TASK ALLOCATION AND AUTOMATION LEVEL IN THE DESIGN OF INTELLIGENT TUTORING SYSTEM AUTHORING TOOLS

Scott Ososky[(a)], Robert A. Sottilare[(b)], & Keith Brawner[(c)]

[(a),(b), (c)] U.S. Army Research Laboratory
[(a)] Oak Ridge Associated Universities

[(a)] scott.j.ososky.ctr@mail.mil, [(b)] robert.a.sottilare.civ@mail.mil, [(c)] keith.w.brawner.civ@mail.mil

ABSTRACT
Intelligent tutoring system (ITS) *authoring tools* are productivity applications that aid humans in developing adaptive tutors. Authoring tools have the potential to reduce the skill, resources, and effort required to produce a tutor, thereby enabling ITS authoring for individuals without programming or instructional design experience. However ITSs continue to evolve and authoring tool designs iterate to keep pace. Authoring tool design has been previously framed as a tradeoff between usability, depth, and flexibility. This current paper, however, takes a different look at the design of authoring tools, through the lens of function allocation and levels of automation. The evolution of authoring tools within the Generalized Intelligent Framework for Tutoring (GIFT) will be used as an example of how tasks can be allocated (and automated) between humans and software. The paper will discuss lessons learned, implications for system design, and considerations regarding human mental models of adaptive tutor authoring processes.

Keywords: intelligent tutoring systems, authoring, automation, function allocation

1. INTRODUCTION

Intelligent tutoring systems (ITSs), or adaptive tutors, are learning systems that can collect data about a learner through assessments, reports, and sensors, in order to select and present optimal instructional content based on a unique learner profile. ITSs have been demonstrated to be more effective than one-to-many instruction (e.g., classroom instruction), approaching the effectiveness of one-to-one human tutoring (VanLehn, 2011). However, ITSs have not been widely adopted in educational settings (Bill & Melinda Gates Foundation, 2015; Murray, 2004) or military training environments (Sottilare, Graesser, Hu, & Holden, 2013). There are many factors that inhibit the adoption of ITSs within these contexts. A lack of usable and accessible authoring tools for ITSs is among those factors.

Much has been written on the topic of theory - and engineering-based efforts to develop authoring tools that provide to potential authors the functions necessary to create tutors without computer science or instructional design knowledge (Aleven & Sewall, 2010; Mitrovic et al., 2009; Olsen, Belenky, Aleven, & Rummel, 2013; Suraweera, Mitrovic, & Martin, 2010). Murray, specifically, has published numerous works on authoring tools over the past two decades. His work includes analysis of the problem (or opportunity) in which design tradeoffs are made in authoring tools between usability, depth, and flexibility (Murray, 1996, 2014). In summary, increasing the power of the authoring tools (i.e., depth), the applicability of the tools to different domains and problem spaces (i.e., flexibility), or the usability of the tools themselves (i.e., learnability, productivity), comes at a cost to one or both of the other two (Murray, 2004). This paper does not endeavor to duplicate the effort of those prior works, rather to explore the design of ITS authoring tools in a slightly different way. Specifically, this paper will delve into the following three questions: a) how should ITS authoring tasks be delegated between humans and software? b) For tasks that are delegated to software, what level of task automation is currently appropriate given the average user's current understanding of ITSs? c) How might automated tutor authoring evolve over the long term? There may not be a clear answers to these questions. Adaptive tutor authoring is a relatively new productivity paradigm. It would be useful to be able to educate current and potential authors about aspects of ITSs and the authoring process in order to build robust mental models of the authoring process and ITSs, in general. Simultaneously, the relative benefit of tutor authoring tools in terms of effectiveness, efficiency, and cost must also be considered. As such, specific efforts in automating parts of the ITS authoring process will also be presented. The current paper will use the authoring tools associated with the Generalized Intelligent Framework for Tutoring (GIFT) as a case study in balancing knowledge-building with automation.

2. BACKGROUND

2.1. Mental Models

Murray (2014) explained that authoring tools should help users build accurate mental models of the ITS building blocks, configurations, and workflow afforded by the authoring tool. This is inherently difficult, because ITSs are evolving, and each ITS will differ in some ways from others. However, mental model theory can provide guidance to approaching this interaction problem en route to an *accurate* mental model for the ITS authoring process. Rouse and Morris (1986) described mental models as "mechanisms whereby humans are able to generate descriptions of purpose and form, explanations of system functioning and observed system states, and predictions of future states" (p. 7). Mental models influence users' expectations regarding a system's functionality and guide user interaction behavior (Ososky, 2013). Human mental models do not need to be complete or even accurate to be applied to a specific system interaction (Norman, 1986). It is important that human authors understand what ITSs are capable of in order to fully realize the potential of adaptive tutors. Thus, it may be necessary to delegate certain ITS authoring tasks to humans in service of cultivating accurate mental models, even if the computer could do the task more efficiently. Therefore, it would be appropriate to dynamically allocate tasks along various levels of automation, supporting human authors of various skill levels and experiences.

