specification is an ongoing initiative started by the Distributed Management Task Force (DMTF) to manage disparate networks, systems, and applications. This widely-adopted standard (Microsoft© operating systems implement this approach, as well as several Linux distributions) stands on the extensible Common Information Model (CIM) to represent managed entities (systems, networks and applications) in a uniform point of view, and addresses the needs for a scalable solution by adopting a distributed architecture of the management components.

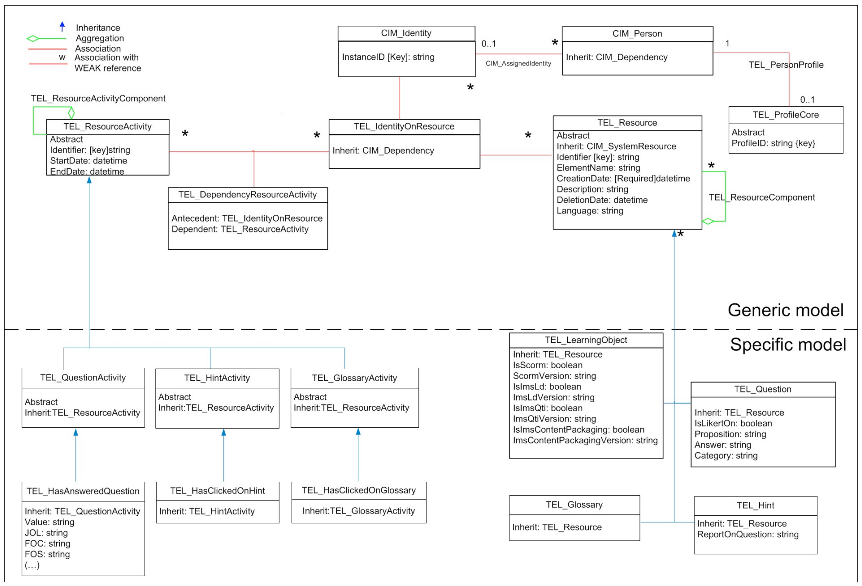


**Figure 3**: Modeling of users, resources and activities.

In order to take into account data resulting from MetaCTAT, we defined a set of classes illustrated on Fig. 3 and modeling users, resources and activities. The abstract class *TEL_ProfileCore* and its sub-classes represent learners' profiles that include the IMS-LIP standard together with some additional information (this model is precisely detailed in (Ramandalahy, 2009)). The model of resources describes questions, hints and glossaries related to a quiz, whereas activities that can be performed on these resources are respectively modeled through the following classes: *TEL_HasAnsweredQuestion*, *TEL_HasClickedOnHint* and *TEL_HasClickedOnGlossary*. The activity class related to questions presents some properties related to the learners' metacognitive experiences. A high abstraction level offering the opportunity of defining specific models according to some specific objectives characterizes the modeling depicted on Fig.3.

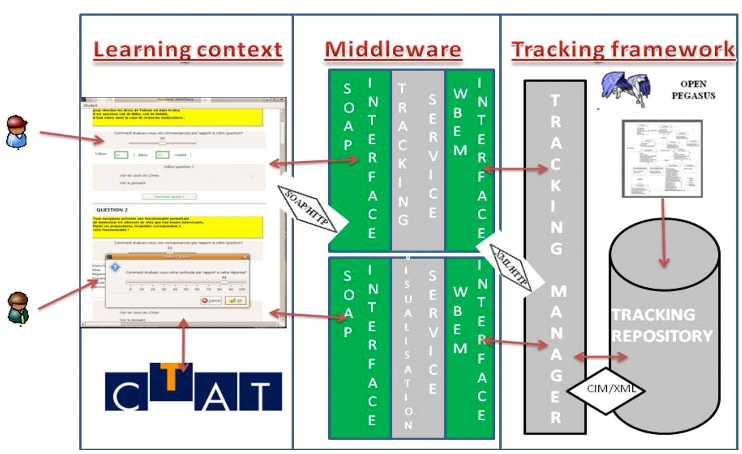


**Figure 4**: Architecture of the global framework.

To make this data model operational, we set up a distributed architecture conform to the WBEM standard and depicted on Fig. 4. The global framework comprises three parts:

- MetaCTAT, which represents the learning context: data modeled in Fig. 3 are collected from this system. An agent embedded into MetaCTAT is responsible for capturing data whenever a user performs an activity, and then forwards information using web services technologies.
- The tracking framework is composed of two WBEM components: a tracking repository is responsible for storing tracking information, whereas a tracking manager is able to manipulate traces stored into the repository. OpenPegasus, a C++ open source implementation of the WBEM standards, has been deployed within our framework.
- The intermediate layer between the learning and the tracking environments offers an easy access to the tracking repository: through the web services of this bridge, learning tools are able to easily provide/retrieve traces into/from the repository.

Thus, when a student performs a task within MetaCTAT (a task includes indication of the JOL, FOC or FOR, suggestion of an answer or consultation of a hint or glossary), several treatments are triggered. First, the integrated agent extracts the information defined into our model and invokes the tracking service of the middleware layer; data that are gathered comprise information about the user, the question, the nature of the performed activity and, according to this last, the values of metacognitive indicators or help resources. Then, the tracking service builds the matching CIM and TEL instances which are finally sent to the tracking manager and stored into the tracking repository.

