Exploring through Simulation the Effects of Peer Impact on Learning

Stephanie Frost[1] and Gord McCalla[1]

ARIES Lab, Dept. of Computer Science, U. of Saskatchewan, Saskatoon, Canada
stephanie.frost@usask.ca, mccalla@cs.usask.ca

Abstract. Simulation modelling helps designers to keep track of many possible behaviours in a complex environment. Having a technique to simulate the effect of peer impact on learning allows designers to test the social effects of their educational software. We implement an agent-based simulation model based on the ecological approach (EA) architecture [9]. The model considers learner attributes, learning object attributes and two styles of peer impact to explore the effects when learners are either positively or negatively impacted by high achieving peers. In this study, we observe different patterns of behaviour based on the style of peer impact and by limiting simulated learners' access to information (the EA metadata). Gaining understanding of these patterns will inform our future work on recommending sequences of learning objects (LOs).

Keywords: simulated learning environments, simulated learners, ecological approach, instructional planning

1 Introduction

Before taking an action in a learning environment, it is important for an intelligent tutoring system (ITS) to have some way of estimating the likelihood that the action will be successful, i.e. that it will benefit the learner(s) involved. To compute such an estimate, there are many dimensions to consider such as: the nature of the content being learned, the pedagogical style of the environment, learning goals, individual learner characteristics, and social factors such as how a learner's own performance can be influenced by knowledge of peer performance. Such complexity is often managed through the use of models.

Simulation modelling can be used by instructional developers for testing their systems; this was identified by VanLehn, Ohlsson and Nason [11] in a survey of possible uses of simulated students. One example is SimStudent by Matsuda et al. [8] which can be used by designers to explore through simulation the effects of various decisions on cognitive tutor design. Whether a model is used "internally" (by an ITS to compute the next action) or "externally" (to evaluate a system design), a challenge remains: How does the model estimate the amount of learning that occurs when a learner interacts with a Learning Object (LO)? In particular, we wanted to explore the impact on learning when learner performance is influenced by the performance of peers. Some learners may become encouraged when observing high peer achievement and perform even better than they would have otherwise. Other learners might become discouraged in the same situation and perform even worse. Having a technique to simulate the effects of peer performance would allow instructional developers to test social effects of their designs. In this paper, we use simulation to explore the behaviours exhibited by two different reactions to peer impact.

We describe our approach in Section 2, followed by the simulation study in Section 3. It is possible to simulate many different kinds of educational software in the ecological approach (EA) architecture [5], and then test the simulation under various conditions to get insight into issues the designer is interested in. Because our model is implemented in the EA architecture, our approach for modelling peer impact can be used across many different styles of learning systems. The data to feed our simulation is synthetic, but could, itself, be modelled on data extracted from actual learner behaviour [5]. We follow with a description of ongoing research that uses simulation for testing and developing a method for recommending sequences of LOs, and conclude with a discussion of our findings.

2 Model Structure

In another paper [5], we have argued that it is not necessary to model every detail of the learning process, but that systems can be tested in a simulation that captures only the most relevant characteristics for a given purpose. Therefore, we take an approach that lets an instructional developer choose different dimensions – such as attributes of the learning objects, aspects of the pedagogical environment, attributes of the learner – and assign weights to each dimension according to the priorities of the developer. This section describes the structure of the simulation model so as to provide background for the experiment around peer impact, described in Section 3.

The EA architecture [9] provides a way to record metadata about learner interactions with LOs. As learners interact with LOs, any information that is known about the learner at the time of the interaction can be saved as metadata and associated with the LO. The EA assumes that each learner is represented by a learner model that contains static attributes (*characteristics*) as well as other data gathered as they interact with the LOs (*episodic*).

We developed an agent-based simulation model with very simple abstractions of learners and LOs. Each learner agent has an attribute, *aptitude-of-learner*, a number between (0,1), which we use to model the range of aptitudes (low to high) different learners have for a given subject matter. In our model, this attribute is assigned at the start of the simulation and does not change, but in future work we plan to create more sophisticated simulations where this attribute is not static. The simulated LOs have an attribute to represent *difficulty level*, which is also a number between (0,1) where higher values represent more difficult material. The simulated LOs are arranged into a random directed acyclic graph to represent prerequisite relationships between the LOs.

