Authoring Tools for Adaptive Training – An Overview of System Types and Taxonomy for Classification

Keith Brawner[1]

[1]Army Research Laboratory, keith.w.brawner.civ@mail.mil

1 Abstract and Introduction:

The Intelligent Tutoring Systems (ITS) community has recently renewed its interest in authoring tools, as evidenced in recent workshops, publications, and developments. Part of this renewal of effort has been in the development of authoring tools systems for maturing products and processes. As an example, the Cognitive Tutor Authoring Tools (CTAT), AutoTutor Script Authoring Tools (ASAT), and Generalized Intelligent Framework for Tutoring (GIFT) Authoring Tool (GAT) are all available for community use and feedback through their various portals. As these authoring tools come available, it is helpful to classify them into groups, for the purpose of study, based upon their similarities and differences.

As part of these recent activities, conversations and discussions from authoring tools experts naturally align themselves to a few basic categories. The research presented in this paper attempts to condense and summarize the state of art, practice, and future along its natural dividing lines. The basic categories of division are the intended users, the intended training environment, the type of content authored, and the level of automation.

The first part of this ontology divides itself according to the intended author. Authors of the various authoring tools range from "super user", to "content developer", to "end user". Each of these users fits a different profile, with different expertise, and different demands. Several tools involved the collaboration between one of more of these categories, such as a tool developed by Carnegie Mellon University, Cognitive Tutor SDK, which uses programmer support and SME authored instruction in order to automatically generated rules for use in student diagnostics. The section of this paper dedicated to various user categories goes into detail about the expected and required expertise for each user category, elaborates on examples of tools constructed for each group, and attempts to predict the direction of future research in each of the categories.

The second part of the ontology divides itself according to the nature of the intended student environment. Student environments can include text-based and plain HTML-delivered websites, simulated environments for skill practice, or fully interactable 3-dimenstional game worlds. Unlike the intended author classification, few tools fall into a category where they target multiple environments. The functions of the authoring tools for each environment varies significantly, and the section of this paper dedicated to details about the expected and required expertise for each environment category elaborates on examples of the tools constructed for each broad

group category and gives basic predictions about the near-term tracks for this research.

The third part of the ontology is divided according to the type of content which requires authoring. Typically, the components of intelligent tutoring which require authoring involve content, assessments, feedback, or within-game adjustments. However, there are fringe areas of authoring which perform these functions in novel manners, such as the use of demonstration by practice for user-authored assessments, or automated feedback generation. Past areas of this technology, such as automatic content reuse for remediation, may now leverage the large body of Internet-accessible content. These novel applications still fit within the overall ontological classification but are noteworthy for their interactions with other portions of this grouping scheme. The section of this paper dedicated to the discussion of the relative difficulties and conflicting goals among the various types of content authoring.

The last component of the ontology is the portion which is centered on automation of the authoring process. Tools in this category are not intended for direct human use, but make use of either machine processes or crowdsourcing to create components of intelligent tutoring systems. The design of such tools is fundamentally different from the other groups of users to warrant further overall discussion.

A single authoring tool may fit in one or more of these categories, but these categories are functionally useful for describing the purpose, construction, and users of modern ITS tools. As an example, a tool which allows content developers to create assessments of student performance within a game world and automatically suggests feedback based on those items fits three categories of tool: one for its user group, one for its purpose, and one for the automaticity of one of its functions. Describing a tool in this detail, however, provides a working definition and taxonomy by which to classify, fairly evaluate, and discuss future authoring tools.

Each ITS development tool logical falls into one of more of these categories. Particular authoring tools will be presented throughout this paper as examples of tools within specific categories.

2 User Category

The first fundamental category from which authoring tools diverge for classification purposes is the category of user which they are built for. Generally, an authoring tool user requires familiarity of a few basic categories: either familiarity of a tool, tutoring/learning theory, the representation of knowledge, or of the actual job/domain/task that the student is supposed to be learning [1]. Each of the authoring tools developed for adaptive training attempts to automate or supplement some part of this functionality in an effort to save time or produce better training.

