LEARNER KNOW THYSELF: STUDENT MODELS TO GIVE LEARNER CONTROL AND RESPONSIBILITY

Judy Kay
Basser Dept of Computer Science
University of Sydney
AUSTRALIA 2006
Tel: +(612) 9351-4502 Fax: +(612) 9351-3838
email: judy@cs.su.oz.au

Abstract

We describe an approach to student modelling where the student can delve into the student model. We describe how this can support learning on several levels. It serves as a basis for planning learning goals, improving communication between the teaching system and the learner, and as an aid to reflection about learning.

There is a growing group of researchers working with accessible and understandable student models. We outline the motivations for such approaches.

This paper presents the underlying philosophy for the design of our student model and shows the type of model we have built. We discuss problems of scrutability we have encountered and the interesting research agenda they define.

1. INTRODUCTION

In 1989 we began an ambitious study: we would monitor all use of the sam text editor. Since our students could use this editor only on our machines, we would have a record of the total use of hundreds of users over several years. We hoped to track the development of expert users of the editor from their very first meeting with it through three years of use. We had several motivations for this study. We expected it would

1. further our understanding of the learning of text editing, a computing activity performed by large populations of users;
2. improve our ability to support life-long learning which should occur both as users tasks change and as their consolidated knowledge and skills can serve as foundations for further learning;
3. enable us to develop techniques for building high quality models of user's knowledge from the huge amounts of very low grade student modelling data we could collect by monitoring all use of the editor;
4 enable us to build computer-based learning support for users of the text editor.

The learning problems suffered by users of tools like text editors have much in common with many interesting and important problems in broader educational settings. There are many barriers that stand in the way of becoming an expert user of a text editor; just as there are many barriers to prevent the student in any learning area from achieving the highest levels of knowledge and understanding. In the case of learning about text editors, one such barrier has been called the *production paradox* (10): the user knows they could work more effectively if they took the time to learn more about the text editor ... but they do not have the time to put into learning!

Why might this be this be so? Most readers of this paper will be non-expert in the use of some basic tools like text editors. They will know some of the reasons for the production paradox. Perhaps the most important is that the cost of learning is too high for the short term rewards. It is just too hard to learn about something that would be really useful for achieving current goals.

It is very appealing to imagine that a knowledgeable coach could look over our shoulder as we work. Then, at just the right moment, the coach could point us in the right direction for learning about an aspect of the editor that is both useful right at that moment and that we are really ready to learn. This is not a new idea. It has been part of the earliest ITS work (for example 9).

Early ITS work was characterised by the goal of emulating excellent individualised human teaching. The individualisation was driven by the student model. In the context of our work with the text editor, a classic teaching system would have the following components:

- expert knowledge of the text editor;
- expertise in teaching about it;
- a good model of the student's knowledge, skills, learning styles and preferences;
- ability to communicate with the text editor user and potential learner.

We know that it is costly and difficult to build such a teaching system. One of the more problematic parts is the student modelling.

Our claim is that there is a much cheaper and highly effective alternative: we can design the student model with the intent that it be externalised in order to

- help the learner become aware of their current knowledge in ways that enable them to
- identify suitable learning goals and
- facilitate pursuing them.

We believe that the goal of modelling the student at the level that an excellent teacher can do is an unnecessarily difficult goal. Computer-based teachers are fundamentally different from human teachers. Because the former are deterministic and driven by a program, it should be possible to state what aspects of the student model affected a particular teaching action. It should equally be possible to state how a change to the student model would affect the teaching in the same situation. We believe that if the student model can be helpful in determining how a machine teaches, it should be possible to make it available to the student so they can improve their own learning through better self-knowledge.

This paper elaborates on our reasons for repurposing the student model as a basis for self-knowledge. First, however, we discuss the reasons for our approach and the character of student modelling information that is appropriate.

2. WHY KNOW THYSELF?

Our primary arguments for helping the learner know their current learning status are utilitarian and pragmatic. It should be easier to build a system if we can share the responsibility for identifying learning goals between the teacher and the learner. At the same time, giving the learner responsibility for their own learning seems to enhance learning effectiveness. This is the thrust of many teaching strategies, for example problem-based learning (2, 14, 24).

