Modeling Training Efficiency and Return on Investment for Adaptive Training

Gregory A. Goodwin[1], Jong W. Kim[1], James Niehaus[3]
ARL HRED[1], Charles River Analytics[2]

ABSTRACT

Adaptive training promises more effective training by tailoring content to each individual. Where non-adaptive training may be just right for one segment of the student population, there will be some students that find it too easy while others find it too difficult. Another, often ignored benefit of adaptive training, is improved training efficiency by minimizing the presentation of unnecessary material to learners. One implication of this is that intelligent, adaptive training should require less time to train a population of learners to a given level of proficiency than non-adaptive training. The gains in efficiency should be a function of several factors including learner characteristics (e.g., aptitude, reading ability, prior knowledge), learning methods employed by the adaptive training system, course content (e.g., difficulty and length, adaptability), and test characteristics (e.g., difficulty, number of items). This paper describes the development of a predictive model for training efficiency based on those factors and how it could be integrated into the Generalized Intelligent Framework for Tutoring (GIFT) architecture. How this model supports return on investment decisions for authors is also discussed.

INTRODUCTION

The Generalized Intelligent Framework for Tutoring (GIFT) is an open-source, modular architecture developed to reduce the cost and skill required for authoring adaptive training and educational systems, to automate instructional delivery and management, and to develop and standardize tools for the evaluation of adaptive training and educational technologies (Sottilare, Brawner, Goldberg, & Holden, 2012a; Sottilare, Goldberg, Brawner, & Holden, 2012b). By separating the components of ITSs, GIFT seeks to reduce development costs by facilitating component reuse.

Meta-analyses and reviews support the claim that intelligent tutoring systems (ITS's) improve learning over typical classroom teaching, reading texts, and/or other traditional learning methods. (Dynarsky et al. 2007; Dodds and Fletcher 2004; Fletcher 2003; Graesser et al. 2012; Steenbergen-Hu and Cooper 2013, 2014; VanLehn 2011). In fact, ITS's have been shown to improve learning to levels comparable to Human tutors (VanLehn et al. 2007; VanLehn 2011; Olney et al. 2012).

While improved training effectiveness is certainly a benefit of ITS technology, another important benefit is improved training efficiency over one-size-fits-all training. The goal of an ITS is to identify the gaps in knowledge specific to each learner so that training can focus on filling just those gaps. One of the problems of one-size-fits-all training is that to insure all trainees can comprehend the instruction, it must be developed for trainees with the least experience, knowledge, and aptitude. Though less costly to develop, the material is presented a pace that is slow and that includes content not needed for more experienced, higher aptitude trainees. An ITS would be expected to reduce the time needed to deliver training to such trainees.

The reduction in time to train (i.e., improved acquisition rate) is an important metric because reductions in training time represent cost savings. This is especially true for military trainees who are paid a salary. Reductions in the time needed to train those trainees save salary costs for both trainees and instructors. For large-volume courses, those savings can be substantial.

All of this highlights the need for a means to model and predict training efficiency gains (i.e., time saved) by ITSs generally and GIFT specifically. Having the ability to model time saved by the use of adaptive, intelligent training, as compared to existing or non-adaptive training would have benefits throughout the lifecycle of a course. During the design of new training, the training developer could more easily make decisions about the relative costs and benefits of adding adaptive features. For example, adding extensive remedial training for easy-to-understand concepts may benefit such a small percent of the population of learners, that the net reduction in training time would be too small to make those features worth the cost of development.

During training delivery, actual trainee data could be used to verify and/or improve the model. For example, suppose the model assumed that learners with an aptitude above criteria A would have a 95% probability of understanding concept B without needing any remediation. Learner data could then be used to validate or adjust that probability. This improved model could then be used to better determine the true time-savings of the course when delivered by GIFT.

During training evaluation and refinement, the disparity between predicted and observed training outcomes could be used to refine the training. For example, if a segment of training proves to be more difficult than anticipated for a group of learners, it is possible that the training segment should be refined or redeveloped.

An example of such a model was developed by McDonnell Douglas (1977). This model incorporated predictor variables in four broad categories: course content (e.g., difficulty, length of content), instructional design (e.g., instructional strategies/techniques), test characteristics (e.g., difficulty, number of items), and trainee characteristics (e.g., aptitude, motivation). The model predicted about 39% of the variability in trainee's first-attempt lesson time for self-paced computer-based instruction.

