The Use Simulated Learners in Adaptive Education

Erik Harpstead, Ph.D.[1], Christopher J. MacLellan, Ph.D.[2], Daniel Weitekamp, B.A.[1] and Kenneth R. Koedinger, Ph.D.[1]
[1]Carnegie Mellon University, USA, *harpstead@cmu.edu*
[2]Soar Technology, Inc., USA, *chris.maclellan@soartech.com*

I. INTRODUCTION

Adaptive instructional software like intelligent tutoring systems have improved learning outcomes for students across several domains. However, building pedagogically effective systems at scale is a challenging problem. In addition to being expensive and technically challenging to build, they rely on accurate models of student learning informed by theories of how people learn. In our work we have explored the use of simulated learners for:

1. Efficiently author instruction at scale
2. Evaluate pedagogical effectiveness of instruction
3. Test theories of how humans learn

In this paper, we describe our computational approach to modeling human learning within the Apprentice Learner (AL) architecture [1] and explore the use of these models in the context of four real-world use cases. Our computational approach learns from interactive instruction—mimicking the learning process of real students. We also discuss methods of evaluating simulated learners present initial evidence of their potential.

II. RELEVANT THEORIES OF LEARNING

There are three lines of theories that are relevant to our work: statistical learning theories, computational learning theories, and instructional learning theories. For purposes of modeling human learning and performance, statistical approaches to cognitive modeling consist of a set of independent acquirable skills known as knowledge components (KC) [2] combined with a mapping of KCs to observable steps in an instructional environment. Methods like Bayesian Knowledge Tracing (BKT) [3], and Additive Factors Model (AFM) [4] use this mapping to predict the probability that a particular student will correctly complete the next step in a problem without assistance. However, one limitation of statistical models is that they require student data in order to estimate key parameters (e.g., students' knowledge and learning rates). This makes it difficult to apply these models prior to collecting data from technology deployments.

To overcome this limitation, we have been exploring the development and use of computational learning theories [1]. Models based on these theories extend prior cognitive architecture research [5] to explain how humans learn from worked examples and feedback. These models only rely on the task structure and presentation of materials to predict the acquisition of skills and misconceptions. Thus, they are able to predict human behavior within adaptive learning technology before any data is collected.

We have leveraged our computational models, such as SimStudent [6] and AL agents [1], to test multiple instructional theories. There are a wide range of instructional approaches [7], such as blocked vs. interleaved instruction [1] and learning-by-teaching [8] and identifying the best instructional approach for a given learning scenario is a major challenge. Our research aims to model human learning across a wide range of domains and tasks under different instructional approaches, to better understand when each approach is best suited to improving pedagogical outcomes.

III. ENABLING TECHNOLOGICAL ADVANCES

Our work on simulated learners is supported by the AL architecture, a framework for modeling the student learning process. Agents created within AL are simulations of inductive learning from examples and feedback. AL agents are designed to support interaction patterns similar to an intelligent tutoring system [9]. Given the state of an interface (Fig. 1) an AL agent will either attempt a problem solving step if it has an applicable skill and process any resulting feedback; or, if the agent does not have an appropriate skill for the context it will request an example of what to do and induce a new skill from the example. AL agents learn by an iterative process of trial and error, bootstrapped by a small number of domain-general operators.

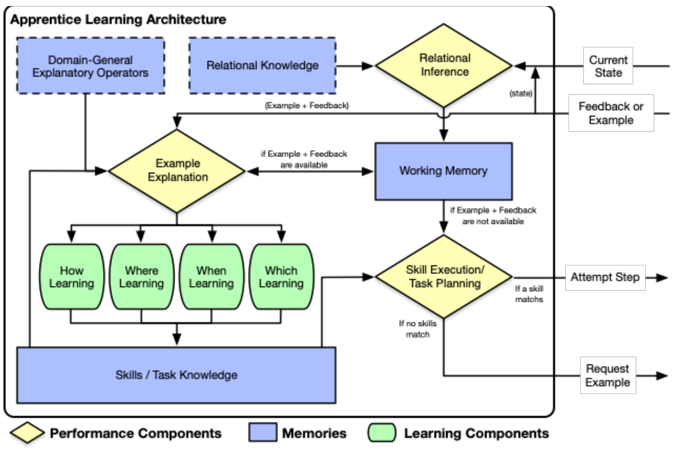

Fig. 1. A diagrammatic representation of the Apprentice Learner Architecture showing the flow of examples and feedback through the different learning mechanisms.

AL agents generate production rules akin to the chunks of procedural knowledge outlined in ACT-R theory [10], by chaining together operators to explain feedback provided by a human instructor or intelligent tutoring system. To achieve this functionality the AL architecture factors learning into four learning mechanisms that each handle different aspects of production rule learning. The *where-* and *when-learning* mechanisms determine the context in which a production rule should fire, the *how-learning* mechanism searches for chains of domain-general operators that explain tutor feedback, and the *which-learning* mechanism determines which production rule should fire if multiple are applicable.