2.2. Task allocation

Task (or function) allocation is the practice of determining which tasks should be delegated to a human, a machine / computer, or some combination of the two. Generally, decisions in task allocation are informed by the principle that humans are better at some tasks than computers, and vice versa (Fitts, 1951). In reality, there are a variety of factors that are considered in task allocation including error rates, fatigue, safety, technology limitations, human values, and human desire to work or learn (see section 2.1). Specifically, in ITS authoring, there is a strong need to build mental models of ITS authoring by humans, specifically instructors, instructional designers, and subject matter experts, because the output of authoring (i.e., tutors) will have far reaching implications in the education and training of learners. Further, since the individual(s) designing the instruction may not be the same individual(s) that administering it, knowledge regarding the functioning of the tutor will need to be communicated. Thus the allocation of authoring tasks between humans and automation (e.g., the authoring tools) should seek to be complementary of one another (Grote, Weik, Wäfler, & Zölch, 1995), supporting general ITS goals, like developing pedagogically sound instruction, providing individualized tutoring, delivering time/cost savings to stakeholders, and so on.

Another aspect of task allocation to consider is the ongoing development of ITS authoring tools. That is the notion of *balanced work*, as described by Hollangel and Bye (2000), resulting from "an adjustment by the working system to the performance demands" (p. 255). Resources are derived from humans, technology, and the organization under which the work occurs. In that paper, performance demands included safety and efficiency in the context of nuclear power production. For ITS authoring, the demand may be better described as efficiency and *user acceptance*. Authors may become more familiar with ITS authoring over time, thus making the process more efficient. However, ITS and their corresponding tools continue to evolve with new features and technologies, which disturb this balance. New features place new burdens on authors, leading to stress and fatigue. Thus, an inefficient (read: frustrating) system may turn authors away from ITSs entirely in favor of easier or "good enough" alternatives to adaptive tutoring (Ososky, 2016). "Good enough" alternatives include non-adaptive learning systems, analog alternatives (e.g., books, flashcards) and/or human-to-human tutoring.

2.3. Automation

Automation, then, is the use of a machine to perform a function or task. Regarding automation of a function for particular system, Parasuraman, Sheridan, & Wickens (2000) suggested the application of a model for types and levels of automation along a 10 point scale ranging from a fully manual to a fully automated task (Table 1). The middle automation layers are further differentiated by the complementary four-stage model of information processing which includes sensory processing, working memory, decision making, and response selection.

Table 1: Levels of Automation of decision and action selection (Parasuraman et al., 2000, p. 287)

10.	HIGH. The computer decides everything, acts autonomously, ignoring the human.
9.	Informs the human only if it, the computer, decides to
8.	Informs the human only if asked, or
7.	Executes automatically, the necessarily informs the human, and
6.	Allows the human a restricted time to veto before automatic execution, or
5.	Executes the suggestion if the human approves
4.	Suggests one alternative
3.	Narrows the selection down to a few, or
2.	The computer offers a complete set of decision/action alternatives, or
1.	LOW. The computer offers no assistance; human must take all decisions and actions.

That model is operationalized within an ITS authoring tool typically in support of efficiency/usability: auto-fill, intelligent defaults, advanced configuration options, templates, etc. With that comes the same concern that may manifest in other automated systems, automation misuse or disuse. Misuse describes failures resulting from an overreliance on automation capability, while the underutilization of automation is known as disuse (Parasuraman & Riley, 1997). Thus, if too much of the authoring system is automated, humans will have little idea of what it is doing (i.e., poor mental model) and will not be able to review/make changes to the tutor once it is produced. Alternatively, if the automation is poor, cumbersome, or not transparent, an author may lose trust in the ability of the tool to build a valid tutor, and discontinue use in favor of some other instructional alternative. Again, a dynamic approach to automation may provide a pathway to a useful and reliable authoring system for potential authors of different skill levels, roles, and/or experiences.

3. CASE STUDY

The widespread adoption of ITSs depends upon the ability for relatively low-skill users to be able to create tutoring systems. These tutoring systems give each user a customized/tailored learning experience, which depends runtime content availability. One of the primary goals for the Generalized Intelligent Framework for Tutoring (GIFT; Sottilare, Brawner, Goldberg, & Holden, 2012), an adaptive tutoring architecture, is to lower the entry skill and reduce the time to author ITSs. Pursuant to this goal, two principle methods are being investigated simultaneously: 1) improving the usability of authoring tools and 2) automating parts of the authoring process to reduce the author's workload. In order to support the former with the latter, the discussion now turns toward the authoring tools contained within GIFT.

3.1. Authoring tools in GIFT

The Generalized Intelligent Framework for Tutoring (GIFT) is "an empirically-based, service-oriented framework of tools, methods, and standards to make it easier to author computer-based tutoring systems (CBTS), manage instruction, and assess the effect of CBTS, components and methodologies" (Sottilare et al., 2013). GIFT is currently under development and includes a number of technologies, features, tools, and methods intended to support a variety of users including instructional designers, authors, instructors, researchers, and learners. The GIFT Authoring Tool (GAT) has undergone a number of major revisions since the project's origin. Each new version of the GAT has endeavored, in part, to improve authoring usability and efficiency, concurrent with the notion of *balanced work.*

3.2. ITS Authoring tasks for humans

High-level tasks associated with tutor authoring and management include: 1.) Defining objectives, 2.) Content curation 3.) Sequencing content and overall "course flow" 4.) Creating discrete-time assessments, and 5.) Creating real-time assessments, 6.) Generating learner reports and, 7.) Analyzing ITS data for purposes of tutor refinement. Each of those tasks will be described in the following sections, through the lens of task automation.