The model execution revolves around an atomic action: the learner's interaction with a LO. This action might occur hundreds or thousands of times during a simulation run, thus creating a multitude of EA metadata from which measurements can be taken. In related work [5], we introduce the term *evaluation function* to describe the function that computes the degree of success as result of an interaction between a learner and a LO. We will use the term *P[learned]* to describe the value that is generated by the evaluation function, i.e. the "probability that the learner learned the LO", or the "system's belief that the learner knows the LO". The P[learned] value is included as part of the EA metadata that is associated with LOs after learners interact with them.

Our evaluation function is a weighted sum, where each term deals with a dimension of learning to be considered. Each dimension of learning is calculated with a mini function. For example, suppose $\text{Learner}_{\text{A}}$ were a novice with *aptitude-of-learner*=0.1. Next, suppose LO_{X} were a fairly easy LO, which implies a high probability of success. We use a mini function, *difficulty-of-LO*, to translate the LO difficulty attribute into a high probability value, giving *difficulty-of-LO*=0.8. Suppose we also wish to take into account that the likelihood of the learner learning the LO is higher if the learner has already viewed prerequisite LOs. Prerequisite information is given in the LO attributes. Our simulation model has a function for *hasPrerequisites* which searches through the EA metadata to discover whether the learner has indeed viewed the prerequisites and returns 1.0 if the answer is yes and 0.0 otherwise. If we want these dimensions to have approximately equal weights, then we can define the evaluation function below and obtain P[learned] as follows:

$$\begin{gathered}(\mathrm{w})(\text{aptitude-of-learner}) + (\mathrm{w})(\text{difficulty-of-LO}) + (\mathrm{w})(\text{hasPrerequisites}) \\ = (0.33)(0.1) + (0.33)(0.8) + (0.34)(1.0) = 0.637\end{gathered}$$

If, on the other hand, we wish to give the aptitude a higher weight, such as 60%, then the new value could be $(0.6)(0.1) + (0.2)(0.8) + (0.2)(1.0)$, or 0.42. As expected, giving greater weight to this learner's low aptitude decreases the P[learned] somewhat. More dimensions can be incorporated so long as the weights sum to 1.0. The evaluation function, implemented as a weighted sum, will provide an estimated likelihood the LO has been learned between (0,1), making it easy to compare averages of such P[learned] values between simulation runs. However, we caution against comparing two simulation runs with different evaluation functions (i.e. different weights or dimensions) because that would be like comparing two numbers with different units of measure.

The independent variables in our experiment are the *aptitude-of-learner* values, the *difficulty level* values, the directed acyclic graph giving prerequisite relationships between LOs, as well as a dimension called *peer-impact*, which is explained in the next section.

3 Experiment

Our experiment is intended to explore through simulation the effects of peer impact on learning. We motivate the experiment by visiting literature around how peers can impact each other's scores.

Students are impacted by their peers even in their ordinary lives. A study was performed by Hanushek et al. [6] to clarify the impacts of peer group characteristics on achievement in the context of family and school factors, race and socio-economic status. Results suggested that students benefitted from higher achieving schoolmates. In contrast, the American Academy of Pediatrics warned that Facebook pages can make some children feel badly because they see themselves as being inferior to their peers [10]. This effect is due to the nature of Facebook, where most users will censor their posts and only share the most positive information about themselves, skewing the view of reality. Along the same lines, Daniel et al. [3] found in a study that learners will usually only participate in online learning activities if they have trust in their peers or some degree of self confidence.

Others have used simulations to study peer effects. Mao et al. [7] used a simulation model to study the impact of social factors in a course where students shared learning materials with each other. The output of Mao et al.'s model was a comparison of the amount of sharing connected to status levels: gold, silver, bronze, common. Populations fluctuated as users began at the common status and gradually transitioned between levels. The paper concluded that simulation models can be useful for developing and improving incentive mechanisms in virtual communities. In a different study, Zhang et al. [12] studied the fluctuation of a population of learners through various activities: registration, activation, action and adaptation. The authors found that learners who participated the most were also the ones most sensitive to changes in the community and had the most fluctuations.