2.1 Super User

Tools intended for ‘super user’ consumption is categorized by programmatic or system assembly ability. Generally speaking, any tool which involves programming, editing of XML documents, manipulations of agent actions, or the creation of objects

within a virtual world becomes a 'super user' tool requiring specific expertise. The primary differentiation for a super user tool is that it is one step removed from programming or configuration (e.g. XML editor is a tool, but Notepad is not).

Carnegie Learning, in the form of the Cognitve Tutor Software Development Kit (CT-SDK) has an example of a tool which is intended for the use of super users to create tutoring systems [2]. The process involves a knowledge engineer, who is familiar with the process of writing Cognitive Tutor rules, to work in conjunction with a domain expert, who is familiar with the processes of novices and experts, and a programmer, who is capable of creating interfaces and systems which work appropriately. This toolset allows the domain expert to articulate knowledge, the knowledge engineer to capture it in the form of Cognitive Tutor processes in procedural software windows, and the programmer to create an environment to execute the knowledge. While this process requires significant expertise, it is speeded enough that it is commercially viable and successful to create tutors.

2.2 Content Developer

Tools intended for content developers primarily revolve around the expertise categories of either a) instructional system design or b) subject matter expertise with a domain. Automated examples such as the content-augmentation process described later, in section 4.1, or the system which uses subject matter expertise and training materials to craft a dialogue tutor, described in section 5.2, can be seen as tools for such individuals.

2.3 End User

There are few existing systems which rely upon the end user to help in the creation of the system. However, the recent trend towards crowd-sourcing includes tools that the students can use, while interacting with the system, to create content for consumption of other students. It is expected that more 'social' tools will be available in the future, as part of the ongoing process of improving tutoring systems.

3 Student Environment

The second fundamental category to distinguish authoring tools is the intended environment for consumption. Student environments can range from multiplayer 3D worlds to simple "page turner" book replacement content, with several types of environments in between.

3.1 Basic Consumption

A basic consumption environment is nothing more than a "page turner", webpage, PowerPoint slide set, video, or other form of static content. While there is little doubt that the presentation of content is part of tutoring, there generally exist few adaptive authoring tools to create this content. There are many tools to create content of this

nature, many environments in which to consume it, and the listing of them is beyond the scope of this paper.

3.2 Enhanced Consumption

Adaptive training tools have the capability to apply enhanced value to existing pieces of training content. Systems such as the Generalized Intelligent Framework for Tutoring (GIFT) [3] and REDEEM [4] operate as "shell tutors" to provide a framework for the import and use of previously existing materials, the addition of feedback, and the embedding of pedagogical practices..

An example of an enhanced consumption authoring tools is the eXtensible Problem-Specific Tutor [5]. This system allows for content authors to highlight certain interactable portions of content on a webpage in order to provide a "tutoring layer" overtop of the existing content. In this manner, an author can use familiar tools for the creation of basic content while creating an overlay of instructional interactions. Later, when the student takes predefined actions, they can receive feedback, additional content, or other information.

3.3 Practice Environment

An example of an authoring tool which is intended for the student practice environment is SimStudent [6]. SimStudent watches expert actions within the environment in order to attempt to generalize rules of human behavior through the creation and population of behavior graphs. Later, these behavior graphs are applied to the assessment of novices. This works very well in procedural domains such as stoichiometry and algebra, as it can easily spot novice mistakes in problem solving order. This can effectively bridge the gap between maximally general Cognitive Tutors and maximally specific example tracing tutors for domains of instruction which involve step-by-step problem solving.

3.4 Virtual Environment

The VECTOR project is an example of an authoring tool designed around the principle of scenario-based training and the incorporation of instructional design workflow into virtual environments. It does this through anchoring instruction explicitly to objects within the game world. It has been applied to ill-defined tasks such as cross-cultural awareness, clinical communication, and negotiation skills [1]. Another example of a system which operates under similar constraints is the Situated Pedagogy (SitPed) authoring tool developed by the Institute of Creative Technologies, which offers the authoring of characters and events from the student's perspective, within a game environment.

