GIFT Cloud: Improving Usability of Adaptive Tutor Authoring Tools within a Web-based Application

Scott Ososky[12], Keith Brawner[1], Benjamin Goldberg[1], and Robert Sottilare[1]
[1]U.S. Army Research Laboratory, [2]Oak Ridge Associated Universities

GIFT Cloud is the recently released web-based application version of GIFT, an open-source computer-based tutoring architecture that supports authoring, deployment, and evaluation of intelligent tutoring system technologies. This paper presents the GIFT Cloud Authoring Tools, through the lens of usability. Each major element within the authoring tools is described, along with usability design considerations that were made in order to reduce occurrence of error, to organize information, and to support end-user goals. The initial release of GIFT Cloud supports an iterative design approach, informed by user data and feedback, with an overall goal of making tutor authoring practical for subject matter experts without computer programming or instructional design knowledge. As such, lessons learned from this release, as well as plans for future research and usability improvements, are discussed.

INTRODUCTION

Intelligent tutoring systems (ITS), or adaptive tutors, are instructional systems that collect learner data through reports, assessments, and sensors in order to optimize learning based on the unique attributes and experience of the learner. 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, ITS 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 ITS in those contexts including a lack of usable and efficient authoring tools for developing tutors.

The Generalized Intelligent Framework for Tutoring (GIFT) is an empirically-based, service-oriented framework of tools and methods to build, deliver, and evaluate ITS across any domain of instruction (Sottilare et al., 2013). GIFT is being developed as an open-source project, available to the public at no cost at GIFTtutoring.org. Its ongoing development is driven in-part by community feedback and user requirements in order to make it easier and faster to author ITSs, manage instruction, predict future learner states, and assess the effect of adaptive instructional techniques through evaluation and experimentation.

Until recently, GIFT was available only as a downloadable program, which required installation and configuration on the course-author's and learners' host computers, respectively. Previous versions of this downloadable version of GIFT consisted of modular user interfaces (UI), some of which required knowledge of computer programming and/or scripting in order to build adaptive tutoring content. That platform was demonstrated to be effective for developing training content and experimental test-beds within a number of domains (Sinatra, Goldberg, & Sottilare, 2014); however, heuristic reviews (Nielsen, 1995) of GIFT's authoring tools identified a number of opportunities for improvement with respect to the way information, content, and workflow is organized and presented to the end-user (Ososky & Sottilare, 2016; Sottilare et al., 2014).

With the recent "alpha" release of *GIFT Cloud* (*Figure 1*), users can create, manage, and access course content through a server-supported, web-based application in a browser. The purpose of the current discussion is to describe ongoing work, lessons learned, and future research directions, with respect to improving the usability of GIFT authoring and adaptive tutoring research, as a function of transitioning to a web-based application. The human factors community may be particularly interested in the ongoing development of tools intended to facilitate research using GIFT-courses. The results from this iterative work are intended to benefit course creators and researchers, respectively, through intuitive and accessible interaction design.

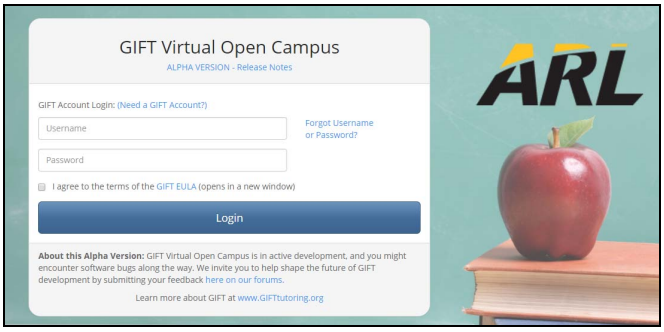

Figure 1. GIFT Cloud log-in, located at cloud.gifttutoring.org

BACKGROUND

ITSs are generally powered by four components, a domain module, a learner module, a tutoring / pedagogical module, and a communication module (Woolf, 2009, pp. 44-45). The inherent complexity of an ITS creates a significant challenge in developing authoring tools that are usable by potential authors without computer programming or instructional design experience (Murray, 2015). That includes creating interfaces for authoring, for instance, the representation of the course flow, the management of learning content, and the structure of assessments. The concept of what an ITS could potentially be is also still evolving. Therefore, the user experience for authoring tools that are developed for a particular ITS platform are often a function of that platform's

depth and/or flexibility for tutoring in a given domain(s) (Aleven & Sewall, 2010; Mitrovic et al., 2009; Olsen, Belenky, Aleven, & Rummel, 2013).

