Defining Complexity in the Authoring Process for Adaptive Instruction

Robert Sottilare[1] & Scott Ososky[1,2]

[1]U.S. Army Research Laboratory, Orlando, FL USA
[2]Oak Ridge Associated Universities, Orlando, FL USA

robert.a.sottilare.civ@mail.mil, scott.j.ososky.ctr@mail.mil

Abstract. Adaptive instruction is computer-based training or education that is tailored to match the difficulty of the content to the states and traits of the learner. Since the individual differences of learners vary widely and contribute greatly to the adaptation decisions by the tutor, adaptive instructional systems (e.g., Intelligent Tutoring Systems – ITSs) need much more content and make many more instructional decisions than non-adaptive instructional systems that instruct all learners only based on their performance level (e.g., low, moderate, high) using identical instructional strategies. Since the authoring of adaptive instruction varies with the complexity of its content and instructional decisions, it is difficult to compare the efficiency of the adaptive instructional authoring tools and methods, and the effort and skill required to use them in the construction of ITSs. This paper puts forth a methodology to assess ITS complexity and operationalize it in an index to enable adaptive instructional scientists to compare authoring tools and methods. The baseline for this initial comparative index is the Generalized Intelligent Framework for Tutoring (GIFT) authoring tools.

Keywords: Adaptive Instruction, Intelligent Tutoring Systems, Generalized Intelligent Framework for Tutoring (GIFT), Authoring Tools

1 Introduction

Adaptive tutors, also known as Intelligent Tutoring Systems (ITSs), deliver instructional (training or educational) content to individual learners or teams of learners that is tailored to match the capabilities, states, and traits of each learner [1]. Adaptive instruction guides the learner(s) based on their individual differences with the goal of optimizing learning, performance, retention, and transfer of skills from instruction to work/operational environments [2]. Since the individual differences of learners vary widely and contribute greatly to the adaptation decisions by the tutor, adaptive instructional systems (e.g., Intelligent Tutoring Systems – ITSs) need much more content and make many more instructional decisions than non-adaptive instructional systems (e.g., computer-based training systems) that instruct all learners and only adapt content and flow based on their performance level (e.g., low, moderate, high).

adfa, p. 1, 2011.

Since the authoring of adaptive instruction varies with the complexity of its content and instructional decisions, it is difficult to compare the efficiency of the adaptive instructional authoring tools and methods, and the effort and skill required to use them in the construction of ITSs. This paper puts forth a methodology to assess ITS complexity and codify it in an index to enable adaptive instructional scientists to compare authoring tools and methods. The baseline for this initial comparative index is the Generalized Intelligent Framework for Tutoring (GIFT) authoring tools [3].

Often, ITS authoring is cited as taking X number of hours to produce one hour of adaptive instruction. The differences in domain complexity, authoring tool usability and author competency make it difficult to compare the efficiency of one ITS toolset versus another. Currently, we have no method to compare the efficiency of processes for ITSs developed with the same authoring toolset.

For example, a three-bedroom house and a skyscraper are both buildings, but the amount of material, skill, and effort to construct them is significantly different. If we wanted to understand and compare the skills and efficiency of two builders, one that built the house and the other the skyscraper, it would be difficult without some measure or index (e.g., building rate). In order to compare the complexity of authoring tasks for building one ITS to another, we need to define what contributes to authoring complexity and establish an index of authoring. In this way, we can compare one authoring task to another, and the performance of one set of authoring tools fairly and objectively to another.

We chose to examine both the performance and skill of the author as well as the complexity of any ITS examined by our index. This was done to be able to compare the effectiveness of authoring tools for low, moderate, and highly skilled authors. As basic measures of ITS authoring performance, we considered efficiency, the time or rate of progress, and effectiveness which involves assessment of the quality of the resulting ITS in terms of the ratio of increases in learning to time on task. For example, a tutor that averages increases in knowledge and skill of 50% and 20% respectively is twice as effective as one that averages increases of 25% and 10% for learners spending the same amount of time in training.