This research, and other research, shows that a learner's score can be impacted by peer performance. We decided to explore this issue by creating a notion of "peer impact", where learners respond differently from one another according to how well other learners are doing in mastering the LOs. This takes the form of a new dimension in our evaluation function called *peer-impact*. Like the other dimensions we discussed in Section 2 (*aptitude-of-learner*, *difficulty-of-LO*, *hasPrerequisites*), this is a function that produces a value between (0,1) to represent a positive or negative impact on P[learned]. In our experiment, we use the following Equation 1 to compute P[learned] each time a learner visits a LO.

$$.25(\text{apt-of-learner}) + .25(\text{diff-of-LO}) + .25(\text{hasPrereq}) + .25(\text{peer-impact}) \tag{1}$$

We created two styles of peer impact called *reinforcing* and *balancing* which refer to a comparison between an individual learner's average P[learned] on the LOs they have viewed so far, compared to the average P[learned] of all learner agents, which we call "class average". The information to compute these

P[learned] averages is obtained from the EA metadata. Each learner is given one of these styles at the start of the simulation and it remains fixed. Future work could explore more sophisticated learner agents where this attribute is not static.

The reinforcing style means that the learner's score is "attracted" to the class average P[learned]. That is, when the class average is higher than their own, the peer impact function for a reinforcing learner produces a value close to 1; thus the learner will perform even better than they would have otherwise. This is a positive feedback loop, because as the learner performs better so does the class average thus further encouraging the learner to do better. If the class average is lower than their own, then the *peer-impact* function gives a value close to zero; thus the learner will do even worse than they would have otherwise.

Balancing is the opposite. In this case, a learner's score is "repelled" from the class average P[learned]. That is, when the class average is higher than the individual's average P[learned], then their score will be pulled down lower than it would have been otherwise. This is a negative feedback loop because when the class average is high, the learner's average goes in the other direction. When the class average is low, then the learner's score will be boosted higher than it would have otherwise. In Figure 1, we show the *peer-impact* function (the values 0.2 and 0.8 were chosen as thresholds to allow clear effects of the two types of learner to emerge).

```
if currentLearner BALANCING
   if class average is HIGHER than mine
      set peerImpact == randomNumBetween(0.0,0.2)
   if class average is LOWER than mine
      set peerImpact == randomNumBetween(0.8,1.0)
if currentLearner REINFORCING
   if class average is HIGHER than mine
      set peerImpact == randomNumBetween(0.8,1.0)
   if class average is LOWER than mine
      set peerImpact == randomNumBetween(0.0,0.2)
```

Fig. 1. Function to generate *peer-impact* for a given learner at a given time in the simulation

The dependent variable in our experiment is the P[learned] values generated by the simulation; we gain insight into whether the peer impact has a positive or negative effect by observing the relative P[learned] values. We varied this experiment under six conditions. We varied the proportions of balancing and reinforcing styles: mostly balancing, mostly reinforcing, and fifty-fifty. For instance, if the model is set to mostly balancing, when new learners are initialized, they have a high chance of being assigned the balancing personality and a low chance of being assigned the reinforcing personality. These three proportions were each run under two difficulty levels: one with mostly easy LOs and high aptitude learners, and the other with mostly difficult LOs and low aptitude learners. These six conditions were hand picked to be representative samples on a curve of possible population mixes that should provide some insight about the effect of these two kinds of personality on the learning environment. We ran each of the six conditions 5 times because our model is stochastic; it produces slightly different results each time even under the same starting conditions.

A typical result is shown in Figure 2 (fifty-fifty, high difficulty with low aptitude learners). Each line represents the average P[learned] of different portions of the simulated learner population: the lightest thin line for all learners, black thin line for the learners who were assigned the reinforcing personality, and the dark grey thin line for the learners who were assigned the balancing personality. Normally, our simulation model would be used to evaluate a particular instructional planning technique, but because this experiment is intended to illuminate peer impact, the order in which LOs are consumed isn't important. Therefore, the simulated learners, of which there are 80, visited random LOs, of which there are 100.