The trace-based system presented in this section supervises learners' activities within the learning context and ensures storage of tracking data within a dedicated repository through an intermediate layer. This 3 tiers architecture can be easily upgraded to N tiers architecture to provide scalability and availability. Indeed, the intermediate layer can be duplicated, whereas the WBEM standard ensures manager-to-manager communications: tracking data stored into various repositories can be exchanged from one system to another. Finally, thanks to the high level of abstraction of the data model, additional information can easily be taken into account.

## Conclusions and future works

The whole framework has just been achieved. MetaCTAT was tested with 12 students enrolled in second year of computer science degree. Bugs have been identified and corrected, but the relative number of sample is not important enough to suggest a theory on effects of JOL, FOC and FOS regarding help-seeking. However, feedbacks of students ware much positive, since they do prefer metaCTAT over the previous application, a web-based testing application using PHP and jQuery plug-in, developed from scratch. CTAT performs well and did took us half of the time needed to implement this application. It is very efficient for creating MTT tool.

Nevertheless, some improvement could be made for this tool particularly the localization problems: some messages remain delivered in English language. The source code of CTAT is not available probably because it uses other components and we noticed a lack of documentation or real-world best practices.

We intend to perform large-scale experimentation by March 2010 in order to synthesize and answer to some research questions such as: does prediction have an impact on real performance? How confidence is related to help seeking? Does confidence is dependent on Prediction?

Depending on the learner profile, help resources will be extended namely ask the expert on which learner seemed to be stuck on certain item could ask for help from these experts. This latter information is stored into the distributed repository.

As we have a generic model, we will be able to personalize this ITS according to the learner profile. It is possible to have different ITSs from this sole framework. And depending on his current profile, questions offered to learners can be changed or modifying hints in order to improve his learning path.

## References

Aleven, V. & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based Cognitive Tutor. In *Cognitive Science* (pp. 147-179), vol. 26.

Aleven, V., McLaren, B. M., Roll, I. & Koedinger, K. R. (2006). Toward meta-cognitive tutoring: A model of help seeking with a Cognitive Tutor. In *International Journal of Artificial Intelligence in Education* (pp. 101-128), vol. 16.

Anderson, J. R., Corbett, A. T., Koedinger, K. R. & Pelletier, R. (1995). Cognitive tutors: Lessons learned. In *Journal of the Learning Sciences* (pp. 167-207), vol. 4(2).

Baker, R. S. J. D., Corbett, A. T., Koedinger, K. R., Evenson, E., Roll, I. & Wagner, A. Z. (2006). Adapting to when students game an intelligent tutoring system. *8th International Conference on Intelligent Tutoring Systems* (pp. 392–401). Berlin: Springer.

Bransford, J. D. Brown, A. L. & Cocking, R. R. (2000). *How People Learn: Brain, Mind, Experience, and School*. Washington, CD: National Academy Press.

Bull, S. & Kay, J. (2008). Metacognition and Open Learner Models. In I. Roll & V. Aleven (Ed.), *Proceedings of Workshop on Metacognition and Self-Regulated Learning in Educational Technologies, International Conference on Intelligent Tutoring Systems* (pp. 7-20).

Efklides, A. (2001). Metacognitive experiences in problem solving: metacognition, motivation, and self-regulation. In A. Efklides, J. Kuhl, & R. M. Sorrentino (Ed.), *Trends and prospects in motivation research* (pp. 297-323). Dordrecht, The Netherlands: Kluwer.

Graesser, A.C., Person, N., Harter, D. & the Tutoring Research Group. (2001). Teaching tactics and dialog in AutoTutor. In *International Journal of Artificial Intelligence in Education* (pp. 257–279), vol. 12.

Koedinger, K. R. & Heffernan, N. (2003). Toward a Rapid Development Environment for Cognitive Tutors. In U. Hoppe, F. Verdejo, & J. Kay (Ed.), *Proceedings of the 11th International Conference on Artificial Intelligence in Education*.

Mayer, R. E. (1992). *Thinking, problem solving, cognition*. New York: W.H. Freeman and Company.

McKendree, J. (1990). Effective feedback content for tutoring complex skills. In *Human Computer Interaction* (pp. 381-413), vol. 5.

Panaoura A. (2007). *The impact of recent metacognitive experiences on preservice teachers' self-representation in mathematics and ITS teaching*. From http://ermeweb.free.fr/CERME%205/WG2/2_Panaoura.pdf

Ramandalahy, T.,Vidal, P. & Broisin, J. (2009). Sharing and reusing open learner profile. In *Proceedings of the IEEE International Conference on Advanced Learning Technologies* (pp. 504-508).

Schraw, G. (1995). Measures of feeling-of-knowing accuracy, a new loook at an old problem. In *Applied Cognitive psychology* (pp. 321-332), vol. 9.

Shute V. (2008). Focus on formative feedback. In *Review of Educational Research* (pp. 153-189), vol. 78.

Walker E., Rummel N. & Koedinger K. (2009). Modeling helping behavior in an Intelligent Tutor for peer tutoring. *14th International Conference on Artificial Intelligence in Education* (pp. 341-348). Amsterdam: IOS Press.

Wang, M.C., Haertel, G.D. & Walberg, H.J. (1993). Toward a knowledge base for school learning, In *Journal of Educational Psychology* (pp. 249-294), vol. 63.