4 Content Authoring

4.1 Content

The majority of content which is consumed in an adaptive training system is created in systems external to the system itself. PowerPoint- and HTML-based training is a prime example of where

An example of a tool created for the modification of content in adaptive training systems, instead of a content creation tool (e.g. PowerPoint™), is the Metadata Authoring Tool (MAT). The MAT is a part of the GIFT standard package of publicly available tools [7]. The purpose of the tool is to enable authors to describe the attributes of created content in a machine-consumable fashion. Authors essentially check boxes which content attributes. During the tutoring process, the Engine for Management of Adaptive Pedagogy (EMAP) matches learner traits and content attributes in a manner which has been validated in the literature to produce learning gains. An automated version of this tool is currently under development from Advanced Distributed Learning [8].

4.2 Assessments

A critical part of any adaptive training system is the ability to assess student competency, either for the purpose of giving feedback, selecting the next scenario, or assigning remediation. The authoring of assessments is usually dependent on the domain of the system which is attempting to assess, but recent efforts have attempted to perform domain-independent assessments. The creation of such domain-independent assessments is usually highly dependent on templates, such as with the AutoTutor Script Authoring Tool (ASAT) [9], or the GIFT Student Information Models for Intelligent Learning Environments (SIMILE) Workbench tool [7]. Each assessment in an intelligent tutoring system ties to a learning objective or cognitive tracking variable.

4.3 Feedback

Most intelligent tutoring systems authoring tools couple the authoring of assessment and feedback, simultaneously creating the way to assess student misconception or poor performance with the actions that the system has available to it. Systems such as SitPed, GIFT Authoring Tool (GAT), and ASAT all allow the authoring of feedback and interactions within the same user interaction.

5 Authoring Process

Lastly, the functionality that divides authoring tools is the process by which they are used. Super user intended tools are primarily human-oriented in nature, while most tools use some combination of machine processes and templates to create interactions.

Finally, there is the potential for highly automated tools to use a wide availability of data to construct interactions without human intervention.

5.1 Primarily Human

The majority of authoring processes are primarily human in nature, utilizing tools to supplement either author knowledge, steps in the authoring process, or as supplemental information. Tools for highly knowledgeable users have a tendency to be especially manual, while tools for content developers have a tendency to be dependent on templates, leaving user tools to be with highly automatic or non-existent. The GIFT architecture has a number of primarily human configuration tools, and is hardly unique in this respect, as most systems involve some amount of system-specific tool for configuration of various variables, assessments, content, and feedback.

5.2 Machine-Assisted

Machine-assisted authoring is the standard for non-super user authoring tools, using the function of the computer to supplement or augment some level of expertise of the author. Tools such as SimStudent [6], once developed for a specific domain, are intended to be used in conjunction with a domain expert in order to create interactive tutoring events. Another example of a adaptive training authoring tool which depends on both automated and human authoring is the Tool for Rapid Automated Development of Expert Models (TRADEM) available at http://trademl.gifttutoring.org [10]. This tool takes a number of pieces of training content, performs a text analysis to determine the key text items, creates a topic map of the core concepts, and requires a content author to fine-tune the results and assessments before exporting to a digital agent. In each of these cases fashion, a framework for the content delivery can be rapidly created and augmented through human use.

5.3 Primarily Automated

Although systems such as TRADEM and automatically constructed concept maps work better with human intervention, they have the potential to ignore human intervention. There is the potential for fully automated intelligent tutoring systems which construct themselves from data, such as the data in DataShop [11]. Technologies such as automatically-created student models from small samples of learner interaction data, or from systems which measure learning gains in order to select the optimal instructional interventions [12] may provide a future path for the creation of adaptive learning content and interactions.

6 Ontology Use and Example Classifications

The Generalized Intelligent Framework for Tutoring (GIFT) has a number of authoring tools available, each of which has its own purpose and is intended for a variety of audiences and environments. As an example, the GAT is a single, web-based, in-

stance where an entire tutoring experience can be created. A GAT-authored course may consist of embedded content assessments, nodes within a course flow, or adaptive branching based on learner experiences.