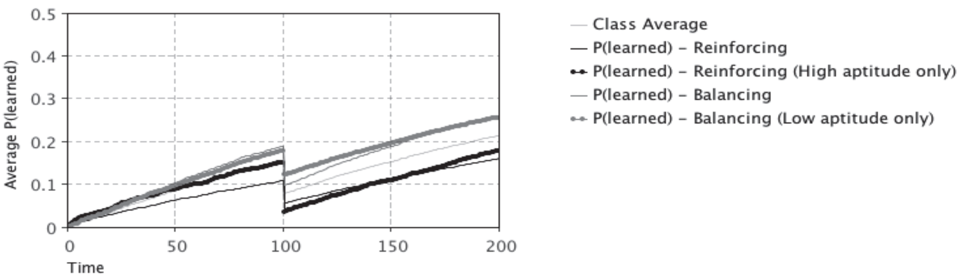

Fig. 2. Typical result

At the start of the simulations, the class average starts at zero. The balancing simulated learners had higher scores in this state because this is the behaviour defined in the evaluation function – that balancing learners do well when the class average is lower than their individual average. The learning gradually increases for both groups as the simulated learners visit more and more LOs. Although the results seem low overall – P[learned] only reaching short of 0.3 – this is due to the number of LOs (100) created in the simulation and the time it would take for learners to visit them all. We ran the simulation again with only 30 LOs and observed the same patterns, but with a steeper slope; the average P[learned] reached around 0.5. This raises interesting questions about whether the amount of time required to learn a set of LOs should actually be represented with a linear function. In reality, learners would get tired or lose interest or change their learning goals. Future work could compare instructional plans with learners having different levels of stamina.

The thick lines in Figures 2 and 3 represent subsets of the balancing and reinforcing personalities whose behaviour we wish to discuss in this experiment. Simulated learners do not have access to the actual class average, but compute the average based on what other simulated learners have allowed them to perceive about their performance. Based on Daniel et al.'s [3] results that confident learners are more likely to share their success, simulated learners with high P[learned] values shared their EA metadata, while those with lower P[learned] values did not. This creates a *suppression effect*, where each simulated learner has access to different information in the computation of how others are doing, depending on which other learners have suppressed information at the time they are computing the average.

The thick grey line shows only the balancing learners with low aptitudes while the thick black line shows only the reinforcing learners with high aptitudes. At the start of the simulation, the thick black line is below the thick grey line: it is perhaps surprising that a group of simulated learners with high aptitudes would have overall lower scores than a group of simulated learners with low aptitudes. We highlight this because it shows that different parts of the evaluation function – *peer-impact*, *aptitude-of-learner* etc. – can dominate at different times. In this case, high aptitude can be dominated by peer impact for reinforcing personalities when the class average is low.

In Figure 3, we observe another interesting phenomenon by injecting 80 more simulated learners halfway though the experiment, a somewhat contrived situation, although one that might happen in the real world if, say, two classes merged partway through a course, or if two study groups in an online course were mashed together, or due to the openness of many online courses (e.g. MOOCs) when new learners can join any time. Under most of the experimental conditions we tried, such as the typical result in Figure 2, although the influx of new learners caused the class average to drop (as expected, because each new learner starts with an average P[learned] of zero), there was no apparent change in the relative ranking of the groups of learners being measured. That is, if the balancing learners had the highest average before the influx, this continued afterward. However, in about a third of the runs with low difficulty LOs and high aptitude learners, the influx of learners caused a *phase shift*: now the thick black line jumps above the thick grey line (see Figure 3). This makes sense: the balancing learners who tend to do more poorly when the class average drops, do just that. The influx also creates a situation where there are now learners with high averages intermingled with learners with zero averages; this creates a different environment than the starting condition where everyone started at zero. Different environmental conditions cause the model to exhibit different behaviour. With the suppression effect deactivated, all learners have access to the same information. In this condition, we observed that the thick grey line overlapped with the thick black line and there was no apparent phase shift (i.e. no lines crossing over).