To understand how GIFT might begin to model and predict training time for learners, it is necessary to understand how training is adapted by this system. GIFT is a framework that modularizes the common components of intelligent tutoring systems. These components include a learner module, an instructional or tutor module, a domain module, and a user interface. One of the main motivations for creating this framework was to lower the cost and labor needed to create intelligent tutoring systems by facilitating re-use of components and by simplifying the authoring process (Sottilare et al., 2012a).

GIFT adapts training using the learning effects model. At the first point of this model, learner data informs the learner state in the learner module. The learner module receives assessments from both sensors and the domain module. The learner state is used to determine the appropriate instructional strategy by the tutor module. The instructional strategy is then interpreted by the domain module and used to determine the domain specific learning activities needed to instruct the learner in that domain. The responses of the learner to that activity then update the learner module which starts the cycle over again.

Developing a predictive model in GIFT is not a straightforward process given the ways that training is adapted to each individual. We should note that our goal is not to predict the single path that a trainee would be expected to take through a specific course, but rather the probability associated with all possible paths through the training for a given learner. From that we can determine the range and distribution of times that would be expected for that learner to complete the training. Taking this one step further, we could apply this to a population of learners and predict the range and distribution of the time for that population to complete that training.

The development and integration of a probabilistic model for predicting time to train into the GIFT architecture is currently in the first phase of a three phase plan. In this paper, we describe work being done in the first phase. In this phase we are developing the structure of the Bayesian probabilistic model, identifying factors that are expected to impact training time, and mapping those to a specific course delivered by GIFT.

In the second phase, we will integrate this model into the GIFT framework and develop the user interface to allow for authoring of new predictive models for other GIFT courses. In the third phase of the work, we will empirically validate the predictive model in GIFT and make adjustments to try to improve it.

METHODS

This section describes our method for modeling adaptive training content and predicting distributions of completion times for both individuals and groups using the GIFT excavator trainer as an example. This course is available with public version of GIFT. The training content includes text, images, video demonstrations, and practice opportunities in a virtual simulator making it a good example of the kind of adaptive training that GIFT can deliver.

An Adaptive Training Course in GIFT: Excavator Training

The excavator training course (Army Research Laboratory, 2015) consists of MS PowerPoint slides with instructional information and questions,and a 3D simulation environment for practice. The excavator training starts with a welcoming message and a set of survey questions that obtain the learner characteristics of motivation, grit, and self-regulatory ability. The GIFT tutor, then, presents the concepts of rules to control the excavator (i.e., Excavator, Boom, Bucket, Arm, and Swing), and corresponding examples. Figure 1 shows the overall structure of the excavator training contents.

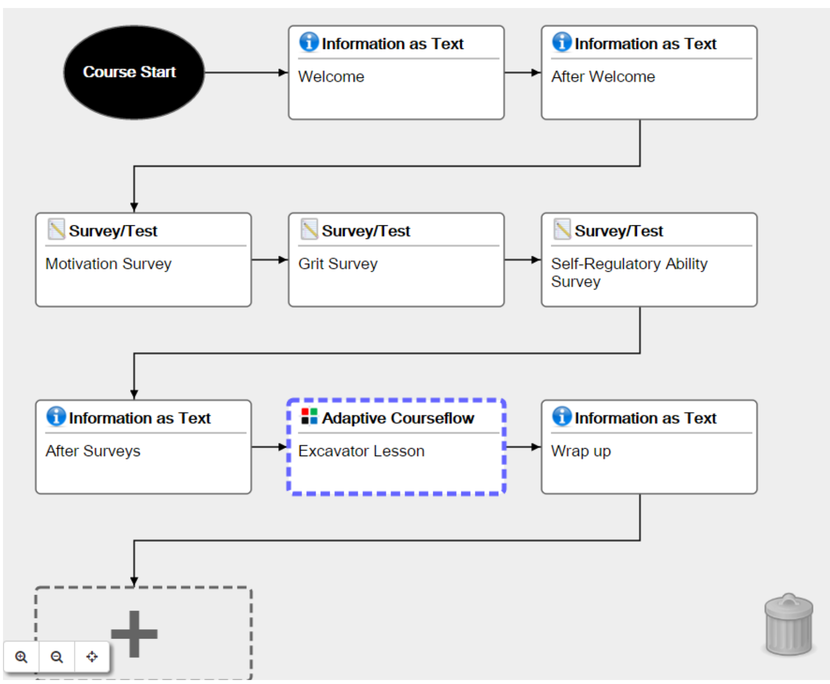


Figure 2. The overall structure of the excavator training course and the adaptive courseflow of the Recall phase in GIFT.

