Active Learner Modelling

Gordon McCalla, Julita Vassileva, Jim Greer and Susan Bull

ARIES Laboratory, Dept. of Computer Science, University of Saskatchewan,
Saskatoon, Saskatchewan S7N 5A9, Canada.
{mccalla, jiv, greer, bull} @cs.usask.ca

Abstract. It is common to think of a "learner model" as a global description of a student's understanding of domain content. We propose a notion of learner model where the emphasis is on the modelling process rather than the global description. In this re-formulation there is no one single learner model in the traditional sense, but a virtual infinity of potential models, computed "just in time" about one or more individuals by a particular computational agent to the breadth and depth needed for a specific purpose. Learner models are thus fragmented, relativized, local, and often shallow. Moreover, social aspects of the learner are perhaps as important as content knowledge. We explore the implications of fragmented learner models, drawing examples from two collaborative learning systems. The main argument is that in distributed support environments that will be characteristic of tomorrow's ITSs, it will be literally impossible to speak of a learner model as a single distinct entity. Rather "learner model" will be considered in its verb sense to be an action that is computed as needed during learning.

1. Introduction

It is still common parlance in intelligent tutoring systems (ITS) to speak of a "learner model", meaning a single global description of a student to be used by an ITS to judge understanding of deep domain content. In this paper we propose an alternative notion of learner model where the emphasis is on the activity and context of modelling, rather than on the global description. Focusing on the activity of learner modelling, we show how the model can be a function used to compute relevant information about one or more learners as needed depending on the purpose, learners involved and available resources. This approach lends itself to the kind of learner modelling often needed in systems coordinating many learners who communicate with one another, who form pairs or groups for learning activities, and who form opinions about one another, thus participating in some form of peer assessment. In such a setting there is

·This research has been partially funded by the Telelearning Network of Centers of Excellence under Project 6.28.

no one monolithic learner model associated with each learner. Rather the knowledge about a learner is distributed among agents who interact with that learner (teachers, other learners, software applications, web-based software agents, etc.) In future, as borders of learning environments disappear and learning environments span the web, many applications and people will hold learner model information about a learner.

Thus learner modelling is the process of assembling and summarizing fragmented learner information from potentially diverse sources. This information can be raw data recorded by a web application, partially computed learner models inferred by an ITS, opinions about the learner recorded by a teacher or peers, or a history of learner actions. The key to making sense of this widely distributed information is the ability to interpret multi-modal information from multiple heterogeneous relevant sources and integrate this just-in-time into a learner model of appropriate granularity. Integration introduces many new requirements for the learner modelling process. In this paper we discuss the implications of this sort of learner modelling.

2. Examples: I-Help and S/UM

We have chosen to illustrate our approach in two systems: I-Help and S/UM.

2.1 I-Help

I-Help provides a student with a matchmaking service to find an online peer to help with a problem [1]. The most recent implementation is based on the Multi AGent Architecture for Adaptive Learning Environment (MAGALE) [2], which uses a decentralized approach in system design and an economic infrastructure to trade knowledge resources. The MAGALE architecture comprises individual personal agents representing each user, and manages a variety of learner models. These models are created and updated by a variety of diagnostic agents. A diagnostic agent can be contacted by another agent to request knowledge about some particular learner. This happens either periodically, or when information from this model is needed. In addition, each personal agent creates models of peers, whose agents the agent has encountered through a help interaction (see Figure 1).

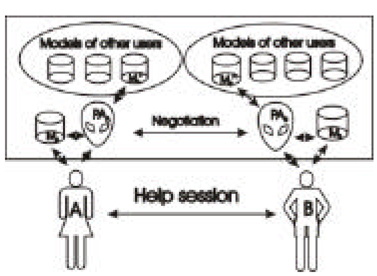

Fig. 1. I-Help: each personal agent maintains a model of its own learner and others encountered