2.1 Life-Long, self-directed learning

Life-long learning must be self-directed: the learner should be able to set personal learning goals, establish plans for achieving them, assess their progress towards goals and revise learning plans. Essentially this involves self-knowledge, planning, reflection and review. To learn to learn effectively, students need to develop the skill of acquiring self-knowledge.

2.2 Getting Teacher and Learner Goals Synchronised

For effective communication between teacher and learner, it is important that the teacher's underlying goals which drive the teaching be consistent with those of the learner. The learner should know what it is they are trying to learn in the current learning context. We have argued elsewhere (12, 13) that in conventional educational settings (with human teachers and students) difficulties occur if there is a mismatch between the teaching goals and the learner's perception of them. If we are to construct effective machine-based learning environments, we need to deal with this problem. Typically, the student model represents various learning goals. Making these available to the student in a suitable form should ameliorate this problem.

3. WHAT TO KNOW

There are many different types of self-knowledge that can facilitate learning. The student could profitably seek to answer questions like these:

What do I know?

> Learning must build upon foundations of existing knowledge. In self-directed learning styles like problem-based learning, an important starting step in planning learning is to identify what is already known and useful. This is partly as a basis of planning the problem solving. It also increases awareness of things that are known and should be kept in mind for the current and potential learning goals.

How well do I know a particular aspect, X?

> There are many degrees of knowing. Some ways of describing this are captured in taxonomies like that of Bloom (1) or Merrill (19). A learner should be able to assess the degree of their competence and knowledge.

What do I want to know? or Do I want to know a particular aspect, Y?

> This means being aware of the current learning goals.

How can I best learn X?

> To plan learning, it is important to know how to go about learning. If the learner has control over the choice of learning approach, they need help is selecting the most effective one for them. A student model may represent learning performance when using various different learning strategies and this may help the student appreciate which enables them to learn best.

3.1 Relativity Theories of Learning

To this point, we have considered the learner in isolation. In fact, students compare themselves with others. Depending upon the student, they might wish to judge their knowledge in terms of

- how much the 'average' student knows - enables the student to see themself against the average peer;
- how much the very top students know - important for students who want to be high achievers as well as for meaningful learners who want to appreciate what the more competent of their peers know;
- how much a particular expert knows - another benchmark that is appropriate for students aspiring to achieving expertise;
- the things known by most experts - given that different experts do not know everything about a complex domain and different experts may have different views, this is the common ground for experts;
- how much a respected teacher recommends knowing - a benchmark from another authority figure;
- the knowledge required to pass the exam - a common goal for students and related to the respected teacher benchmark;

If we are to assist the student in understanding their student model and using it as a basis for setting learning goals, we need to enable them to see themself in relation to an appropriate standard. For different students and in different contexts, the type of standard will differ.

3.2 Peer Models

The first two bases for comparative student models in Section 3.1 relate the student's model to that of their peers. These examples above are by no means exhaustive. The critical point is that the student's ability to assess or know themselves may only be meaningful in relation to what other students know. Studies like our long term observation of text editor users can give detailed comparative models. Far more modest enterprises can provide simpler comparative models. For example, experienced teachers can often recall common misconceptions and levels of achievement at particular points in the learning process. In particular, examination performances may document this. Alternatively, the student may inspect the models of arbitrary peers, an approach that has been explored by Bull (4, 5).

3.3 Domain Expert or Overlay Models

Overlay student models have been part of the earliest teaching systems. They give one important basis for comparisons of knowledge. If the student aspires to become an expert, they need to see an expert's knowledge as their goal. They will assess themselves against that in setting new learning goals and measuring their progress.

The definition of an expert model is no simple matter as our list in Section 3.1 indicates: experts can have different views and even experts do not know everything. Although many teaching systems have created student models from overlays on expert models, this problem has not been addressed. This may be due to the fact that student models have been hidden and any one expert model is probably good enough for the system to teach from. (And, of course, there is a significant cost in modelling one expert!). The interesting matter of how much of a domain model is known or agreed on by experts arises when we want to share the model with the student.

