

Towards Deeper Integration of Intelligent Tutoring Systems: One-way Student Model Sharing between GIFT and CTAT

Vincent Alevan¹, Jonathan Sewall¹, Juan Miguel Andres², Octav Popescu¹, Robert Sottolare³, Rodney Long³,
Ryan Baker²
Carnegie Mellon University¹, University of Pennsylvania², US Army Research Laboratory³

ABSTRACT

There are strong potential benefits to be had by integrating intelligent tutoring systems (ITSs) with each other, but few instances of successful integration are known in the literature. Given the central role that student models play in ITS architectures, and given that different ITS platforms tend to have their own student models, a key challenge is exchanging and mapping student models. Our project focuses on integrating GIFT and CTAT, both widely used ITS authoring and delivery environments. The specific goal of our project is to create, as a proof-of-concept, adaptive capabilities for an edX MOOC, using GIFT and CTAT within edX. We have created an initial version of this integration, in which GIFT supports the outer loop and simple interactive activities, while CTAT (under GIFT outer loop control) supports more complex problem-solving activities. As a student works with a CTAT tutor, whenever CTAT updates its own student model, the updates are sent also to GIFT, so that GIFT's outer loop can take advantage of a complete and up-to-date view of the student's knowledge as it selects appropriate remedial activities. The main student model elements are mapped in a 1:1 manner between CTAT's and GIFT's student model, even if the general problem of creating such mappings is hard. This one-way student model sharing is achieved with an extended use of the LTI standard. The main contribution is a proof-of-concept demonstration of ITS integration, limited in a number of ways (e.g., for the time being, the student model is communicated in one direction only), but exciting in its possibilities for joining the ITS functionality of different ITS platforms.

INTRODUCTION

Intelligent tutoring systems can be authored, increasingly, with efficient and easy-to-learn authoring tools, such as the Generalized Intelligent Framework for Tutoring (GIFT), (Brawner, 2015; Goldberg & Hoffman, 2015; Goldberg, Hoffman, & Tarr, 2015; Sottolare, 2012), the Cognitive Tutor Authoring Tools (CTAT) (e.g., Alevan et al., 2016), and others (Cai, Graesser, & Hu, 2015; Mitrovic et al., 2009; Razzaq et al., 2009). Although different ITSs tend to share a core of tutoring behaviors (VanLehn, 2006; 2016), they often have complementary strengths and focuses. As noted in Baker (2016), the challenge of developing a single form of adaptivity is often sufficiently high that some ITSs focus on just one form of adaptivity apiece; other ITS include multiple forms of adaptivity, but not always the same forms (Alevan, McLaughlin, Glenn, & Koedinger, 2017). GIFT, for example, offers an adaptive outer loop that covers a wider range of pedagogy than CTAT; it also offers tools for easy authoring of questions to test recall of concepts and APIs for integrating sensors and training applications. CTAT, on the other hand, offers possibilities for crafting highly adaptive step loops, responsive to students' strategies and errors, and offers an adaptive outer loop that supports cognitive mastery.

A promising approach to building effective, innovative, adaptive learning technologies would therefore be to bring together ITS systems to leverage the strengths found in each (integrating GIFT and CTAT, for instance). Potential advantages could be, speculatively, that more adaptive tutoring systems could be authored more easily, that systems with more sophisticated pedagogical approaches could be authored, and that the choice of pedagogy could be better matched to the instructional goals.

ITS interoperability has long been viewed as desirable (Brusilovsky, 1995), but has proven elusive, now forming the basis of one of the BLAP prizes in Learning Analytics (Baker, under review). A small number of interesting instances exist (Alevén & Rosé, 2004; Koedinger, Suthers, & Forbus, 1998), but the main ITS platforms are still separate. Given the central role that student models play in ITS architectures (Bull & Kay, 2016), sharing or mapping student models should be a focal point in the integration of ITSs: if tools could share their student models, then tutors created with these tools should be able to make better adaptive decisions from the richer, more complete information available (Alevén et al., 2017; Woolf, 2009), and make better decisions sooner when a student starts in a new platform (Sosnovsky et al., 2007). However, the student models used in different ITS platforms tend to differ in the types of student characteristics they assess, the ontologies that they use to represent these characteristics, their methods for updating the model, and the data they require. The many differences make integration of student models, at least as a general problem, quite a daunting prospect. In the current project, we explore a small but interesting instance of this challenge.

Specifically, our project focuses on creating a MOOC, within the edX platform, that is adaptive to students' knowledge growth in ways that edX courses are not. We do so by integrating GIFT and CTAT with each other, and embedding them together within edX, so that GIFT's and CTAT's combined adaptive tutoring functionality is available in the MOOC. We carry out this integration and demonstrate its feasibility in the context of the edX MOOC "Big Data and Education" (BDEMOOC), created and taught by the last author. In the current paper, we focus on the GIFT/CTAT integration, as we reported on the integration into edX in prior publications (Alevén, Baker, et al., 2017). Also in prior work, we made it possible for GIFT to invoke CTAT tutors in a manner adaptive to a student's knowledge growth as assessed by GIFT (Alevén et al., 2018).