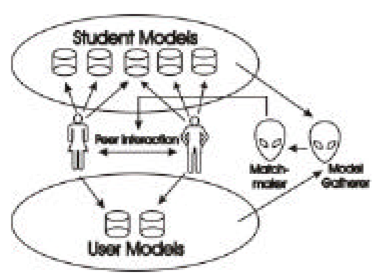

Fig. 2. S/UM: each user maintains their user model, and contributes to student models of numerous peers

2.2 S/UM

S/UM [3] also offers a matchmaking service to students, but its focus differs from I-Help. S/UM is concerned with matching learners who may offer or wish to receive feedback on some aspect of their work, or who may want to collaborate or cooperate in their learning. The aim is to arrange partnerships to promote reflection through peer interaction and peer modelling. A major goal is that the feedback *givers* should also benefit, by reflecting while evaluating a peer. The additional relationship of cooperation in S/UM concerns a double feedback/help situation: X helps Y on A; Y helps X on B. Collaboration takes its usual sense of two learners working together on a common problem or task. Peer interactions may take place either on-line or off-line.

The S/UM architecture focuses on student and user models used by a 'model gatherer' and the matchmaker. The single modeller-modellee relationship does not hold: representations are constructed from self-evaluation by the model's owner–i.e. the modeller is also the modellee [4]; and by contributions from peer modellers after peer interaction [3]. The model gatherer organises these model fragments, generating an appropriate synopsis of model contents from the multiple information sources (e.g. more weight to recent entries and assessments from competent peers). This synopsis may be of interest to the model's owner for reflection; to the matchmaker for finding suitable partners; to peers who may browse information about potential partners. A single student model may comprise many entries from different peer sources, and a single learner may contribute to any number of peer models (see figure 2).

3. Integration in Learner Modelling

As illustrated in the above systems and many others, it is often inconvenient, unproductive, or computationally difficult to maintain a single consistent global model about each learner. In I-Help learner models are derived as needed according to the person or people being modelled, the agent who is modelling, and the end use or purpose of the model [5]. In S/UM learner models are aggregated for presentation to peer viewers. We believe this emerging trend of deriving learner models from distributed model fragments will increase as learners interact with more widely distributed learning resources and applications on the Web. Continuous contact between learners and technology will allow for fine-grained tracking of learners' activities under different circumstances and by different modelling agents. The problem for learner modelling will be making sense out of too much knowledge, rather than trying to make do with too little [6]. Thus the need for integrating learner model fragments will grow, and the ideal of maintaining a single monolithic learner model for each learner will be seen as less desirable (and likely intractable).

We believe the fragmented, distributed learner model will have a significant impact on learner modelling research. The main question is how to manage the information:

- how to find the agent who has a relevant model depending on the context and the purpose for which the model is needed;
- how to make sense of possibly inconsistent or even contradictory data;
- in general how to interpret models created by other agents.

The focus is shifted from the model itself to the process of modelling, i.e. the learner model is thus not so much a noun as a verb. The learner model is computed "just in time" [7] and only makes sense in the context of who is being modelled and for what.

For clarification we introduce a simple notation. We can think of a learner model as a function: *learnerModel (a, L, p, r)*, where:

a is the agent doing the modelling,
L is the set of learners participating in the modelling activity,
p represents the purpose of the model, and
r corresponds to the computational resources (time, space, etc.) which are available at the time the model is being created.

It may also be useful to think of *learnerModel* as a **method** of the agent doing the modelling. From this viewpoint, the notation might be: *a.learnerModel(L, p, r)*.

It is important to note that this notation has no ambition of broad generality, nor do we intend to make a contribution to computational mathetics [8]. There is some overlap of our approach and the notion of runnable learner models. Indeed our learner model function implies that the learner model is a computation. The distinction is that our approach permits the computation to work on partially computed learner models drawn from diverse sources in addition to just-in-time computation with raw data.

4. The Different Purposes of Learner Modelling

Learner models can have a variety of purposes. They form a set of partially computed models describing fragments of knowledge about learners. The aggregate of **all** such *fragmented* models, if such a thing could be computed, would be the complete and definitive model of all learners associated with a system. We not only believe this aggregate could be very hard to compute, but we also believe it is not necessary for most purposes. We now investigate the various purposes of learner modelling.