3.4 Teaching-Expert Models

This class of comparative models covers the last two cases in Section 3.1. It enables the comparison against an expert to be tempered by advice from a respected authority in the form of a teacher. We would expect that this would normally be a subset of some expert models. But it might also model common misconceptions that the teacher would expect a student to have overcome at certain stages of their learning. In the case of sam, for example, we observed that many of our students seemed to think that the best way to quit the editor was by killing the sam-window. We consider this a misconception and model it.

4. THE STUDENT MODEL AS A FOUNDATION FOR SELF-KNOWLEDGE

Literature on student modelling is quite consist in defining a student model as a machine's representation of its beliefs about the student. See, for example Holt, Dubs, Jones and Greer (16.) Of course, this is normally assumed to be an internal representation intended only for use by a program.

There is a growing body of work recommending that student models be made available to the student as well as exploring ways to do this effectively (3, 4, 5, 6, 7, 15, 20, 21, 22).

We have explored several approaches to making the student model accessible to the learner. Figure 1 shows one example of a summary form of a student model for the sam text editor. This interface is from one overviewer for the student model. It gives a quick overview of whether the user appears to know the editor elements displayed. The dark squares indicate the user knows that element and the white ones indicate lack of knowledge. Nested shapes, like the nested squares for *default-size_k* indicate there is conflicting information about this aspect and the overviewer could not determine whether the user knew it or not. The overviewer allowed users to 'correct' the model and get explanations about it.

The structure we have used for the student model of Figure 1 is highly utilitarian: it classifies the things one might know about sam in terms of their overall utility. The aspects highest on the screen are the most basic. This user's knowledge is mainly restricted to basic aspects. One anomaly is the xerox command shown somewhat below the middle of the display.

Note that this overviewer collapses many aspects of the editor. Each round node represents several components of the model. The overviewer has heuristics for selecting the most 'interesting' aspects to display. So, for example, this user was deemed not to know any of the 'powerful' aspects so the summary node for these is shown as not known and it is not expanded. By contrast, the simpler aspects are expanded so that

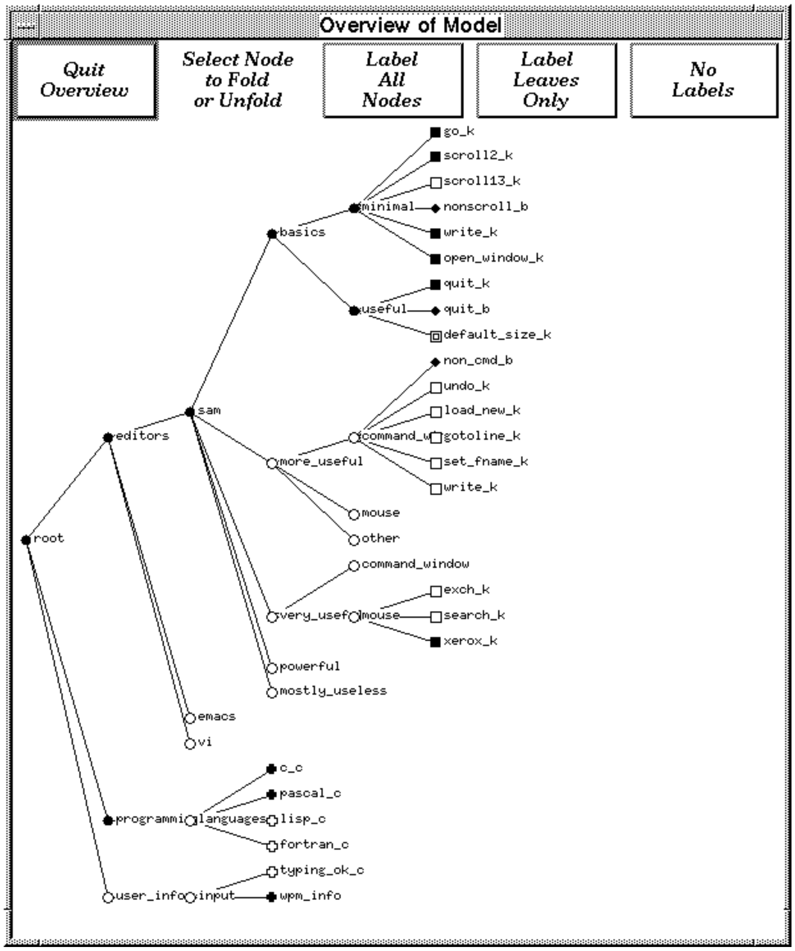

Figure 1. Overview of a student model of editor knowledge

the user can see that some aspects are judged known and others not.

