Approximate Modelling of the Multi-dimensional Learner*

Rafael Morales, Nicolas van Labeke, and Paul Brna

The SCRE Centre, University of Glasgow
11 Eldon Street, Glasgow G3 6NH, United Kingdom
Tel +44 (141) 330-3490, fax +44 (0)141 330 3491
{r.morales,n.vanlabeke,paul.brna}@scre.ac.uk

Abstract. This paper describes the design of the learner modelling component of the LEACTIVEMATH system, which was conceived to integrate modelling of learners' competencies in a subject domain, motivational and affective dispositions and meta-cognition. This goal has been achieved by organising learner models as stacks, with the subject domain as ground layer and competency, motivation, affect and meta-cognition as upper layers. A concept map per layer defines each layer's elements and internal structure, and beliefs are associated to the applications of elements in upper-layers to elements in lower-layers. Beliefs are represented using belief functions and organised in a network constructed as the composition of all layers' concept maps, which is used for propagation of evidence.

1   Introduction

The description given by the ADL Initiative of modern e-learning systems combines a content-based approach from computer based training with adaptive educational strategies from intelligent tutoring systems [1]. This mixture of approaches produces tensions in the design of these systems, particularly in the design of their learner modelling subsystem, which aim at supporting a wide range of adaptive educational strategies—e.g. from coarse-grain construction of e-books to tailored natural language dialogue [2]—with a general lack of what is traditionally afforded by ITS systems: painfully designed and dynamically constructed learning activities capable to provide large amounts of specific and detailed information on learner behaviour.

This type of e-learning systems make heavy use of pre-authored educational content to support learning, hoping to capitalise on the expertise of a variety of authors in producing educational materials. However, educational content is for the most part opaque to learner modelling due to the absence of domain expert subsystems to query about it. The information available is hence reduced to what is explicitly provided by
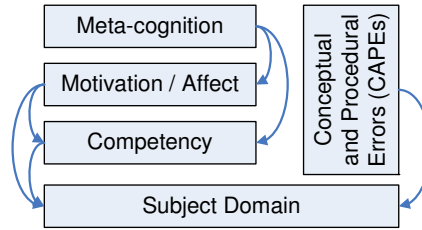
---

authors as *metadata* [3]. Unfortunately, provision of metadata is a heavy burden on authors because it amounts to work be done twice: to say something and to say what it was said. The more detailed and accurate the metadata, the more work there is to do. Consequently, metadata tends to be subjective and shallow, with a well-intentioned drive towards standardisation thwarted—from a learner modelling point of view—by the shallowness of current metadata standards such as LOM [4].

Guidance to learners through educational content in e-learning systems tends to jump in between two extremes: predefined paths and content browsing. From a learner modelling perspective, both situations are mainly equivalent, since neither of them accommodates to learner modelling needs. While in some ITS systems learner modelling can lead learner progress through the subject domain, in these type of e-learning systems it has to be *opportunistic*, taking advantage of whatever information becomes available. A learner modelling subsystem in this conditions has to do more with less, answering questions about the learner on the basis of scarce and shallow information, hopefully without pursuing blind over-generalisation.

In this paper we describe the consequences of requirements and working conditions as sketched above on the design of the Extended Learner Modeller (xLM) for LEACTIVEMATH, an e-learning system for mathematics [2]. xLM was required to model motivational and affective dispositions of learners towards a subject domain and related competencies, as well as learners' meta-cognition of their learning. Our approach to the problem can be summarised in terms of four characteristics: (i) a generic, layered and multi-dimensional modelling framework, (ii) tolerance to vague and inconsistent information, (iii) squeezing of sparse information and (iv) open learner modelling.