4.1 Reflection

learnerModel(*a*: learner's personal agent; *L*: learner and other relevant learners; *p*: to find out how the learner is viewed; *r*: might not need real time response)

Making the contents of learner models accessible to students can be used to promote *reflection* on the target domain [9-11]. With the broader information in fragmented models in multi-user systems, such reflection may concern not only domain content, but may also be focused on other issues, e.g. "how do other learners view me?"

"How do other learners view me?" may refer to social issues such as helpfulness in I-Help, perhaps to assist someone in rethinking their attitude to the group; or for learners to compare their performance with their peers in S/UM. They may wish to see how well they are doing compared to the average student, or they may wish to view possibilities attainable by high achievers [12]. Students may also reflect on reactions of others who have viewed their work, leading to better understanding of difficulties. Finally, helpers may also benefit by reflecting on their own knowledge or the helpee's knowledge, when giving feedback.

4.2 Validation

learnerModel(*a*: modelling agent; *L*: the learner whose model is validated, the agents whose models are used for comparison; *p*: to confirm some of the beliefs in the initial model created about the learner, to leverage others, to add new beliefs; *r*: will probably take place off line, so lots of time and resources)

Learners can make use of various learner model viewpoints to confirm or deny opinions/knowledge. This could be used to confirm domain knowledge, and also to find out other people's opinions about a person's social characteristics. *Validation* is probably a special kind of reflection, distinguished by the learner starting with an opinion, rather than with a blank request. In I-Help validation would take place by direct agent interactions; in S/UM, it occurs through learner requests for feedback.

With so many distributed user models, questions of validity and consistency arise. Ensuring global consistency seems impossible and unnecessary. However, if each person, component or agent maintains its own models and is indifferent to how other agents model the same users, there is no advantage to multiple models. If an agent can communicate with other agents about its models, it can benefit from their experience, extend and validate its model (see also [13]). This is easier when agents are validating models created for the same purpose, with a similar modelling function. It is harder with data collected by an agent for a different reason, with a different function.

4.3 Matchmakers

learnerModel(*a*: matchmaking agent; *L*: learner and potential partners; *p*: to find appropriate peer; *r*: must complete in "real time" (I-Help) / need not complete in "real time" (S/UM))

In both I-Help and S/UM the system finds a ready, willing and able partner for a particular learner and learning need. Locating a suitable partner is handled by an agent we call the *matchmaking* agent.

Depending on the matchmaking agent *a* and the purpose *p*, the modelling function *learnerModel* may differ and different features *L* of the learner and potential peer helpers may be relevant for matching. For example, matching with the purpose of finding a peer helper may use the models of the potential helpers' knowledge and social characteristics (helpfulness, class ranking, eagerness) only, or it could also use the helper's and helpee's preferences. Matching with the purpose of finding partners in a collaborative project (p_1) may be done by another agent, a_1 which uses the same user characteristics *L*, but a different modelling function, $learnerModel_1$, which searches for knowledge and social characteristics which complement each other.

The modelling function *learnerModel* may depend on the agent who does the modelling, *a*, as will usually be the case, since it is easier to design smaller matchmaking agents specialized for one modelling function and purpose only. However, in the general case, there can be also more complex agents, able to create models of other agents for different purposes and with various alternative modelling functions.

4.4 Negotiation

learnerModel(a_1: helpee's personal agent; L: learners known by the agent; p: to obtain a fair price for help; r: must complete in "real time")

learnerModel(a_2: helper's personal agent; L: learner associated with the help request; p: to obtain a fair price for help; r: must complete in "real time")

In I-Help two personal agents can interact and *negotiate* for various reasons. This can be part of the matchmaking process [14], but can also occur between agents for other reasons, such as knowledge sharing where agents can acquire information directly from other agents so that one or both can work "better".