3.2.1. Define learning objectives

GIFT represents concepts to be learned as either a flat list (i.e., no hierarchical relationship) or a hierarchy. The lowest level in the hierarchy of concepts require an assessment to determine if the learner has mastered that concept. Leaves may be defined as any concept without a child. Concepts at the leaf level may be rolled up to determine proficiency in higher level concepts. For example, the assessments for concepts noted as circles, parabolas, ellipses, and hyperbolas may be used to assess the parent concept "Examples of Conic Sections and their properties." This means that each leaf in the hierarchy or item on a list that is assessed contributes to authoring workload as it requires the development of an assessment (e.g., knowledge, skill test, or real-time assessment coupled with an external environment). Currently, this task is fully performed by the human (i.e., Level 1 automation). Even if the system provided ontologies for high-level concepts to form hierarchies, the manner in which those hierarchies may be ultimately structured will likely differ between course designers.

3.2.2. Content curation and creation

Content curation and, subsequently, content sequencing are two tasks that humans are primarily responsible for in GIFT, with little oversight from system. Content creation / curation involves gathering all of the multimedia that will be displayed to the learner, whether created from scratch or taken from another source.

GIFT makes the assumption that content curation is an integrated part of the authoring process. In other words, the efforts of finding, retrieving, and organizing content must be performed within the system. We assume this must be done with all content. Authored content need not be created from scratch, but does have to be managed. Some content (e.g., presentation material, surveys, quizzes, multimedia, or simulation scenarios) can be reused. The need for content is primarily driven by the total number of concepts, which include terminal learning objectives (TLOs) and enabling learning objectives (ELOs). In GIFT, concepts can be organized in a hierarchical or linear structure.

3.2.3. Content sequencing and course-flow

Content sequencing involves building the timeline of events that the learner will encounter (Figure 1); in GIFT this is known as the *course flow*. The course flow is made up of a series of events known as *course objects.*

The current authoring experience is based about a visual course building interface within the Course Creator. From within this interface, all other core aspects of course authoring are accessible to the user. The course flow timeline interface was redesigned based on a flow chart (or discrete event process) metaphor with simple drag-and-drop functionality. The visual structure of the course more accurately suggests the sequencing functions that are available to course authors. The design intent was to evoke a *mental model* of similar, more familiar interfaces in order to make tutor authoring more learnable for new users. The system provides a set of all possible course objects, displayed in the toolbox on the left-hand side of the interface (similar to automation level 2). Authors can drag and drop objects onto the timeline in any position. Objects already on the timeline can be re-ordered or deleted as needed.

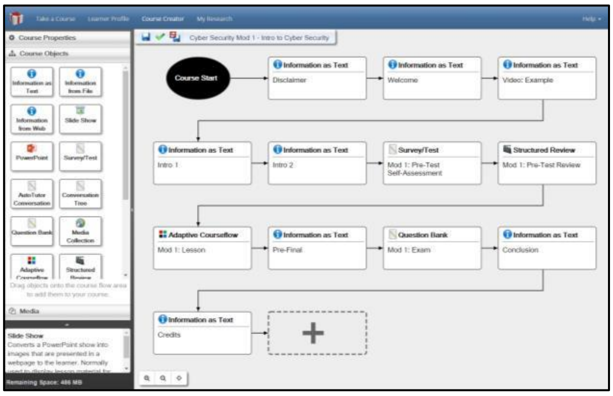

Figure 1: Human authors are responsible for sequencing course objects, which contain multimedia, assessments, digital game scenarios, and other instructional events.

The current authoring experience is based about a visual course building interface within the Course Creator. From within this interface, all other core aspects of course authoring are accessible to the user. The course flow timeline interface was redesigned based on a flow chart (or discrete event process) metaphor with simple drag-and-drop functionality. The visual structure of the course more accurately suggests the sequencing functions that are available to course authors. The design intent was to evoke a *mental model* of similar, more familiar interfaces in order to make tutor authoring more learnable for new users. The system provides a set of all possible course objects, displayed in the toolbox on the left-hand side of the interface (similar to automation level 2). Authors can drag and drop objects onto the timeline in any position. Objects already on the timeline can be re-ordered or deleted as needed.

3.2.4. Author discrete-time assessments

Creating content for assessment is another task that is, in part, delegated to the author; however, parts of this process are in planning to increase in level of automation. Discrete-time assessments include surveys and question banks, as well as the logic that supports adaptive branching through a course. In the case of the former, that is similar to a media-content creation task (Figure 2). Human authors must generate a set of questions (and responses) for a concept(s) within the course. The author must also designate the correct response(s) and manner in which the question will be scored. If the question is contained within a question bank, the author may also assign a difficulty level to the question.