GIFT, specifically, is a series of modules (domain, pedagogical, learner, sensor), which use standard communication methods to exchange information between them via request-driven, event-driven, or periodic messages. Content in GIFT is separated from executable code. A set of extensible markup language (XML) files provide the configuration for a particular course, including the use of learning content and external training applications, where applicable (Sottilare et al., 2013). Given this modular design, GIFT could be reconfigured to use different sensors, external programs, or even different pedagogical engines depending on the needs of the course author and learning domain. The design of GIFT lends itself to the reuse of tutors as well.

PRACTICE INNOVATION

The challenge in creating authoring tools from a usability perspective is to bridge the gap between complex ITS design principles and authors' mental models for creating tutors. Potential tutor-authors involved in training and education may come with a variety of skills and backgrounds, further shaping their mental models of the authoring process. Mental models are the frameworks that guide human understanding of a system's purpose and function, as well as the interpretation of system states and prediction of future states (Rouse & Morris, 1986). Mental models also influence users' expectations regarding a system's functionality as well as subjective assessments of systems (Ososky, 2013). Authoring tools in GIFT have improved significantly since the initial release of GIFT: from writing XML code, to guided XML editing, to form-based editing, and now toward a more object-oriented design of a course with the release of *GIFT Cloud*.

Authoring within a Web-application

First, GIFT Cloud is a web-application, which eliminates the need to download and configure software on the host computer. Tutors and their associated content may be uploaded and saved to the server and/or referenced with web-links, which allows an author to work on their courses from any PC with an internet connection. Authors' content and work are secured by GIFT Account access. (GIFT Accounts are created on the project site: https://gifttutoring.org). Moving GIFT into a web-based environment is an important step toward enabling advanced functionality for users including content curation and collaborative authoring.

Transitions: The Building Blocks of GIFT

Each course object within GIFT is referred to as a *transition*. Transitions include: (a) Guidance, simple messages displayed to a learner; (b) Lesson Material, in the form of documents and/or web-resources; (c) Survey, to present pre-prepared or dynamically generated assessments, (d) Training Application, to interface with external programs such as PowerPoint or Virtual Battle Space (VBS); and (e) Branch Point, which configures the Engine for Management of Adaptive Pedagogy (EMAP). Transitions can be added in sequence, inserted into an existing course, or re-ordered using a familiar drag-and-drop technique (*Figure 2*).

Many of the transitions are configured using standard, familiar form elements (e.g., drop-down menus, radio buttons, and checkboxes) and guided dialogs (e.g., wizards). The use of these elements is intended to reduce the potential for user error (the original version of GIFT relied on properly written XML code), and increase the learnability of the tools by presenting possible configuration options for each transition for recognition (rather than recall) by the author. The usability of the interface is also supported by the addition of help and tooltips at the point-of-need, triggered by mouse-over or clicking help icons located throughout the interface.

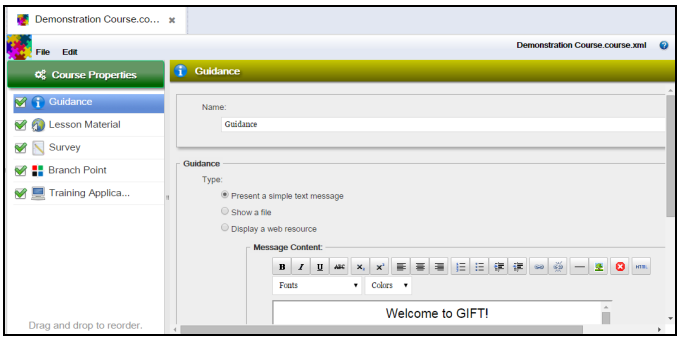

Figure 2: Building courses with transitions (left); configuration of a Guidance message (right)

Creating Surveys in GIFT

The UI for creating surveys and question banks is also web-enabled. The survey authoring system supports a variety of question and response types. A pop-up window guides the user through creating individual questions using drop-down menus with a dynamic display of available options for each question type. As a question is created, a preview of the item is displayed to the author in order to verify the presentation of the question to the learner in an efficient manner.