2 Modelling Framework

In xLM, a learner model is a collection of beliefs about the learner's states and dispositions arranged along five dimensions (figure 1): subject domain, competency, motivation, affect and meta-cognition. Each of these dimensions is described in a concept map which specifies the individual factors in the dimension that are relevant to learning and considered in learner models. The maps also specify how the different factors and attributes relate to each other. For example, in the current implementation the subject domain is a branch of mathematics known as Differential Calculus and breaks down into domain topics such as *function*, *derivative* and *chain rule* (a particular instance of *differentiation rule* that produces *derivative*s); competency is mathematical and decomposes according to the PISA framework [5]; and motivation decomposes into factors that are considered to influence learner motivation such as the interest, confidence and effort put into learning the subject domain and related competencies. The layered structure of learner models specifies how the modelled dimensions of learners interact with each other. At the bottom of the stack lies the subject domain, underlining the fact that learner modelling occurs within a subject domain, even if a learner model does not hold any belief about the subject domain *per se*. but about learner dimensions *applied to* the domain. On top of the subject domain are the layers of competency, motivation, affect and meta-cognition, each one relying on the lower layers for expressing a wide range of beliefs about the learner. For example, mathematical competencies on the subject

Fig. 1. A multi-dimensional and layered structure for learner models.

domain (e.g. the learner's level of problem solving with respect to the chain rule), motivational dispositions towards the subject domain (e.g. the learner's level of confidence with respect to differential calculus) or towards competencies on the subject domain (e.g. the learner's level of effort with respect to solving problems with the chain rule).

Conceptual and procedural errors (CAPEs) is a sixth but special dimension. CAPEs are not generic as competencies, but each one specific to particular domain topics. Moreover, neither motivation nor affect nor meta-cognition apply to CAPEs, under the assumption that they are not perceived by learners.

3 Levels and Beliefs

Let us start our explanation of the type and representation of beliefs held in xLM learner models by considering a particular belief concerning the learner's *competency to posing and solving mathematical problems with derivatives*. Let us also assume that the mathematical competencies of learners can be measured in a discrete scale of four levels, from an entry level I to a top level IV— only three levels are described in [5] but LEACTIVEMATH uses four. A learner's mathematical competency is assumed to *be at* one of this levels, having achieved and passed all previous levels, hence a belief on the learner's competency to pose and solve problems with derivatives becomes a statement about *the level that the learner's competency is (more likely) at*.

In the same way, every belief in xLM is about a learner's level on something—as far as that something can be expressed as the application of upper dimensions to lower dimensions in the learner model structure (figure 1). In the current implementation, all dimensions in xLM are measured in a similar scale of four levels.

There are many ways to represent beliefs such as this, from symbolic representations using a logical formalism [6] to numeric representations such as using probability distributions [7]. In xLM, a belief in a learner model is represented and updated using a numeric formalism known as the Transferable Belief Model (TBM) [8], a variation of Dempster-Shafer Theory (DST) which is based on the notion of *belief functions* [9]. A first difference between a probability distribution and a belief functions is that, while the former assigns a number in the range of $[0, 1]$ to each possible state of the world—i.e. each level the learner's competency could be at—the latter does the same but to each set of possible states of the world.

More formally, if $\Theta = \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}\}$ is the set of all possible states of the world, then a probability distribution is a function $p : \Theta \to [0, 1]$ while a belief function $b : 2^\Theta \to [0, 1]$ maps the set of all sets of levels in Θ into $[0, 1]$.

A belief in a learner model can be represented at least in three different ways, as a *mass*, a *certainty*[1] or a *plausibility* function [8, 9]. Although they are equivalent, a mass function is the easiest to manipulate computationally and is hence the one used in xLM. If s_0 is a set of levels, say $s_0 = \{\mathrm{III}, \mathrm{IV}\}$, the mass of s_0, or $m(s_0)$ can be interpreted *objectively*, as the support the evidence gives to the case that the true learner's competency level is in the set s_0 (i.e. it is either III or IV) without making any distinction between the elements of the set[2]. *Subjectively*, it can be interpreted as the part of the belief that pertains exclusively to the likelihood that the true learner's competency level is in s_0, without being any more specific towards either of the levels.

A mass function in xLM is generally required to satisfy the requirement that the sum of all its values must be one; i.e.

$$m : 2^\Theta \to [0, 1] \quad \text{such that} \quad \sum_{s \in 2^\Theta} m(s) = 1. \tag{1}$$

However—in accordance with TBM and differently from DST—it is not required that the mass assigned to the empty set to be zero. Such a mass is interpreted in xLM as the amount of *conflict* in the evidence accumulated. The mass assigned to the set of all levels Θ is generally interpreted as the amount of complete *ignorance* in a belief.