Within a question bank, the human author must also specific the logic for the number and type of questions that should be randomly displayed to the learner at runtime. That is the point at which we are developing more automation to handle the logic set by the author. In the past would have been responsible for calculating and setting the question selection and scoring criteria for the survey, but usability analysis determined that this created a burden on the author to remember the type and count of each question type (e.g., easy, medium, hard) as well as the possible points available for a particular survey administration. Now, the GAT is moving toward an automated process by which these sums are calculated on-the-fly and dynamically update when changes are made to the survey content. This seemingly superficial change significantly reduces the possibility for display and/or scoring errors during runtime.

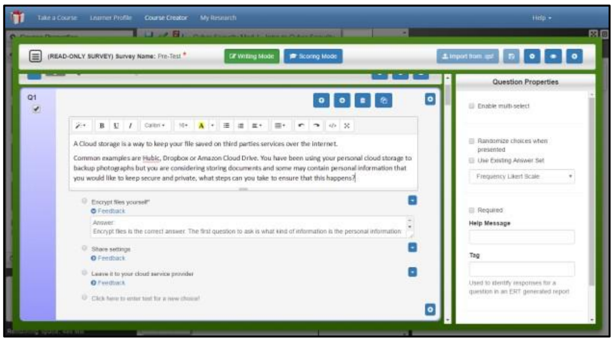

Figure 2: Human authors create questions, the system selects questions to display and calculates scores

One way in which GIFT provides adaptive tutoring is via the *adaptive courseflow course object*. In that object, each concept must be tied to content presented to the learner as part of Merrill's component display theory (CDT; Merrill, 1983) implemented within the GIFT authoring schema. For a set of concepts, this CDT content includes information about *rules* (facts, principles), *examples* (models of successful behavior), recall (an assessment also known as a check on learning or a knowledge test), and *practice* (opportunities to apply knowledge and develop skill). Recall and practice are relevant to this current paper.

For the recall quadrant, GIFT assesses domain knowledge. For the practice quadrant, both knowledge and skill may be assessed as part of an interactive experience (e.g., simulation, serious game; see sections 3.2.5 and 3.3.2, respectively). The human author is responsible for setting thresholds at both the recall and practice phases at which the learner will either move onto the next activity, or return to either the rules or examples phases for remediation in the form of additional content. That content comes from the same pool of content described earlier (in 3.2.2), however the content must be tagged with appropriate metadata in order for the tutor to dynamically select content for presentation at runtime. Currently, much of this selection logic is transparent to authors, and was coded based on empirical research and review of the relevant literature. It would be similarly beneficial if agents within the tutoring system were able to make intelligent suggestions regarding the metadata tagging of media content (i.e., automation level 5). Recommendations for such agents can be found in other work (Brawner, 2015).

3.2.5. Author real-time assessments

Real-time assessment in GIFT refers to the monitoring of learners while engaged in an interactive experience, which could be anything from dynamic slide show content, to an immersive game or simulation environment. Those experiences may occur within the practice phase of an adaptive course flow course object (see 3.2.4) or as a standalone activity. The process of creating real-time assessments is inherently complex and contains a number of moving parts. The technical details of how GIFT communicates with external applications is outside the scope of this discussion; however, the relevant aspect of that function is that authors are responsible for interpreting messages and creating assessment logic from inputs provided by the interactive application. Real-time assessment of an interactive application includes four steps to be defined by the author (Figure 3): 1) scenario properties, 2) tasks and concepts 3) instructional strategies and 4) state transitions. An example of this for a virtual excavator simulator is shown in the following figure.

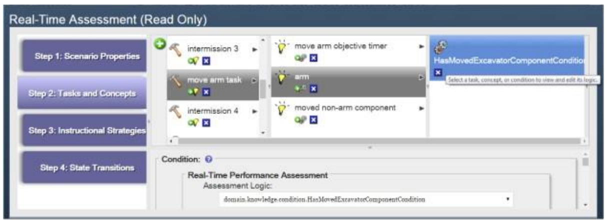

Figure 3: The four step process to create real time assessments for external learning environments is currently a highly manual process.

There, the GAT provides support around automation level 3 (refer back to Table 1 for a refresher), in which options available to the human author in the latter steps of this process are constrained based on human inputs in the earlier part of the authoring process. However, this may be a point at which *balanced work* is disturbed by the sheer complexity of this process, and the potential for errors during configuration. To that end, the GAT development team is currently working toward a more streamlined workflow for creating real-time assessment of practice environments and other external applications. By providing more of a guided experience through this process, and reaching level 4/5 of automation, the memory burden on the human author can be reduced. In other words, instead of the human author working from a blank slate, an intelligent set of options can be presented from which the author can evaluate. Thus, authors free up cognitive resources to work on other tasks, such as designing feedback or remedial content to improve the quality of the instruction for the learner.

