Quick Summary – AuthorIT and TutorIT

JOSEPH M. SCANDURA

Director of Research, TutorITweb, Emeritus Professor, University of Pennsylvania, USA

This article summarizes the current status of AuthorIT authoring and TutorIT delivery platforms available at www.TutorITweb. It is based on two recent publications, and includes a short history of developments along with references and relationships to the goals established for the GIFT framework established by the Army Learning Model (ALM). Also included is a review of deterministic theorizing in the Structural Learning Theory (SLT), summarizing how we got where we are.

Keywords: Structural Learning Theory, Structural Learning, Structural Analysis, Bloom’s 2-Sigma, Universal Control Mechanism, AuthorIT, TutorIT, GIFT, TICL, knowledge representation, ACT-R, BIG DATA, deterministic theory, higher order knowledge

AuthorIT and TutorIT have been the subject of a large number of articles beginning in 2001 (see Scandura References). A short history follows to provide context. Two recent articles focus on AuthorIT and TutorIT technologies based on the Structural Learning Theory (SLT) and are especially relevant (Scandura, 2016, Scandura & Novak, 2017).

These articles help clarify distinctions between AuthorIT/TutorIT and adaptive learning systems based on intelligent tutoring (ITS) and those based on BIG DATA. ITS generally start with a learning theory (e.g., ACT-R in Anderson, 1995) and add tutoring to it. Systems based on BIG DATA (e.g., Knewton) attempt to automate the entire teaching-learning process using large data bases (Scandura, 2017).

Current Status. — As detailed in Scandura (2013a, 2014, 2017), AuthorIT & TutorIT rest on a fundamentally different theoretical foundation than either ITS or BIG DATA. AuthorIT and TutorIT were explicitly designed from inceptions to

*Corresponding author: scandura@scandura.com
model interactions between teachers (tutors) and learners as a whole. Two recent articles offer quick entry.


This article focuses on answering four basic questions that good human tutors must ask, explicitly or implicitly: 1) How can one represent what needs to be learned for success? 2) How can one determine what an individual student does and does not know that needs to be learned, 3) How do students use their existing knowledge to acquire new knowledge and 4) How can one to put this all together into a comprehensive and operational theory of teaching and learning.

AuthorIT and TutorIT provide answers to each of these questions. AuthorIT provides a systematic tool for creating knowledge representations (1 above). TutorIT takes knowledge representations as input and makes all tutoring decisions automatically. More specifically, TutorIT takes knowledge representations created with AuthorIT as input and automatically pinpoints what each student knows at each point in time (2 above). Although technically incomplete, TutorIT also knows how students use the information they have to create new knowledge so it theoretically can predict problem solving on novel problems (3 above). Collectively, AuthorIT and TutorIT provide a comprehensive operational system of teaching and learning (4 above).

My paper (2016) then goes into detail showing how and why AuthorIT authoring and TutorIT delivery technologies work. They have made it possible in a highly cost effective manner to build a broad variety of TutorIT tutorials, tutorials that interact with students as might a good human tutor. AuthorIT authoring platforms have been refined over a period of years and used to build a broad variety of TutorIT tutorials. These authoring platforms along with many TutorIT tutorials are now ready for release. Those interested in field testing or further research are encouraged to contact the author.

**Historical Developments leading to SLT.** — The second recent article “AuthorIT & TutorIT: Attacking Bloom’s 2-Sigma Problem from a Different Perspective” with Elena Novak (2017) shows that AuthorIT and TutorIT represents a fundamentally different approach to building and delivering adaptive learning systems. Intelligent Tutoring Systems (ITS) guide students as they solve problems. BIG DATA systems make pedagogical decisions based on average student performance. Decision making in AuthorIT and TutorIT is designed to model the human tutoring process as a whole. Just as good human
tutors need to know the subject matter, AuthorIT is used to systematically identify what students need to learn for success. TutorIT takes this information as input and makes all pedagogical decisions automatically. This dramatically reduces the time and expense of building adaptive learning systems, while simultaneously having the potential of ensuring predetermined levels of mastery.