The Adaptive Course Flow object in GIFT (formerly known as the Engine for Management of Adaptive Pedgagoy – EMAP, e.g., Sottilare, 2014; Goldberg, 2015) supports adaptive capabilities for training based on the Component Display Theory (CDT, Merrill, 1983). The CDT supports a general framework of skill training that progresses through two types of learning activities, each with two categories: expository (rules and examples) and inquisitory (recall and practice). According to Merrill, learners should progress through these four quadrants in order starting with rules (presentation of general principles), then to examples (presentation of a specific instance), then to recall (declarative knowledge test of the trainee's comprehension), and finally to practice (opportunity for the trainee to perform the skill). By sorting learning activities into these four quadrants, adaptive training systems like GIFT can apply the CDT to any domain as long as content for that domain is so labeled.

Modeling the Content of Adaptive Training

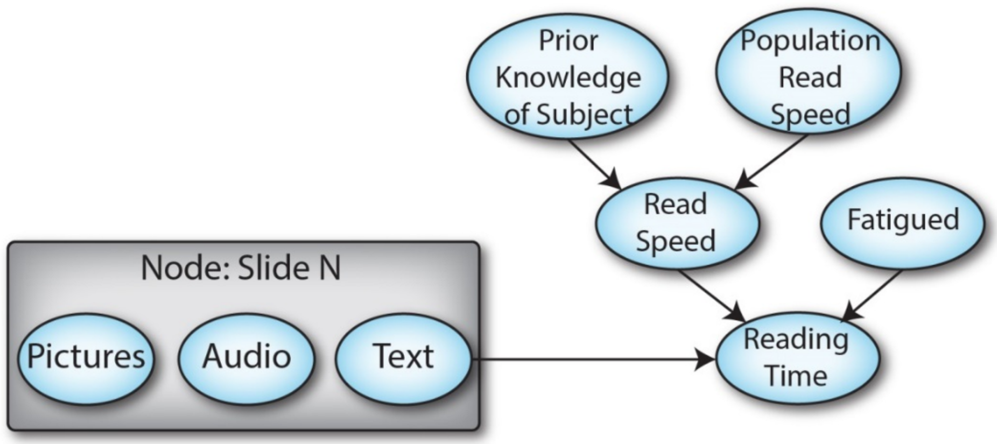
To model the content of adaptive training, we use the *Methodology for Annotated Skill Trees* (MAST) skill trees. The "skeleton" of the skill tree breaks down entire procedures into constituent steps, tasks, and subtasks. Annotations are added to the procedure model. For example, consider completing a set of questions in the excavator tutor that features hints and feedback. This step includes tasks for reading the introduction to the problems, each problem, reading hints, and reviewing feedback. Critical for adaptive training, the MAST procedure model represents not only the base procedure of answering each question correctly without hints, but also the optional hints and feedback steps, variations, and multiple potential paths among questions as chosen by GIFT. Annotations within the MAST skill tree include the following additional information for each step, task, and subtask.

- *Information Elements:* Information or knowledge needed by the trainee to perform the actions required by the skill tree node. These requirements are commonly called the "knowledge map" in ITS literature. In the example of completing a set of GIFT questions, this is the knowledge used to answer the question correctly.
- *Instructional Resources:* Resources to teach the skills needed to perform the actions required by the node. In the question example, these are pointers to additional training content.
- *Skill Priorities*: Ratings of the difficulty and criticality of the skills needed to perform the actions required by the node. These ratings enable training systems to prioritize skills for training and optimize ROI. In the question example, ratings express the criticality of answering the questions correctly to the overall learning goals.
- *Assessments*: Methods of assessing the skills required by the node. These methods enable training systems to determine trainee ability. In the question example, assessment methods include secondary measures of trainee cognitive workload, motivation, or affect that may influence completion time.
- *Decision Making Models*: Computational models of how the procedure steps, tasks, and subtasks are chosen and ordered. These models enable some of the adaptation logic to be represented in the skill tree. In the question example, these models encode the rules for providing hints, providing feedback, and selecting the next question.
- *Completion Time Data:* describes a distribution of completion time based on past data or an estimate of completion time based on type. This data will be used to train the prediction algorithms

We use a probabilistic model to represent the different factors and instructional strategies that impact the completion time of a MAST module, as well as probabilistic inference techniques to determine a distribution of a course completion time. Not only must our model represent relationships between variables and paths in the MAST skill tree, but it must also recognize and model the impact of time as well; many variables can change as the trainee is completing a training module. Building this model consists of two basic steps: developing a model that estimates the completion time for nodes in the MAST skill tree, and temporally linking these models together to enable inference of the entire module completion time.