To understand the complexity of any ITS that might be authored, we examined three contributing factors: 1) task complexity, 2) tutor authoring tool usability, and 3) author competency and interaction. We examined these factors in relationship to the Generalized Intelligent Framework for Tutoring (GIFT) authoring process to provide context, but these principles could be applied to any ITS authoring system. Factors that contribute to increased task complexity include: the number of concepts or learning objectives; the amount and diversity of content (including media and surveys) required/curated/created/used in the instruction; and the number of assessments, decisions or adaptations required during the instruction.

Factors which decrease complexity include: automation and ease of use. What parts of the authoring process can be handled by an artificially intelligent method? Is it clear to the author what the authoring process is and should be done next? The competency of the learner is inversely related to their time on task or contact time with the ITS, and therefore influences the perception of the ITSs effectiveness. Other learner behaviors have variable influence on the effectiveness of the ITS. For example, off-task behaviors (e.g., doing other than what they should be doing, sleeping, or daydreaming) negatively impact ITS effectiveness while learner familiarity and confidence with the instructional environment have positive effects. An examination of the GIFT authoring tools and processes reveals how task complexity, tutor usability, and learner competency and interaction might be used to define ITS authoring complexity. This paper puts forward a concept for assessing ITS complexity as a method to compare the effectiveness of ITS authoring systems.

2 Examining Task Complexity in the ITS Authoring Process

As part of our quest to define complexity in the ITS authoring process, we begin by examining elements of task complexity. Our goal is to provide a practical method for defining ITS complexity within GIFT and this begins with concepts or learning objectives. GIFT represents concepts to be learned as either a list (i.e., no hierarchical relationship) or a hierarchy as shown in Fig. **1**.

In GIFT, the lowest level in the hierarchy of concepts (shown in Figure 1 as leaves) 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 complexity as it requires the development of an assessment (e.g., knowledge, skill test, or real-time assessment coupled with an external environment).

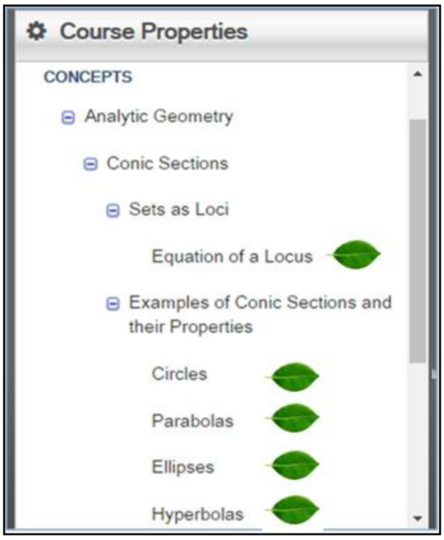

Fig. 1. Hierarchical Concepts in GIFT

In addition to concept-specific content, there may be amplifying content/media that is presented to the learner to provide context or background as shown in Figure 2. The complexity of the authoring task varies based on the amount of content that must authored/found/retrieved/implemented in GIFT.

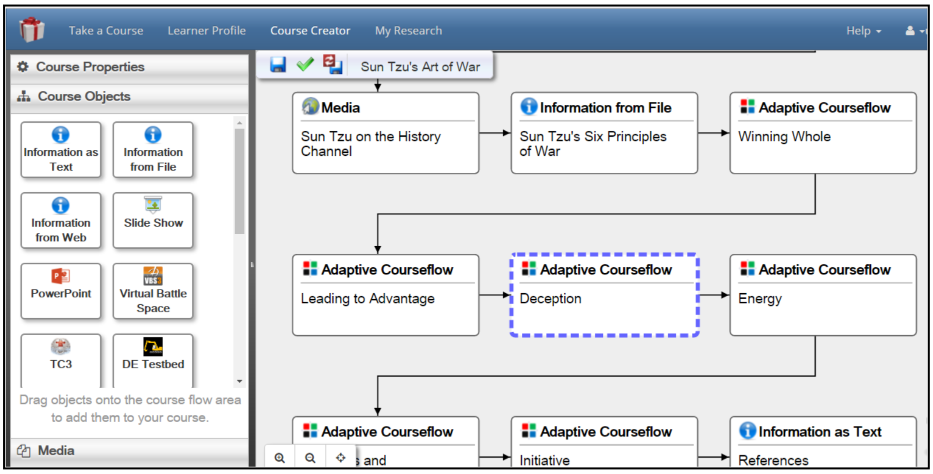

Fig. 2. Adaptive Course flow and Course Objects for a GIFT course on Sun Tzu's Art of War

In GIFT's *adaptive courseflow* object, each concept must be tied to content presented to the learner as part of Merrill's component display theory (CDT) [4] 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).