The Scandura and Novak (2017) article also lists and categorizes selected references for those unfamiliar with the evolution, application and refinement of the Structural Learning Theory (SLT), its origins and current status along with how and why it was developed.

Traditional ITS like most research in the behavioral sciences is based on theory designed to provide a stochastic accounting of observables. By way of contrast, classical linguistics, software engineering, etc., even psycho-linguistics, have a deterministic foundation (Chomsky, 1957, 1968; Miller & Chomsky, 1963). This is true whether the research focuses on individuals or groups.

Stochastic theorizing in behavioral science (including tutoring) leads to a host of costly, time consuming problems that need to be addressed. The goal is to explain behavior as it exists in the real world under real world conditions -- with all its complications but only stochastic precision.

AuthorIT and TutorIT technologies build on the fundamentally different set of assumptions on which SLT rests. SLT focused on deterministic theorizing from its inceptions -- focusing on behavior under idealized conditions. See my linked article on “Deterministic Theorizing in Structural Learning: Three Levels of Empiricism”. Published in 1971, this paper became a citation classic but was largely ignored in ITS development beginning a decade later. http://ticl.coe.uh.edu/3FF1D85E-F66C-45D1-9D6D-1BBCAE379E17.pdf.

To justify deterministic theorizing, in talks as well as papers on SLT, I asked readers to imagine Galileo at the Leaning Tower of Pisa. Instead of dropping a large and a small iron ball, suppose he had dropped an iron ball and a feather. He’d obviously have gotten fundamentally different results. Instead of concluding that ALL objects fall at the same rate, physics might have developed in a fundamentally different direction -- into a science of droppings, comparing the rates of fall of various kinds of objects.

Our research demonstrates that we face the same decision in developing human like intelligent tutoring systems (ITS). Is it best to build ITS based on a stochastic foundation, introducing the equivalent of epicycles to explain the many variations observed in the real world? Or, as we have done in SLT, explain and predict behavior under idealized conditions, attributing observed variations under real world conditions to errors of measurement.
AuthorIT/TutorIT build on the same scientific reasoning: Instead of inserting random factors into the underlying (SLT) theory itself, as has been done with ITS, deviations from theory are attributed to errors of measurement. The focus in authoring is on what needs to be learned for success. Pedagogical decisions in TutorIT are based on what any given student does and does not know at each point in time and what the student needs to progress.

Other decisions made when authoring and implemented during tutoring are strictly secondary. Secondary concerns include things like the type of media used, wording of the questions asked, promoting student reflection, encouraging learners to discover, amount and nature of assistance given, amount of required review, etc.

This does not mean that such factors may not be desirable, or even important. They may indeed facilitate, reduce instructional time or otherwise improve learning (or not). However, ALL are secondary to whether or not a student has actually learned what the author intends (and is testing for). In short, how a student achieves mastery is strictly secondary. What matters is results.

Ignoring theoretical side issues is precisely what early physicists did. Of course, later work revealed limitations (e.g., relativity and quantum theory to address issues going well beyond everyday thinking). Nonetheless, think how much deterministic theorizing in physics has accomplished over centuries. Even today, basic ideas prevail in so many areas of application.

With respect to adaptive learning, I can say with certainty that there is no way we otherwise could have developed AuthorIT and TutorIT, much less build as many “ITS” as we have with such limited funding. One can only hope that others will see the potential.

There is obviously a lot more that can be said on this subject, but for present purposes, the following references cover:

1. Formative research beginning with analysis of the new math and research in mathematical and experimental psychology leading to a focus on rule learning.

2. This research laid the ground work for the Structural Learning Theory, first presented as a deterministic theory at the third in a series of annual Structural Learning Conferences at Penn in 1970 and published in 1971. This work was followed by refinement and expansion based a series of basic and applied studies published in a variety of journals in experimental, educational, developmental psychology, artificial intelligence and structural learning (the research led to a new journal initiated by ZP. Dienes an internationally known mathematics educator). My initial books on structural learning (Scandura, 1973, 1976) have just been republished by Taylor and Francis, September 2017.
3. While experimental research in structural learning continued at a slower pace, emphasis for a variety of reasons switched to applications in software engineering. The development of several major software engineering systems beginning with PRODOC (PROfessor with his DOCtoral students) ultimately led to key refinements in SLT (to be revealed in my next paper).