The basic structure of the survey authoring system consists of three parts, organized into tabs within the interface (*Figure 3*). *Questions* make up the content of *Surveys* and/or *Question Banks*. Collections of Surveys make up a *Survey Context.* The Survey Context is the object that links surveys to individual courses. The design intent of this implementation is the reuse of questions and surveys. Response sets (e.g., True / False) can also be reused within the survey authoring system to aid in the efficiency of question creation. This design also enables search and filtering options within the interface.

At present, the survey authoring system is within a visually distinct interface from the other authoring tools. However, this legacy system does communicate information to the core authoring tools in order to populate relevant fields with the names of surveys and survey contexts. Tighter integration of the survey authoring system within the new authoring tools is planned for a future version of GIFT Cloud. For instance, it is expected that authoring efficiency will be improved with the ability to author survey materials directly

within the core authoring tools without switching to a visually separate and different survey authoring system.

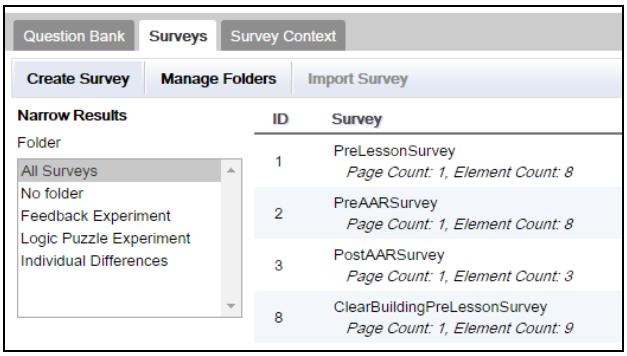

Figure 3: Survey authoring system with Surveys tab active

EMAP: The Pedagogical Engine

The Engine for Management of Adaptive Pedagogy (EMAP) is the default learning engine behind GIFT (Goldberg et al., 2012). Its structure is based on David Merrill's (1983) Component Display Theory (CDT), and organizes learning around four fundamentals categories: learning *Rules* of a domain, seeing *Examples* of the rules applied, *Recalling* declarative and procedural information associated with the rules, and *Practicing* the application of those rules in a novel context. The four categories of CDT are encapsulated within GIFT's authoring workflow as a *Branch Point* transition, where content, questions, problems / scenarios, and pedagogical practices are configured for automatic selection and execution by GIFT during course runtime.

The UI is logically organized around those four categories (*Figure 4*). The available concepts to be taught are pulled automatically from the course properties configuration panel. The Rule and Example quadrants are populated with media content, which is selected using a familiar file-browser interface.

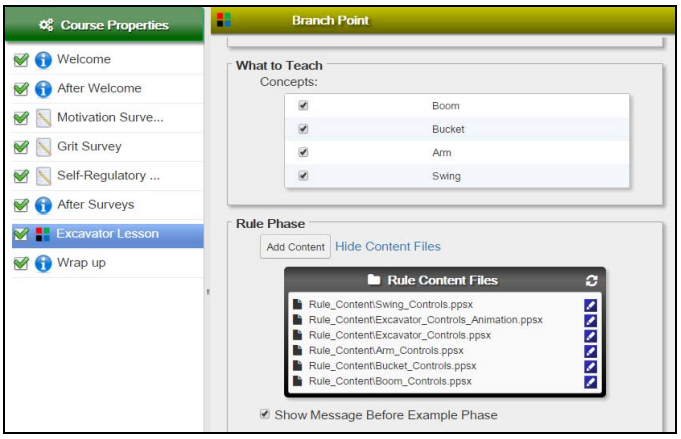

Figure 4: Organization of Branch Point UI

The recall branch (*Figure 5*) allows an author to use an existing survey, or configure a dynamic assessment from a Question Bank based on question difficultly. Criteria for establishing an expertise level (i.e., novice, journeyman, expert) is also configured in this section. Each element is visually blocked with different color headings to separate the similar options for each expertise level.