In this case we have 2 agents performing the modelling. They are personal agents involved in negotiation, let's say a_1 and a_2. a_1 develops a model of user L_2 and a_2 develops a model of user L_1. The purposes p_1 and p_2 of modelling may be identical (in the case of MAGALE, to better predict the reaction of the opponent in negotiation), or may differ. The same applies to the modelling functions. However, in a more general and complex case, when for example two agents are negotiating about the models of their users, the purposes / functions may be completely different.

Various versions of I-Help have been deployed to experiment with reflection, validation, matchmaking and negotiation. To achieve real time response we have computed minimal and partial models, with both content and social dimensions. Other "proof of concept" experiments in negotiation [14], supporting the helper [15] and visualizing models [16] have shed more light on these functions in use. S/UM emphasizes reflection and larger scale models of content. We aim to integrate the S/UM and I-Help approaches in a distributed environment, to further illuminate these issues. Other "classical" purposes of learner modelling e.g. diagnosis, assessment, context adaptation are also consistent with this active, procedural view of modelling.

5. What Processes and Techniques are Needed to Learner Model?

With this perspective of learner modelling as distillation and integration of fragments of data and models, the important activity changes from model building to model management. The focus expands from diagnosis of behaviour and representation of learner information to retrieval of appropriate model fragments and their integration to suit the purpose. Thus learner modelling consists of several processes, including:

- retrieval - gathering suitable data, processes, learner model fragments from various sources that would be relevant to the learners and purposes of the learner modelling process.
- integration - aggregating and abstracting learner model fragments (and possibly additional raw data) into coarser-grained, higher-level learner model fragments. Integration across all possible information about a learner might result in a single monolithic learner model. However, computational resources would likely preclude such comprehensive integration, and the purpose of the modelling would rarely require a monolithic learner model.
- interpretation - using the result of learner modelling for some purpose. The result of the learner modelling/integration process is a knowledge structure

that is to be interpreted by applications requiring learner model information. These processes will necessarily be idiosyncratic to the purpose required.

We will focus on retrieval and integration in this section. Many of the interpretation issues have already been covered in the discussion of purposes in section 4.

5.1 Retrieval

Since there are multiple models of various aspects of every learner, developed by different agents with different purposes under different resource constraints, it would be helpful to make use of all this information when a learner modelling need arises. How can one retrieve an appropriate model or collection of models? If several candidate models are available, which should be chosen? What should be done if candidates have contradictory contents? Two criteria will likely be most relevant in retrieving models: who created the model (a) and for what purpose (p). E.g. if an agent a_0 (of learner L_0) wants to learn the qualities of learner L_1 with respect to programming in C++, it will ask other agents that a_0 trusts and that know something about L_1. From these it will select agents who have models developed with the same purpose, i.e. evaluation of L_1's knowledge in C++. This means only users who have interacted with L_1 in the context of C++ will be queried. Another criterion, which can be considered as supplementary to the first, and will probably be more difficult to implement, is to look for agents with a similar modelling function (*a.learnerModel*). In this way an agent may seek models developed by trusted agents, or agents with similar evaluation functions. Finally, the time resources under which the model was created could regulate retrieval. A model created in a rush might be less adequate than one developed over a longer period of time and with more computational resources.

5.2 Integration

We use the term "integration" in a broad sense, more like "mediation" introduced in information systems [17], to denote the integration of diverse and heterogeneous information sources, achieved by abstracting away representational differences and integrating individual views into a common model. This integration captures the requirement for combining learner model fragments into coherent explanations. In its most complete sense, this process is complex, domain dependent, and resource intensive. Fortunately it is often only necessary to get an approximation of a learner's cognitive or social state derived from a few bits of raw data. Sometimes all that is needed is to confirm that a new bit of evidence is consistent with prior inferences.

Integration involves aggregation and abstraction of data and partial models. It demands that a domain ontology has been chosen and model elements are tagged according to that ontology. Integration of information is even more difficult than retrieval, as it requires interpretation and summarization of data retrieved from the model fragments to be integrated. This interpretation depends on the agents that created the model fragments, and moreover on the models of these agents created by the agent performing the integration, on their modelling functions and on the purposes of modelling. Suppose agents a_1, a_2 and a_3 had each created a model of L_0's eagerness, and L_4 wants to aggregate this information. L_4's agent (a_4) will interpret information from each of the three agents depending on its model of L_1 and L_2 and L_3's evaluation functions (i.e. how capable are they of accurately judging L_0's eagerness). Figure 3 shows how this integration might occur.