We now extend this work so the CTAT tutor can send its up-to-date student model to GIFT. This model captures a student's mastery of knowledge components targeted in the instruction (Alevén et al., 2016; Alevén & Koedinger, 2013). We enabled GIFT to map CTAT's student model onto its own student model, which (among other things) captures similar knowledge components. This way, GIFT's outer loop has up-to-date information about a student's skill level on which to base the adaptive selection of learning activities. Although one could envision other ways of combining GIFT and CTAT, this particular way plays to the strengths of both tools, as discussed in more detail below.

In the current paper, we address the following questions: What leverage is there in enabling CTAT to communicate its student model to GIFT? What adaptive tutoring behaviors might now be easier to author than before? How can the two student models be mapped to each other? How can their integration be accomplished technically? What are the limitations of this means of integrating student models, and how might they be addressed in future work?

ADVANTAGES OF INTEGRATION: TARGETED TUTORING BEHAVIORS

In this section, we describe the student experience that we implemented within the BDEMOOC as a proof-of-concept demonstration of the new GIFT/CTAT integration. One could envision more complex forms of adaptive instruction based on this integration, but we wanted to start simple. Specifically, we added a new pattern of adaptive instruction that includes examples and learn-by-doing activities for week 1 of the 8-week BDEMOOC, implemented as a short GIFT course embedded within the overall edX course. The course used as its outer loop GIFT'S Engine For Management of Adaptive Pedagogy (EMAP), which implements Merrill's component display theory (CDT) quadrants (Goldberg et al., 2015). Generally speaking, in EMAP a student first enters the optional Rules quadrant to receive direct explanation of the concepts (e.g., in a video lecture), then proceeds to the Examples quadrant to see instances of application of the concepts. Next, in the Recall quadrant, the student answers questions associated with individual concepts. The optional Practice quadrant specifies activities by which the student can learn or demonstrate skill with applying the concepts. The Remediation "quadrant" (not explicit in the original CDT) provides concept-specific materials for review if the student's Recall or Practice performance does not meet expectations. An author defines the quadrants by configuring GIFT's Adaptive Courseflow object with the concepts and materials to be presented.

In our edX MOOC, week 1 includes 6 short GIFT courses, each with an Adaptive Courseflow object having a video lecture (Rules content) explaining concepts and techniques in educational data mining (the subject of the course) and providing slides with examples and recall questions on these concepts. We split the material into individual GIFT courses in order to provide questions after each lecture, instead of after all 6 lectures, and to let students use edX to navigate to individual lectures at will.

New in the 2019 edition of our MOOC is a 7th GIFT course at the end of week 1, with an Adaptive Courseflow object configured as shown in Figure 1. This course covers concepts and skills related to Decision Trees and k NN (for k -Nearest Neighbor). The course's principal purpose is to permit GIFT to provide adaptive practice with a CTAT tutor, and then depending on the success of the student's learning in the CTAT tutor, to present opportunities for remedial studying of examples, and remedial additional practice with a second tutor. To this end, CTAT communicates its student model to GIFT, to summarize the state of student learning resulting from the tutor activity.

Although an Adaptive Courseflow typically starts in the Rules quadrant, the new course's Adaptive Courseflow object omits explicit Rule content, to avoid repeating material from the video lectures earlier in the week. Its Examples quadrant (top left in Figure 1) provides detailed Powerpoint slides illustrating the application of the two algorithms (Decision Trees and k NN). The Recall quadrant (top right in Figure 1) has click-through screens instead of questions, again to avoid redundancy with the questions asked earlier. The Practice quadrant (bottom right in Figure 1) offers two CTAT tutors as practice applications: the primary tutor, presented first, covers both algorithms. If the student's skill level after exiting the primary tutor is still Novice on either algorithm, then the remediation quadrant (bottom left, Figure 1) lets the student review the example slides for just that algorithm. After remediation, the student will do a secondary tutor that covers both algorithms. We would have preferred to have two separate tutors for remedial practice, one for each algorithm, but GIFT disqualifies from remedial use any Practice application that fails to cover *all* Practice quadrant concepts, even those already mastered. Even so, the student still receives adaptive content due to the integration of CTAT tutors in the Remediation Quadrant.