A display of a student model like this can serve as a basis for the learner to reflect on their knowledge. The student model in Figure 1 indicates that the student knows only a small amount of sam's functionality.

Our student model interfaces could provide more detailed information about the student model. One aspect of this is that the student could access a description of the meaning of each component of the model. So, for example, the student could access a detailed explanation of what the node labelled xerox_k means.

Another dimension of the detail we make available to the student is illustrated in Figure 2. This shows the list of evidence the system holds about one component of an hypothetical student model. The first eight lines are various forms of evidence indicating the modelled aspect is true. The last three are opposing pieces of evidence. As the display indicates, the user can select a line to get a detailed explanation of its meaning. For example, the first line indicates the evidence was **Given** by user them**self** using the **viewer** program at the time shown at the right. Other evidence items are based on observations of the user. If a student studies this list of evidence, they can recall the various recorded events that informed the student model.

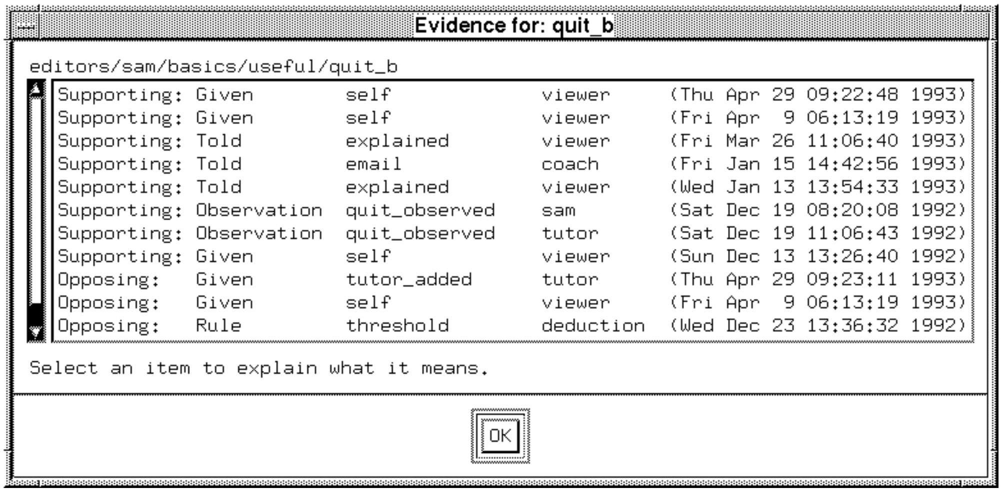

Figure 2. Example of evidence

5. RESEARCH AGENDA

As section 2 and 3 suggest, there are many interesting and difficult issues is employing student models as a basis for assisting the learner in achieving self-knowledge.

5.1 Representations for Students Models

Some representations are better for supporting student reflection than others. We have used an endorsement-like evidence-based approach since it preserves a record of all the events used to conclude about the student. It enables us to provide a list like that in Figure 2.

An open question is the trade-off between making a student model that is effective for the use of teaching systems and making it accessible to the student. Should we omit aspects from the model if we find it too hard to display them for the student? This is a particularly difficult question when the student model is generative, as for example in TAGUS (21) where the system can infer beliefs not explicitly stored in the model. It is impractical (and typically impossible) to display such a model in its entirety, with all inferences expanded and explicitly shown.

5.2 Visualisation

Our work with sam involved only ninety concepts and it was easy to organise these into a hierarchy. This made it feasible to display the model as in Figure 1, with manageable amounts of information on the screen at any one time and an intuitive navigation mechanism to the hidden parts of student model.

In general, student models may be larger than this. If there are hundreds of concepts or if they do not form a natural hierarchy, the interface to the model will need to employ visualisation tools.

5.3 Multiple views and perspectives

Figure 1 shows one view of the text editor. There are many others. For example, if a user knows another editor very well, it might be sensible to organise the sam concepts in terms of how well they match the known editor. Those concepts that are identical might be in one group, those that are completely different in another and those that are similar but likely to cause interference in learning in yet another. Depending on which editors the user knows, the organisation would differ. A quite different perspective on the sam concepts could be in terms of the primitive functions of editors. This would group the facilities for moving around the file, with the most basic and highly sophisticated classed together. Another class would have the commands for managing the interface to the file system, and so on.