4 Evidence

Evidence for learner modelling comes into xLM in the shape of *events* representing what has happened in the learners interaction with educational material and the rest of LEACTIVEMATH. Some events originate inside xLM, as is the case for events generated by its Situation Model (the component of xLM in charge of modelling the local motivational state of learners) and Open Learner Model (the graphical user interface to learner models that supports inspectability of the models and challenging of beliefs).

Events are raw evidence that needs to be *interpreted* in order to produce mass functions that can be incorporated into beliefs in learner models using a *combination rule* [10]. Currently, two categories of events are interpreted by xLM and their evidence incorporated into beliefs: *behavioural* events and *diagnostic* events. Events in the first category simply report what the learner has done or achieved, whereas events in the second category report a judgement of learner levels produced by some diagnostic component of LEACTIVEMATH.

---

[1] More commonly known as a *belief* function, but we use *certainty* to avoid confusion with the more general notion of belief in xLM.

[2] Imagine Theta is an international tennis team composed by Indian Papus (I), Japanese Takuji (II), Scottish Hamish (III) and French Pierre (IV). They have played against the American team and you hear on the radio than all of them, but an European one, have lost their matches. If this is all the evidence you have about who in Theta won its match, it certainly supports the case that the player is in the set {Hamish, Pierre} but does not distinguish between them.

Listing 1. Example metadata for an exercise.

```
<exercise id="K3_TIMSS" for="deriv_higher/second_deriv">
  <metadata>
    <Title xml:lang="en">Acceleration of a straight forward movement</Title>
      <depends-on><ref xref="deriv/diff_f"/></depends-on>
      < difficulty  value="easy"/>
      <competency value="think"/>
      <competency value="model"/>
      <competencylevel value="simple_conceptual"/> <!-- equiv. to II -->
       ...
  </metadata>
  ...
</exercise>
```

4.1 Interpretation of Behavioural Events

Given an event of type ExerciseFinished, reporting that a learner has finished an exercise with some *success rate*, xLM interprets it to produce evidence for updating the learner model. The resulting evidence would be a collection of mass functions over the following set of sets of levels $2^{\Theta} = \{s | s \subseteq \Theta\}$. However, given the fact that levels are ranked, it makes no sense to have mass for sets that are not intervals (e.g. $\{I, III\}$). In other words, the *focus* of *m* is always going to be a subset of

$$\Phi = \{\{I\}, \{II\}, \{III\}, \{IV\}, \{I, II\}, \{II, III\}, \{III, IV\}, \\ \{I, II, III\}, \{II, III, IV\}, \{I, II, III, IV\}\}. \tag{2}$$

For the particular case of an event of type ExerciseFinished in the current implementation of xLM, these levels are actually *competency levels*. In the general case, the nature of the levels will depend on the nature of the belief the evidence is relevant to.

In order to make the explanation easier to follow, let us consider a concrete case: interpreting an ExerciseFinished event that resulted from the learner finishing an exercise with metadata as in listing 1. An exercise of this type comes with its own additional information, including learner identifier, exercise identifier and success rate achieved by the learner in the exercise (in the range $[0, 1]$).

Beliefs Addressed By interpreting events of type ExerciseFinished, xLM generates direct evidence for beliefs grounded on the subject domain topics the exercises are related to. For example, the exercise K3_TIMSS is for a learning object with identifier deriv_higher/second_deriv, which is mapped to the domain topic *second_derivative* that represents the abstract notion of second order derivative. The metadata listed above indicates the exercise depends on the learning object deriv/ diff_f which is mapped to the topic *derivative* that stands for the abstract notion of derivative. Consequently, all direct evidence produced from an ExerciseFinished event from this exercise will be evidence for beliefs grounded on the topics *derivative* and *second_derivative*.

Metadata indicating which mathematical competencies the exercise evaluates or trains on provide further details of which beliefs should be affected by the event. For our example, this are beliefs on competencies think (think mathematically) and model (model mathematically). Therefore, the beliefs to be directly affected by our example of event would be *beliefs related to the learner's thinking or modelling mathematically with first or second derivatives*. These could be, in principle, beliefs on the learner's competencies, their motivational or affective dispositions on these competencies, or their meta-cognition on these competencies. No belief on a conceptual or procedural error is directly affected, since the event does not provide any information on CAPEs. The current implementation of xLM depends on events produced by its Open Learner Model component for updating beliefs on meta-cognition, on events produced by its Situation Model component for updating beliefs on motivational dispositions, and on events produced by LEACTIVEMATH Self-Report Tool for updating beliefs on affective dispositions. Consequently, only evidence for beliefs on learners' mathematical competencies on the subject domain are produced from events of type ExerciseFinished.