	GAT	MAT	TRADEM	CT-SDK	Sim-Student	xPST	VECTOR	SitPed
User								
Super User				X	X	X	X	
Content Developer	X	X	X	X		X	X	X
End User								X
Environment								
Basic Consumption	X	X	X			X		
Enhanced Consumption	X			X				
Practice Environment	X	X		X	X			X
Virtual Environment	X	X					X	X
Content								
Content	X	X	X	X	X	X	X	X
Assessments	X		X	X	X	X	X	X
Feedback	X			X	X	X	X	X
Process								
Human	X	X		X		X	X	X
Automated					X			
Both			X		X			

7 Conclusion and Future Research

The reuse of ITS systems for learning research reduces the overall need for authoring tools and enables the field to continue to develop system-specific systems. ITS authoring tools generally do not exist in large enough numbers to be considered a subfield, but general software engineering design principles are emerging. These principles, such as the division of tool and function, a tool for each function, the limitation of complexity per user allow the field to be categorized by function. The most recent work on the subject catalogues these items in great detail [13].

A set of general-purpose authoring tools for a general-purpose intelligent tutoring system remains a difficult goal, but the GIFT GAT, AutoTutor ASAT, and xPST are all now publicly available. In addition to these tools, increasing automation, such as from TRADEM and SimStudent, is removing the task of authoring from the user. The author believes that these trends of traditional design, domain generality, system

generality, and increased automation, will dominate the ITS authoring tools field in the coming years.

8 References

1. Zachary, W., Santarello, T., Engimann, J., Hu, X.: Authoring by Whom? Two Perspectives on Its Authoring Tools. Intelligent Tutoring Systems 2014, Honolulu, Hawaii (2014)
2. Brawner, K., Sinatra, A.: Intelligent Tutoring System Authoring Tools: Harvesting the Current Crop and Planting the Seeds for the Future. In: Intelligent Tutoring Systems. Springer, (2014)
3. Sottilare, R.A., Brawner, K.W., Goldberg, B.S., Holden, H.K.: The Generalized Intelligent Framework for Tutoring (Gift). (2012)
4. Ainsworth, S., Major, N., Grimshaw, S., Hayes, M., Underwood, J., Williams, B., Wood, D.: Redeem: Simple Intelligent Tutoring Systems from Usable Tools. Authoring Tools for Advanced Technology Learning Environments, pp. 205-232. Springer (2003)
5. Gilbert, S., Blessing, S.B., Kodavali, S.: The Extensible Problem-Specific Tutor (Xpst): Evaluation of an Api for Tutoring on Existing Interfaces. In: V. Dimitrova, R. Mizoguchi, B. du Boulay, Graesser, A. (eds.) Proceedings of the 14th International Conference on Artificial Intelligence in Education, pp. 707-709. IOS Press, Amsterdam, Netherlands (2009)
6. MacLellan, C.J., Koedinger, K.R., Matsuda, N.: Authoring Tutors with Simstudent: An Evaluation of Efficiency and Model Quality. In: Intelligent Tutoring Systems, pp. 551-560. Springer, (2014)
7. Army Research Laboratory: Gift 2014-1x, Generalized Intelligent Framework for Tutoring Release Page. http://www.gifttutoring.org (2014)
8. https://github.com/adlnet/DECALS
9. Susarla, S., Adcock, A., Van Eck, R., Moreno, K., Graesser, A., Group, T.R.: Development and Evaluation of a Lesson Authoring Tool for Autotutor. In: AIED2003 supplemental proceedings, pp. 378-387. (2003)
10. Ray, F., Brawner, K., Robson, R.: Using Data Mining to Automate Addie. In: Educational Data Mining 2014. (2014)
11. Koedinger, K.R., Brunskill, E., de Baker, R.S.J., McLaughlin, E.A., Stamper, J.C.: New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization. AI Magazine 34, 27-41 (2013)
12. Liu, Y.-E., Mandel, T., Brunskill, E., Popovic, Z.: Trading Off Scientific Knowledge and User Learning with Multi-Armed Bandits.

13. Sottilare, R., Graesser, A., Hu, X., Brawner, K.: Design Recommendations for Intelligent Tutoring Systems: Authoring Tools, www.gifttutoring.org. In Press. (2015)