Figure shows part of an example model for estimating the completion time of a node in a MAST skill tree. This example shows some contributing factors that could be used by PAST Time to estimate the time it takes for a trainee to read the text on the slide. There are also variables that estimate the time to process the pictures and audio on the slide, but that these have been omitted from this example for brevity.

The model includes a Reading Time variable, which represents the time it takes for the user to read the text. The value of this variable is a function of the amount of text on the slide, the speed at which the trainee can read the text (Read Speed), and the current alertness of the trainee (Fatigued). These relationships are probabilistic. For example, if a trainee normally reads at 100 words per minute, there are 100 words in the text, and the trainee is tired, the reading time of the trainee could be distribution uniformly from 1 to 2 minutes. The reading speed of the trainee is also a non-deterministic variable that depends on how much prior knowledge the trainee possesses about statistics about how fast the general population of trainees read.



ut the subject, and

Figure 2: Example model for estimating the time to read Text on a Slide node.

One of the benefits of building a probabilistic model to represent the completion time is that not all of the information in the model is needed to estimate the completion time. For example, if we know how much prior knowledge the user has about the subject (for example, from a pre-instruction questionnaire), we can post that knowledge as *evidence* to the model that would be taken into account when estimating the completion time. If we do not possess that information, we can treat the variable as *latent* and use a prior distribution to represent the state of the variable. For example, we can estimate that only 20% of trainees taking the course have prior knowledge of the subject. These prior distributions can be estimated from the literature review or expert knowledge, and then *learned* over time based on the outcomes of actual testing.

Figure 3 shows a portion of a MAST skill tree for the excavator training GIFT course. This skill tree focuses on the information elements that most heavily influence the completion time. On the left, the overall course on Excavator is the root of the tree structure. Its children are the different topics covered by the course, including the Boom Movement topic. This topic features a number of slides with Pictures, Audio, and Text components. Individual trainees may vary in the amount of time they spend examining the Pictures, whether or not they listen completely to the Audio, and the amount of time taken to read the Text. Trainees may also choose to view optional Slides explaining concepts that they may not be familiar with, adding more time. If trainees fail to demonstrate sufficient knowledge in the quiz or fail to complete the simulation tasks appropriately, they are sent back to the beginning of the Boom Movement topic on Slide 1, adding significant time to completion of the course. This model may be expanded to represent a maximum number of failures before the trainee either moves to a different topic or ends the course.

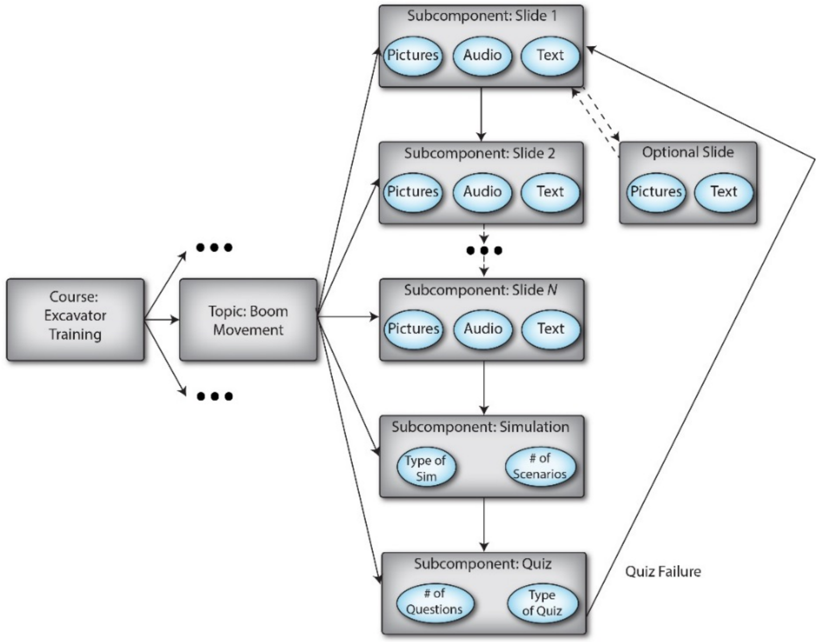


Figure 3: High-level design of a MAST skill tree of a GIFT module with representations of individual instructional elements, branching content, and variables that influence completion times.

After reviewing the Slides, the trainees are asked to practice their skills in Simulation. The MAST model of the simulation can be either a complex procedure describing the steps needed to complete the scenario and optional steps that may or may not contribute to the overall goal. The MAST simulation model may also be simple, representing just the type of simulation and the number of scenarios. To save modeling time and effort, these MAST models are constructed with only the level of detail needed to sufficiently and accurately predict the completion time.

