A Model for Content Sequencing in Intelligent Tutoring Systems Based on the Ecological Approach and Its Validation Through Simulated Students

John Champaign and Robin Cohen
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada N2L 3G1
jchampai@cs.uwaterloo.ca,rcohen@ai.uwaterloo.ca

Abstract

In this paper, we present an algorithm for reasoning about the sequencing of content for students in an intelligent tutoring system. Our motivating influence is McCalla's ecological approach which advocates attaching models of learners to the learning objects they interact with, and then mining these models for patterns that are useful for various purposes. In particular, we record with each learning object those students who experienced the object, together with their initial and final states of knowledge, and then use these interactions to reason about the most effective lesson to show future students based on their similarity to previous students. We validate our approach in a context of simulated students, providing details of the model of learning used in the simulation and the results obtained in order to demonstrate the value of our model. As a result we offer a novel approach for peer-to-peer intelligent tutoring from repositories of learning objects.

Introduction

One direction for designing intelligent tutoring systems is to make use of peer-to-peer assistance. Several researchers have developed models that enable a student to learn on the basis of feedback from peers (Read et al. 2006) or on the basis of the experiences that peers have had in learning similar material (Vassileva 2008). Recently, McCalla has proposed an ecological approach for the design of intelligent tutoring systems (McCalla 2004) in which he advocates that student learning be achieved on the basis of the experiences of previous students but in a way that evolves over time, as the students themselves adjust with their learning.

Two central challenges in the design of intelligent tutoring systems are compiling the material for the lessons and determining the best methods to use, for the actual teaching of those lessons. We observe in particular that it is desirable to provide a framework for determining the material to be taught that does not rely on experts hand-coding all the lessons. Indeed, that particular approach presents considerable challenges in time and effort.

We are therefore interested in making use of existing repositories of information (books, webpages, etc.) towards the design of lessons for students. But we are also interested in facilitating the learning of those lessons through peer interactions. As such, we focus on the subtask of content sequencing and propose an algorithm for selecting appropriate content (learning objects) to present to a student, based on previous learning experiences of like-minded students. We discuss how this model honours McCalla's proposed ecological approach for the design of intelligent tutoring systems. Next, we introduce the methodology of simulating students in order to illustrate the effectiveness of our particular approach. We then discuss the value of this particular methodology for validating intelligent tutoring systems in general. We conclude with a comparison to existing research, including existing peer-to-peer approaches, to demonstrate important differences in our aims and our results.

Copyright © 2010, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Background

McCalla's ecological approach to e-learning systems (McCalla 2004) is described as "attaching models of learners to the learning objects they interact with, and then mining these models for patterns that are useful for various purposes." Using techniques inspired by collaborative filtering, the basis of this approach is to identify which users in a system are similar to each other, to then preferentially recommend learning objects that similar students have found useful. Learning objects are used in the ecological sense, which differs from the traditional usage of the term in that instead of attaching static ontologies to objects as metadata, models of previous interactions (i.e. presenting learning objects to students) are used in their place. These models are then actively interpreted, with the meaning derived by real-time processes, as information about the object is needed.

Since McCalla's ecological approach is intended to primarily be a general philosophy for designing intelligent systems, individual researchers need to then create their own algorithms and systems to embody this approach. This paper is an effort to do so for the problem of content sequencing, explored through a framework of simulated students.

Our Approach

Our proposed algorithm for determining which learning objects to present to students is presented in Algorithm 1. We assume that we are tracking a set of values, *v[j,l]*, representing the benefit of the interaction for user *j* with learning