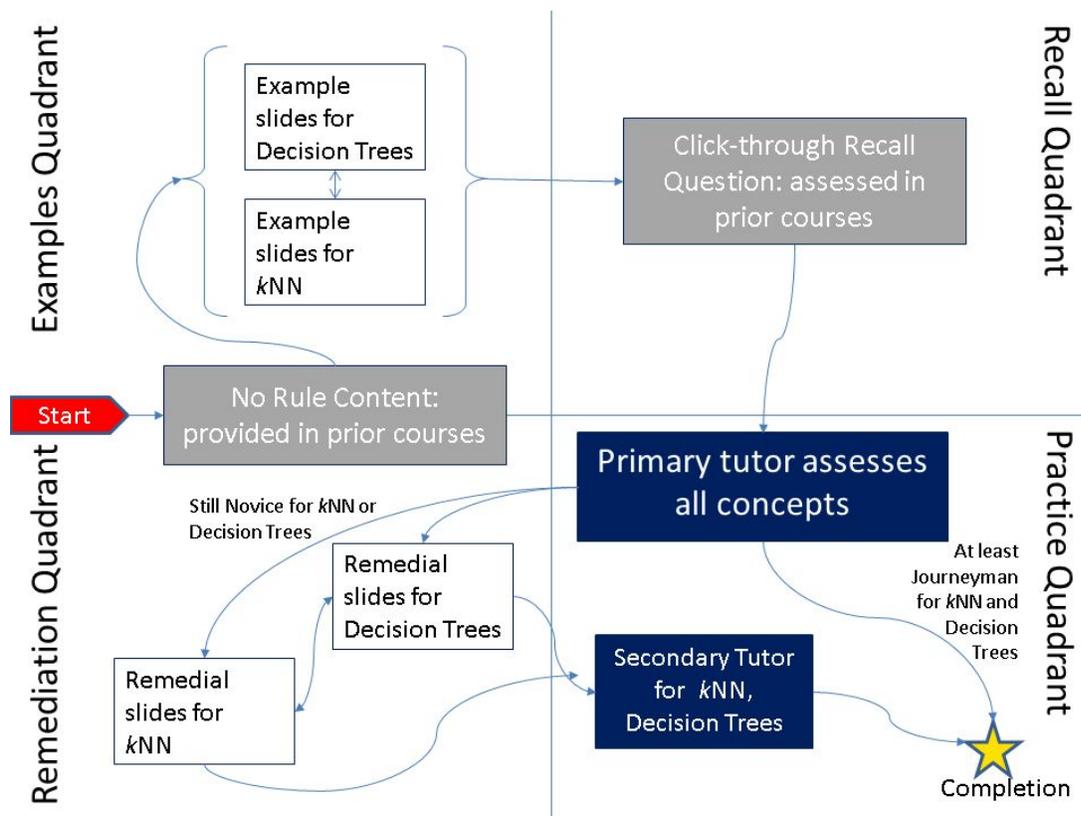


Figure 1: Week 1 Adaptive Courseflow object with CTAT tutors as Practice applications. The GIFT concepts and corresponding CTAT skills refer to the data mining algorithms Decision Trees and k NN (for k -Nearest Neighbor).

Even if this pattern of adaptive instruction - complex tutored problem-solving adaptively combined with remedial examples and remedial tutored problem solving - is simple, it extends what GIFT and CTAT easily do separately, and capitalizes on strengths of each tool. The complex tutored problem-solving activities authored with CTAT could not easily have been authored in GIFT, as GIFT does not support a non-programmer approach to creating user interfaces or adaptive inner loops, as CTAT does (Alevan et al., 2016). On the other hand, the adaptive interleaving of problem solving and declarative instruction, based on Merrill's quadrants, could not have been authored as easily in CTAT, because its standard adaptive outer loop option, namely, cognitive mastery based on Bayesian Knowledge Tracing (Corbett & Anderson, 1995), is geared towards problem solving only, without declarative instruction interleaved. CTAT does not represent the quadrant structure (an author would have to write a custom outer loop), and is not geared towards embedding external learning objects such as Powerpoint slides. We note that our proof-of-concept pattern of instruction realizes (in a new, more adaptive way) one of the Cognitive Tutor principles (Anderson, Corbett, Koedinger, & Pelletier, 1995; Koedinger & Corbett, 2006), namely, to "Provide instruction in the problem-solving context." In previous Cognitive Tutors, the declarative instruction was provided in the classroom (Koedinger, Anderson, Hadley, & Mark, 1997), or was embedded in the tutor as static text pages, though without adaptive sequencing.

STUDENT MODEL MAPPING

We saw two principal questions with respect to the semantics of GIFT's and CTAT's student models: How do key elements of CTAT's student model (mastery probabilities for KCs) correspond to the richer set of categories for representing knowledge in GIFT's student model? Second, how can CTAT's KC mastery probabilities be mapped onto the three mastery levels used in GIFT's student model (Novice, Journeyman, Expert) for use with adaptive decisions, in a manner that respects the semantics of these categories, as intended by the GIFT designers?

