

# Externalising Learner Models

A. Paiva

INESC, IST - Technical University of Lisbon, Portugal.

and Department of Computing, Lancaster University,

Lancaster LA1 4YR, UK

Telephone (0524) 65201, Fax (0524) 381707

amp@cbl.leeds.ac.uk <http://cbl.leeds.ac.uk/amp/personal.html>

J. Self

Department of Computing, Lancaster University

Lancaster LA1 4YR, UK

R. Hartley

Computer Based Learning Unit, University of Leeds,

LS2 9JT Leeds, UK

## Abstract

This paper deals with the problem of externalising learner models, which is the process of making the learner models and embedded processes visible to external agents.

We will tackle this problem by introducing the approach followed in the learner modeling system TAGUS which allows the creation of learner models in an independent way. The main idea relies on the fact that TAGUS can be seen as an independent system or agent, and therefore the content of the learner model can be communicated to other computational agents, and consequently externalised.

The mechanism to externalise learner models is embedded in TAGUS by the use of Viewers which are subsystems that allow access to the learner model extracting the appropriate information to the external agent. These Viewers provide a kind of interface between TAGUS, that is responsible for maintaining the learner model, and the external agent, who is going to use the externalised model.

## 1 Introduction

Learner Models ( $\mathcal{LM}$ ) are representations of some characteristics and attitudes of the learners, which are useful for achieving the adequate and individualised interaction established between computational environments and students. They are constituted by descriptions of what is considered relevant about the actual state of knowledge of a learner, providing information for the learning environment to adapt itself to the individual learner.

The quantity of techniques and approaches that exist for acquiring these learner models and the richness of their content and representation suggests that they can be used for other purposes than merely the adaptation of the interaction to the learner.

Some of these other uses were suggested in (VanLehn, Ohlsson & Nason 1994) and can be for example assessment, teacher training or formative evaluation. However, some of these other uses of the learner models involve the need to communicate the  $\mathcal{LM}$  to the outside, or, what we call, the need to "externalise the learner model". In this paper we explore the process of externalising learner models so that they can be used for many possible situations by different agents.

A learner model is externalised if it can be inspected, used and manipulated not only by the system that uses it but also by a different agent or the human agent that is modeled.

This paper is organised as follows: first we will discuss the applications of the process of externalising learner models; then, we will present the approach taken in TAGUS's (Paiva & Self 1994b) to represent learner models using a cognitive approach to express mental states. Then we will present some examples of how TAGUS can be used in the process of externalising learner models. Finally, we will also discuss some of the possible advantages of this approach and relate it with other work in learner modeling.

## 2 Possible uses of Externalised Learner Models

Externalising learner models is the process of making the learner model (acquired and represented within a modeling system) accessible and usable by other agents different from the system that usually maintains it.

The main idea is illustrated in Figure 1 which shows a situation where three users can access the learner model: the learner; the teacher and the systems designer.

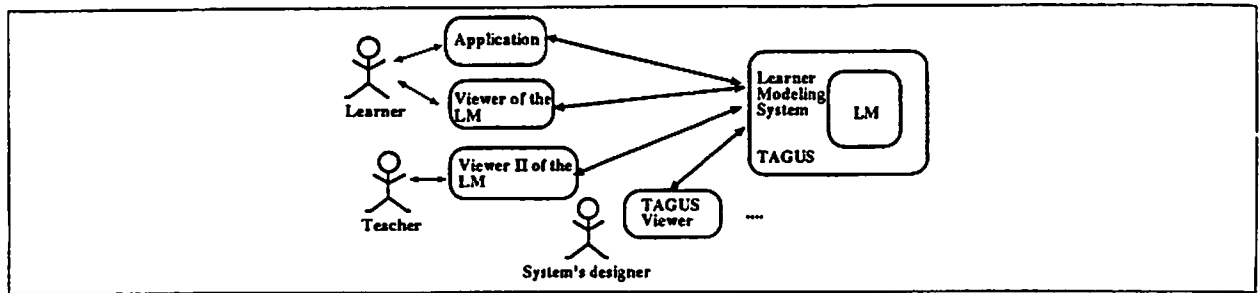


Figure 1: External Users accessing the Learner Model

The idea of allowing learners to inspect models of expert knowledge, presented in (Goldstein 1982), is a well known approach and was introduced in the *glass-box expert* where the fundamental characteristic is that the expert reasoning is inspectable by students. What we propose here is not the learner inspect the knowledge of an expert, but rather that the content of the knowledge ascribed to that learner, i.e. the learner model, be inspectable by other agents, including the learner. Some of the possible advantages and uses of this externalisation are:

**Testing the learner modeling process:** Learner modeling processes can be a compound of many different techniques, some more appropriate to certain types of situations and others more adapted to others. The process of specifying and constructing the mechanisms for creating and maintaining the learner models are greatly improved if some tools are provided to facilitate the access to and the change of these models, giving some transparency to the modeling techniques used (in this case, the external agent is the systems designer and we will illustrate this case later in this document).

**Self assessment** Students like to know how far they have reached in the learning sessions, and what are the problems they haven't solved and need to. The externalisation of the learner model can provide the learner with access to her own model, and therefore give her a way of assessing her learning status and performance. Self assessment mechanisms have proved as well to be good vehicles to stimulate motivation (for example (Anderson 1994) introduced an assessment ruler in the interface of a ILE for that purpose).

**Promote reflection** One of the main goals of Intelligent Learning Environments is to provide environments such that the learners not only can engage in problem solving activities but also are able to reflect upon the activities performed. For instance in (Collins & Brown 1988) some functionalities systems can provide that improve in different ways the process of reflection are presented. The functionalities illustrated are: Imitation; Replay and Abstracted Replay. It is argued in (Collins & Brown 1988) that tools which allow the learner to replay their performance (or abstract replay) and imitate some performance, provide a good way to improve reflection over such performance. In order for a tool to provide such functionalities it needs to externalise what is known about the learner (her actions and what has been inferred about her).

**Interactive diagnosis** Diagnosis is one of the most important and difficult activities the learner modeling subsystems need to perform when creating learner models. In many situations the process of deciding what to store about the learner is hard, depends on uncertain mechanisms and in many cases is crucial for the interaction and motivation of the learner. One of the ways to deal with the complexity of this diagnosis process is to make it interactive and therefore the learner herself can play a role in the diagnostic procedure. For that to be possible, the learner model must be usable and changeable by the learner herself. So, whenever the learner modeling system is unable to decide upon which beliefs to trust, the learner may be able to help in this process.

**Students assessment by the teacher** The collections of learner models constructed by the learner modeling systems can be given (externalised) to the teachers such that an of evaluation of the population of students can be made by them using those externalised models.

**Teacher training** As (VanLehn et al. 1994) has argued, many situations occur where teachers can test some techniques with simulated students. If the process of learning is simulated by the learner modeling system, and it can be externalised to the teacher, then it is possible to test teacher practices and intervention techniques to determine which ones perform best according to the simulated student.

### 3 Externalising the Learner Models maintained by TAGUS

Externalising a learner model is a simple idea: make the model accessible to be understood by other agents different from the system itself. Although the idea is simple, its realization is not as easy as it may at first glance appear. Indeed, we are dealing with a problem of knowledge communication between agents.

One of our main goals is to achieve mechanisms such that external agents can access the externalised models that satisfy the following properties:

- *Understandable* : the content of the model should be clear and motivated by cognitive based assumptions that describe humans mental states, so that it can be understandable by other agents.
- *Transferable* : the system that allows the externalisation must be able to communicate knowledge with other agents.
- *Usable* : the externalised models must have a degree of abstraction that makes them usable by the other agent. For example, if the model is too fine grained, and detailed, it might be useless for a self assessment tool.

Because TAGUS acts as an independent agent of the application, the process of externalising a learner model is a process of knowledge communication between the system that keeps the model (TAGUS) and the agent that is going to use it. So, TAGUS is a system that acts as a server for other applications or computational agents, providing the functionalities of acquiring, storing and changing the learner model by other agents.

#### 3.1 The content of the Learner Model in TAGUS

One of the questions one may ask about this external model is whether the content of the model can be relevant to the other agents. Learner Models may contain a diverse set of entities about a learner, such as: actions; goals; plans; characteristics; beliefs ; problem solving strategies; stereotypes and justifications.