Algorithm 1 Pseudocode For Collaborative Learning Algorithm

```
Input the current-student-assessment
for each learning object: do
   Initialize currentBenefit to zero
   Initialize sumOfBenefits to zero
   Input all previous interactions between students and this learning object
   for each previous interaction on learning object: do
      similarity = calculateSimilarity(current-student-assessment, interaction-initial-assessment)
      benefit = calculateBenefit(interaction-initial-assessment, interaction-final-assessment)
      sumOfBenefits = sumOfBenefits + similarity * benefit
   end for
   currentBenefit = sumOfBenefits / numberOfPreviousInteraction
   if bestObject.benefit < currentBenefit then bestObject = currentObject
end for
if bestObject.benefit < 0 then bestObject = randomObject
```

object *l*. *v[j,l]* is determined by assessing the student before and after the interaction, and the difference in knowledge is the benefit. We also record for each learning object the previous interactions of students with that object, in terms of their initial and final assessments.[1] We assume that a student's knowledge is assessed by mapping it to 18 concrete levels: A+, A, A-, ... F+, F, F-, each representing $\frac{1}{18}$th of the range of knowledge.

The anticipated benefit of a specific learning object *l*, for the active user, *a*, under consideration would be: [2]

$$p[a,l] = \kappa \sum_{j=1}^{n} w(a,j)v(j,l) \tag{1}$$

where *w(a,j)* reflects the similarity $\in (0,1]$ between each user *j* and the active user, *a*, and κ is a normalizing factor. $\frac{1}{|n|}$ was used as the value for κ in this work where *n* is the number of previous users who have interacted with learning object *l*. *w(a,j)* was set as $\frac{1}{1+difference}$ where difference is calculated by comparing the initial assessment of j and the current-student-assessment, and assigning an absolute value on the difference of the letter grades assigned. So the difference of A+ and B- would be 5 and the difference of D+ and C- would be 1. This is shown as the calculateSimilarity function in Algorithm 1.

v(j,l) is also computed using a difference, not an absolute difference but an actual difference (between the initial and final assessments). For example, *v(j,l)* where *j* is initially assessed as A+ and finally assessed at B- would be -5, while where *j* is initially assessed at B- and finally assessed at A+ would be 5. This is shown as the calculateBenefit function in Algorithm 1.

In the absence of other criteria, a user *a* will be assigned the learning object *l* that maximizes *p[a,l]*. In the case that the maximum *p[a,l]* is a negative anticipated benefit, a random learning object will be assigned to the user.

[1]The algorithm would be run after an initial phase where students are learning through the use of a set of learning objects. These students' experiences would then form the basis for instructing the subsequent students.

[2]Adapted from (Breese, Heckerman, and Kadie 1998)

Example

Here we provide a simplified example for illustration. Suppose we track each learning object, LO with [index, [StudentID, initial assessment], [final assessment]]. After multiple interactions with 3 students, S1, S2 and S3:

LO[1; S1(B,C), S3(B,A+)]   LO[4; S1(C,A-), S2(B,B)]
LO[2; S1(A,A), S3(C,A-)]   LO[5; S3(C+,B)]
LO[3; S2(B-,A)]

At this point the system is slightly positive on the benefit of LO[1] for B students (because one time it raised a student to A+, and another time it lowered a student to C). It is neutral on LO[2] for A students (the lesson didn't change the student's assessment), and very positive for C students (since it raised a C student to an A-). Similarly LO[3] is good for B- students, LO[4] is very good for C students and neutral for B students, and LO[5] is good for C+ students.

Suppose the system were now considering which lesson to recommend for a student, S4, with current-student-assessment of B+. Per Eq. (1), it would consider LO[1] to have a currentBenefit of $(\frac{1}{1+1} \times -3 + \frac{1}{1+1} \times 4) \div 2 = \mathbf{0.25}$, LO[2] a currentBenefit of $(\frac{1}{1+2} \times 0 + \frac{1}{1+4} \times 5) \div 2 = \mathbf{0.67}$, LO[3] a currentBenefit of $\frac{1}{1+2} \times 4 = \mathbf{1.33}$, LO[4] a currentBenefit of $(\frac{1}{1+4} \times 5 + \frac{1}{1+1} \times 0) \div 2 = \mathbf{0.5}$ and LO[5] a currentBenefit of $\frac{1}{1+3} \times 2 = \mathbf{0.5}$. In this situation, LO[3] would be recommended. After the system's interaction between S4 and LO[3] there will be more information to reason about with future students. The next B+ student will be assigned to LO[3] if S4 has a positive experience, but will instead be assigned to LO[2] if S4 has a neutral or negative experience with LO[3]. This assumes that no additional students use these learning objects in between S4's interactions.