Within the rules and examples phases of the adaptive courseflow object GIFT delivers content (e.g., media, presentations, audio, text) to the learner to support the acquisition of knowledge. For the recall quadrant, GIFT assesses domain knowledge and for the practice quadrant, both knowledge and skill may be assessed as part of an interactive experience (e.g., simulation, serious game). Figure 3 provides details for the recall phase or check on learning. For the concept called *deception*, the learner's knowledge of information presented in the rules and examples phases is assessed through random selection of questions from a question bank of 25 questions in which 2 each (easy, medium, and hard) are presented to the learner. GIFT may also have a fixed survey/test in which the author only generates the number and difficulty level of the questions needed. Either way, the complexity of authoring is tied to the number of assessment questions generated.

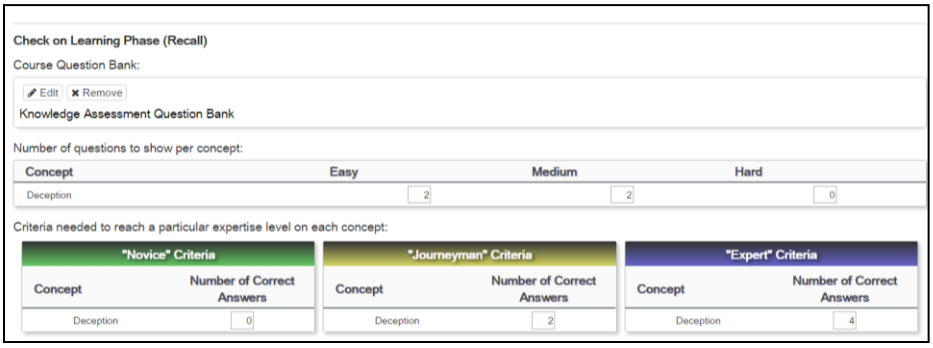

Fig. 3. Details of Adaptive Courseflow Object as part of GIFT Authoring Process

In the practice quadrant, GIFT is capable of using either an existing environment, one already integrated though the GIFT gateway or the author will need to develop a new gateway interop to support the exchange of information between GIFT and the application. Reuse of the already integrated applications reduces the authoring burden, but is limited to about 6 publicly available practice environments at the time of this publication.

Each practice environment will require a real-time assessment which includes four steps to be completed by the author to define: 1) scenario properties, 2) tasks and concepts (Figure 4), 3) instructional strategies (Figure 5), and 4) state transitions (Figure 6) as shown below for a virtual excavator trainer.

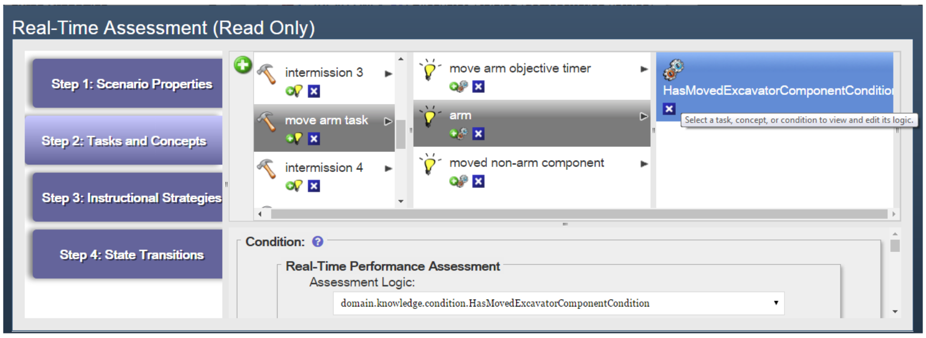

Fig. 4. Defining Tasks and Concepts for Real-time Assessment in External Environments