Once these probabilistic models are defined, they can be used to compute a distribution over the course completion time. To generate this distribution, a modeler first provides knowledge about a trainee, group of trainees, or a module as evidence to the model. This could be statistical information obtained from the trainees from a pre-course questionnaire, or data obtained from prior training. Then, given the posted evidence, the user can apply standard probabilistic inference techniques (e.g., variable elimination, importance sampling, Metropolis-Hastings, support computation, most probable explanation (MPE), and particle filtering) to generate a distribution over the completion time of the module. These specific methods are included in the Figaro libraries. Statistical moments of this distribution (e.g., mean and variance) can be easily computed and presented to a module designer.

A significant advantage of combining this probabilistic modeling with the MAST skill tree representation is the capability to ascribe time to individual models, and perform "what if" analysis by adding or removing components. For example, a node for a module requiring detailed arithmetic may take little time in and of itself, but it may be fatiguing, causing significant downstream effects in terms of overall training completion time.

RESULTS

Implementing the Adaptive Training Models

The probabilistic model is being implemented using Charles River Analytics' open source probabilistic programming language, Figaro™ (Pfeffer 2012), to construct and learn probabilistic models of the relationships between these factors. The use of Figaro will greatly simplify the authoring of these models which can be complex and require a high degree of experience by users who may not be experts in probabilistic reasoning.

Figure 4 shows an example Figaro program that creates the completion time model for the node slide shown previously in Figure . Note that the probabilities and values in this program are notional. First, we define the amount of text in the node as 1000 characters. Then, we define two latent variables, one representing the prior knowledge of the trainee and the other representing typical reading speeds. In this case, we specify that a trainee has prior knowledge with 0.2 probability, and the trainee's reading speed is normally distributed around 100 characters a second. Next, we define the actual reading speed of this trainee. In this example, if the trainee has prior knowledge of this subject, we increase their reading speed by a value normally distributed around 50 characters a second. We next represent the fatigued state of the trainee (0.4 probability that the trainee is fatigued). Finally, we define the reading time of this node as the amount of text divided by the reading speed of the trainee; if the trainee is fatigued, however, we assume they can only read at 50% capacity. To use this model to estimate the completion time of the module, we use Figaro's built-in importance sampling algorithm to sample the model and print the distribution over the reading time variable. Observe that invoking an inference algorithm to estimate the completion time is a single line of code, and any other Figaro inference algorithm can be substituted into this program with no other changes.

```
val text = Constant(1000.0)
  val priorKnowledge = Flip(0.2)
  val populationReadSpeed = Normal(100.0, 50.0)
  val readSpeed = If(priorKnowledge,
    populationReadSpeed ++ Normal(50.0, 25.0), populationReadSpeed)
  val fatigued = Flip(0.4)
  val readingTime = If(fatigued,
    text / (readSpeed * Constant(0.5)), text / readSpeed)

  val algorithm = Importance(10000, readingTime)
  algorithm.start
  println(algorithm.distribution(readingTime))
```

Figure 4: Figaro program that models reading time of a Slide node.

Figaro probabilistic programming is useful in this context for a number of reasons: We can automatically build a model given a specification of the MAST skill tree, the trainee model, and a set of known relationships. Prediction based on the model is already coded in Figaro's inference algorithm, so additional effort is not required to use the model. Figaro supports the creation of dynamic Bayesian networks that model the temporal processes of variables, simulating fatigue and practice effects. We can continuously learn using these models; the probabilistic programs are flexible enough to update relationships between variables based on historical or dynamic data. Figaro's encapsulation mechanism enables easy creation of reusable components. Trainee models and MAST skill trees can be reused for future prediction models. It is embedded in a general purpose language, Scala, which allows the creation of front end graphical interfaces that can edit and invoke the models created in Figaro.

Figure 5 shows the results of running this Figaro model. The distribution of reading times has three modes. At about 7 seconds, individuals that have prior knowledge and are not fatigued read the slide quickly. At 10-11 seconds are individual that have no prior knowledge and are not fatigued. At 20-21 seconds are individuals without prior knowledge and who are fatigued, reading slowly to absorb more information. An instructor may use a model like this one to examine how individual slide contents may be processed by a class of students, and make small changes to the presentation to increase learning efficiency.

Figure 5: Probability Density of Reading Times for One Slide.