Even though the observed patterns are merely a result of the evaluation function implementation – that is, the model is simply doing what it was programmed to do – it helps system designers to keep track of the different possible

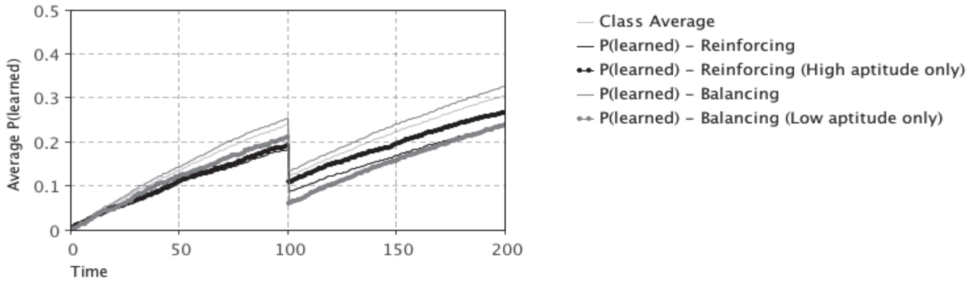

Fig. 3. Condition showing phase shift

behaviours as they try to design systems to support learning in all of these conditions: low or high aptitude learners, easy or difficult material, peer effects, prerequisites and many other possible dimensions, with each behaving differently in different situations. Without simulation, it is unlikely we would have made our observations about the phase shift as well as the observation about the high aptitude reinforcing learners having lower scores than low aptitude balancing learners. These observations reveal the specific circumstances that instructional developers should address in order to maximize the expected learning. For example, the system could be programmed to intervene when it detects that the current class average will push a learner's expected outcome in an undesirable direction. When the class average is higher than an individual's average, the scores of other learners should be displayed more prominently for the balancing learners but not for the reinforcing learners.

Through this experiment, we have also shown that simulations can be used to test unexpected situations. Future experiments could test for influxes of new LOs instead of new learners. Other variations could look at adding or removing LOs to impact the difficulty level of the course or the level of expertise of peer learners. When we injected a herd of simulated learners, we observed some surprising results. But, by examining the underlying dynamic behaviour as the simulation proceeded, we could actually explain why these results happened, thus gaining more intuition about learning that would help to better inform an experiment that might be carried out with real learners.

4 Other Research Directions

In ongoing work, we are also developing a technique for recommending sequences of LOs. Instructional planners have been built that explore different kinds of sequencing such as sequencing things of the same type, like "lessons" or even sequencing several types of activities, like presentations and assessments [1]. Our method involves using the EA metadata to identify "trails" of LOs. We are investigating the use of user-based and item-based approaches to generate recommendations of these trails using Apache Mahout [1]. Using information captured in the EA metadata, we create metrics for giving sequences a score to reflect the quality of the sequence, for example does P[learned] increase or decrease over the sequence. We are also exploring changes to the evaluation function to favour sequences that suggest coherence, such as trails that give learners a view of the big picture before going into the details. Sequences with high scores are then used as a basis for recommending sequences to other learners. Our study will examine whether learners receiving sequence recommendations see any improvement over learners receiving one LO recommendation at a time.

Other work in simulating recommender systems for learning systems has been done by Drachsler et al. [4]; but the main difference is that this work did not involve sequences, peer impact or the EA architecture. Champaign [2] uses the ecological approach architecture to use the experiences of past learners to suggest sequences of LOs for future learners while also studying the impact of peer ratings, which are not the same as our peer impact because our peer impact is linked to the evaluation function.

Even with the simplistic models of learners and LOs we have presented so far, the peer impact experiment demonstrates the combinatorics of the various features is already becoming too complex to rely on human intuition; this is one of the main reasons for simulation modelling.