To achieve aggregation we must be able to represent and reason about a modeller's objectivity and priorities (expressed in the modelling function *learnerModel*). We must also be able to represent circumstances under which modelling is done. This is different from *p* (the purpose for which the model was created). Here we are more interested in the interpersonal relationship between modeller and modellee at the moment the model was created: whether they were in a cooperative or adverse relationship, close or distant, whether the modeller was observer or collaborator, whether they had common or different goals, as well as the general result of the situation (positive or negative, success or failure). This implies that complex reasoning may happen during integration. The good news is that global integration will rarely (if at all) be required. Integrating learner models will be done mostly by various agents (*a*) with a certain purpose (*p*), for a small subset of partial goal-related models (*L*), and under certain time constraints (*r*). In a narrow context this can be feasible.

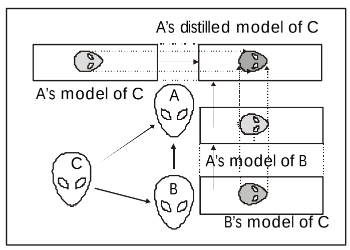

Fig. 3. Integration in A's model of C

Many AI techniques can possibly enter into the retrieval and integration processes:

- *belief revision*, to be able to incorporate new evidence into models personal agents keep about their learner. This belief revision is entirely local to the personal agent doing it, however, and will be done in the context of end use. The big issue will be whether to just add information without interpretation, and then put it together when there is an end use, or to have a separate belief revision process run occasionally like a garbage collection algorithm.
- *knowledge representation*, to capture both social and content knowledge. For many purposes knowledge will only need to be fairly shallow, so perhaps many of the deep KR problems can be avoided. Semantics will necessarily have to be procedural, in the sense that final meaning is totally relative to the procedures using the knowledge. A consistent ontology would simplify the representation process. Unfortunately, the likelihood of fine-grained ontologies remaining consistent across the diversity of applications and knowledge sources we envision would be small. The ability to merge, abstract and reason about ontologies will thus become important issues.
- *information retrieval and information filtering*, that is getting knowledge from the environment when needed, often very quickly.

- *knowledge externalization*, that is putting knowledge into a form that can be easily understood by the learner(s) or end users. This may vary from learner to learner and from one end use to another. Techniques for *knowledge visualization* will be useful here [16].
- *data mining* techniques to find patterns within and between agents' models and raw data.
- *group modelling techniques*, to find characteristics shared among many personal agents [18]. This will need to be retrieved by means of agent-agent negotiations, and will support collaborative styles of learning.
- *Bayesian belief networks* [19], useful for integrating multi-modal, multi-source evidence and propagating beliefs using a well-defined process.

Despite the daunting list of techniques and apparent complexity of learner modelling, we believe learner model computation to be tractable in many circumstances.

6. Conclusion

This paper argued for a revised view of "learner model" as a computation (the verb sense of "model"), rather than a data structure. We argued that in the new distributed computational architectures such a view will not only be useful, but necessary. Learner modelling will be a fragmented activity, performed on demand as a function of the people being modelled, purpose of modelling, and resources available. Learner modelling will occur for many reasons, extended from the traditionally narrower focus on diagnosis and assessment. For many purposes learner modelling computations will compute social as well as content aspects of learners. This should be easier than in the past given the vast amount of information that will be available about learner interaction in the emerging information technology intensive world.

These revised ideas about learner modelling will shift the learner modelling research agenda. Techniques such as retrieval, integration, and interpretation will be much more important. Many interesting research issues surrounding these techniques will have to be explored. In a fragmented, distributed, and universally accessible technological environment, learner modelling will increasingly be viewed as essential to building an effective system, but will also increasingly be seen to be tractable as new techniques emerge. Nevertheless, as our experiments have already shown, it will not be necessary to resolve all of these issues in order to usefully learner model.