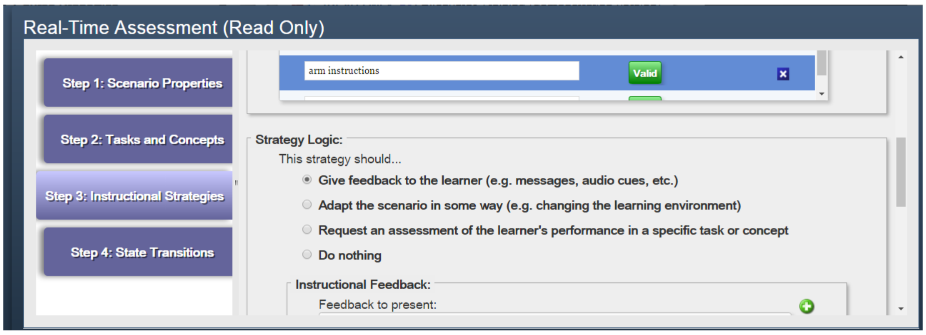

Fig. 5. Defining Instructional Strategies in response to Real-time Assessment in External Environments

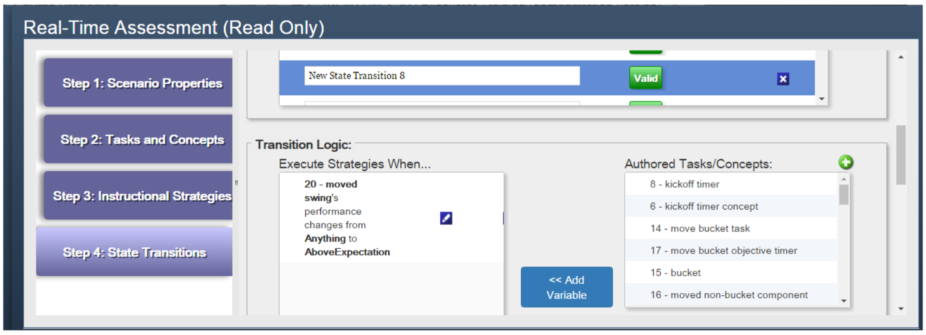

Fig. 6. Defining State Transitions during Real-time Assessment in External Environments

3 Examining ITS Authoring Complexity

Authoring complexity is largely a function of the type of tutor that needs to be produced, with respect to dimensions like those described in the previous sections. Authoring tools, then, are productivity applications that aid a developer (or team of developers) in the creation of tutors. If all authoring tools were created equally, more complex tutors will require more time and effort to create than less complex tutors. Therefore, authoring tools seek to reduce the time and effort required to develop tutors, through functions that provide various levels of automated support in organizing content, setting conditions for assessments, configuring adaptations, and so on.

While the core models / modules of adaptive training systems [5] are well established, the functions with which tutors are built and the modes through which they can be delivered is still evolving. Authoring tools, therefore, are also evolving with ITS platforms. That creates difficulty in establishing a generalized model of authoring complexity, as individual authoring tools can differ widely from one another [6-9].

Murray has written that that authoring tools "are highly complex educational software applications used to produce highly complex software applications" [10]. His work described challenges in developing tutor authoring tools with respect to design tradeoffs between *usability*, *depth*, and *flexibility* [11, 12]. 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 [13]. Those characteristics provide a suitable reference for examining authoring complexity through the lens of the authoring tools, using GIFT as our reference baseline.

3.1 Authoring Tool Usability

The GIFT authoring tools have evolved the task of authoring a tutor from the direct manipulation of extensible markup language (XML) code to an object oriented visual interface similar to those associated with developing discrete event simulation models. Redesigning GIFT's authoring tools were motivated, in part, by the desire to allow users to create tutors without requiring specific knowledge of instructional design or computer programming. Further, mental model theory served as one of the core principles of newer user-centered interface designs in two important ways: 1. Prior versions of GIFT authoring tools were structured very closely to the system conceptual model, creating a burden on the author to configure system variables that were tangential to the task of creating a tutor [14]; and 2. There simply isn't another productivity or content creation task that is a suitable and/or complete analogy for tutor authoring, however current designs leveraged familiar interaction patterns and thoughtful interface representations from various productivity applications in order to assist the user in developing accurate mental models of the authoring process [15].