The four separate learning mechanisms of the AL architecture are designed to be modular, allowing for easy testing of different hypotheses of student learning. For example, an AL agent might instantiate *when-learning* using a decision tree that is retrained for each new example, or it might use an incremental categorization learner such as TRESTLE [11] which incorporates new examples incrementally. Additionally, a single algorithm may fulfill multiple roles within the architecture. For example, *when-* and *where-learning* may be combined.

IV. Real World Applications

We have explored the application of simulated learners to three use cases: tutor authoring, cognitive crash testing, and learning theory testing. It is a widely known that authoring cognitive models for adaptive technologies is a difficult and time consuming process [12]. Our simulated learning models enable teachers and other non-programmers to efficiently author tutoring systems by teaching a simulated agent as they would a student—through tutoring rather than programming—and the agent automatically learns a cognitive model that can power adaptive instruction [13].

Once adaptive software is developed, it is difficult to know if it achieves desired learning outcomes. Previous work has explored how A/B testing can be used to evaluate which version of an adaptive system is better [14], but testing technology with real users is an expensive and time consuming process. Our learning models enable researchers to simulate A/B experiments by generating entire cohorts of simulated students to "crash test" different versions of instructional technologies prior to deploying them to real students.

After learning technologies have been deployed, we can leverage them to collect educational data to improve our understanding of human learning. Approaches such as A/B testing only provide limited information to make design decisions between specific versions of a product. However, by evaluating how well the behavior of alternative computational models fits human behavior we can empirically test and improve our underlying theories of how humans learn. These improvements to our understanding of the learning process synergistically feed back into the other use cases—better supporting the design and evaluation of future learning technologies.

V. Evidence of Potential Impacts

We evaluated the use of simulated learners for the use cases. First, we applied simulated agents to tutor authoring. Fig. 2 demonstrates the time it would take to author an algebra tutor using either Example Tracing [15] or an AL agent. We found that AL cut authoring time in half compared to Example Tracing, which already reduces authoring time by 75-80% over hand authoring [15]. We also applied simulated learners to discovering KC models. We trained simulated learners on items across three domains (see Table 1) and labeled items using the learned skills [16]. We found that simulated learners produced KC models that fit human data as well as, or better, than human-generated models. These findings suggest simulated agents can drastically reduce the cost of building tutors.

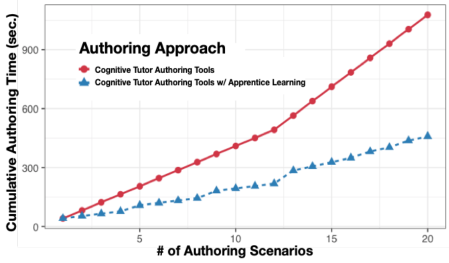

Fig. 2. A comparison of domain model authoring time for Cognitive Tutor Authoring Tools with and without AL agents (for an algebra tutor)

TABLE I. Model Fit Statistics of SimStudent-Generated and Human-Generated Knowledge Component Models in Terms of Their Fit to Human Data

	Human-Generated Model AIC	Simulated-Student Model AIC
Algebra	6534.07	**6448.1**
Stoichiometry	17380.9	**17218.5**
Fraction Addition	**2112.82**	2202.02

Next, we evaluated the use of simulated learners for predicting the outcome of controlled A/B experiments. We simulated learners using two versions of a fraction arithmetic tutor and gave them a simulated posttest [17]. We found that these simulations are able to correctly predict the outcome of an actual classroom A/B experiment (Fig. 3) and demonstrate how simulated learners can support evaluation of learning technologies prior to deployment.

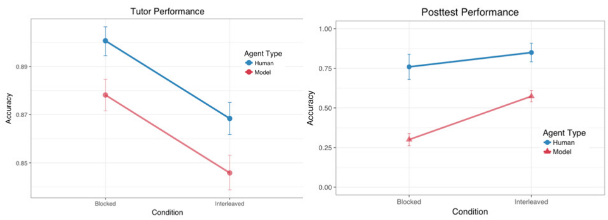

Fig. 3. Simulated agents are able to predict the main experimental effects of classroom A/B experiments, such as predicting that students practicing fractions problems in an interleaved order will perform worse in the tutor but better on a posttest.

Finally, we explored the use of AL for improving our theories of learning. We compared two theories of how students learn: one positing they revise their skills by considering all previous applications (non-incremental) and one positing they revise skills only considering new examples (incremental). We instantiated two variants of simulated students that embody each theory and simulated behavior across seven tutoring systems. We found the incremental model better predicts human behavior, suggesting it is a better model of the human learning

[17] (Fig. 4). These results demonstrate the use of AL for testing and improving our understanding of human learning.