Generation of Evidence Once the beliefs to be affected by an event have been identified, the next step is to generate the corresponding evidence: a mass function per belief over the sets of levels in Φ. Although most of the metadata for an exercise could have an impact on the evidence to be produced, for the case of an ExerciseFinished event in the current implementation of xLM only a subset of the metadata is taken into account: the relationship between the exercise and the belief addressed (i.e. whether the exercise is *for* the topic the belief is about, or only *depends on* it), the competency level of the exercise, the difficulty of the exercise and the success rate reported in the event.

The competency level of an exercise is used to determine who should find the exercise either very easy or very difficulty, and who may find it otherwise (i.e. easy, medium or difficult, the remaining terms in LEACTIVEMATH vocabulary for metadata on the difficulty of exercises). For example, being K3_TIMSS an easy exercise for competency level II means it should be a very easy exercise for any learner with competency level IV. Furthermore, we have assumed that would be the case for any exercise for competency level II, and table 1 presents the initial estimation of the difficulty of an exercise for a learner, given the competency levels of the exercise and the learner. To fill the still empty cells in table 1 we use the metadatum for difficulty of the exercise. A possible interpretation of this metadatum is given in table 2 for the case of exercises for competency level II.

Table 1. Effect of the metadata value for competency level on the estimated difficulty of an exercise for a learner at a given competency level.

Competency level of exercise	Competency level of learner			
	I	II	III	IV
I	-	-	very easy	very easy
II	-	-	-	very easy
III	very difficult	-	-	-
IV	very difficult	very difficult	-	-

Table 2. Effect of the metadata value for difficulty on the estimated difficulty of an exercise for competency level II for a learner at a given competency level.

Exercise		Difficulty of the exercise for a learner at a given competency level			
Competency level	Difficulty	I	II	III	IV
	VE	M	VE	VE	VE
	E	D	E	VE	VE
II	M	VD	M	VE	VE
	D	VD	D	E	VE
	VD	VD	VD	M	VE
VE: very easy, E: easy, M: medium, D: difficult, VE: very difficult					

At this stage, the metadata for an exercise such as K3_TIMSS supports estimates of how difficult the exercise would be for learners with different competency levels. In particular, we can see that K3_TIMSS does not discriminate between learners with competency level III or IV. Hence mass should not be assigned to levels III nor IV alone in any evidence generated from it, but only to the set $\{III, IV\}$ or sets containing it.

Here is the point when we need to translate from the qualitative tags denoting difficulty to quantitative measures. In other words, we need to estimate, for every rate of success r, the probability P of achieving r given difficulty d. We need to estimate $P(r|d)$. We use scaled normal distributions

$$P(r) = \delta e^{-(r-\mu)^2/2\delta^2} \tag{3}$$

with parameters determined by difficulty as specified in table 3. They assign a 0.5 probability to being completely successful ($r = 1$) in a very easy exercise, to being moderately successful ($r = 0.75$) in an easy exercise, to being just fair ($r = 0.5$) in a medium exercise, to being unsuccessful ($r = 0.25$) in a difficult exercise and to being completely unsuccessful ($r = 0$) in a very difficult one. For example, if the success rate reported in an ExerciseFinished event for exercise K3_TIMSS is $r = 0.8$ then we get the following probabilities per competency level: 0.2730 (I), 0.4975 (II) and 0.4615 (III and IV).

Table 3. Parameters for probability assignment functions per difficulty value.