If we want students to think about their knowledge in different ways, it might be valuable to present different views of the same knowledge. These may correspond to the different views taken by different pieces of teaching software. On the other hand, it might constitute a whole level of infrastructure exclusively to support student access to the student model.

Quite another set of views can be snapshots of learning. If a learner wants to review their progress and achievements, they could ask for a display of the student model as it was at various important points in time. This approach could be supported by Mr Collins (8) and by the our representation since it is feasible to reconstruct the model as it would have appeared at various times.

5.3 Ontological Explanations

Students need to be able to determine what the student model is modelling. For example, our model shows xerox: if the learner is to make sense of our display and the student model, they need to be able to find out what xerox means.

Any student model is based upon an ontology of concepts the system designer deemed important. We require that each component of the student model has an explanation that can be accessed by the interfaces to the model.

5.4 Teaching Strategies to Exploit the Student Model

To this point, we have assumed the learner will study their user model. In our sam work, we mailed students, telling them that they could access a model of their knowledge of sam. Of the eighty-two students mailed, forty-six (56%) made some use of it. Detailed analysis of the heavier users indicated that some students explored their model extensively (11).

Of course simple exploration of the model is not at all the same thing as using it effectively for learning. There are many ways that this might be encouraged:

- Ask the student to use the model to identify areas to revise. We would expect the student to search for aspects they had studied and the model showed as not well known. The student may want to explore the reason that the model makes this judgement.
- Ask the student to reflect on their knowledge, summarise what they consider they know and do not know and then compare this with student model (13). Where there is a mismatch, one might expect the student to delve into student model to see the bases for the machine's reasoning.
- A human teacher might sit with the student to study student model. Similarly, a peer (4).
- We might integrate an assessment and model access. For example, a student might be asked to construct a concept map (17) as a diagnostic task that feeds into a student model. At suitable points in the process, the student might inspect the student model which would represent the system's inferences about their knowledge.
- Create opportunities for learner to disagree or negotiate with the machine. These serve as a tool for reflection (6).

6. CONCLUSIONS

Our sam study tracked over two thousand users for at least a year and several hundred over a three year period. We found that the largest number of commands were learnt in the early periods of use and that more subtle but significant learning continued throughout the time that students continued to use the editor (18). However, the amount learnt by most students were quite modest. For example, in the first three year cohort, only about a third of the students ever used the undo-command. Our study has enabled us to build models of the 'average' student. We began the work with clear models we expected an effective user would achieve. Both could be useful benchmarks for students to compare themselves against. We also explored the ways to build the student model as a support for students to be able to learn what they know.

Self (23) has identified four major roles for the student model: prediction and planning; diagnosis and remediation; negotiation and collaboration; interaction and communication. In all these, the focus is on a program using the student model. We have argued for an additional role for the student model as a first class citizen of the learning environment.

We have argued the importance of student self-knowledge, especially for life-long, self-directed learning. We have discussed the ways the student model can support the student in introspecting on what they know, deciding on learning goals and reviewing their progress.

We have also identified a range of challenging issues to be explored:

- how best to make the model available to the student;
- whether we must or should restrict the scope of modelling to match those aspects that can be made available to the student;
- how to ensure the student can understand a modelling ontology;
- how to select and highlight the most relevant and interesting parts of the students model;
- how to establish learning environments that enable the student to focus effectively on the model.

ACKNOWLEDGMENTS

This paper describes work done in collaboration with several other people: Ronny Cook, Richard Thomas, Noroja Parandeh-Ghebi, Kathryn Crawford and David Benyon.