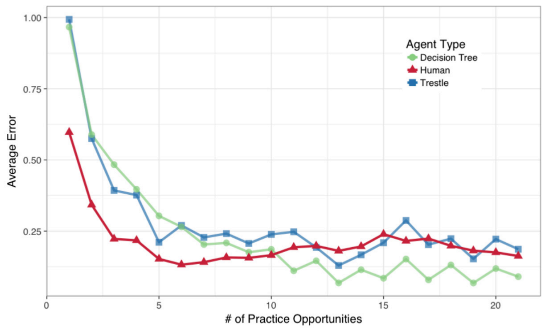

Fig. 4. A comparison of non-incremental (decision tree) and incremental (Trestle) apprentice learning agents and human learning (across seven domains).

VI. SUMMARY

We describe recent advances in simulating human learning and discuss our applications of simulated learning agents to three real-world use cases: tutor authoring, evaluating the effectiveness of instruction, and testing learning theories. We start by reviewing the Apprentice Learner (AL) architecture, a framework for generating and testing simulated learners. We then review relevant learning theories and describe our real-world use cases. Finally, we present evidence from multiple simulated learner studies highlighting their potential impact for reducing tutor authoring costs, predicting the outcomes of A/B experiments before deployment, and testing and improving theories of how people learn.

REFERENCES

[1] C. J. MacLellan, E. Harpstead, R. Patel, and K. R. Koedinger, "The Apprentice Learner Architecture: Closing the loop between learning theory and educational data," in *Proceedings of the 9th International Conference on Educational Data Mining - EDM '16*, 2016, pp. 151–158.

[2] K. R. Koedinger, A. T. Corbett, and C. Perfetti, "The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning," *Cogn. Sci.*, vol. 36, no. 5, pp. 757–798, Jul. 2012.

[3] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapted Interact.*, vol. 4, no. 4, pp. 253–278, 1995.

[4] H. Cen, K. Koedinger, and B. Junker, "Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement," in *Proceedings of the 8th International Conference on Intelligent Tutoring Systems - ITS 2006*, 2006, pp. 164–175.

[5] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cogn. Syst. Res.*, vol. 10, pp. 141–160, 2009.

[6] N. Li, N. Matsuda, W. W. Cohen, and K. R. Koedinger, "Integrating representation learning and skill learning in a human-like intelligent agent," *Artif. Intell.*, vol. 219, pp. 67–91, 2015.

[7] K. R. Koedinger, J. L. Booth, and D. Klahr, "Instructional Complexity and the Science to Constrain It," *Science (80-.).*, vol. 342, no. 6161, pp. 935–937, Nov. 2013.

[8] N. Matsuda, E. Yarzebinski, V. Keiser, R. Raizada, W. W. Cohen, G. J. Stylianides, and K. R. Koedinger, "Cognitive anatomy of tutor learning: Lessons learned with SimStudent.," *J. Educ. Psychol.*, vol. 105, no. 4, pp. 1152–1163, 2013.

[9] K. VanLehn, "The Behavior of Tutoring Systems," *Int. J. Artif. Intell. Ed.*, vol. 16, no. 3, pp. 227–265, 2006.

[10] J. R. Anderson, *Rules of the Mind*. Psychology Press, 2014.

[11] C. J. MacLellan, E. Harpstead, V. Aleven, and K. R. Koedinger, "TRESTLE: A Model of Concept Formation in Structured Domains," *Adv. Cogn. Syst.*, vol. 4, pp. 131–150, 2016.

[12] T. Murray, "An Overview of Intelligent Tutoring System Authoring Tools: Updated Analysis of the State of the Art," in *Authoring Tools for Adv. Tech. Learning Env.*, Murray, S. E. Ainsworth, and Blessing, Eds. 2003, pp. 493–546.

[13] N. Matsuda, W. W. Cohen, and K. R. Koedinger, "Teaching the Teacher: Tutoring SimStudent Leads to More Effective Cognitive Tutor Authoring," *Int. J. Artif. Intell. Educ.*, vol. 25, no. 1, 2014.

[14] R. Liu and K. R. Koedinger, "Closing the loop: Automated data-driven cognitive model discoveries lead to improved instruction and learning gains," *J. Educ. Data Min.*, vol. 9, no. 1, pp. 25–41, 2017.

[15] V. Aleven, B. M. McLaren, J. Sewall, M. van Velsen, O. Popescu, S. Demi, M. Ringenberg, and K. R. Koedinger, "Example-Tracing Tutors: Intelligent Tutor Development for Non-programmers," *Int. J. Artif. Intell. Educ.*, vol. 26, no. 1, pp. 224–269, Mar. 2016.

[16] N. Li, E. Stampfer, W. W. Cohen, and K. R. Koedinger, "General and Efficient Cognitive Model Discovery Using a Simulated Student," in *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*, 2013, no. 2009.

[17] C. J. MacLellan, "Computational Models of Human Learning: Applications for Tutor Development, Behavior Prediction, and Theory Testing," Carnegie Mellon University, 2017.