First, a brief look at what these student models contain. GIFT (cf. the EMAP explanation at [https://giftutoring.org/projects/gift/wiki/Engine_For_Management_of_Adaptive_Pedagogy_\(eMAP\)_2018-1](https://giftutoring.org/projects/gift/wiki/Engine_For_Management_of_Adaptive_Pedagogy_(eMAP)_2018-1)) decomposes expertise into concepts but also recognizes affective state and has a notion of behavior state. Concepts may be hierarchical, where a single overarching concept decomposes into a tree of finer-grained concepts, but this multi-level modeling is not required: concepts may instead be enumerated in a simple single-level list. Assessment of a student's mastery of each concept is maintained with respect to 1 or 2 measures, **Cognitive Knowledge** and **Cognitive Skill**. The latter is defined as the "ability to execute." For each concept GIFT maintains separate assessments of Cognitive Knowledge and Cognitive Skill as one of Novice, Journeyman or Expert. The assessment drives adaptive decisions within GIFT's Adaptive Courseflow object.

CTAT's student model is a set of independent knowledge components (KCs), also called skills. For each, CTAT records a probability that the student has mastered it, based on their prior performance; a single threshold (0.95 by default) indicates mastery. This technique for modeling students' knowledge, originally developed in Cognitive Tutors for personalized problem selection, tries to model especially procedural knowledge. Knowledge components are fine-grained: their scope can be refined empirically by observing error rates on questions thought to require the same knowledge (Aleven & Koedinger, 2013; Anderson et al., 1995).

For our proof-of-concept system, as an initial position we simply make a 1:1 correspondence between CTAT's knowledge components and the Cognitive Skill assessment of the lowest-level GIFT concepts (that is, the leaf concepts if the GIFT course uses hierarchical concept modeling). Both seem meant to capture procedural knowledge. To map CTAT probabilities onto GIFT's Novice-Journeyman-Expert levels, we let a GIFT author set probability ranges for the 3 levels in the GIFT Authoring Tool. We are still experimenting with the actual ranges to use for Novice, Journeyman and Expert, as it is hard to find a principled basis for this choice. We have considered equating the expertise level needed for promotion in the GIFT course with CTAT's mastery threshold, and we have set GIFT's Expert level provisionally to the 0.95 probability of mastery threshold in CTAT. But, so far, we have set the Journeyman level, again provisionally, at 0.75 probability. This initial approach may be simplistic, but it permits GIFT to use its full adaptive decision-making capabilities in Adaptive Courseflow objects that include CTAT LTI tools as practice applications. Our discussion below explores the limitations of our initial integration. Our recommendations suggest straightforward changes to GIFT that would permit finer-grained adaptive decisions.

TECHNICAL INTEGRATION

In this section, we describe how we implemented the one-way student model communication (from CTAT to GIFT), using the LTI interoperability standard.

GIFT accommodates external learning activities via two different mechanisms. Heretofore, most integrations have required custom Java-language gateway programs that conform to an interface specified in the Domain Knowledge File ([Domain Knowledge File](#)). The use of Java on the client makes these programs inconvenient to deploy over the World Wide Web, however. Therefore, we decided to use GIFT's second mechanism for integrating external activities, namely, its implementation of the Learning Tools Interoperability (LTI) Tool Consumer interface. In prior work on our project (Aleven et al., 2018), ARL enabled GIFT to accommodate learning activities that adhere to the LTI v1.1.1 standard (IMS 2012). Figure 2 illustrates our use of this integration, where GIFT itself is an LTI Tool Provider to edX, due to yet earlier work on our project (Aleven et al. 2017).