Parameters	Difficulty				
	very easy	easy	medium	difficult	very difficult
μ	1	0.75	0.5	0.25	0
δ	0.5	0.5	0.5	0.5	0.5

An straightforward way of translating this probabilities into a mass function would be by normalising the probabilities obtained above and assigning them to the singletons $\{I\}$, $\{II\}$, $\{III\}$ and $\{IV\}$. However, as it was said before, exercise K3_TIMSS does not distinguish between learners with competency levels III or IV, hence it does not provide evidence for the learner having any of these levels in particular but, in

any case, just of having any of them. Furthermore, what should be done if all probabilities above where the same? A possibility is to consider the exercise as unable to discriminate between the possible levels of competency of the learner, hence providing no new evidence at all. Technically, this means the mass distribution in this case should be the one corresponding to *total ignorance* (or complete lack of evidence): $m(\Theta) = 1$ and $m(s) = 0$ for all other $s \neq \Theta$.

We can generalise these two cases to an iterative method for calculating a mass functions from probabilities:

1. If $s = \{l_1, l_2, \ldots, l_n\}$ is the set of levels with non zero probabilities, then make $m(s)$ equal to the smallest probability, $m(s) = \min(p(l_1), p(l_2), \ldots, p(l_n))$.
2. For every level l_i in s make its probability equal to $p(l_i) - m(s)$.
3. Remove from s all levels with re-calculated probability equal to zero and start again at step (1).
4. Finally, scale all $m(s)$ uniformly, so that the total mass $\sum_{s \subseteq \Phi} m(s) = 1$.

The application of this method to the case of exercise K3_TIMSS with success rate of 0.8 and the probabilities calculated above produces the following mass function:

$$m(\{I, II, III, IV\}) = 0.549, \quad m(\{II, III, IV\}) = 0.379, \quad m(\{II\}) = 0.072,$$
$$m(s) = 0.0 \text{ for any other } s \subset \Phi.$$

In words, this is weak evidence for the learner being at competency level II (simple-conceptual) and stronger evidence for they being at a competency level in $\{II, III, IV\}$. However, this evidence includes a fair amount of ignorance that suggest it is still plausible for the learner to be at any competency level, including level I.

Each belief that have to do with topics trained on, or evaluated by the exercise would receive such an evidence. Beliefs concerning topics the exercise depends on would receive a *discounted* evidence with increased ignorance:

$$m'(s) = d \times m(s) \text{ for all } s \neq \Theta \text{ and } m'(\Theta) = m(\Theta) + (1-d), \tag{4}$$

where d is a discount factor in between zero and one.
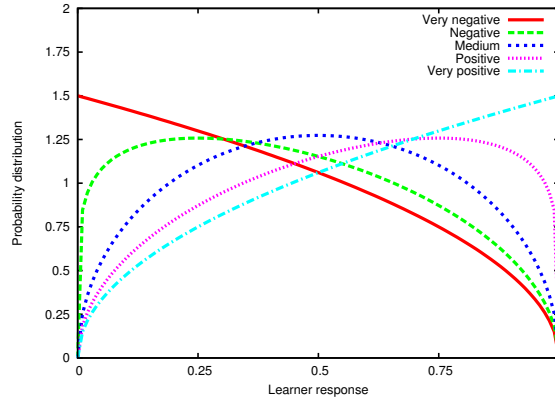
4.2 Interpreting Diagnostic Events

The interpretation of diagnostic events by xLM is simpler than the interpretation of behavioural events given the fact that an estimation of the learner level is included in diagnostic events. Based on how much xLM trust the source of the event, the original estimation of the learner level is transformed into a probability distribution over the set of levels Θ. This probability distribution is then transformed into a mass function following the same procedure as explained in section 4.1.

Consider, for example, the case of LEACTIVEMATH Self-Report Tool, which is presented to learners every time they complete an exercise so that they can provide feedback on their states of liking, pride and satisfaction—which are assumed to be with respect to the exercise just finished. The values input by the learners are reported to xLM in SelfReport events. Then xLM transform a single value per factor into a probability distribution by choosing a suitable Beta distribution from the collection shown

in figure 2. The mass function resulting from the interpretation of the event would constitute evidence for beliefs on the learner's affective dispositions towards the subject domain topics and competencies that result from considering the exercise metadata.

Fig. 2. Beta distributions for each value that can be reported by learners using the Self-Report Tool. The intervals $[0, 0.25]$, $[0.25, 0.50]$, $[0.50, 0.75]$ an $[0.75, 1.0]$ correspond to the levels I, II, III and IV, respectively.