Validation of Approach

Experimental Setup

In order to simulate the learning achieved by students, we used the following approach. Let *UK[j,k]* represent user *j*'s knowledge of $k \in$ K, such that *UK[j,k]* $\in [0,1]$. In the context of this work what "knowledge" means will be abstract. As most readers will be familiar with a typical computer science undergraduate curriculum, a concrete example from a computer science 101 course would be a knowledge of recursion currently recorded to be at 0.67. This would be interpreted as the student has an understanding of 67% of the course content dealing with recursion.

Let *LOK[l,k]* represent some learning object *l*'s target instruction level of knowledge *k*, such that *LOK[l,k]* $\in$ [0,1]. Viewing this more concretely, say a specific learning object is a 90 minute lab where students work at a computer on a set of exercises designed to broaden their understanding of recursion. The students have used recursion on lists and lists of structures; however in this lab they are required to apply recursion to solve mathematical problems, such as determining whether a number is prime. The target instruction level might be 0.68 since the students have a fairly detailed understanding of recursion, reinforced by 2 previous labs on the subject, but are still gaining an understanding of the nuances of the subject, and how to apply it to real world and mathematical problems.

Learning objects also have an impact, which can be positive or negative. The negative impact was introduced to simulate the possibility of misinformation from a poor quality learning object or a learning object that does a good job teaching one concept, while undermining the understanding of another concept.

Let *I[l,k]* $\in \mathcal{R}$, represent the impact of learning from learning object *l* on the knowledge *k*, that is, in the optimal case how much the learning object can adjust a student's knowledge *k*. The impact can be thought of as, for a student at the target level, what is the expected learning benefit of the object. This is information used by our approach to simulate the learning that is occurring. For the example of the lab that targets students with an understanding of 0.68 in recursion, perhaps the 90 minute lab is expected to improve this understanding by 0.11 to a 0.79 understanding of recursion and subsequent labs, lectures and assigned readings over the rest of the term are intended to bring the students to a knowledge level of 1.0.

After an interaction with an object *l*, a user *j*'s knowledge of *k*

$$\Delta UK[j,k] = \frac{I[l,k]}{1 + (UK[j,k] - LOK[l,k])^2} \quad (2)$$

This has the implication that the impact of a lesson is at a maximum when the student's knowledge level matches the target level of the learning object[3]. As the two values differ, the impact of the lesson exponentially decreases. Two concrete examples might be a student who didn't understand the previous lab and skipped the recommended readings, and therefore doesn't have the necessary background to benefit fully from the current lab on recursion, or a student who is a self-taught programmer and had a deep understanding of recursion before beginning the course. Since these two students aren't the target audience of the lab, it seems reasonable to expect that they won't benefit as fully as the target audience.

Based on this change, the user's knowledge in that area is updated as:

$$UK'[j,k] = UK[j,k] + \Delta UK[j,k] \quad (3)$$

The user's average knowledge can then be calculated as:

$$\overline{UK}[j] = \frac{1}{|K|} \sum_{k \in K} UK[j,k] \quad (4)$$

Testing Our Approach

For our experiment, variable numbers of simulated students and a static set of learning objects were allowed to interact over a set number of trials, arbitrarily chosen as 100. Each simulated student was randomly assigned values for each of 6 knowledges, each evenly distributed in the range [0,1]. A multi-dimensional structure for knowledge was used to ensure randomly generated students were distinct from one another, and to provide a rich model for simulated learning. In a real world context, this can be thought of as students with a better understanding of one part of a course of study compared to another part of the same course. Each learning object was randomly assigned a value for the target level of instruction for each knowledge, evenly distributed in the range [0,1]. Impact values were assigned to learning object knowledges, randomly and evenly distributed in the range [-0.05, 0.05]. The values of -0.05 and 0.05 were chosen such that no single learning object could radically change a user's knowledge level (at most it can adjust it by 5%).