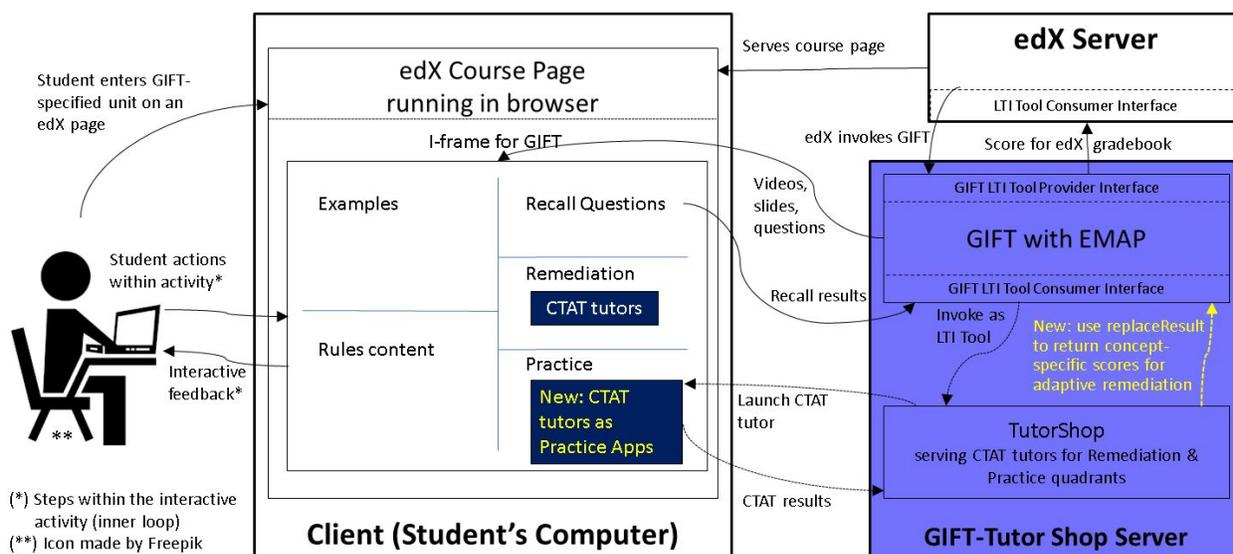


Figure 2: Architecture and control flow in the BDE MOOC week 1 course, with new features shown in yellow. GIFT is invoked by edX as an LTI Tool Provider and in turn launches CTAT tutors as Practice applications whose individual skill recalculations update concept-specific assessment levels in GIFT.

In our current work, CTAT tutors (introduced into the Practice quadrant) update GIFT's assessment of cognitive skill per concept. We make this update dependent on CTAT's calculation of probability of mastery of a corresponding knowledge component. To communicate the value, we use the LTI v1.1.1 specification's **replaceResult** request, by which a tool (here, CTAT), can return a single numeric score, with a label, to the tool consumer (GIFT). This single-score limitation presented a problem, since the GIFT course and the CTAT tutor generally track several concepts and knowledge components, respectively, each with different values. But we noted that the LTI specification permits the tool to issue the **replaceResult** request more than once, and we found that our off-the-shelf library implementations of the LTI interface permitted us to vary the labels on these requests. So, to permit CTAT to return multiple values to GIFT, we make special use of the current standard by allowing the tool to send many **replaceResult** requests, each with a concept-specific label and value.

In our integration, CTAT reports changes in skill mastery estimates while the student is working on a tutor, as soon as they happen, not just upon exiting the tutor. Part of the rationale is that the session could end abruptly at any time: the tutor might not have a chance to send final scores. A second reason is that the same student might be logged into GIFT in multiple sessions concurrently--even perhaps inadvertently. A common usage pattern on the World Wide Web is for a user to open a new browser tab to attend to some matter while in the midst of a task on another tab; after some time passes and more open tabs accumulate, it could be easy for a user to forget what tasks were in progress and so begin anew in a new tab on a site already active on another tab. Step-by-step reporting of student model updates helps student model values remain up-to-date for access from concurrent sessions.

After the time of our implementation (and the initial submission deadline for this paper), the LTI Assignment and Grade Services Specification v2.0 (<https://www.imsglobal.org/spec/lti-ags/v2p0/>) was published. This standard offers richer reporting capabilities. During our work it was neither finalized nor freely available; now we look forward to the development of support libraries to promote its adoption. Our v1.1.1 workaround was easy to implement with existing freely-available libraries, and it let us prototype a useful extension to GIFT's capabilities.

DISCUSSION

In this paper, we present a new GIFT/CTAT integration with one-way student model sharing between GIFT and CTAT, a novel feature, relative to our own prior work and to prior work in the field. Our approach is to exchange a student model in one direction and to map student model values with 1:1 correspondence between the main student model elements. We present a proof-of-concept demonstration of this integration within one of the units of the summer 2019 edition of the edX course "Big Data and Education." In this unit, our integration supports new adaptive interleaving of complex problem solving and declarative instruction.

Our proof-of-concept represents a relatively simple special case of a more complex problem. First, it is limited in its curricular scope: it covers only a portion of one week's worth of instruction within the BDEMOOC, captured in one instance of GIFT's Adaptive Courseflow object. Nonetheless, similar extensions could be repeated in other course chapters.

Another limitation may be that in our project, we have pretty much a best case situation (albeit one that may not be uncommon) in which a single content development team works with all the different tools being integrated. This situation no doubt makes it easier to create student models whose elements map 1:1 than it would be otherwise. Things may be very different when integrating, post-hoc, two ITSs from different authors. Under those circumstances, the student models might not be as easy to align, and some form of ontological translation may be necessary. Perhaps even greater benefits from integration accrue when integrating existing systems; more content may be involved, for example. Then again, we do not know how likely that kind of integration scenario is.

