but there is evidence that it may accelerate the authoring process and produce more accurate cognitive models than hand authoring. One demonstration explored the benefits of a traditional programming by demonstration approach to authoring in SimStudent versus a programming by tutoring approach [2]. In the latter, SimStudent asks for demonstrations only at steps where it has no relevant productions. Otherwise, it performs a step and asks the author for feedback as to whether the step is correct or not. Programming by tutoring was found to be much faster than programming by demonstration (77 minutes vs. 238 minutes) and produced a more accurate cognitive model whereby there were fewer productions that produced over-generalization errors. Programming by tutoring is now the standard approach because of its improved efficiency and effectiveness. Better efficiency is obtained because many author demonstrations are replaced by SimStudent actions with a quick yes-or-no response. Better effectiveness is obtained because these actions expose over-generalization errors to which the author responds "no" and the system learns new if-part preconditions to more appropriately narrow the generality of the modified production rule.

A second demonstration of SimStudent as an authoring tool [22] compared authoring in SimStudent with authoring example-tracing tutors in CTAT. Tutoring SimStudent has considerable similarity with creating an example-tracing tutor except that SimStudent starts to perform actions for the author, which can be merely checked as desirable or not, saving the time it otherwise takes for an author to perform those demonstrations. This study reported a potential savings of 43% in authoring time.

5 Evaluating a Simulated Learner as a Teachable Agent

Simulated learner systems can be more directly involved in helping students learn when they are used as a teachable agent whereby students learn by teaching [cf., 23]. Evaluating the use of an SL in this form ideally involves multiple steps. One should start with an SL that has already received some positive evaluation as a good model of student learning (see section 2). Then incorporate it into a teachable agent architecture and, as early and often as possible, perform pilot students with individual students [cf., 24 on think aloud user studies) and revise the system design. Finally, for both formative and summative reasons, use random assignment experiments to compare student learning from the teachable agent with reasonable alternatives.

Using SimStudent, we built a teachable agent environment, called APLUS, in which students learn to solve linear equations by teaching SimStudent [25]. To evaluate the effectiveness of APLUS and advance the theory of learning by teaching, we conducted multiple *in vivo* experiments [25,26,27,28]. Each of the classroom studies have been randomized controlled trials with two conditions varying one instructional approach. In one study [25], the self-explanation hypothesis was tested. To do so, we developed a version of APLUS in which SimStudent occasionally asked "why" questions. For example, when a student provided negative feedback to a step SimStudent performed, SimStudent asked, "Why do you think adding 3 here on both sides is incorrect?" Students were asked to respond to SimStudent's questions either by selecting pre-specified menu items or entering a free text response. The results showed that the amount and the level of elaboration of the response had a reliable correlation with students' learning measured by online pre- and post-tests.

6 Conclusion

We outlined four general purposes for simulated learners (see Table 1) and reviewed methods of evaluation that align with these purposes. To evaluate an SL as a precise theory of learning, one can evaluate the cognitive model that results from learning, evaluate the accuracy of error predictions as well as prior knowledge assumptions needed to produce those errors, or evaluate the learning process, that is, the changes in student performance over time. To evaluate an SL as an instructional test, one should not only evaluate the SL's accuracy as a theory of student learning, but should also perform human experiments to determine whether the instruction that works best for SLs also works best for human students. To evaluate an SL as an automated authoring tool, one can evaluate the speed and precision of rule production, the frequency of over-generalization errors and the fit of the cognitive models it produces. More ambitiously, one can evaluate whether the resulting tutor produces as good (or better!) learning than an existing tutor. Similarly, to evaluate an SL as a Teachable Agent, one can not only evaluate the system features, but also perform experiments on whether students learn better with that system than with reasonable alternatives.

Simulated learner research is still in its infancy so most evaluation methods have not been frequently used. We know of just one such study [29] that evaluated an SL as an instructional tester by following up a predicted difference in instruction with a random assignment experiment with real students. It used an extension of the ACT-R theory of memory to simulate positive learning effects of an optimized practice schedule over an evenly spaced practice schedule. The same experiment was then run with human students and it confirmed the benefits of the optimized practice schedule. Such experiments are more feasible when the instruction involved is targeting simpler learning processes, such as memory, but will be more challenging as they target more complex learning processes, such as induction or sense making [31].

The space of instructional choices is just too large, over 200 trillion possible forms of instruction [32], for a purely empirical science of learning and instruction to succeed. We need parallel and coordinated advances in theories of learning *and* instruction. Efforts to develop and evaluate SLs are fundamental to such advancement.