Each experimental condition was repeated for 20 iterations, and the mean of the average knowledges of all students after each trial was graphed (see Results). In this context, the average knowledge might be thought of as, if a final mark for the CS 101 course was to be assigned to a student based on their current understanding of the course content, what would it be? A student's final mark will be based on their knowledge of a number of areas, such as introductory data structures, recursion, sorting, introductory proofs, and programming in a specific language.[4] In order to explore the value of our approach, we graph the performance of several algorithms to select a learning object for each student, over a number of trials. After each trial, the average knowledge under each condition is compared.

Random Association: Two reference points were created to compare our approaches against. One reference was created by associating each student with a randomly assigned learning object each trial. Given that any intelligent approach to matching students with objects should outperform blind chance, this was viewed as the lower limit.

Greedy God: The other reference point, the greedy god reference point, was created by giving the algorithm full access to the fine-grained knowledge levels of the students and learning object, testing what the outcome would be for every possible interaction, then choosing the best interaction

[3]This does not imply a desire to maximize performance as early as possible. A high impact learning object may succeed in conveying knowledge to a student by confusing them (for example). The end result of this process will be an increase in understanding, i.e. the increase in knowledge level represents.

[4]We make the simplifying assumption that all knowledges contribute equally to this progress assessment and calculate it as the average knowledge.

for each student for each trial. The results, based on an omniscience not typically available in real world learning environments, was viewed as a ceiling on the possible learning benefit of any approach.

The impact values and target levels of objects are used for the reasoning of the greedy god algorithm. In contrast, for the following three algorithms, our ecological approach is used to select the learning objects to be presented to users (so based on their similarity to previous students who have experienced these objects and on the benefit that these students derived). We assume that as each simulated student is assigned a learning object, that student's interaction with the object can be used as the "previous experience" to which subsequent students are matched. Using our ecological approach, learning objects presented to students should end up being ones that have an effective combination of impact value and target level (i.e. beneficial to those previous users and at a somewhat similar level of knowledge).

Raw Ecological: For the raw ecological approach, 3 trials were run where each student was randomly assigned to a learning object. For the remaining 97 trials, each student was matched with the learning object best predicted to benefit her knowledge, as detailed in Algorithm 1[5]. The initial three trials with random associations were used to provide rough information about the learning objects and students, which was used and refined over the course of the remaining trials.

Pilot Group: For the pilot group ecological approach, a subset of the students (10%) were assigned as a pilot group, and for their 100 trials were systematically assigned to learning objects to explore their impact. These interactions, along with the accumulation of their own interactions, were used by the remaining 90% of the students to reason about the best sequence using Algorithm 1.

Simulated Annealing: Our third ecological approach was inspired by simulated annealing, which in turn was inspired by the metallurgical approach of heating and cooling to induce change in a material. For this approach, we had a "cooling" period, which was the first 1/2 of the trials. During this period, for every student, there was an inverse chance, based on the progress of the trials, that they would be randomly associated with a lesson; otherwise, the ecological approach detailed in Algorithm 1 would be applied. For example, in the first trial, every student would be randomly associated with a learning object, but by the 25th trial, each lesson would have a 50% chance of being randomly associated. After the cooling period was over (the 50th trial), every student was repeatedly assigned to a learning object by ecological reasoning.

Results

As seen in Figure 1, the random associations of students with learning objects is clearly and consistently shown to be an inferior approach to improving the average knowledge of a group of students, as expected. Similarly, an omniscient sequencer using perfect knowledge of students, learning objects and the outcome of a potential interaction (greedy god) can consistently produce the greatest learning benefit.