The integration of the two student models may incur a certain level of what we might call semantic friction. Do thresholds on CTAT KC probabilities capture what the GIFT designers meant by the categories of Novice, Journeyman, and Expert? Some degree of semantic mismatch might be inevitable and perhaps unresolvable. We would like to think, however, that the current thresholds (set at .75 for Journeyman and .95 for Expert) align reasonably well with the intent of the GIFT designers, although

some further scrutiny of this issue might be in order, informed perhaps by prior work in the Knowledge/Skills/Abilities (KSA) doctrine, on which the GIFT student model draws.

A further limitation is that we present no data to support the point that the current integration - although supported by instructional design principles - is actually benefiting students. The current work should be viewed as exploratory, focused mainly on technical issues. We are collecting data in this summer's run of the BDEMOOC, and plan to discover the different pathways that students take through the course materials. We do not expect however that the new adaptive pattern just in week 1 of the BDEMOOC will lead to measurably different outcomes (e.g., learning gains, retention rates). It may be better to wait with a more rigorous evaluation of student learning until more content has been moved into this (and similar) adaptive patterns.

A final limitation of the current integration is that it supports one-way communication of the student model only, namely, from CTAT to GIFT. Full integration would require two-way communication, so CTAT can be cognizant of information about the student inferred from performance in other GIFT activities, an interesting avenue for future work, as discussed below.

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

In this paper, we solved technical issues regarding the one-way sharing and 1:1 mapping of student models, as one way in which one might integrate GIFT and CTAT. As a proof-of-concept, we demonstrated this integration by adaptive interleaving of problem-solving practice with remedial example studying and problem solving. These adaptive tutoring behaviors would be harder to author in either GIFT or CTAT alone. The main contribution of the work is that it demonstrates benefits of simple one-way student model sharing between ITS platforms, one of very few demonstrations in the literature of ITS integration. It demonstrates, as well, that integration of ITSs, generally viewed as both highly desirable and highly challenging, does not always need to be exceedingly difficult.

Our plans for future work are as follows: After running the edX course (planned for late spring and early summer 2019), we will analyze the course data to get a sense for the functioning of the new adaptive mechanisms in the course that are made possible by the newly-implemented student model sharing. We will check how students' paths through the course changed and whether they became more effective, and to learn how well participation held up across adaptive transitions.

If the new v2.0 LTI standard with richer reporting capabilities gains broad adoption, we recommend its implementation in GIFT and in CTAT. Independent of that, we recommend two near-term enhancements and suggest some longer-term ideas. First, for the near term, it would be good to extend the GIFT Authoring Tool to permit concept-specific ranges for translating LTI numeric scores into GIFT's Novice-Journeyman-Expert expertise levels. That is, instead of a single slider to set a single tuple of Novice-Journeyman-Expert ranges for *all* concepts, it would be useful to be able to set these ranges *per* concept. Then authors would have flexibility to establish concept-specific performance expectations. Perhaps a way could be found that such flexibility would help reduce some of the semantic friction discussed above.

A second near-term recommendation for GIFT would be to enhance the Practice quadrant's content selection algorithm to account for cognitive skills already mastered. As mentioned above, when a student re-enters the Practice quadrant after remediation, GIFT currently considers only those applications

associated with *all* concepts covered by the Practice phase. The student might avoid repetitive work if GIFT were able to choose a Practice application that covered only those concepts for which the student has not yet met expectations.

Further out, we note that GIFT's student model also includes Affective State. Recent work in CTAT has made it easy to integrate detectors that infer, from the student-tutor transaction stream, variables regarding student affect, unproductive persistence, disengaged behaviors such as gaming the system, and so forth (Holstein et al., 2018). It would be valuable, in the future, to investigate how these additional variables could be shared between CTAT and GIFT in a generalizable manner.

Another key direction would be to communicate the student model from GIFT back to CTAT, so the student model would be shared in bi-directional fashion, and the tutoring behavior in a CTAT tutor could adapt to what the student learned (or did not learn) in GIFT activities. Straightforward extensions of the LTI implementations of GIFT and CTAT (see `readResult`, below) would permit CTAT to receive the GIFT student model. If we assume, as in our current one-way integration, that GIFT skills would be mapped 1:1 to CTAT KCs, a key issue would be: How should we translate GIFT skill levels (novice, journeyman, expert) into CTAT KC probabilities? It seems inevitable that in the mapping we would lose "resolution." CTAT has greater precision in its student model values (which represent probabilities) than GIFT, a downside of GIFT's choice to distinguish only three mastery levels in its student model. A possible way around this loss of resolution might be for the two systems to each maintain their own student model, and to share student model *updates*, rather than the student model itself - so each system could update its student model based on events that happened in the other system. While that solution might maintain resolution, and perhaps avoid semantic friction of the kind described above, it would be more difficult to implement; we did not explore it.