3.2.6. Author tutor interactions

Tutor interactions refer to the tutor's responses to the learner as the learner interacts with the instructional content. For a survey or quiz, tutor interactions take the form of feedback based on the learner's responses. The tutor may also take action to provide remediation to the learner, in the case of an adaptive courseflow sequence (see 3.2.4). Real-time assessment tutor interactions are much more open ended. Tutors can alter the scenario in an interactive simulation in response to learner behavior, provide encouragement or coaching via a synthetic avatar, or suggest additional instruction on previous concepts. While tutors are expected to interact with the learner without direct human intervention, human authors are currently responsible for manually specifying most of the feedback messages, scenario adaptations, etc. during the authoring process.

3.3. ITS authoring tasks for computers

Automating different functions in GIFT associated with authoring has been explored through different approaches with varying degrees of success including, automated content development from text sources and the development of wizards to guide inexperienced authors. This section discusses current functions in GIFT that seem appropriate for allocation to automation at higher levels. Early portions of the project maintained many of these tasks as human tasks – the exact tasks, formats, structures, and other details were still being defined. The modern case study of GIFT involves many tasks which were originally human-performed, but are now automated.

3.3.1. System-level functions

While not a major step of authoring, *per se*, we identify system level functions as those tasks that are necessary to the operation of ITS platform, but not necessarily relevant to the goal of creating a tutor. Those include, for example, file management on a server, maintaining users and files within a database, and generating configuration scripts to power the tutor. We will take a closer look at that last example.

At the system-level of GIFT, a set of extensible markup language (XML) files provide the configuration for a particular GIFT course, including the use of learning content, assessments, and external training applications (Sottilare et al., 2013). The current version of the GAT is a cloud-hosted, web-based productivity application (known as GIFT Cloud) that uses flow charts and forms to help users to structure and configure tutors. In terms of function allocation, the construction of the XML code is being automated entirely by the GAT, with virtually no oversight by the author to verify the syntax of the code is error-free. In earlier, downloadable versions of GIFT, however, humans were responsible for writing and maintaining the underlying XML code.

In most cases, save for administrators and power users, system level functions should be automated by the system, and only informing the human when there is an error (automation level 7). This helps to provide balanced work, by reducing the burden on the human author, and allocating more human resources to tasks relevant to tutoring. In writing, this may seem obvious, however many ITSs are borne from development projects, where authoring interfaces (if available at all) are driven by engineering needs, not user-centered design. ITS development, by its nature, outpaces the development of authoring interfaces and tools. Constant monitoring throughout the ITS development project is therefore necessary to ensure that humans are not accidently assigned system-level tasks.

3.3.2. Creating practice and training content

Much of Army training differs from traditional ITS content (e.g., problem-based mathematics and physics tutors) in that it often requires conceptual knowledge (why you are doing something) in addition to procedural knowledge (what to do). At the U.S. Army Research Laboratory (ARL), we are seeking new methods to reduce the skill and time required to author scenario-based simulations and serious games to allow GIFT to automatically author variants of existing training scenarios which are relevant to the authors defined learning objectives. That is similar to content creation (see 3.2.2), however the workload scales exponentially in comparison when learners require an abundance of unique opportunities for interactive practice and evaluation.

The method to address that challenge is called automated scenario generation (ASG; Zook et al., 2012) or evolutionary scenario generation (ESG; Luo, Yin, Cai, Zhong, & Lees, 2017). This method focuses on how to use information from a "parent" scenario to generate hundreds or thousands of "child" scenarios and then rank order the child scenarios according to their relevance to a set of author-defined learning objectives (see 3.2.1).

The automated scenario generation method described would allow a GIFT-based tutor to customize (e.g., change difficulty level of the scenario) in real-time based on the learner's state (e.g., performance or emotion) or traits (e.g., personality) to optimize their learning, retention, and transfer of skills from training to the operational or work environment. This method would allow ITS developers who want to integrate GIFT with training simulation or serious games (e.g., Virtual Battle Space) to expand existing training capabilities to facilitate adaptive instruction with minimal additional burden on the scenario author.

ASG and ESG are good examples of ideal collaboration between humans and automated authoring tools. At automation level 5/6 the system is providing intelligent suggestions to the human regarding potential scenarios, and the human is able to review those suggestions and make manual changes, as needed.

3.3.3. Reporting and Analytics

There are two authoring-related tasks that occur after a tutor has been created: reporting and analytics. The tasks are described as authoring-related because either may result in changes being made to an existing tutor, or providing a foundation for a new tutor. For purposes of the current discussion, *reporting* refers to the collection and presentation of data resulting from learners interacting with a course (e.g., instructor dashboard). *Analytics*, here, refers to the aggregation of learner and system data, perhaps in a historical context, over a longer-term (e.g., training effectiveness, return on investment).

Those tasks are combined here because, within GIFT, they are currently accomplished from within the same interfaces in the GAT. At present, these processes are highly manual (automation levels 1 / 2). An event reporting tool is available within the *My Research* section of GIFT. This tool allows humans to build customized reports by selecting from a set of available data features to compile into a spreadsheet file (Figure 4). Currently, the human is responsible for processing the data in whatever manner they choose. This task has remained mostly manual because GIFT is highly domain independent, meaning that tutors may generate potentially many different types of data.