Contrasting our ecological techniques (which would each be feasible in a real educational setting) with these reference points, provides illumination on the usefulness of the ecological approach in this setting. Reasoning intelligently, in this manner, has produced greater knowledge in a shorter number of trials for the group of students as a whole compared to a random association.

As asserted by (McCalla 2004), we see his predicted impact that with more learners the ecological approach's performance improves. A correlation of improved performance with an increase in number of learning objects was also seen. This makes intuitive sense: if an intelligent tutoring system (ITS) is given a larger repository of learning object to assign, we would expect it to be able to find objects better suited to a particular student.

While Figure 1 seems to show superior performance of the ecological approach with a pilot group, it is important to remember that 10% of the class was used as a pilot group for this experimental condition. These were not included in the average assessed knowledge. Their increased knowledge, which would be roughly equivalent to the lack of increase shown by random associations, is omitted and the improved performance of the remaining students can be viewed as at the expense of the pilot group.

The "Simulated Annealing" technique was interesting as it underperformed the other two techniques during its "cooling period" but quickly gained ground after the cooling period was complete. This is due to the randomness added to the during the cooling period leading to a greater exploration of the possible interactions between learning objects and students. This improved understanding of the two groups could then be used when reasoning about which students to match with which learning objects in later trials. In the largest condition (50 students and 100 learning objects), simulated annealing matched the performance of the pilot group condition, without sacrificing 10% of the class. With the correct choice of cooling periods, this technique shows promise for delivering comparable long term performance at the expense of early progress for the entire group instead of no progress for a pilot group.

Discussion

In this paper, we have proposed an algorithm to select learning objects to present to students, based on the previous experiences of other, similar students. As such, the tutoring of each new student can be viewed as peer-based, to the extent that student learning is enabled by previous peer interactions. Other researchers have explored a peer-based approach to intelligent tutoring. For example, the COMTELLA project at the University of Saskatchewan investigated recommendation of academic papers and similar resources. In the early phases of the work (Vassileva 2008) there was difficulty in getting users to accurately provide metadata when entering papers in the system. Subsequent

[5]Since each student's knowledge is now multi-dimensional the difference calculated to determine benefit is now a sum of the differences of each knowledge dimension. The numeric values for knowledge are converted to one of the concrete letter grade levels before performing the computation.

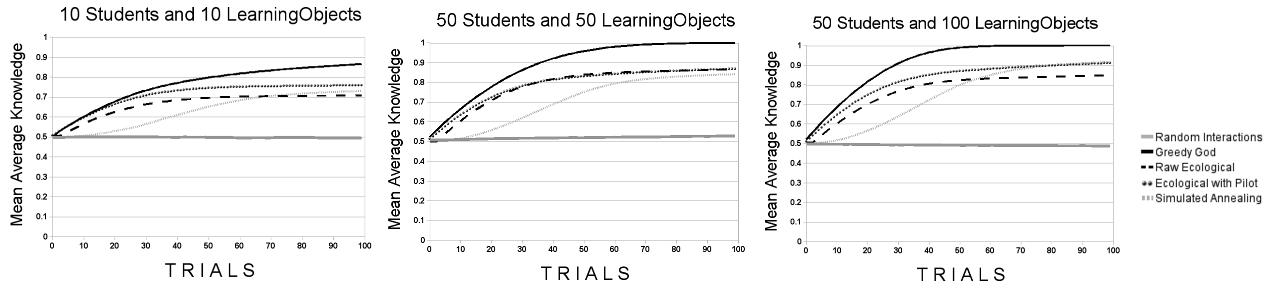

Figure 1: Comparison of 5 Approaches to Sequencing Learning Objects

work (Cheng and Vassileva 2006) has focused on providing incentives to encourage users to interact positively with the system. In contrast, our work avoids soliciting explicit feedback, and reasons based on typical usage.