In implementing two-way student model sharing, we may need to account for the possibility that a student may at times be working on two different activities simultaneously, as described above. If simultaneous activities share targeted knowledge components, then the updated student models sent from the Tool (e.g., CTAT) to the Consumer (GIFT) may clobber each other. This issue can be avoided by having the Tool first query the Consumer's student model for its current value(s) before computing an updated value and sending it back to the Consumer. Similarly, the Tool should query the Consumer's student model before and, based on the result of the query, update its own student model before using it for its own adaptive pedagogical decisions. CTAT would need some straightforward modifications to make these queries. In short, we see a clear path to two-way student model sharing between GIFT and CTAT, assuming that the 1:1 mapping of student model elements will remain appropriate.

Finally, the most exciting future work is exploring what new student experiences can be authored with the new GIFT/CTAT integration. Some attractive scenarios might involve CTAT's cognitive mastery outer loop, which has been very well-studied in the EDM literature, and proven to be effective (Corbett, McLaughlin, & Scarpinato, 2000), or more flexible interleaving of problem solving and declarative instruction. These scenarios might require additional flexibility in, and authoring capabilities for, how GIFT adaptively traverses the quadrants in the Adaptive Courseflow object, and may benefit from ways of modeling links between procedural and conceptual knowledge, already represented in GIFT's student model, but not in CTAT's.

REFERENCES

- Aleven, V., Baker, R., Blomberg, N., Andres, J.M., Sewall, J., Wang, Y., Popescu, O. (2017). Integrating MOOCs and Intelligent Tutoring Systems: edX, GIFT, and CTAT. In *Proceedings of the Fifth Annual GIFT Users Symposium*, May 2017.
- Aleven, V., & Koedinger, K. R. (2013). Knowledge component approaches to learner modeling. In R. Sottolare, A. Graesser, X. Hu, & H. Holden (Eds.), *Design recommendations for adaptive intelligent tutoring systems* (Vol. I, Learner Modeling, pp. 165-182). Orlando, FL: US Army Research Laboratory.
- Aleven, V., McLaren, B. M., Sewall, J., van Velsen, M., Popescu, O., Demi, S., Ringenberg, M., & Koedinger, K. R. (2016). Example-tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education*, 26(1), 224-269. doi:10.1007/s40593-015-0088-2.
- Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2017). Instruction based on adaptive learning technologies. In R. E. Mayer & P. Alexander (Eds.), *Handbook of research on learning and instruction* (2nd ed., pp. 522-560). New York: Routledge.
- Aleven, V., & Rosé, C. P. (2004). Towards easier creation of tutorial dialogue systems: Integration of authoring environments for tutoring and dialogue systems. In J. Mostow & P. Tedesco (Eds.), *Papers from the Workshop on Dialog-Based Intelligent Tutoring Systems*, held in conjunction with ITS 2004 (pp. 1-7). Maceió, Brazil.
- Aleven, V., Sewall, J., Popescu, O., Sottolare, R., Long, R., & Baker, R. (2018). Towards adapting to learners at scale: Integrating MOOC and intelligent tutoring frameworks. In *Proceedings of the Fifth Conference on Learning @ Scale*, June 2018..
- Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, 4(2), 167-207.
- Baker, R.S. (2016) Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence and Education*, 26(2), 600-614.
- Baker, R.S. (under review) Some challenges for the next 18 years of learning analytics. Manuscript under review.
- Brusilovsky, P. (1995). Intelligent learning environments for programming: The case for integration and adaptation. In J. Greer (Ed.), *Proceedings of the 7th World Conference on Artificial Intelligence in Education (AIED '95)*, (Vol. 95, pp. 1-8). Washington, DC: Association for the Advancement of Computing in Education.
- Bull, S., & Kay, J. (2016). SMILI[©]: A framework for interfaces to learning data in open learner models, learning analytics and related fields. *International Journal of Artificial Intelligence in Education*, 26(1), 293-331. doi:10.1007/s40593-015-0090-
- Cai, Z., Graesser, A. C., & Hu, X. (2015). ASAT: AutoTutor script authoring tool. In R. Sottolare, A. Graesser, X. Hu, & K. Brawner (Eds.), *Design recommendations for adaptive intelligent tutoring systems* (Vol. III, Authoring Tools, Chap. 17, pp. 199-210). Orlando, FL: US Army Research Laboratory.
- Corbett, A. T., & Anderson, J. R. (1995). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4), 253-278.
- Corbett, A., McLaughlin, M., & Scarpinato, K. C. (2000). Modeling student knowledge: Cognitive Tutors in high school and college. *User Modeling and User-Adapted Interaction*, 10, 81-108.
- Holstein, K., Yu, Z., Sewall, J., Popescu, O., McLaren, B. M., & Aleven, V. (2018). Opening up an Intelligent Tutoring System development environment for extensible student modeling. In C. P. Rosé, R. Martínez-Maldonado, H. U. Hoppe, R. Luckin, M. Mavrikis, K. Porayska-Pomsta, B. McLaren, & B. du Boulay (Eds.), *Proceedings, 19th International Conference on Artificial Intelligence in Education, AIED 2018* (Part 1, pp. 169–183). Cham, Switzerland: Springer. doi: 10.1007/978-3-319-93843-1_13