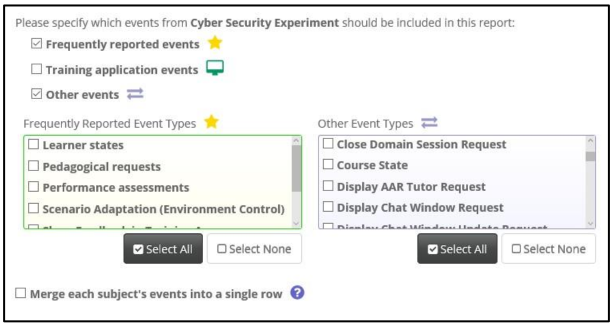

Figure 4: GIFT Cloud's event reporting tool allows authors to select from either suggested or all possible options.

We are currently investigating best practices in automating this process in the form of dynamically configurable instructor dashboards, and enhanced visualization tools for analytics. The design intent is to reduce the amount of time needed to generate insights from data, as well as reduce the skill level required in order for humans to leverage insights.

4. DISCUSSION

Using GIFT as a case study, we have identified major functions related to adaptive tutor authoring. Each function is evaluated through the lens of function allocation between humans and the system, and the degree to which automated functions within the system support human authors. A primary aim of the present paper was to present evidence and make recommendations for the ongoing design of ITS authoring tools, in general.

4.1. Summary of GIFT authoring automation

The current case study was presented in the context of where the GAT is now with respect to automation. Target levels of automation for those tasks, near or on the horizon, were also described. Those are summarized in Figure 5.

With the exception of system-level functions, it is suggested that most ITS authoring tasks be automated no higher than level 5. This is the point at which the human still approves all decisions and/or suggestions made by the system. Stated differently, it is recommended that a human remains "in the loop" regarding all content and experiences that a learner may encounter when interacting with the adaptive tutor. There are several reasons for this: 1.) Intelligent tutoring, while supported by decades of research, is still relatively novel to some involved in the process of creating and managing instruction. Therefore, keeping the human in the loop allows them to build knowledge of an otherwise novel system, while also developing trust in the efficiency and effectiveness of adaptive tutors. 2.) The concept of what an ITS is (and could be) is still evolving, in both features and intelligence. Automating too much of the process may inhibit authors from making technical and creative contributions to the community at-large. 3.) Existing educational models rely upon instructors who are responsible for the student experiences – learning, or failure to learn, is the responsibility of the instructor. The current instructional model is more "sage on the stage" than "guide on the side." In the future, especially in alternative models of education, the level of automation recommendation may change to reflect higher levels of automation, however more empirical evidence is required to allow systems to autonomously act on system suggestions without human supervisory approval.

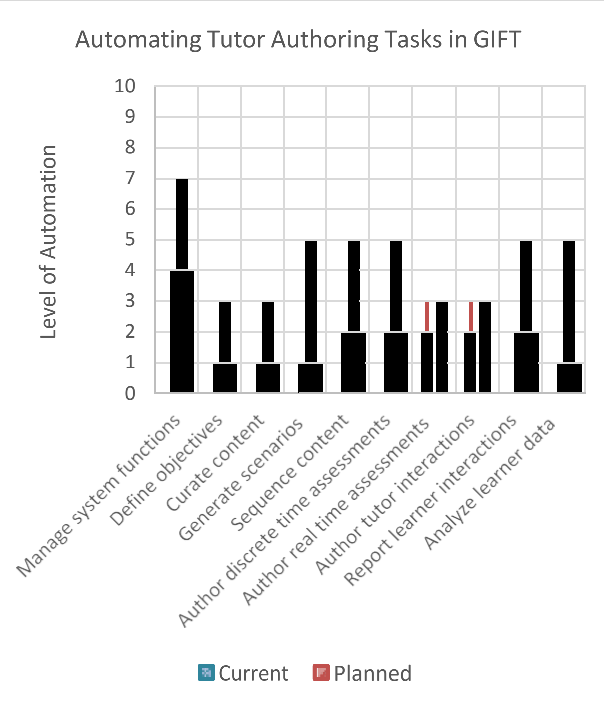

Figure 5: Current vs. planned tutor authoring function level of automation in GIFT

4.2. Looking toward the future

The authors would like to end on a discussion of the "art of the possible" – what is the maximum amount of automation possible at each level and what are the probable technologies involved at the high level? This paper presented a view of how a human author affects the system operation, while previous work outlines the operation of system-level functions (Brawner, Sinatra, & Sottilare, in press), or existing system-level function already developed (Brawner, Heylmun, & Hoffman, 2017).

First, it is worthwhile to mention that a discussion of learner and instructional modeling has been removed from this work. Other than the configuration of a small number of settings, for which defaults exist, there is nothing that the human user needs to do in order to capture emotional state data from a Microsoft Kinect, or to select the type of instruction used in remediation from a failed quiz. Learner-specific instructional strategies can now be applied directly from provided content without *any* interaction from the human user – removing this functionality from the discussion of instructor roles.