4.3 Propagation of Evidence

Interpretation of events such as the ones described in sections 4.1 and 4.2 provides direct evidences for some beliefs in a learner model. These evidences are propagated to the relevant parts of the learner model following the associations between elements in the maps for each layer in the learner modelling framework (section 2). An iterative algorithm is used for belief propagation which is a simple adaptation of Shenoy-Shafer algorithm for belief-functions propagation [11]. In every iteration, all beliefs that have received updated messages (with adjusted evidences) re-calculates its own and checks whether these have changed significantly (given a predefined threshold and a method for comparing mass functions). If that is the case, and its messages are not full with ignorance (beyond another predefined threshold) then it propagates the evidence to their neighbours. The iterative process ends when no more messages have been exchanged or when a predefined maximum number of iterations have been reached.

5 Conclusions and Future Work

A learner modelling subsystem called xLM has been presented in this paper which tries to capture the multi-dimensional nature of learners. xLM uses a collection of dimensions—subject domain, conceptual and procedural errors, competency, motivation, affect and meta-cognition—defines them using concept maps and arranges them

in layers. Together, the internal maps and layered framework provide a rich structure for organising beliefs about learners. Beliefs are represented using belief functions, which allow the representation of ignorance, uncertainty and conflict in evidence and beliefs. Together with a simple algorithm for propagation of evidence, xLM is the first implementation of learner models with belief functions networks we are aware of, providing in this way an alternative to Bayesian networks for learner modelling [7, 12–14].

A first complete implementation of xLM has been delivered early this year. Nevertheless, there are many issues to be revised and parameters to be adjusted before xLM reaches maturity. First, competency level and difficulty are seen as two granularities in the same scale, like metres and centimetres. This may be a misinterpretation of the nature of competency levels, which seems to represent more qualitative changes than difficulty [5]. Secondly, a core but minimum subset of metadata is taken into account while interpreting events, which could be expanded for better. Thirdly, careful analysis and evaluation of the probability assignments, probability distributions and the propagation algorithm are necessary to improve the modelling process.

References

1. Advanced Distributed Learning: Sharable Content Object Reference Model Version 1.2: The SCORM Overview. (2001)
2. LeActiveMath Consortium: Language-enhanced, user adaptive, interactive elearning for mathematics (2004) http://www.leactivemath.org.
3. National Information Standards Organization: Understanding Metadata. (2004)
4. Institute of Electrical and Electronics Engineers: IEEE 1484.12.1 Draft Standard for Learning Object Metadata. (2002)
5. Organisation for Economic Co-Operation and Development: The PISA 2003 Assessment Framework. (2003)
6. Self, J.A.: Dormorbile: A vehicle for metacognition. AAI/AI-ED Technical Report 98, Computing Department, Lancaster University, Lancaster, UK (1994)
7. Zapata-Rivera, J.D., Greer, J.E.: Inspecting and visualizing distributed bayesian student models. In Gauthier, G., Frasson, C., VanLehn, K., eds.: Intelligent Tutoring Systems: Fifth International Conference, ITS'2000. Number 1839 in Lecture Notes in Computer Science, Springer Verlag (2000) 544–553
8. Smets, P., Kennes, R.: The transferable belief model. Artificial Intelligence **66**(2) (1994) 191–234
9. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press (1976)
10. Sentz, K., Ferson, S.: Combination of evidence in dempster-shafer theory. Sandia Report SAND2002-0835, Sandia National Laboratories (2002)
11. Shenoy, P.P., Shafer, G.: Axioms for probability in belief-function propagation. In Shachter, R.D., Levitt, T.S., Kanal, L.N., Lemmer, J.F., eds.: Proceedings of the Fourth Anual Conference on Uncertainty in Artificial Intelligence, North-Holland (1990) 169–198
12. Conati, C., Gertner, A., VanLehn, K.: Using bayesian networks to manage uncertainty in student modeling. User Modeling and User-Adapted Interaction **12**(4) (2002) 371–417
13. Bunt, A., Conati, C.: Probabilistic student modelling to improve exploratory behaviour. User Modeling and User-Adapted Interaction **13**(3) (2003) 269–309
14. Jameson, A.: Numerical uncertainty management in user and student modeling: An overview of systems and issues. User Modeling and User-Adapted Interaction **5**(3–4) (1996) 193–251