- IMS Global Learning Tools Interoperability™ Implementation Guide Final Version 1.1.1 (2012), at <https://www.imsglobal.org/specs/ltiv1p1p1/implementation-guide>.
- Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8(1), 30-43.
- Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning sciences to the classroom. In R. K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61-78). New York: Cambridge University Press.
- Koedinger, K. R., Suthers, D. D., & Forbus, K. D. (1998). Component-based construction of a science learning space. In B. P. Goettl, H. M. Half, C. L. Redfield, & V. J. Shute (Eds.), *Intelligent Tutoring Systems, Fourth International Conference, ITS '98* (pp. 166-175). Lecture Notes in Computer Science 1452. Berlin: Springer Verlag.
- Sottolare, R., Long, R. and Goldberg, B. (2017). Enhancing the Experience Application Program Interface (xAPI) to improve domain competency modeling for adaptive instruction. *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, April 2017, 265-268.
- Goldberg, B., Hoffman, M. and Tarr, R. (2015). Authoring Instructional Management Logic in GIFT Using the Engine for Management of Adaptive Pedagogy (EMAP). *Design Recommendations for Intelligent Tutoring Systems: Authoring Tools and Expert Modeling Techniques*, 319-334.
- Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., & McGuigan, N. (2009). ASPIRE: an authoring system and deployment environment for constraint-based tutors. *International Journal of Artificial Intelligence in Education*, 19(2), 155-188.
- Razzaq, L., Patvarczki, J., Almeida, S. F., Vartak, M., Feng, M., Heffernan, N. T., & Koedinger, K. R. (2009). The Assistent Builder: Supporting the life cycle of tutoring system content creation. *IEEE Transactions on Learning Technologies*, 2(2), 157-166.
- Sottolare, R. and Gilbert, S. (2011). Considerations for tutoring, cognitive modeling, authoring and interaction design in serious games. Authoring Simulation and Game-based Intelligent Tutoring workshop at the Artificial Intelligence in Education Conference (AIED) 2011, Auckland, New Zealand, June 2011.
- Sottolare, R., Brawner, K., Goldberg, B. & Holden, H. (2012). The Generalized Intelligent Framework for Tutoring (GIFT). US Army Research Laboratory.
- Sottolare, R., Goldberg, B., Brawner, K., & Holden, H. (2012). A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In *Proceedings of the Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2012.
- Sottolare, R., Holden, H., Goldberg, B., & Brawner, K. (2013). The Generalized Intelligent Framework for Tutoring (GIFT). In Best, C., Galanis, G., Kerry, J. and Sottolare, R. (Eds.) *Fundamental Issues in Defence Simulation & Training*. Ashgate Publishing.
- VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3), 227-265.
- VanLehn, K. (2016). Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education*, 26(1), 107-112. doi:10.1007/s40593-015-0056-x
- Vygotsky, L.S. (1978). *Mind in Society: The development of higher psychology processes*. Cambridge MA: Harvard University press.

ABOUT THE AUTHORS

***Dr. Vincent Aleven** is a Professor of Human-Computer Interaction at Carnegie Mellon University. His research focuses on adaptive learning technologies, with a grounding in cognitive theory and theories of self-regulated learning. He and his colleagues have created the Cognitive Tutor Authoring Tools (CTAT), which has been used to build a very wide range of intelligent tutoring systems, many of which have been used in real educational settings.*

***Mr. Miggy Andres-Bray** is a 4th year PhD student at the University of Pennsylvania, working on the MOOC Replication Framework.*

***Dr. Ryan S. Baker** is Associate Professor at the University of Pennsylvania's Graduate School of Education and directs the Penn Center for Learning Analytics. He teaches the MOOC Big Data and Education.*

***Mr. Rodney Long** is a Science and Technology Manager at the Army's Soldier Center - Simulation and Training Technology Center in Orlando, Florida and conducts research in adaptive training technologies.*

***Dr. Octav Popescu** is a Senior Research Programmer/Analyst in Carnegie Mellon's Human-Computer Interaction Institute, where he is in charge of TutorShop, the Learning Management System part of the Cognitive Tutor Authoring Tools project.*

***Mr. Jonathan Sewall** is a project director at CMU's Human-Computer Interaction Institute, working on the Cognitive Tutor Authoring Tools (CTAT).*

***Dr. Robert Sottolare** leads adaptive training research within ARL's Learning in Intelligent Tutoring Environments (LITE) Lab and is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT).*