In the case of TAGUS, the *LM* stored describes several types of attitudes and characteristics of the learners. As far as their knowledge state is concerned, we adopted a representation that distinguishes between the *beliefs* held by the learner, her reasoning capabilities, which are explicitly represented as *reasoners*, and her knowledge about strategies for solving problems, explicitly represented by *monitors*. So, the cognitive model adopted stores a learner's state of knowledge stratified in a *belief level* (a model of the learner's beliefs) *reasoning level* (a model of the reasoning and problem solving capabilities of the learner) , *monitoring level* (a model of the learner's tactics and control in solving problems) and *reflection level* (a model of the learner's methods for changing the other levels).

For example, in the domain of Algebra, TAGUS was used to maintain a model of a learner for a simple ILE (see also Figure 5), and an example of a model maintained is the following:

```
b('singleValue(43)').
b('equation(x=45)').
b('singleVariable(x)').
b('equation(2+x=45)').
%---REASONERS
final :: solve(equation(LHS,RHS)) <-
          b(singleVariable(LHS)) and
          b(singleValue(RHS)) .
%-----MONITORS
solved(Goal) if applicable(final,Goal)
                then apply(final,Goal).
continue(Goal) if applicable(Reasoner,Goal)
                then apply(Reasoner,Goal).
giveUp(Goal) if b(tooComplicated(Goal))
                then giveUp.
giveUp1(.) if b(tooManySteps)
```

then giveUp.

This example shows some beliefs held by the learner, such as  $b(\text{singleVariable}(x))$ , which states that the system is assuming that the learner knows that "x" is a single variable. It also shows one type of reasoning activity defined as a reasoner, final, expressing the situation where the learner is able to determine when an equation is solved. This model also presents some monitors, such as  $\text{giveUp}(\text{Goal})$ , showing the strategy whereby the learner gives up solving a goal if the equation to be solved is too complicated.

Apart from the beliefs, reasoners and monitors, the *LM* also contains information about actions, goals, characteristics of the learner, such as her age, or any other features that are needed for the interaction. For example, a goal stored in the above model is  $\text{solve}(\text{equation}(2*x=45))$ .

### 3.2 The steps in the externalisation

The externalisation of the *LM* stored by TAGUS is carried out by Viewers. Because the *LM* may contain many features which are not relevant to other agents the Viewers need to filter which information to provide to the outside, and select in which form (language) to provide such information. Meanwhile, TAGUS behaves as a server of those Viewers, providing them the appropriate functionalities to manipulate the *LM* stored.

For instance the actions performed by the learner may not need any filtering or they may be completely irrelevant for the outside (if the action is too fine-grained as in the case where the actions stored are the use of the left button to select a certain operation). Considering the following learner model :

$$LM = \{ b(\text{'equation}(x=45)'), b(\text{'singleValue}(43)'), b(\text{'singleVariable}(x)'), \\ b(\text{'equation}(2+x=45}') )$$

a filter is a function such that only the relevant information is selected:

$$Filter\_Student\_Assess(LM) = FLM$$

such that *FLM* has the properties of being understandable by the learner and using a language that the learner understands. In this case, the *FLM* can be for example:

$$FLM = \{ \text{single value} : 45, \quad \text{single variable} : x, \\ \text{equation} : 2 + x = 45 \}.$$

Note that the belief that  $b(\text{'equation}(x=45)'),$  is not externalised because it is assumed that as far as the learner is concerned  $x=45$  is a solution and not an equation. However, it is kept within the model as shown for for problem solving reasons.

Perhaps the main difficulty in the externalisation is the language problem. If the environment is just a problem solving domain, the content of the model are symbols which do not signify any relevant fact to the learner, they are only meaningful to the system to choose how to continue, as in the above example. In one of TAGUS Viewers the problem of language communication is dealt by asking the other agent (that can be the application) about the language to be used. This solution is also followed by some knowledge communication protocols (as KQML (Finin 1993)) where the language is one argument within the exchanged messages.

### 3.3 Externalising Problem Solving and Changes

In TAGUS the problem solving performed by the learners can be simulated within the model using the beliefs, reasoners and monitors as an expression of the knowledge of the learner. The simulation executed by TAGUS results in the construction of a proof of a goal (the problem to be solved). Thus, the process of externalising the problem solving involves using the proofs constructed and transferring them to the external agent.

In Figure 2 an example of such a simulation is shown. In this case, the result is shown to the person that is testing the modeling processes, therefore the proof is shown by showing the justifications for all the steps in the proof, goals, subgoals, reasoners used and justifications for each step.