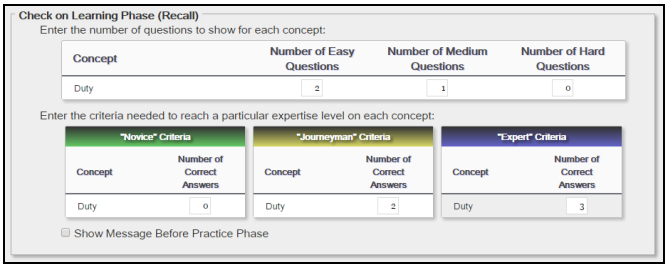

Figure 5: Configuration of the Recall quadrant

Dynamic real-time assessment with applications

In addition to the delivery of media content in the Rule and Example quadrants, GIFT Cloud can leverage external applications through either the Practice quadrant of the Branch Point transition or the Training Application transition (*Figure 6*). Dynamic assessment with external applications represents the inner-loop of tutoring within GIFT. During course run-time, a lightweight Java applet transmits simulator environment information to the tutoring system for real-time assessment, remediation, and feedback.

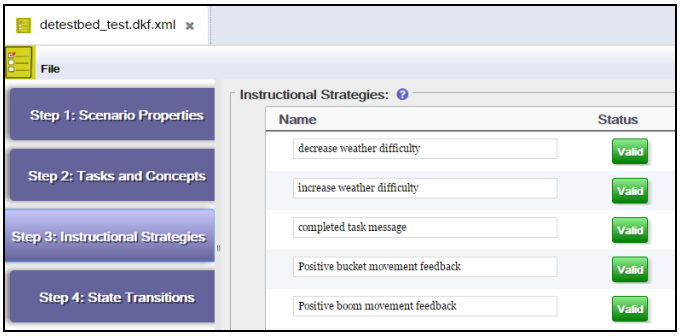

Figure 6: Authoring tool for dynamic assessments

Authoring real-time assessment with external applications (e.g., simulations) is arguably one of the more complex tasks within GIFT. In order to begin to increase the usability of this particular authoring tool, a number of considerations have been made within the UI. The workflow is represented as a series of four major steps. Information clutter is reduced by only displaying form elements to the author based on the options selected (non-selected elements are hidden until activated). Where relevant, portions of the UI containing advanced options are also available, but partially hidden (i.e., collapsed). Finally, some real-time validation of certain fields also takes place in this interface using text plus color coded boxes to alert the author to where attention is needed.

Support for Experimentation

The Experiments UI may be of particular interest to the human factors community. While not a tutor authoring tool, per se, the Experiments UI augments the functionality of

authoring. It also represents a way in which user-centered design is driving the creation of new functions and interfaces.

These tools were originally created out of the need to determine the efficacy of the various portions of domain, learner, or pedagogical modeling approaches. However, a web-based study can be created from any existing GIFT course for any purpose. When an "experiment" is created, GIFT creates a locked copy of the source course to prevent future modification and ensure experimental control. The UI then generates a unique access link for the experiment (*Figure 7*). Participants anonymously access the study using the link, without a GIFT account, supporting data anonymity. The researcher can pause and resume access to the study in order to comply with data collection timelines. Finally, tools are available which semi-automatically aid the researcher in downloading participant data and compiling custom reports that include GIFT log files and other session data.

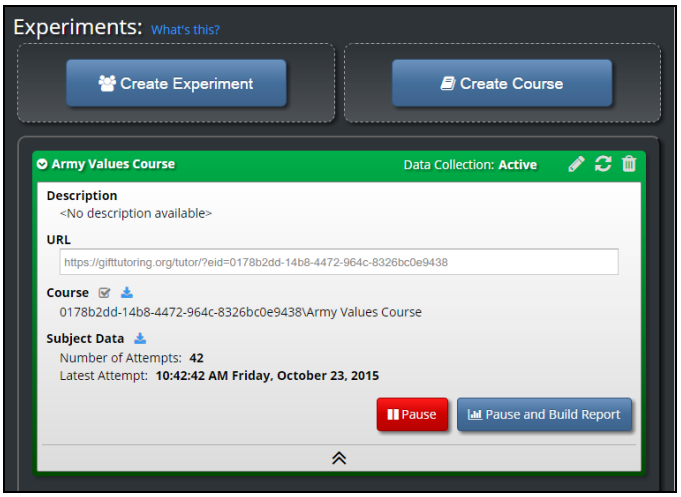

Figure 7: Information panel with the Experiment UI

FINDINGS

Each release of GIFT provides an opportunity to gather user insights and feedback that are leveraged in future iterations of GIFT, including the web-based GIFT Cloud application. While positive and significant progress has been made with respect to the usability of the authoring tools in the alpha release of GIFT Cloud, this web-based version of GIFT created further opportunities for design improvements.