5 Conclusion

We created simulated learners whose overall learning was influenced by one of two styles of peer impact. Our study demonstrated that different patterns emerge when when simulated learners change their own behaviour based on the behaviour of the group and when these learners have limited access to information due to others' ability to suppress their EA metadata. In some conditions, a phase shift occurred from the initial situation where the class average is zero to a new situation with some learners having relatively high averages. The simulated learners prior to the influx had higher averages because they had the opportunity to visit LOs before the arrival of the new simulated learners. One style of peer impact is not universally better or worse than another, but each has advantages in different circumstances. It is important for instructional developers to understand such patterns. In future work, the use of simulations with the EA architecture will shed more light on peer impact and will allow us to also factor in the effects of different kinds of sequence recommendations.

The EA metadata make it easy to look deeply into the underlying dynamics and identify the conditions that create such behaviours. The EA metadata also allow us to change the inputs of the simulation and take measurements, as we did to compare the P[learned] averages between learners with different styles of peer impact. By using the EA architecture for the simulation studies, the later construction of a real learning system is made easier if the real system also uses

[1] http://mahout.apache.org/

the EA architecture. That is, if the real system also stores information about a learner's interaction with a LO as metadata associated with the LO, then estimating the likelihood of success for a real learner follows the same methods used by developers to estimate the success of simulated learners.

Acknowledgements
We would like to thank Dr. Julita Vassileva for discussions with the first author about using simulations for instructional planning and for insights on applying social concepts in the simulation model; Graham Erickson for his ideas about the evaluation function; and Dr. Nathaniel Osgood for discussions about cause loop diagrams and agent-based modelling in AnyLogic. We also wish to acknowledge the Natural Sciences and Engineering Research Council of Canada for funding some aspects of this research through a Discovery Grant to the last author.

References

[1] Brusilovsky, P. and Vassileva, J.: Course sequencing techniques for large-scale web-based education. International Journal of Continuing Engineering Education and Lifelong Learning, 13, 75-94 (2003).
[2] Champaign, J.: Peer-based intelligent tutoring systems: a corpus-oriented approach. Ph.D. Thesis, University of Waterloo, Waterloo, Canada (2012)
[3] Daniel, B., McCalla, G., Schwier, R.: Social Network Analysis techniques: implications for information and knowledge sharing in virtual learning communities. Int. J. of Interactive Media in Education, 2(1), 20-34 (2008)
[4] Drachsler, H., Hummel, H., and Koper, R.: Using simulations to evaluate the effects of recommender systems for learners in informal learning networks. In Learning, 3 CEUR Workshop Proc., 404-423 (2008).
[5] Erickson, G., Frost, S., Bateman, S., and McCalla, G.: Using the ecological approach to create simulations of learning environments. To appear in Lane, H.C., Yacef, K., Graesser, A., Mostow, J. (eds.), Proc. 16th Int. Conf. on AIED, Memphis (2013)
[6] Hanushek, E., Kain, J., Markman, J., Rivkin, S.: Does peer ability affect student achievement? Journal of Applied Economics, 18, 527-544 (2003)
[7] Mao, Y., Vassileva, J., and Grassmann, W.: A system dynamics approach to study virtual communities. In Proc. 40th Annual Hawaii Int. Conf. on System Sciences, HICSS '07, pp.178a-, Washington, DC, USA, IEEE Computer Society (2007)
[8] Matsuda, N., Cohen, W.W., Sewall, J., Lacerda, G. and Koedinger, K.R.: Predicting students performance with SimStudent that learns cognitive skills from observation. In Luckin, R., Koedinger, K.R., and Greer, J. (eds.), Proc. 12th Int. Conf. on AIED, Marina del Rey, 467-476 (2007)
[9] McCalla, G: 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.
[10] Tanner, L: Docs warn about teens and 'Facebook depression'. Associated Press. http://www.msnbc.msn.com/id/42298789/ns/health-mental_health/t/docs-warn-about-teens-facebook-depression/ Accessed April 13, 2013.
[11] VanLehn, K., Ohlsson, S., and Nason, R.: Applications of simulated students: an exploration. Int. J. Artificial Intelligence in Education, 5, 135-175 (1996)
[12] Zhang, Y., and Tanniru, M.: An agent-based approach to study virtual learning communities. Hawaii International Conference on System Sciences, 1(11c) (2005)