In order for the viewers to be able to construct these simulations TAGUS provides the following services:

- *SimulateProblemSolving\_ULM* : In this case, the other agent gives a goal (a new situation) to TAGUS and the result given by TAGUS is a proof of that goal (even if the learner cannot solve the problem, as in the example of Figure 2). So:

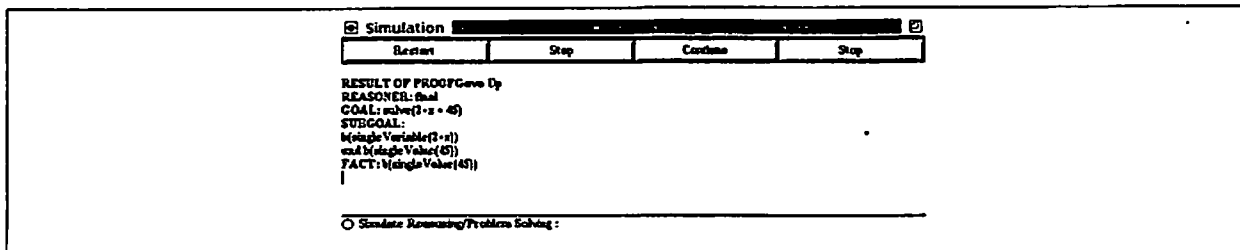


Figure 2: Simulation of a proof

#### *SimulateProblemSolving\_ULM : Situation \* LM → Proof*

A *Proof* given by TAGUS is a sequence of actions that the learner can carry out, in order to achieve the desired goal. For TAGUS to simulate such a proof, it uses a meta-reasoner (which is one of the components of TAGUS). For example, in Figure 2 the result of a proof constructed by TAGUS and shown by a viewer to simulate a learner solving the equation  $2 + x = 45$  is shown. Because the only reasoning capability ascribed to the learner is the reasoner final (see example in section 3.1), which identifies the final state of an equation, the simulation shows that the learner will give up trying to solve that equation. Meanwhile, all the entities kept in the model by TAGUS have justifications associated which are kept by a subsystem AMMS (see (Paiva & Self 1994a)). These justifications can also be shown to the external agent within the Viewer of the model, as in the above example.

- *SimulateChange\_ULM* : TAGUS can also simulate changes performed by the learner herself. A learner change is a detected change of knowledge in the learner, for example, when a learner acquires a new fact, or when a learner revises her beliefs (the process of simulating conceptual change by the LMs in TAGUS is further developed in (Paiva, Self & Hartley 1994)).

The two above services are used by the viewers to obtain the dynamics of the models.

### 3.4 Changing the Learner Model using the Viewers

The role of the viewers is not only to allow access to the information of the LM but also to change it, whenever that is possible. Because TAGUS provides some functionalities to inspect and change the content of the LM by the other agents, these functionalities (or services) can all be used by the viewers. The main services provided by TAGUS to update the content of the model are (see (Paiva & Self 1994b)):

\* **Add information to the LM** Whenever the external agent wants TAGUS to include some new information in the LM, the viewer will need to use the service *expand*. The *expansion* of the LM will result in a new LM, but no inferences will be drawn from the new added content.

\* **Revise information in the LM** Whenever the information of the LM needs to be changed in order to include some new facts inconsistent with the actual LM, the *revision* service can be used. One should note that the revision process in TAGUS does guarantee that the new fact is going to be included in the model. This change is similar to the performative "achieve" in KQML (Finin 1993) and in a similar way embedded implicitly in the "tell" messages of BGP-MS (Kobsa & Pohl 1994).

\* **Tell information to the LM** Whereas the *add* service assumes that the content is definitely going to be added to the LM, a *tell* service gives the TAGUS a new observation or situation of the environment in which the user and learner is embedded. This means that TAGUS is not asked to add something to the model, but rather it is told about a new situation, and it is its role to decide what to do with this new information.

\* **Contract information in the LM** The service associated with the removal of information from the ULM is called *contraction*.

The above services are a small subset of the update facilities provided by other knowledge communication languages such as KQLM, however, they allow the Viewers to access and change this special type of model maintained by TAGUS.