References

1. B S Bloom et al, *The Taxonomy of Educational Objectives: The Classification of the Educational Goals. Handbook I: The Cognitive Domain,* McKay Press, New York (1956a).
2. D Boud and G Feletti, *The challenge of problem based learning,* Kogan Page, London (1991b).
3. P Brusilovsky and E Schwarz, "User as student: towards an adaptive interface for advanced web-based applications" in *User Modeling, Proc of Sixth International Conference UM97,* ed. A Jameson, C Paris, and C Tasso, Springer (1997c).
4. S Bull and E Broady, "Spontaneous peer tutoring from sharing student models," *Artificial Intelligence in Education,* pp. 143 - 150, IOS Press (1997d).
5. S Bull and P Brna, "What does Susan know that Paul doesn't? (and vicecersa): contributing to each other's student model," *Artificial Intelligence in Education,* pp. 568 - 570, IOS Press (1997e).
6. S Bull and H Pain, "Did I say what I think I said, and do you agree with me? : Inspecting and Questioning the Student Model," *Proceedings of World Conference on Artificial Intelligence in Education,* pp. 501 - 508, AACE, Washington DC, USA (1995f).
7. S Bull and M Smith, "Using targeted negotiation to support students learning," *Proceedings of International Conference on Computers in Education,* Singapore (173g - 181).
8. S Bull, P Brna, and H Pain, "Extending the scope of the student model," *User Modeling and User-Adapted Interaction,* 5, 1, pp. 44 - 65 (1995h).
9. R R Burton and J S Brown, "An investigation of computer coaching for informal learning activities," *Intl J of Man-Machine Studies,* 11, pp. 5-24 (1979i).
10. J M Carroll, *The Nurnberg Funnel: designing minimalist instruction for practical computer skill,* MIT Press, Cambridge, Mass (1990j).
11. R Cook and J Kay, "The justified user model: a viewable, explained user model" in *UM94: User Modeling Conference,* pp. 145-150, Hyannis, Cape Cod, USA (1994k).
12. K Crawford and J Kay, *Shaping learning approaches with intelligent learning systems,* 3, pp. 1472-6, Proc Intl Conf for Technology in Education, France (1992l).
13. K Crawford and J Kay, "Metacognitive processes and learning with intelligent educational systems" in *Perspectives on Cognitive Science,* ed. P Slezak, T Caelli, and R Clark, pp. 63-77, Ablex

(1993m).

14. C A Curtis, A Carver, R A Howard, and W D Lane, *A methodology for active, student controlled learning: motivating our weakest students,* Philadelphia, Pennsylvania (1995n).

15. P Dillenbourg, “The computer as constructorium: tools for observing one’s own learning” in *Knowledge Negotiation,* ed. R Moyse and M T Elsom-Cook, pp. 185 - 198, Academic Press, London (1992o).

16. P Holt, S Dubs, M Jones, and J Greer, “The state of student modelling” in *Student Modelling: The Key to Individualized Knowledge-Based Instruction,* ed. Jim E Greer and Gordon I McCalla, pp. 3 - 35, Springer-Verlag (1994p).

17. J Kay, “An explicit approach to acquiring models of student knowledge” in *Advanced Research on Computers and Education,* ed. R Lewis and S Otsuki, pp. 263-268, Elsevier, North Holland (1990q).

18. J Kay and R Thomas, “Studying long term system use,” *Comm of the ACM,* 4, 2, pp. 131-154, ACM (1995r).

19. M D Merrill, “Component Display Theory” in *Instructional Design Theories and Models,* ed. C Reigeluth, Erlbaum Associates, Hillsdale, NJ (1983s).

20. M Murphy, “Learner modelling for intelligent CALL” in *User Modeling, Proc of Sixth International Conference UM97,* ed. A Jameson, C Paris, and C Tasso, pp. 301 - 312, Springer (1997t).

21. A Paiva, J Self, and R Hartley, “Externalising Learner Models,” *Proceedings of World Conference on Artificial Intelligence in Education*, pp. 509 - 516, AACE, thesis (1995u).

22. J Self, “Bypassing the Intractable Problem of Student Modelling: Invited paper” in *Proc 1st Intl Conf on Intelligent Tutoring Systems,* pp. 18 - 24, Montreal (1988v).

23. J Self, “Formal approaches to student modelling” in *Student Modelling: The Key to Individualized Knowledge-Based Instruction,* ed. J Greer and G McCalla, pp. 295 - 352, Springer-Verlag (1995w).

24. D R Woods, J D Wright, T W Hoffman, R K Swartman, and I D Doig, “Teaching problem solving skills,” *Engineering Education,* 66, 3, pp. 238-243 (1975x).