References

1. VanLehn (1991). Two pseudo-students: Applications of machine learning to formative evaluation. In Lewis & Otsuki (Eds.), *Adv Res on Comp in Ed* (pp. 17-26). Amsterdm: Els.
2. Matsuda, Cohen, Koedinger (2015). Teaching the teacher. *Int J of AI in Ed, 25*, 1-34.
3. Li, Matsuda, Cohen, Koedinger (2015). Integrating representation learning and skill learning in a human-like intelligent agent. *AI, 219,* 67-91.
4. Anzai, Y. & Simon (1979). The theory of learning by doing. *Psych Rev, 86 (2)*, 124-140.
5. Anderson, J.R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Hillsdale: Erl.

6. Laird, Newell, & Rosenbloom (1987). Soar. *AI, 33*(1), 1–64.
7. Langley & Choi (2006). A unified cognitive architecture for physical agents. In *Proc of AI*.
8. Newell, Allen. 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard U. Press.
9. Tenenbaum, J. B., Griffiths, T. L., & Kemp.C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, *10*, 309-318.
10. MacLaren & Koedinger (2002). When and why does mastery learning work: Instructional experiments with ACT-R "SimStudents". In *Proc of ITS*, 355-366. Berlin: Spr-Ver.
11. Li, Stampfer, Cohen, & Koedinger (2013). General and efficient cognitive model discovery using a simulated student. In *Proc of Cognitive Science*. (pp. 894-9) Austin, TX.
12. Martin, Mitrovic, Mathan & Koedinger (2011). Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-Adapted Int, 21*(3), 249-283.
13. Stamper, J.C. & Koedinger, K.R. (2011). Human-machine student model discovery and improvement using data. In *Proc of AI in Ed*, pp. 353-360. Berlin: Springer.
14. Koedinger, Baker, Cunningham, Skogsholm, Leber, Stamper, (2010). A Data Repository for the EDM community: The PSLC DataShop. In *Hdbk of Ed Data Min*. Boca Rat: CRC.
15. Brown, J. S., & VanLehn, K. (1980). Repair theory. *Cognitive Science, 4,* 379-426.
16. Matsuda, Lee, Cohen, & Koedinger, (2009). A computational model of how learner errors arise from weak prior knowledge. In *Proc of Cognitive Science*. pp. 1288-1293.
17. Booth, J.L., & Koedinger, K.R. (2008). Key misconceptions in algebraic problem solving. In Love, McRae & Sloutsky (Eds.), *Proc of Cognitive Science,* pp. 571-576.
18. VanLehn, K. (1982). Bugs are not enough. *Journal of Mathematical Behavior, 3(2)*, 3-71.
19. Matz, M. (1980). Towards a process model for high school algebra errors. In Sleeman & Brown (Eds.), *Intelligent Tutoring Systems* (pp. 25-50). Orlando, FL: Academic Press.
20. Li, N., Cohen, W. W., & Koedinger, K. R. (2012). Problem Order Implications for Learning Transfer. In *Proceedings of Intelligent Tutoring Systems*, 185–194.
21. Aleven, V., McLaren, B., Sewall, J., & Koedinger, K. R. (2009). Example-tracing tutors: A new paradigm for intelligent tutoring systems. *Int J of AI in Education, 19,* 105-154.
22. MacLellan, Koedinger & Matsuda (2014). Authoring Tutors with SimStudent: An Evaluation of Efficiency and Model Quality. *Proc of Intelligent Tutoring Systems,* 551-560.
23. Biswas, G., Schwartz, D., Leelawong, K., Vye, N. (2005). Learning by Teaching: A New Agent Paradigm for Educational Software. *Applied Artificial Intelligence, 19*, 363-392.
24. Gomoll, K., (1990). Some techniques for observing users. In Laurel B. (ed.), *The Art of Human-Computer Interface Design*, Addison-Wesley, Reading, MA, pp. 85-90.
25. Matsuda, Yarzebinski, Keiser, Raizada, William, Stylianides & Koedinger (2013). Cognitive anatomy of tutor learning. *J of Ed Psy, 105(4)*, 1152-1163.
26. Matsuda, Cohen, Koedinger, Keiser, Raizada, Yarzebinski, Watson, Stylianides (2012). Studying the effect of tutor learning using a teachable agent. In *Proc of DIGITEL*, 25-32.
27. Matsuda, Griger, Barbalios, Stylianides, Cohen, & Koedinger (2014). Investigating the effect of meta-cognitive scaffolding for learning by teaching. In *Proc of ITS,* 104-113.
28. Matsuda, Keiser, Raizada, Tu, Stylianides, Cohen, Koedinger (2010). Learning by Teaching SimStudent: In *Proceedings of Intelligent Tutoring Systems,* 317-326.
29. Ritter, Anderson, Koedinger, & Corbett (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin &Review, 14(2)*, 249-255.
30. Pavlik & Anderson (2008). Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied, 14*, 101-117.
31. Koedinger, Corbett, Perfetti (2012). The KLI framework. *Cog Sci, 36 (5),* 757-798.
32. Koedinger, Booth, Klahr (2013). Instructional complexity. *Science, 342*, 935-937.