The Viewers can use these services as in the case presented in Figure 3. In this example, a "special user" (system designer) can inspect the model of the learner by activating the viewer of the LM using

the interface (presented in Figure 3) to manipulate the model. In this interface, the set of functionalities provided uses the services provided by TAGUS, and allows the system designer to test and evaluate the model kept by TAGUS.

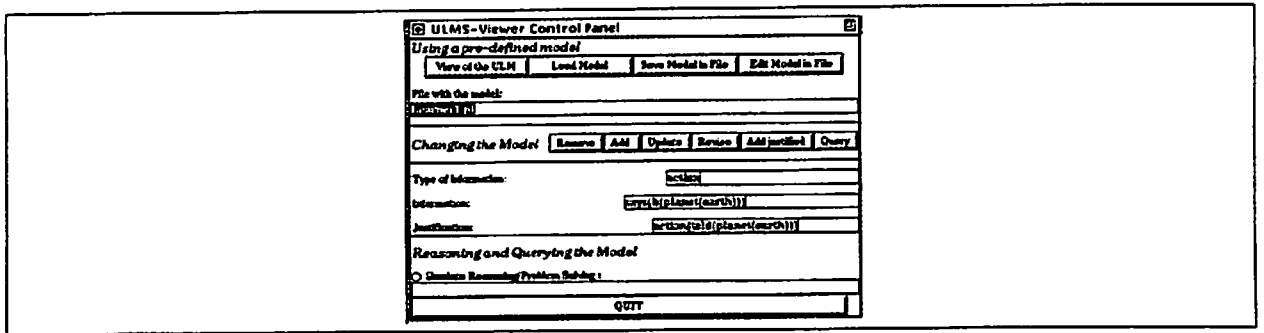


Figure 3: Control Panel of a LM Viewer

### 3.5 Externalising application specific *LMs*

Finally, in another situation TAGUS provides services to two systems at the same time, an ILE and a Viewer, allowing the externalisation by the Viewer of the model corresponding to the learner using that ILE. The ILE interacts with a student via a graphical interface, where a student solves problems (algebraic equations) by using a predefined set of operators, such as 'Divide both sides by' (see Figure 4). A number of equations are established beforehand, and they are characterized by different degrees of complexity and emphasising one or another operator. The environment provided gives the learner a vast set of operators to choose from which are used to execute each step during solving of the problem. The path taken is shown to the learner, who can backtrack and choose another way of solving the same equation.

From that interaction, a model of the student is being built based on some acquisition mechanisms that were beforehand included in TAGUS. The communication between this simple learning environment and TAGUS is also based on the set of functionalities TAGUS provides. This allows the ILE to give to TAGUS the actions the learner performed. At the same time, TAGUS keeps a record of those actions and uses them for determining which level the learner belongs to.