The usability of authoring tools differently benefits users of various skill levels and experience. The usability authoring tool characteristic might be further divided into *learnability* and *efficiency*. For novice users, authoring tools must be learnable, GIFT has an opportunity to improve upon learnability by making it easier for new authors to figure out what to do first/or next. Likewise, authoring tools must not be intimidating or frustrating to the point where a user gives up (a point at which subjective authoring complexity supersedes objective complexity). The GIFT authoring tools support that notion by displaying only the most common and straightforward functions by default, a technique referred to as *progressive disclosure* [16]. Authors only see what they need to see, and can explore further into the interface as they become more comfortable, or require advanced functionality.

Progressive disclosure is also related to authoring efficiency, which is beneficial for author of all skill levels. Spending less time scanning an interface for a specific option allocates more effort toward the actual authoring task. Lightbown [16] also noted that a balance must be found between progressive disclosure and *excise*, which is the physical effort involved in using the interface (e.g., mouse movement, clicks, visually scanning the UI). As an example, the latest update to GIFT's authoring interface includes an improved survey editing experience, which reduces the amount of physical effort required to quickly create questions.

Finally, authoring tool usability feeds into the larger notion of an authoring user experience (UX), which seeks to support the author from concept to deployment. In this area we consider elements external to the authoring tools including publications and documentation, course management / organization, and community building through forums and face-to-face meetings [17, 18].

Depth and flexibility are the other two aspects of the authoring tool design tradeoff space. Those might collectively be referred to as *authoring tool complexity*. It is the position of one of the authors of this paper that separation can be placed between authoring tool complexity and usability [17]. Further, complex authoring tools can be usable, given a thoughtful focus on usability in a way that does not provide a detriment to depth and flexibility. For what good is depth and flexibility if no one can (or wants to) use the authoring tools?

3.2 Authoring Tool Depth

Depth in authoring tools refers to the "structural or casual depth of any of the ITS modules" [11]. Depth varies within the authoring tools for each of the various modules of GIFT. Within the learner module, GIFT can track a variety of affective (e.g., anxiety, arousal) and performance based variables. These are configurable through the authoring tool user interface, but GIFT provides default configurations for those variables, as well as the logic with which to interpret them within the pedagogical module. The variables tracked within the learner module, as well as the logic contained with the pedagogical module, was determined by research and literature review. The author does not have to edit these modules, unless they wish to do so.

The domain module offers, perhaps, the greatest depth within the GIFT authoring tools. GIFT currently uses two primary mechanisms for adaptation. The first is through the adaptive courseflow object (Figure 3, above), a discrete-time adaptation capability based on a learner's proficiency within the object. The depth of this experience can quickly expand, given even a modest number of concepts to assess, and learning the content required to sufficiently populate the adaptations for this object. The second core mechanism for adaptation within GIFT is the real-time assessment engine most commonly associated with external training applications or sensors. This aspect of GIFT is highly configurable for a variety of external applications, subject to the data that is able to be exchanged between GIFT and the application, as well as the variables that the author wishes to assess (see Figures 4-6, above).

Finally, GIFT provides variable depth within the authoring tools for the tutor user interface (TUI) module (Figure 7). GIFT supports a variety of different learning content types including web-based and local resources (e.g., PDF, PowerPoint shows). The author can also construct different types of interactions using virtual avatars, and branching conversations. The overall look and feel of the TUI (to the learner) is not currently directly configurable; however, authors can use built-in rich-text editing tools or write custom HTML to add styling to individual elements within a course.