Figure 6 shows the probability density of reading times over three slides with the student having increased chance of fatigue (40%, 45%, and 50%) on each successive slide. In this simulation, only a small portion of the students are in the fastest group, completing three slides in about 20 seconds. The bulk of the students range from 25-55 seconds for these three slides, with three modes in this range covering the combinatorics of prior knowledge and different possible fatigue states on each slide. Also, a significant portion of the students takes longer than 55 seconds, with a possibility of up to 76 seconds to complete. An instructor can use this model to examine the differential effects of fatigue, prior knowledge, and reading speeds of a heterogeneous group of students, and adjust the learning content or course expectations accordingly.

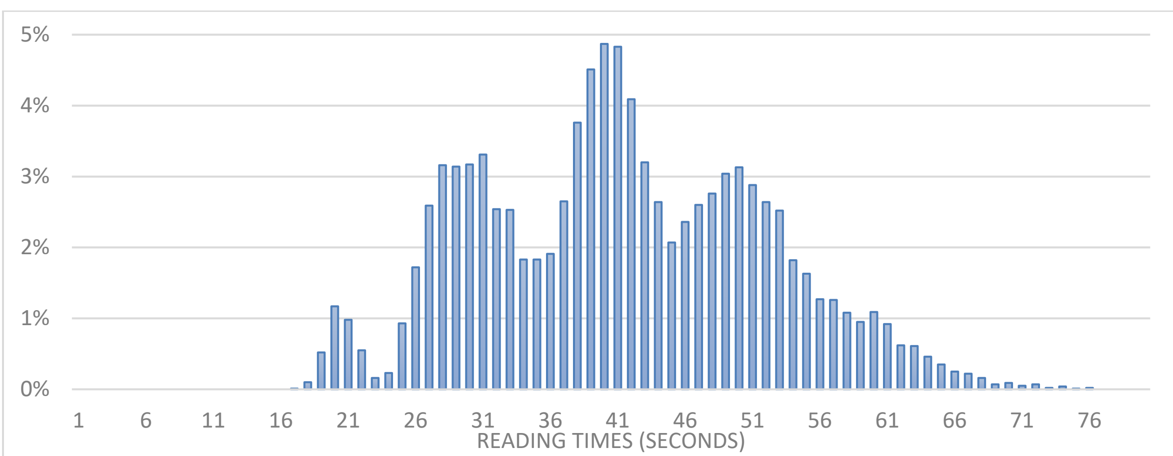


Figure 6: Probability Density of Reading Times for Three Slides with Increasing Chance of Fatigue.

This modeling can reveal underlying properties of the adaptive learning content that may be counter-intuitive at first glance. For example, the most likely reading speed of a single slide (according to the first model) is about 10 seconds. For three slides, one might assume 10 * 3 = 30 seconds, but the distribution in Figure 10 shows the mean of the predicted time about 41 seconds with significant standard deviation. Allotting only 10 seconds on average per slide in a course would prevent about two-thirds of students from completing all of the course content.

The adaptive training content with significant remedial steps has a much wider variance of completion times. We hypothesize that retraces through previous material (e.g., reviewing the boom operation slides) will be performed much faster than the initial trace. Trainees may also be able to optimize their reading and comprehension strategies if they know how they will be tested and what the consequences for failing are. Therefore, later sections in an adaptive training course (e.g., excavator bucket handling after boom handling) may have significantly different variable interactions than earlier sections, as trainees learn the training structure.

DISCUSSION

We believe that including a capability to predict training time for trainees in GIFT has several significant advantages for accelerated learning. First, it facilitates return on investment (ROI) calculations by enabling the author to determine training time reductions resulting from the addition of adaptive features. Second, it provides a means for GIFT to monitor student progress against an expected timeline. Students who take much longer to complete training than expected may not be fully engaged in the training or may be having difficulty with the material. These are conditions that might prompt a response by GIFT. Finally, it can play a role in quality control of GIFT courses. For example, if segments of a course take much longer than expected across multiple trainees, GIFT could flag those sections for review by the course author to insure that the material is presented clearly.

Determining the ROI for training is not always easy. As Fletchter and Chatham (2010) put it, how does one determine the benefit of a pound of training? In some cases it may be fairly straight forward. For example, one might measure the increase in revenue produced by the introduction of new training for a sales staff. While this may work for commercial businesses, the military is not a profit making organization, therefore one must look at other factors like cost avoidance to get a measure of ROI.