Preliminary feedback regarding GIFT Cloud has revealed that, while authoring within individual transition elements can be relatively intuitive, the *overall* authoring workflow in GIFT is not intuitive. Perhaps it is unclear because the interface does not strongly suggest a specific workflow to the author; and/or, perhaps authors are unfamiliar with creating ITSs in general, therefore lacking an appropriate general mental model to apply to this specific set of authoring tools. Further investigation into authors' understanding of ITS as well as their understanding of relationships between transitions and other GIFT objects is needed to guide future improvements.

A related finding suggested that support for collaborative authoring can increase the effectiveness of GIFT Cloud. It is an anticipated that *teams* of authors will be creating tutors, which may include instructional designers, content mangers, and subject matter experts. Collaborative authoring features would leverage the authoring responsibilities of those roles, and consider the differences in users' mental models of authoring. With this information, the usability of the authoring tools might be improved with default views and specialized tools based on each authoring role.

Feedback also revealed that there are opportunities to reduce excise within the system. This is to be somewhat expected with an alpha release of an application. For example, the amount of physical effort required in mouse movement, mouse-clicks, etc. can be reduced when creating individual questions within the survey authoring system. Reducing the time to author an individual survey question by even a few seconds can increase the efficiency of authoring overall, across the number of questions authored for tutor assessments. Additionally, reducing excise throughout the system should increase user satisfaction with the authoring tools, in general.

Tasks such as authoring simulation-based real-time assessments remain inherently difficult in comparison to other authoring tasks. While the current design aids the author, for instance, in organizing information and preventing errors, there exists an opportunity to improve the learnability of the task through the design of the interface. This includes imparting to the author what the capabilities of the tool are, as well as how authors can translate their goals into actions within these powerful sections of the authoring system.

Finally, feedback from the community has identified the need for continued work on *help* and related supporting materials. The current release of GIFT Cloud provides on-demand help in the form of tool-tips and function descriptions. Future situated aids might include semi-transparent overlays that would be shown upon first accessing a particular tool, or tutorial videos that demonstrate capabilities of tools.

DISCUSSION

Interacting with any aspect of GIFT Cloud, including the authoring tools, documentation, forums, etc. should endeavor to build knowledge, confidence, and trust in user's ability to plan and execute authoring goals within the GIFT Cloud authoring system. To that end, future research will expand the concept of usability to a comprehensive user experience for GIFT Cloud. We are investigating the best methods for cultivating authoring knowledge within the user-community which include tutorial videos, dynamic web-based help, and highly polished demonstration courses that can be inspected and deconstructed for use in the creation of new tutors.

In the current version of the authoring tools, much of the ontology of the GIFT architecture is exposed to authors. This may be desirable for expert authors, but may be confusing or intimidating to novice authors. Therefore, efforts to help bridge the gap between GIFT's system conceptual model and users' mental models of authoring are underway. Those efforts are expected to result in the design of tools that align more closely with authors' understanding of and expectations for completing authoring tasks within GIFT Cloud. This also includes building course templates and increasing the availability of configuration wizards within the system.

Another way to tackle the complexity of authoring a tutor is by exploring different techniques for representing the structure of a tutor within the authoring tools. This consideration is not unique to GIFT; Murray, for instance, has described the problem space in detail over a series of publications (Murray, 2014, 2015; Murray, Woolf, & Marshall, 2004). GIFT currently presents tutor elements as a list that can be re-ordered. Flowcharts and event-based model structures are being investigated for their viability in representing the structure of tutors created within the GIFT platform (*Figure 8*). Such structures may aid authors in understanding course creation from a high-level perspective, and can also enable more efficient navigation between objects within a given tutor.