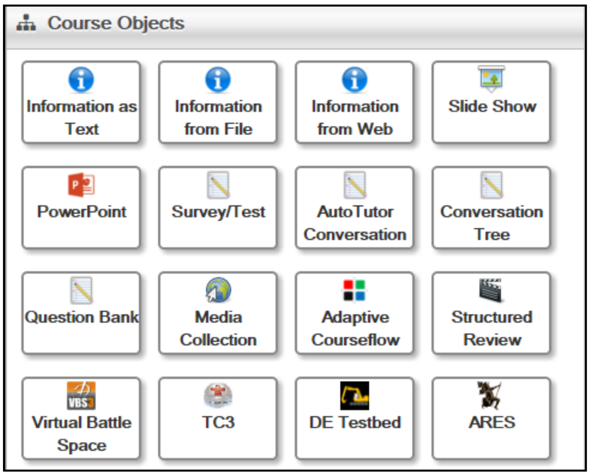

Fig. 7. Part of GIFT's growing list of course objects. Each is highly configurable, providing depth to how content can be displayed to the learner through the tutor-user interface.

3.3 Authoring Tool Flexibility

Flexibility refers to the ability to "author a diversity of types of" tutors [11]. This characteristic of tutor authoring tools may be the most difficult to quantify, especially in the case of GIFT. GIFT was built to be domain independent, meaning that the same authoring tools could be used to generate tutors for cognitive tasks, psychomotor tasks and so on. The GIFT authoring tools provide a number of intelligent defaults; however, it is possible to create new features in support of creating new types of tutors. Some of these manual options are natively available within the current authoring tool interfaces (e.g., altering learner or pedagogical models), while others may require some external development (e.g., interfacing with a new physiological sensor).

GIFT is an open-source platform, meaning that developers can extend the functionality of GIFT to accommodate new training applications, deliver tutors through different platforms (e.g., virtual and augmented reality), or build new pedagogy based on a preferred learning theory. From that perspective, GIFT is highly flexible. The caveat, however, is that a developer would also need to create authoring tool interfaces (as opposed to hard-coded solutions) to support those enhancements.

Given the rapid pace at which tutors, in general, continue to evolve, flexibility may be most at odds with usability. It takes time, resources, and testing to develop new tutor functionality. Often, the actual authoring tool supporting the use of the new functionality is one of the last pieces to fall in place, because the tool cannot be truly finished until all of the configurable parameters are known.. From a usability perspective, support material and error-prevention measures cannot be established until the limits of the new functionality is well understood. That being said, GIFT continues to expand to meet new challenges across the ITS waterfront. Once a new function is developed and integrated into the baseline, that functionality becomes available to all users. The most direct example of this is GIFT's interoperability with external applications. Some back-end development is required in order to establish a communication gateway between the two systems. However, because GIFT is modular, additional development in other modules would not be necessary. GIFT Components like these are also re-usable once this is done, thus expanding the overall flexibility of the authoring tools for all GIFT users.

4 Examining Author Competency

As part of our examination of complexity in the authoring process, we discuss the impact of author competency on the ITS authoring process in GIFT. While the author's familiarity and expertise influence the time needed to develop a tutor using GIFT or other authoring tools, the author's competency does not affect the complexity of the tutor. We acknowledge that some factor may be needed to accurately compare the authoring process for a novice, journeyman, and expert developing the same tutoring content with the same authoring tools, our understanding of what this factor might be has no empirical basis yet.

5 Applying a Comparative Index to the GIFT Authoring Tools

Specifically, within the GIFT authoring, we have identified several variables contributing to the complexity of building ITSs. We make the assumption that content curation is an integrated part of the authoring process. In other words, we don't break out the effort to find, retrieve, and organize content. We assume this must be done with all content. So, if you need content, you must curate it, but you might not build it from scratch. 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 (or learning objectives) designated "TC" and associated leaf nodes designated "LN". Leaf nodes within a hierarchical or non-hierarchical set of concepts (learning objectives) are nodes without children.

Assuming each major concept requires an adaptive courseflow object, the author is responsible to curate content for use in all phases of learning (rules, examples, recall, and practice). This might be done manually or with the use of curation tools. As we review the application of our model of complexity with respect to GIFT authoring, we will refer to examples provided previously in figures above. In Figure 1, we show a hierarchical set of concepts for analytic geometry in GIFT. A tutor built around these concepts would have TC = 9 and LN = 5. While each concept requires content, not all concepts require assessments. Typically, the lowest levels of concept, LNs are where assessments are authored (e.g., surveys, real-time assessments).