The GIFT system, and the base of instructors / instructional designers that use it, is not necessarily ready for this approach, but it is technically possible to automate the author out of these functions. As an example, if we assume that the author can provide or point to seed content (such as a textbook, deck of PowerPoint slides, etc.), the identification of instructional objectives can be automated via natural language processing technologies. Using the same technologies, the content can be curated into which portions of it address which objectives. Further use of text processing technologies such as Latent Dirichlet Allocation (LDA) and Latent Semantic Allocation (LSA) can personalize learning paths based on observed evidence or relation to other topics (Lahti, 2010). In total, a fully automatic authoring process (level 5+) may be possible, as discussed elsewhere (Olney, Brawner, Pavlik, & Koedinger, 2015). Further, an example of "full" (level 5) automation may consist of the techniques and associated technologies described in Table 2.

Table 2: Summary of higher automation authoring tasks and associated technologies

Authoring technique	Enabling technology
An analysis of a subset of Wikipedia pages, including the linked-in references	Learning objectives identified LSA / LDA technologies used
An analysis of the learner profile and mapping to the learning objectives	Simple difference calculations
A path mapping of learning objectives through content	A* ("best first") search algorithm
Generation of discrete time assessments	Generation of assessment questions Generation of distractors / incorrect LDA, Ontology search technologies such as syntactic tree kernels
Generation of simulation events	Using seed scenarios, scenario generation techniques, simulated students for scenario assessments Multiple AI/search technologies, no clear choices

Authoring technique	Enabling technology
Generation of realtime assessments	Deviation from expert performance criterion Simple metrics such as standard deviations
Analysis of learner data	Typical reporting – grades, trends, etc. Typical algorithms, clustering of student groups, etc.
Self-Improving systems	Learning appropriate instructional interventions over time Markov decision processes, policy adjustments

5. CLOSING THOUGHTS

This paper presented a view of the three questions stated earlier: How should ITS authoring tasks be delegated between humans and software? What level of automation is appropriate for tasks that are delegated to software? How might tutor authoring be automated in the future? The answer to all questions is that the system (automation) and user have a collaborative relationship – the system shouldn't ever present unknown materials to a learner, but suggestions, automation, and assistance is valuable during the authoring process. While there is potential to fully automate ITS creation, human tools to create ITSs will be in use and practice for the foreseeable future. Level 5 automation, where the human has approval ability, but not time-sensitive veto ability, appears to be the "sweet spot" for the various activities of content creation. Simultaneously, the management of pedagogy, learner profiles, reporting, and other tasks can be left to the machine alone.

ACKNOWLEDGMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory. The statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

REFERENCES

Aleven, V., & Sewall, J. (2010). Hands-on introduction to creating intelligent tutoring systems without programming using the cognitive tutor authoring tools (CTAT). *Proceedings of the 9th International Conference of the Learning Sciences-Volume 2*, 511-512.

Bill & Melinda Gates Foundation. (2015). Teachers know best: What educators want from digital instructional tools 2.0. Retrieved from http://collegeready.gatesfoundation.org/2015/12/what-educators-want-from-digital-tools-2-0/

Brawner, K. (2015). Authoring agent based tutors *Design Recommendations for ITS* (Vol. 3). Orlando, FL: U.S. Army Research Laboratory.

Brawner, K., Heylmun, Z., & Hoffman, M. (2017). The GIFT 2017 Architecture Report. *5th Annual GIFT Users Symposium*.

Brawner, K., Sinatra, A., & Sottilare, R. (in press). Motivation and Research In Architectural Intelligent Tutoring. *Simulation and Process Modelling Special Issue of "New trends of simulation and process modeling in multiple domains"*.

Fitts, P. M. (1951). Human engineering for an effective air-navigation and traffic-control system.

Grote, G., Weik, S., Wäfler, T., & Zölch, M. (1995). Complementary allocation of functions in automated work systems. *Advances in Human Factors/Ergonomics, 20*, 989-994.

Hollnagel, E., & Bye, A. (2000). Principles for modelling function allocation. *International Journal of Human-Computer Studies, 52*(2), 253-265.

Lahti, L. (2010). Personalized learning paths based on Wikipedia article statistics. *2nd International Conference on Computer Supported Education (CSEDU 2009)*.

Luo, L., Yin, H., Cai, W., Zhong, J., & Lees, M. (2017). Design and evaluation of a data-driven scenario generation framework for game-based training. *IEEE Transactions on Computational Intelligence and AI in Games*.

Merrill, M. D. (1983). Component display theory. *Instructional-design theories and models: An overview of their current status, 1*, 282-333.

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*, 155-188.

Murray, T. (1996). Having it all, maybe: Design tradeoffs in ITS authoring tools. *Intelligent Tutoring Systems*, 93-101.

Murray, T. (2004). Design tradeoffs in usability and power for advanced educational software authoring tools. *Educational Technology, 44*(5), 10-16.

Murray, T. (2014). Theory-based authoring tool design: considering the complexity of tasks and mental models. In R. A. Sottilare, A. C. Graesser, X. Hu & K. Brawner (Eds.), *Design Recommendations for Intelligent Tutoring Systems* (Vol. 3 Authoring Tools and Expert Modeling Techniques, pp. 9-29). Orlando, FL: U.S. Army Research Laboratory

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction* (pp. 31-61). Hillsdale, NJ: Lawrence Erlbaum Associates.