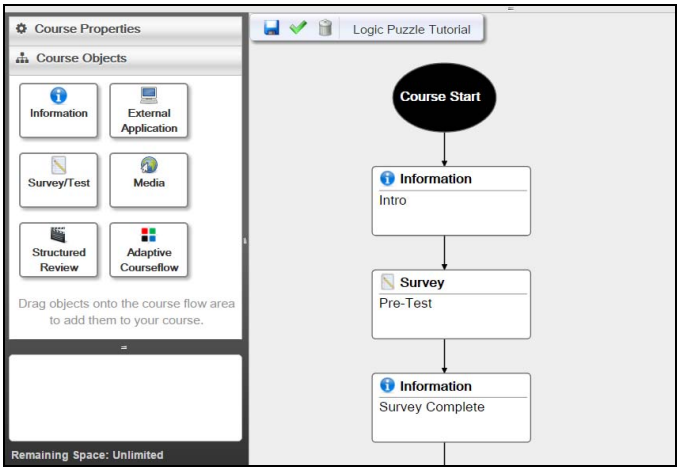

Figure 8: Prototype of new flowchart UI for course authoring

Finally, there is value in examining development tools in other complex software domains, such as game development tools (Lightbown, 2015). Games are similarly complex and dynamic software products that require powerful tools. In that industry, saving time and resources (e.g., budget, human-power) in development are of critical importance. Game development tools are widely used, have community support, and are in a mature state. Case studies and lessons learned from game development tools can provide design inspiration to future iterations of the GIFT Cloud web-application.

CONCLUSION

One important design goal of the authoring tools associated with GIFT is to allow authors to create adaptive tutors without any knowledge of computer programming or instructional design. Usable interfaces are on the critical path to this goal; specifically, interfaces that promote system learnability, and allow users to create tutoring content in an efficient manner. GIFT Cloud provides an important step toward that goal, by increasing the availability of the authoring tools and enabling the ability to more rapidly design and implement interfaces that are informed by user feedback. Design for usability will continue to be an iterative effort, in parallel with the development and improvement of system functions, for future versions of the web-based GIFT Cloud application and the GIFT platform, in general. More about the GIFT project can be found at www.gifttutoring.org.

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/

Goldberg, B., Brawner, K., Sottilare, R. A., Tarr, R., Billings, D. R., & Malone, N. (2012). Use of evidence-based strategies to enhance the extensibility of adaptive tutoring technologies. *The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC)*.

Lightbown, D. (2015). *Designing the User Experience of Game Development Tools*. Boca Raton, FL: CRC Press.

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. (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

Murray, T. (2015). Coordinating the Complexity of Tools, Tasks, and Users: On Theory-based Approaches to Authoring Tool Usability. *International Journal of Artificial Intelligence in Education*, 1-35.

Murray, T., Woolf, B., & Marshall, D. (2004). Lessons learned from authoring for inquiry learning: A tale of authoring tool evolution. In J. C. Lester, R. M. Vicari & F. Paraguacu (Eds.), *Intelligent Tutoring Systems 2004, LNCS 3220* (pp. 197-206). Heidelberg, Germany: Springer.

Nielsen, J. (1995). 10 Usability Heuristics for User Interface Design Retrieved 27-Feb-2015, from http://www.nngroup.com/articles/ten-usability-heuristics/

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., & Sottilare, R. A. (2016). *Heuristic evaluation of GIFT authoring tools*. Manuscript submitted for publication.

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.

Sinatra, A. M., Goldberg, B. S., & Sottilare, R. A. (2014). The Generalized Intelligent Framework for Tutoring (GIFT) as a tool for human factors professionals. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting 58*, 1024-1027.

Sottilare, R. A., Brawner, K., Giammaria, C., Mucke, K., Sohn, A., & Matthews, M. D. (2014). *An evaluation of the Generalized Intelligent Framework for Tutoring (GIFT) from an author's perspective*. (ARL-SR-0303). U.S. Army Research Laboratory.

Sottilare, R. A., Graesser, A. C., Hu, X., & Holden, H. (2013). *Design Recommendations for Intelligent Tutoring Systems: Volume 1 Learner Modeling*. Orlando: U.S. Army Research Laboratory.

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

Woolf, B. P. (2009). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Burlington, MA: Morgan Kaufmann.