We chose to simplify the modeling of content development complexity because the complexity of content development processes varies so widely. For example, building a slide for a presentation is a much different task from building a three-dimensional, interactive, immersive virtual simulation. To simplify our model, we chose to use interactive multimedia instruction (IMI) levels [19] already defined in

GIFT as a meta-data element of our ITS authoring complexity model. The IMI schema in GIFT is described by four levels as shown below in Table 1.

Table 1. Interactive Multimedia Instruction (IMI) Levels.

- Level 1 – low interaction and low user control: primarily passive with minimal action required by the learner
- Level 2 - limited interaction, low user control: some recall required
- Level 3 – significant interaction and moderate user control: primarily requires learner to make decisions, solve problems or interpret results
- Level 4 – full interaction and user control: real-time interaction and responses to complex cues; learner required to demonstrate specific skills with measurable results

The following factors were identified through our review process as required to define ITS complexity in GIFT:

- TC = total number of concepts defined by the author
- LN = Leaf Nodes = total number of concepts without children which require assessments
- CDT = Component Display Theory Phases = usually four (rules, examples, recall and practice)
- W, X, Y, Z = number of separate pieces of content for each CDT phase respectively under a given concept
- IMI = IMI Level of Content

Based on these factors, we determined that ITS complexity is the sum of the complexity for each of the concepts (Equation 1), and the complexity for each concept is dependent on the number of pieces of content available to the learner for each CDT phase and the IMI level of each piece of content. For the recall phase, the number of pieces of content developed is equal to the number of questions in the question bank, survey, or check-on-learning. This led us to formulate the following equations:

$$\text{ITS Complexity} = \sum_{a=1}^{a=TC} \text{Complexity Concept}_i \quad (1)$$

$$\text{ITS Complexity} = \left.\begin{aligned}&\sum_{a=1}^{a=TC} \Big(\sum_{b=1}^{b=W} (\text{IMI for Rules}_{ab}) + \sum_{c=1}^{c=X} (\text{IMI for Examples}_{ac}) \\ &+ \sum_{d=1}^{d=Y} (\text{IMI for Recall}_{ad}) + \sum_{e=1}^{e=Z} (\text{IMI for Practice}_{ae}) \Big) + \text{LN}\end{aligned}\right\} \quad (2)$$

$$\text{where} \left\{\begin{aligned}&\text{Rules}_{ab} = \text{content \#b for Concept}_a \text{ in the Rules Phase} \\ &\text{Examples}_{ac} = \text{content \#c for Concept}_a \text{ in the Examples Phase} \\ &\text{Recall}_{ad} = \text{content \#d for Concept}_a \text{ in the Recall Phase} \\ &\text{Practice}_{ae} = \text{content \#e for Concept}_a \text{ in the Practice Phase}\end{aligned}\right.$$

6 Next Steps

In this paper we have highlighted three primary components that influence the authoring process: the tutor itself, the authoring tools used to build the tutor, and the competency of the author. We have discussed, mostly qualitatively, what factors into each of those areas, and have started to identify how each of these aspects affects one another. However, continued work is needed to formalize this logic into a quantifiable comparison metric, and to refine such a metric through research and case studies.

Regarding authoring tools specifically, more work is needed in service of quantifying their value. We used the characteristics of usability, depth, and flexibility to differentiate authoring tools from one another, but it is likely that those three areas can be further subdivided to offer greater detail to our metrics. Usability, for instance, includes learnability and efficiency. Depth and flexibility might be further categorized at the ITS model level. Determining how to segment authoring tool properties, and determining the relative importance of each could yield a useful taxonomy. Such a taxonomy could be used to compare authoring tools directly to one another, or even serve as a blueprint for identifying opportunities for future authoring tool development.

Regarding, complexity across the entire authoring process, future work should continue to refine the models for relationships between pairs of components, along a theoretical X-Y axis. For example, there may be points at which an author's competency is high enough that a more usable set of authoring tools offer little additional benefit. Likewise, different sets of authoring tools may seem relatively comparable with simple tutors, but those same platforms may actually be vastly different in efficiency with more complex tutors. Identifying the critical features of these relationships may help to determine where the best effort can be put forth in future research into the complexity of ITS development.