Determining this can be quite difficult as one rarely has before and after data on the operational impact of training. In rare cases it can be found. For example, Fletcher and Chatham (2010) examined the benefits of Top Gun training given to pilots during the Vietnam war. Because of this training, kill ratios of Navy pilots improved from 2.4 enemy kills per loss up to 12.5 enemy kills per loss. The authors determined that the training had reduced U.S. losses by about 10-12 aircraft during the war When they looked at the cost of procuring and employing that many aircraft during the war, they calculated that the training had saved the Navy about $132 million dollars for an ROI of about 2.5.

Determining the ROI for adaptive vs. non-adaptive training in terms of cost avoidance measures in an operational context would be very difficult. Adaptive training is still relatively new and opportunities to do side-by-side comparisons with traditional non-adaptive training are virtually non-existent. Rather than trying to quantify an impact in the operational environment however, we can look at the impact in a training environment. Specifically, one of the key advantages of adaptive training would be to reduce the overall time needed to deliver the training to a population of trainees.

A challenge for authors of adaptive training is determining how *adaptive* the training should be. While adding adaptive features can potentially save training time, it also increases the cost of development. How does one determine, when the training is adaptive enough? Using an ROI metric can help to answer this question. On one hand is the cost of adding the adaptive feature. On the other hand is the value of the time saved by that adaptive feature. The value of that time could be calculated by looking at the total salary paid to the trainees over that time (e.g., 1,000 trainees/year x .5h/trainee x $35/h = $17,500/year). So, as long as the cost of adding the adaptive feature was less than value of the time saved, there would be a positive ROI and therefore justification for adding that particular adaptive feature.

As can be seen, our model supports this strategy for the design and development of adaptivetraining in GIFT by helping to predict the effect of adaptive features on the training time for a known population of learners.

There are several challenges we may face as we develop this model. First, the initial MAST skill tree may not contain sufficient variables to predict adaptive training completion times. Our initial literature review and analysis have identified a potential set of most influential variables, but these variables may not be reflective of the completion time upon closer inspection. We will mitigate the identified risk by widening the scope of task models to incorporate more predictive variables if necessary.

Second, while the model predictions may be highly accurate, there is a risk that the system will be too difficult or time consuming to use for some or all of the target populations of instructional designers, course managers, and instructional staff. We mitigate this risk by conducting a requirements analysis early in the effort to closely examine the needs of these user groups and design our system and interfaces to best meet those needs. We will apply human factors and user-centered design and understand the challenges of and methods for developing highly useful and usable decision-aiding tools for practitioners.

Third, while this approach combines state of the art probabilistic approaches and identifies key variables from the literature and past experience, there is a potential that the initial predictions will not sufficiently account for the variability of trainee completion times. We plan to mitigate this risk by incorporating historical data early and adjusting the analysis techniques to capture the maximum amount of variability from data that can be reasonably collected in the field.

When complete, this will be the first system to predict the completion times of GIFT and to enable effective assessments of the ROI that is useful for key design and implementation decisions of an adaptive training system. It includes an innovative application of the procedure skill modelin the MAST skill tree to flexibly represent the adaptive training content for analysis. It is the first application using a probabilistic programming language (i.e., Figaro) to predict completion times for adaptive training technologies, including both unobserved latent variables and temporal factors, such as trainee fatigue, boredom, or flow.

REFERENCES

Army Research Laboratory. (2015). GIFT 2015-1, Generalized Intelligent Framework for Tutoring Release Page. Retrieved from http://www.gifttutoring.org

Dodds P, Fletcher JD. Opportunities for new “smart” learning environments enabled by next generation web capabilities. Journal of Educational Multimedia and Hypermedia. 2004;13:391–404.

Dynarsky M, Agodini R, Heaviside S, Novak T, Carey N, Camuzano L, Sussex W. Effectiveness of reading and mathematics software products: findings from the first student cohort; March Report to Congress 2007. [accessed 2015 May 15]. http://ies.ed.gov/ncee/pdf/20074005.pdf.

Fletcher JD. Evidence for learning from technology-assisted instruction. In: O’Neil HF, Perez R, editors. Technology applications in education: a learning view. Mahwah (NJ): Erlbaum; 2003. p. 79–99.

Fletcher, J. D., and R. E. Chatham. 2010. “Measuring Return on Investment in Military Training and Human Performance.” In Human Performance Enhancements in High-Risk Environments, edited by J. Cohn and P. O’Connor, 106–28. Santa Barbara, CA:

Praeger/ABC-CLIO.

Goldberg, B., & Hoffman, M. (2015). Adaptive course flow and sequencing through the engine for management of adaptive pedagogy (EMAP). In *Proceedings of the AIED Workshop on Developing a Generalized Intelligent Framework for Tutoring (GIFT): Informing Design through a Community of Practice* (pp. 46-53). Madrid, Spain.