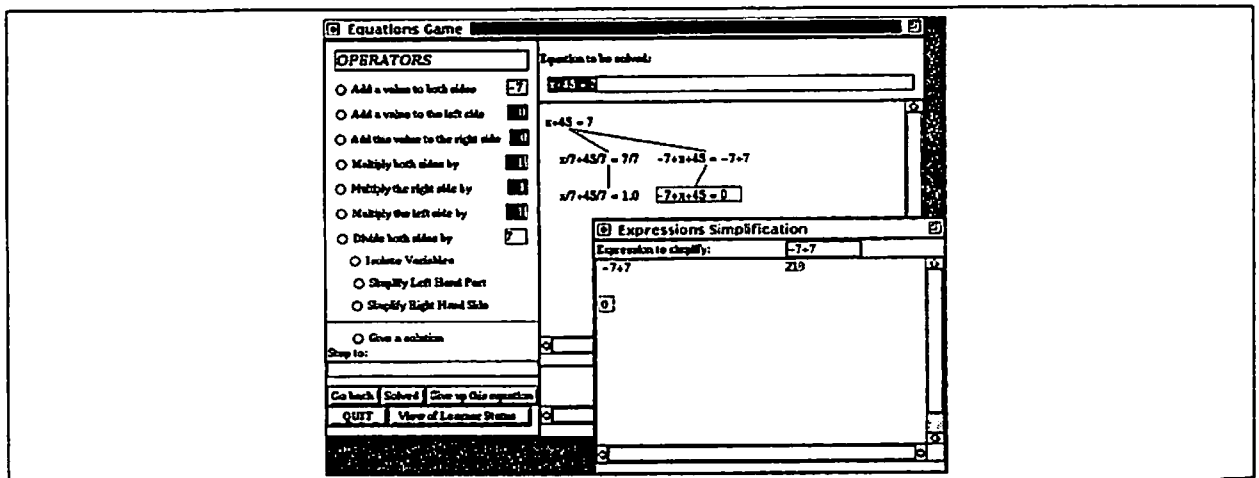


Figure 4: An environment to solve equations

Meanwhile, the constructed learner model can equally be inspected by using the appropriate Viewer (see Figure 5), allowing any agent (including the learner) to trace the content of the learner model being acquired while the learner solves the problems presented by the ILE. This type of externalisation can be explored in the future as a way to promote reflection and self assessment.

## 4 Related work

Externalising learner models, in particular to the learners themselves, has already been explored in simple situations. For example, in the area of user modeling, some experiments have already been carried out (see (Kay 1994)) in giving the users access to the model maintained by the system. In this case, the access is supported by a set of tools to enable the user to see the model in particular a model viewer which is a graphical interface given to the user to manipulate the model represented as a graph. The results presented in (Kay 1994) show that it is encouraging the idea of giving the users their own models. The externalisation made in this case was nevertheless based on an simple model, and tied up with the application and with one viewer to be used only the the learners.

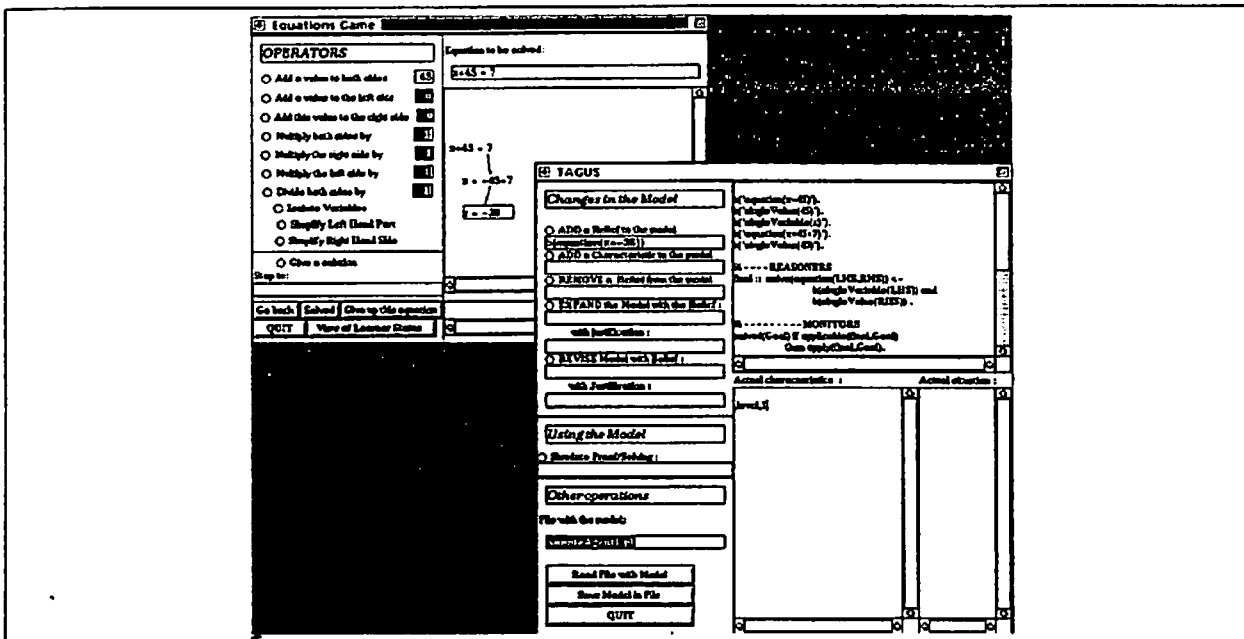


Figure 5: Example of an ILE and the use of TAGUS

In the case of the learner's modeling the idea of externalising to the learners can be found as a new way to promote reflection in the learners. For example, in the case of SHERLOCK (Lesgold, Lajoie, Bunzo & Eggen 1992) students engage in a problem of troubleshooting, and after the students have solved the problem, they can use RFU to reflect on their performance by reviewing their work with the assistance of the computer based coach (see (Katz, Lesgold, Eggen & Gordin 1992)). The purpose of the RFU is to help students to internalise the curriculum issues that SHERLOCK teaches. During the RFU the students can replay their actions step by step. In the case of TAGUS, because the system is independent from the application, the student herself can set up a situation to be simulated for her own model, and see what happens, and not simply replay the performed actions.