7 References

1. Sottilare, R. (2014). Fundamentals of Adaptive Intelligent Tutoring Systems for Self-Regulated Learning. In Proceedings of the *Interservice/Industry Training Simulation & Education Conference*, Orlando, Florida, December 2014.
2. Sottilare, R.: Elements of a Learning Effect Model to Support an Adaptive Instructional Framework. In R. Sottilare & S. Ososky (Eds., 2016) 4th Annual GIFT Users Symposium (GIFTSym4). Army Research Laboratory, Orlando, Florida, (2016). ISBN: 978-0-9977257-0-4.
3. Sottilare, R.A., Brawner, K.W., Goldberg, B.S. & Holden, H.K. The Generalized Intelligent Framework for Tutoring (GIFT). Orlando, FL: U.S. Army Research Laboratory – Human Research & Engineering Directorate (ARL-HRED), 2012.
4. Merrill, M. D. *The descriptive component display theory*: Educational Technology Publications, Englewood Cliffs, NJ., 1994.
5. Woolf, B.P.: Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning. Morgan Kaufmann, Burlington, MA (2009)
6. Olsen, J.K., Belenky, D.M., Aleven, V., Rummel, N.: Intelligent tutoring systems for collaborative learning: Enhancements to authoring tools. In: Artificial Intelligence in Education, pp. 900-903. Springer, (2013)
7. Aleven, V., Sewall, J.: Hands-on introduction to creating intelligent tutoring systems without programming using the cognitive tutor authoring tools (CTAT). In: Proceedings of the 9th International Conference of the Learning Sciences-Volume 2, pp. 511-512. International Society of the Learning Sciences, (2010)
8. Suraweera, P., Mitrovic, A., Martin, B.: Widening the knowledge acquisition bottleneck for constraint-based tutors. International Journal of Artificial Intelligence in Education (IJAIED) 20, 137-173 (2010)
9. Mitrovic, A., Martin, B., Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N.: ASPIRE: an authoring system and deployment environment for constraint-based tutors. International Journal of Artificial Intelligence in Education 19, 155-188 (2009)
10. Murray, T.: 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 (2015)
11. Murray, T.: Theory-based authoring tool design: considering the complexity of tasks and mental models. In: Sottilare, R.A., Graesser, A.C., Hu, X., Brawner, K. (eds.) Design Recommendations for Intelligent Tutoring Systems, vol. 3 Authoring Tools and Expert Modeling Techniques, pp. 9-29. U.S. Army Research Laboratory Orlando, FL (2014)
12. Murray, T.: Having it all, maybe: Design tradeoffs in ITS authoring tools. In: Intelligent Tutoring Systems, pp. 93-101. Springer, (1996)
13. Murray, T.: Design tradeoffs in usability and power for advanced educational software authoring tools. Educational Technology 44, 10-16 (2004)
14. Ososky, S., Sottilare, R.A.: Heuristic evaluation of GIFT authoring tools. US Army Research Laboratory, Orlando, FL (2016)
15. Ososky, S., Brawner, K., Goldberg, B., Sottilare, R.: GIFT Cloud Improving Usability of Adaptive Tutor Authoring Tools within a Web-based Application. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, pp. 1389-1393. SAGE Publications, (2016)
16. Lightbown, D.: Designing the User Experience of Game Development Tools. CRC Press, Boca Raton, FL (2015)

17. Ososky, S.: Practical requirements for ITS authoring tools from a user experience perspective. In: Schmorrow, D.D., Fidopiastis, M.C. (eds.) Foundations of Augmented Cognition: Neuroergonomics and Operational Neuroscience: 10th International Conference, AC 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings, Part II, pp. 55-66. Springer International Publishing (2016)
18. Ososky, S.: Designing the user experience of the GIFT Cloud authoring tools. In: Sottilare, R., Ososky, S. (eds.) Proceedings of the 4th Annual GIFT Users Symposium, vol. 4, pp. 145-156. U.S. Army Research Laboratory, Orlando, FL (2016)
19. Schwier R, Misanchuk ER. Interactive multimedia instruction. Educational Technology; 1993.