Graesser AC, Conley M, Olney A. Intelligent tutoring systems. In: Harris KR, Graham S, Urdan T, editors. APA Educational Psychology Handbook: vol. 3. Applications to Learning and Teaching. Washington (DC): American Psychological Association; 2012. p. 451–473.

McDonnell Douglas Corporation. A survey and analysis of military computer-based training systems: (A two part study). Vol II: A descriptive and predictive model for evaluating instructional systems; 1977. Defense Advanced Research Projects Agency. [accessed 2015 October] http://www.dtic.mil/dtic/tr/fulltext/u2/a043358.pdf.

Merrill, M. D. (1983). Component display theory. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: An overview of their current status* (pp. 282-333). Hillsdale, NJ: Lawrence Erlbaum Associates.

Olney AM, Person NK, Graesser AC. Guru: designing a conversational expert intelligent tutoring system. In: Boonthum-Denecke C, McCarthy P, Lamkin T, editors. Cross-disciplinary advances in applied natural language processing: issues and approaches. Hershey (PA): Information Science Publishing; 2012. p. 156–171.

Pfeffer, A. (2012). Creating and manipulating probabilistic programs with Figaro. Workshop on Statistical Relational Artificial Intelligence (StarAI).

Sottilare R, Brawner KW, Goldberg BS, Holden HK. The generalized intelligent framework for tutoring (GIFT). Orlando (FL): Army Research Laboratory (US); Human Research and Engineering Directorate (HRED); 2012a [accessed 2015 May]. https://gifttutoring.org/attachments/152 /GIFTdescription_0.pdf.

Sottilare R, Goldberg BS, Brawner KW, Holden HK. A modular framework to support the authoring and assessment of adaptive computer-based tutoring systems (CBTS). In: Proceedings of the Interservice/Industry Training Simulation and Education Conference; 2012 Dec 3–6; Orlando, FL. Arlington (VA): National Defense Industrial Association; 2012b.

Sottilare, R. A. (2014). Using learner data to influence performance during adaptive tutoring experiences. In *Proceedings of International Conference on Augmented Cognition--HCII2014* (pp. 265-275). Crete, Greece: Springer.Steenbergen-Hu S, Cooper H. A meta-analysis of the effectiveness of intelligent tutoring systems on K-12 students’ mathematical learning. Journal of Educational Psychology. 2013;105(4):970–987.

Steenbergen-Hu S, Cooper H. A meta-analysis of the effectiveness of intelligent tutoring systems on college students’ academic learning. Journal of Educational Psychology. 2014;106:331–347.

VanLehn K, Graesser AC, Jackson GT, Jordan P, Olney A, Rosé CP. When are tutorial dialogues more effective than reading? Cognitive Science. 2007;31(1):3–62.

VanLehn K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist. 2011;46 (4):197–221.

ABOUT THE AUTHORS

Dr. Gregory Goodwin *is the acting Branch Chief of the Creative and Effective Technologies Branch at the Army Research Laboratory – Human Research and Engineering Directorate in Orlando, FL. For the last decade, he has worked for the Army researching ways to improve training methods and technologies. He holds a Ph.D. in Psychology from Binghamton University and an M.A. in Psychology from Wake Forest University.*

Dr. Jong Kim *is a postdoctoral fellow at the US Army Research Laboratory, Orlando, FL. Kim received his PhD in Industrial Engineering at Pennsylvania State University. His research interests lie in the area of Cognitive Science* *and Engineering. Kim is interested in theories of cognition for the development of intelligent systems. Kim developed a skill retention theory (D2P: Declarative to Procedural) that is being applied to implement a series of intelligent tutoring systems for the Navy in collaboration with Penn State and Charles River Analytics.*

Dr. James Niehaus *is a senior scientist at Charles River Analytics. Dr. Niehaus's areas of expertise include artificial intelligence, training systems, and health technology. Dr. Niehaus is currently the principal investigator on multiple efforts developing new training and health technologies. He is currently leading projects to: enhance physical therapy with video games and virtual coaches and investigate the neural, cognitive, and behavioral relationship between implicit learning and intuition. Dr. Niehaus is collaborating on projects to: identify and model critical skills in laparoscopic surgery, develop a training system for combat medic tourniquet application, and develop training for first responder trauma assessment. Previously, he has worked on projects concerning narrative generation, adaptation and individualization of training content, cultural interaction with virtual agents, and cognitive models of discourse comprehension. Dr. Niehaus has a B.S. in computer science from the College of Charleston and a Ph.D. in computer science from North Carolina State University..*