Olney, A. M., Brawner, K., Pavlik, P., & Koedinger, K. R. (2015). Emerging trends in automated authoring. In R. A. Sottilare, A. Graesser, X. Hu & K. Brawner

(Eds.), *Design recommendations for adaptive intelligent tutoring systems: Learner modeling* (Vol. 3, pp. 227-242).
Olsen, J. K., Belenky, D. M., Aleven, V., & Rummel, N. (2013). Intelligent tutoring systems for collaborative learning: Enhancements to authoring tools. *Artificial Intelligence in Education*, 900-903.
Ososky, S. (2013). *Influence of task-role mental models on human interpretation of robot motion behavior.* (Doctoral Dissertation). University of Central Florida, Orlando, FL.
Ososky, S. (2016). Designing the user experience of the GIFT Cloud authoring tools. In R. Sottilare & S. Ososky (Eds.), *Proceedings of the 4th Annual GIFT Users Symposium* (Vol. 4, pp. 145-156). Orlando, FL: U.S. Army Research Laboratory.
Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 39*(2), 230-253.
Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans, 30*(3), 286-297.
Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin, 100*(3), 349.
Sottilare, R., Brawner, K. W., Goldberg, B. S., & Holden, H. A. (2012). *The Generalized Intelligent Framework for Tutoring (GIFT). Concept paper released as part of GIFT software documentation.* Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED) Retrieved from https://gifttutoring.org/attachments/152/GIFTDescription_0.pdf.
Sottilare, R., Graesser, A. C., Hu, X., & Holden, H. (2013). *Design Recommendations for Intelligent Tutoring Systems: Volume 1 Learner Modeling*. Orlando: U.S. Army Research Laboratory.
Suraweera, P., Mitrovic, A., & Martin, B. (2010). Widening the knowledge acquisition bottleneck for constraint-based tutors. *International Journal of Artificial Intelligence in Education (IJAIED), 20*, 137-173. doi: 10.3233/JAI-2010-0005
VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems and other tutoring systems. *Educational Psychologist, 46*(4), 197-221.
Zook, A., Lee-Urban, S., Riedl, M. O., Holden, H. K., Sottilare, R. A., & Brawner, K. W. (2012). Automated scenario generation: toward tailored and optimized military training in virtual environments. *Proceedings of the international conference on the foundations of digital games*, 164-171.

AUTHORS' BIOGRAPHIES

Scott Ososky, Ph.D. is a Postdoctoral Research Fellow assigned to the U.S. Army Research Laboratory (ARL) within the Human Research and Engineering Directorate (HRED) at ARL Orlando. His current research examines mental models of adaptive tutor authoring, including user experience issues related to productivity tools and interfaces within the adaptive tutor authoring workflow. His prior work regarding mental models of human interaction with intelligent robotic teammates has been published in the proceedings of the Human Factors and Ergonomics Society, HCI International, and SPIE Defense & Security annual meetings. Dr. Ososky received his Ph.D. and M.S. in Modeling & Simulation, as well as a B.S. in Management Information Systems, from the University of Central Florida.

Dr. Robert A. Sottilare leads adaptive training research at the US Army Research Laboratory where the focus of his research is automated authoring, instructional management, and analysis tools and methods for intelligent tutoring systems (ITSs). He is a co-creator of the Generalized Intelligent Framework for Tutoring (GIFT), an open source, AI-based adaptive instructional architecture. He is the lead editor for the *Design Recommendations for Intelligent Tutoring Systems* book series and the founding chair of the GIFT Users Symposia. He is a program committee member and frequent speaker at the Defense & Homeland Security Simulation, Augmented Cognition, and AI in Education conferences. Dr. Sottilare is a member of the AI in Education Society, the Florida AI Research Society, and the American Education Research Association. He is a faculty scholar and adjunct professor at the University of Central Florida where he teaches a graduate level course in ITS design. Dr. Sottilare is also a frequent lecturer at the United States Military Academy (USMA) where he teaches a senior level colloquium on adaptive training and ITS design. He has a long history of participation in international scientific fora including NATO and the Technical Cooperation Program. Dr. Sottilare is the recipient of the Army Achievement Medal for Civilian Service (2008), and two lifetime achievement awards in Modeling & Simulation: US Army RDECOM (2012) and National Training & Simulation Association (2015).

Keith Brawner, Ph.D. is a researcher for the Learning in Intelligent Tutoring Environments (LITE) Lab within the U. S. Army Research Laboratory's Human Research & Engineering Directorate (ARL-HRED). He has 11 years of experience within U.S. Army and Navy acquisition, development, and research agencies. He holds a Masters and PhD degree in Computer Engineering with a focus on Intelligent Systems and Machine Learning from the University of Central Florida. His current research is in machine learning, cognitive architectures, learning technologies, and ITS architecture. He manages and advises research in adaptive training and architectural programs towards next-generation training.