A different approach that involves externalisation but with a different goal is taken by EXPITS (Takeuchi & Otsuki 1992), which is a system where the model of the learner maintained can be inspectable, and the mechanisms of modeling tested. However, its utility is confined to the exploration of learner modeling techniques with simple learner models (overlays) representations.

Finally, the work on OLAE (Martin & VanLehn 1993), uses as well mechanisms of externalising the bayesian learner model to an external assessor.

## 5 Conclusions

In this paper, we explore the idea behind the process of making learner models accessible to several different users or agents: externalising learner models. To do so, we presented a system called TAGUS where learner models and learner modeling processes are independent from the application. This independence is used in the construction of Viewers for the Learner Models, so that its accessibility is given to several agents. This system, TAGUS can be seen as an agent that interacts with other computational

agents, providing a set of services for acquiring and maintaining learner models. We illustrated some of the functionalities of TAGUS, that are fundamental for the externalisation of the learner models.

## References

- Anderson, J. (1994), Cognitive tutors: Lessons learned (presentation), in B. Goodman, A. Kobsa & D. Litman, eds, 'Fourth International Conference on User Modeling'.
- Collins, A. & Brown, J. S. (1988), The computer as a tool for learning through reflection, in H. Mandl & A. Lesgold, eds, 'Learning Issues for Intelligent Tutoring Systems', Springer-Verlag.
- Finin, T. (1993), Specification of the kqml agent-communication language, Technical report, The DARPA Knowledge Sharing Initiative.
- Goldstein, I. P. (1982), The generic graph: a representation of the evolution of procedural knowledge, in D. Sleeman & J. Brown, eds, 'Intelligent Tutoring Systems', Academic Press.
- Katz, S., Lesgold, A., Eggen, G. & Gordin, M. (1992), 'Modeling the student in sherlock ii', *Journal of Artificial Intelligence in Education* 3, 495-518.
- Kay, J. (1994), The justified user model: a viewable, explained user model, in B. Goodman, A. Kobsa & L. D., eds, 'Proceedings of the International Conference on User Modeling 1994', MITRE Corporation, USA.
- Kobsa, A. & Pohl, W. (1994), 'The user modeling shell system bgp-ms', *User Modeling and User-Adapted Interaction*.
- Lesgold, A., Lajoie, S., Bunzo, M. & Eggen, G. (1992), Sherlock: A coached practice environment for electronics troubleshooting job, in J. Larkin & R. Chabay, eds, 'Computer-Assisted Instruction and Intelligent Tutoring Systems', Lawrence Erlbaum Associates.
- Martin, J. & VanLehn, K. (1993), Olaf: Progress toward a multi-activity, bayesian student modeler, in P. Brna, S. Ohlsson & H. Pain, eds, 'Artificial Intelligence and Education, 1993', AACE- Association for the Advancement of Computing in Education.
- Paiva, A. & Self, J. (1994a), A learner model reason maintenance system, in T. Cohn, ed., 'ECAI'94 - Proceedings of the European Conference on Artificial Intelligence', John Wiley & Sons, Ltd.
- Paiva, A. & Self, J. (1994b), Tagus - a user and learner modeling system, in B. Goodman, A. Kobsa & D. Litman, eds, 'Fourth International Conference on User Modeling', The MITRE Corporation, USA.
- Paiva, A., Self, J. & Hartley, R. (1994), On the dynamics of learner models, in T. Cohn, ed., 'ECAI'94 - Proceedings of the European Conference on Artificial Intelligence', John Wiley & Sons, Ltd.
- Takeuchi, A. & Otsuki, S. (1992), Expits: an experimental environment on its, in C. Frasson, G. Gauthier & G. McCalla, eds, 'Intelligent Tutoring Systems', Lecture Notes in Computer Science, Springer-Verlag, pp. 124-131.
- VanLehn, K., Ohlsson, S. & Nason, R. (1994), 'Applications of simulated students: An exploration', *Journal of Artificial Intelligence in Education* 5(2), 135-175.