Another example of peer-based learning is that of (Labeke, Poulovassilis, and Magoulas 2008), which uses collaborative filtering to match students based on their "lifelong learning". Matching to similar users occurs based on life events, such as a specific degree at a certain university or working at a specific company. Their results are presented transparently in a user centric approach where users can investigate the "trails" of similar users. Matching was done by using string metrics where life events are encoded into a token based string which is used to reason about similarities between users. Our work is distinct from the above approach, however, in a number of ways. Obtaining a history, and accurately categorizing a user's life events, will be a time consuming process that may be difficult to convince users to undertake. In contrast, our approach uses typical ITS interactions and doesn't elicit anything specific from the user. In their system user histories must be continually updated, with the ongoing issue of out-of-date user profiles. The data used by our system can be easily gathered in real-time by usage of the system and will be as up-to-date as their last usage of a learning object.

A different style of peer-based learning, called COPPER (Read et al. 2006), approaches the problem of ICALL. It intelligently matches students, and assigning them specific roles for their interaction, through a Bayesian approach, and allows them to help one another learn. Their approach could easily be integrated with ours, where an interaction between students is a learning object. While their approach is useful in real-time, it doesn't allow students to independently learn from the experience of previous student interactions with the system. Our system, in contrast, reasons using the entire experiences of all previous students, not just the current, online students.

In this paper, we have introduced experiments that simulate student learning, in order to validate our proposed model for content sequencing in intelligent tutoring systems. Other intelligent tutoring systems researchers have explored the value of simulating students.

(VanLehn, Ohlsson, and Nason 1996) discusses the authors' experiences with simulated students and the methods that can be used to assist in education. The authors claim that this is useful not only for providing a collaborative learning partner for a student but also for instructional developers to test systems that they develop, including early development where trials with human students may not be feasible. The authors highlight grain-size as an important spectrum for considering simulated students. For example, an example of fine-grained knowledge in physics is knowing the existence of tension in a string, when a string is tied to a body; an example of large-grained knowledge in physics would be simply knowing the law of conservation of energy.

Our system uses a granularity outside of this range, which we would term coarse-grained. As an example, a student might be modeled as having a 0.67, which could mean, for instance, that the student has enough knowledge to receive a 67% mark in Physics 101 or that they understand enough knowledge to complete 67% of the projects.

(VanLehn, Ohlsson, and Nason 1996) also specifically track and formally represent, for each student, their knowledge before learning, the behaviour during the learning, the instruction and the student's knowledge after learning. In contrast, we are interested in tracking behaviour with respect to learning objects, and focus on modeling the student's knowledge before and after interactions with those learning objects.

Another research group used what they call learning curve analysis to analyze how their simulated student performed (Matsuda et al. 2007). They measured the accuracy of production rules, in terms of successfully matching a step in solving the problem, compared to number of training problems or frequency of learning opportunities. We follow a similar approach in the evaluation of our work, where we use the resulting learning curves to contrast educational environments.

Conclusion and Future Work

In this paper, we have presented a concrete approach for selecting content for students in environments that make use of McCalla's ecological approach to intelligent tutoring – selecting learning objects based on other students' previous interactions. We make use of simulated students to verify the value of our proposed approach.

Since a study based on simulated learning is not confirmed to reflect real world learning, we note that extrapolations of results beyond the context of our current work should be made carefully. However, in addition to being expensive to develop, ITS are expensive to evaluate (VanLehn, Ohlsson, and Nason 1996). Our work should be viewed as a benchmark of a concrete implementation of an ecological approach to curriculum sequencing, as an overview for an approach to perform an inexpensive first evaluation of an ITS using a simulation, and as a suggestion for techniques that may be useful for systems working with human students. As such, our work serves to demonstrate the feasibility and the value of using simulated students in the design of an ITS.

There are obvious limits to how closely simulated students can correspond to real learners. We have by necessity been forced to omit issues such as learner motivation, affect and other emotional considerations which occupy the ITS research community. Yet, the abstractness of our model allows us to take a broad perspective on interpreting interactions, and the small impact derived from student interactions with inappropriate objects can be viewed as the student refusing to work with the object or quickly giving up on the interaction. For future work, we would explore experimenting with real students, to contrast those results with the results obtained in this work. Other worthwhile future work would include comparing and contrasting the impact of variations on the algorithm used to model learning, within our simulations. Several of these variations are discussed below.