References

1. Greer, J., McCalla, G., Cooke, J., Collins, J., Kumar, V., Bishop, A. and Vassileva, J. (1998) The Intelligent HelpDesk: Supporting Peer Help in a University Course, Proceedings ITS'98, San Antonio, Texas, LNCS No1452, Springer Verlag: Berlin pp.494-503.
2. Vassileva J., J. Greer, G. McCalla, R. Deters, D. Zapata, C. Mudgal, S. Grant (1999) A Multi-Agent Approach to the Design of Peer-Help Environments, in S. Lajoie and M. Vivet (eds.) Artificial Intelligence and Education, IOS Press: Amsterdam, 38-45.

3. Bull, S. (1997) A Multiple Student and User Modelling System for Peer Interaction, in R. Schäfer & M. Bauer (eds) ABIS-97: 5 GI-Workshop, Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Universität des Saarlandes, Saarbrücken, 61-71.
4. Bull, S. (1998) 'Do It Yourself' Student Models for Collaborative Student Modelling and Peer Interaction, in B.P. Goettl, H.M. Halff, C.L. Redfield & V.J. Shute (eds) Intelligent Tutoring Systems-ITS98, Springer-Verlag, Berlin Heidelberg, 176-185.
5. Vassileva, J.I., Greer, J.E., McCalla, G.I. (1999) Openness and Disclosure in Multi-agent Learner Models, in Proceedings of Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling, International Conference on AIED, Le Mans, France.
6. McCalla, G.I. (2000) The fragmentation of culture, learning, teaching and technology: implications for artificial intelligence in education research agenda in 2010. Int Jnl of AIED.
7. Kay, J. (1999). A Scrutable User Modelling Shell for User-Adapted Interaction. Ph.D. Thesis, Basser Department of Computer Science, University of Sydney, Sydney, Australia.
8. Self, J. (1990) Theoretical foundations for intelligent tutoring systems, Int Jnl of AIED 1(4).
9. Bull, S. & Pain, H. (1995) "Did I say what I think I said, and do you agree with me?": Inspecting and Questioning the Student Model, in J. Greer (ed), Proceedings of World Conference on AI in Education, AACE, 501-508.
10. Dimitrova, V., Self, J. & Brna, P. (1999) The Interactive Maintenance of Open Learner Models, in S.P. Lajoie & M. Vivet (eds), Artificial Intelligence in Education, IOS Press.
11. Paiva, A., Self. J. & Hartley, R. (1995) Externalising Learner Models, in J. Greer (ed), Proceedings of World Conference on AI in Education, AACE, 509-516.
12. Kay, J. (1997) Learner Know Thyself: Student Models to give Learner Control and Responsibility, in Z. Halim, T. Ottmann & Z. Razak (eds), Proceedings of International Conference on Computers in Education 1997, AACE, 18-26.
13. Maes, P. (1994) Agents that Reduce Work and Information Overload, Communications of the ACM 37(7), 31-40.
14. Mudgal, C., Vassileva, J. (to appear) An Influence Diagram Model for Multi-Agent Negotiation, Proceedings of International Conference on Multi-Agent Systems, Boston.
15. Kumar, V., McCalla, G., Greer J. (1999) Helping the Peer Helper. S. Lajoie and M. Vivet (eds.) Artificial Intelligence and Education, IOS Press, Amsterdam, 325-332.
16. Zapata-Rivera, J.D. & Greer, J., (this volume), Inspecting and Visualizing Distributed Bayesian Student Models.
17. Wiederhold, G. & Genesereth, M. (1997) The Conceptual Basis for Mediation Services, IEEE Expert.
18. Hoppe, H.-U. (1995) The use of multiple student modelling to parameterise group learning, in J. Greer (ed), Proceedings of World Conference on AI in Education, AACE, 234-241.
19. Reye, J. (1999) Student Modelling based on Belief Networks. Int Jnl of AI in Education, 11.