Learning Models: Consider different models for learning, such as (Ohlsson 1993; VanLehn 1990), when simulating the interaction between learning objects and students. This would allow both our simulation approach to be considered from the perspective of various learning theories, and to provide an initial assessment of the impact of an ecological approach to systems built based on these descriptions of learning.

The Cold Start Problem: Investigate various initial conditions and the impact on system calibration. E.g. Expose all students to the entire system vs. some students to the entire system, vs. some students to part of the system, vs. all students to part of the system vs. known students to the entire system. Perhaps if exposing students to part of the system, investigate different "growth rates" of incorporating the rest of the system into the "active lessons".

Repository: Connected to the cold start problem, investigate techniques for using a repository of information, such as a collection of instructional videos or multiple recommended texts on a subject as a basis for automatically generating the core of an ITS.

Annotations: Model the impact of simulating students while allowing annotation of learning objects. These student-made modifications to the lesson will be intelligently shown (or not shown) to future students to adjust their learning. This will be validated using the simulation approach presented in this paper.

Curriculum Sequencing: Look at the raw ecological, pilot group and simulated annealing techniques and their impact on curriculum sequencing. After a group of students have used the system extensively, it should be possible to extract "stereotypes". By looking at the recommended sequence, it should be possible to take a student from the beginning of the course to the end with an automatically generating curriculum for future students who fit that stereotype.

Scaling: Look at how the system (and its growth) changes as more students and/or learning objects are added.

Another thread for future research is to expand our algorithm to include other modeling of learners and learning objects to determine characteristics of both that can be used to influence which object to present.

Acknowledgements

Thanks to Simina Brânzei and Lachlan Dufton for helpful advice and to NSERC's Strategic Networks of Research hSITE project for funding.

References

Breese, J. S.; Heckerman, D.; and Kadie, C. 1998. Empirical analysis of predictive algorithms for collaborative filtering. 43–52. Morgan Kaufmann.

Cheng, R., and Vassileva, J. 2006. Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. *User Model. User-Adapt. Interact.* 16(3-4):321–348.

Labeke, N.; Poulovassilis, A.; and Magoulas, G. 2008. Using similarity metrics for matching lifelong learners. In *ITS '08: Proceedings of the 9th international conference on Intelligent Tutoring Systems*, 142–151. Berlin, Heidelberg: Springer-Verlag.

Matsuda, N.; Cohen, W. W.; Sewall, J.; Lacerda, G.; and Koedinger, K. R. 2007. Evaluating a simulated student using real students data for training and testing. In Conati, C.; McCoy, K. F.; and Paliouras, G., eds., *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, 107–116. Springer.

McCalla, G. 2004. The ecological approach to the design of e-learning environments: Purpose-based capture and use of information about learners. *Journal of Interactive Media in Education: Special Issue on the Educational Semantic Web* 7:1–23.

Ohlsson, S. 1993. The interaction between knowledge and practice in the acquisition of cognitive skills. In Meyrowitz, A., and Chipman, S., eds., *Foundations of knowledge acquisition: Cognitive models of complex learning*. Norwell, Massachusetts, USA: Kluwer Academic Publishers. 147–208.

Read, T.; Barros, B.; Bárcena, E.; and Pancorbo, J. 2006. Coalescing individual and collaborative learning to model user linguistic competences. *User Modeling and User-Adapted Interaction* 16(3-4):349–376.

VanLehn, K.; Ohlsson, S.; and Nason, R. 1996. Applications of simulated students: An exploration. *Journal of Artificial Intelligence in Education* 5:135–175.

VanLehn, K. 1990. *Mind Bugs – The Origins of Procedural Misconceptions*. MIT Press.

Vassileva, J. 2008. Toward social learning environments. *IEEE Trans. Learn. Technol.* 1(4):199–214.