4. This work in turn laid the ground work for both revisiting the Structural Learning Theory with a new set of formal tools and development of AuthorIT and TutorIT.

**Relationship to GIFT Goals.** — Finally, it seems worth noting where AuthorIT and TutorIT stand with respect to the ambitious goals set for GIFT (see Sottilare, This issue). GIFT offers a broad framework for building adaptive learning systems. It includes a number of technologies within this framework. While reportedly open to alternative tutoring engines (e.g., TutorIT), GIFT’s (Sottilare, this issue) current default tutoring engine, eMAPS, is based on Merrill’s Component Display Theory (Merrill, 1983).

As follows, AuthorIT and TutorIT offer alternatives for essentially all GIFT goals.

**Goal 1: Developing Efficient Authoring Processes.** — AuthorIT authoring systems have demonstrated an extremely high degree of efficiency. The number and variety of TutorIT tutorials developed by a small team never larger than part time people with limited resources over so short a time is unprecedented (see TutorITweb.com). This track record is even more unusual given the fact that many of these TutorIT tutorials were developed in parallel with development of the authoring systems used to create them. One of the major factors in this efficiency was automating pedagogical decision making. Elimination of the need to program pedagogical decision making alone resulted in a major decrease in the amount of work required.

**Goal 2: Developing Effective Instructional Decisions.** — TutorIT decision making is designed to ensure that every student who completes a given TutorIT tutorial will necessarily have demonstrated predetermined levels of mastery of the content in question. A student may not complete a given tutorial (e.g., by giving up, not having enough time, etc.). However, if he or she does complete a given TutorIT tutorial, that student will necessarily have demonstrated mastery.

**Goal 3: Modeling Learner and Team Trends and Competency.** — The main goal of AuthorIT authoring and TutorIT tutoring is to model the human tutoring process. This holds whether training takes place with individuals or in
groups. In addition to individual tutoring, AuthorIT supports the development and TutorIT the delivery of team tutoring. In the latter case, various members of a team could be required to collaborate while performing various tasks in parallel. TutorIT can even allow students to ask for more information during the course of training.

**Goal 4: Building Rapport and Engagement with Learners.** — This is largely a matter of using appropriate instructional support. AuthorIT and TutorIT take a step in this direction by supporting the inclusion of a broad variety of media for this purpose.

**Goal 5: Modeling Collective Instructional Domains.** — Although use has been limited, TutorIT has been used successfully in a group or classroom setting. The only difference between this and the individual case is that someone, or the group as a whole, must agree on and/or submit answers.

**Goal 6: Expanding Adaptive Instruction to a Broader Array of Task Domains.** — Universal applicability is an inherent characteristic of AuthorIT and TutorIT. They are designed first and foremost for broad application. Although examples involving higher order rules are limited, TutorIT potentially supports broad domains involving multiple, potentially even novel domains not initially identified during development (see use of higher order SLT rules). A broad range of student response types also is supported – text entry, multiple-choice, selection of one or more objects on a screen, moving objects on the screen. While TutorIT supports Text To Speech (TTS), it does not currently support speech recognition. The latter is a specialized area where plugins are potentially available.

**Goal 7: Evaluating the Effectiveness and Efficiency of Adaptive Instructional Systems.** — Effectiveness of any given TutorIT tutorial is an inherent characteristic. TutorIT is designed to ensure mastery. Efficiency also is inherent in the way TutorIT operates. Tutoring continues until predetermined levels of mastery have been achieved. Mastery criteria are adjustable as desired.

**Goal 8: Supporting Distributed/Mobile Learning.** — AuthorIT and TutorIT were originally designed for the desktop. They have since been ported to and are currently supported on the web. Plans also are underway for enabling mobile execution.

**Concluding Remarks.** — Bob’s paper (this issue) provides a very broad framework for Adaptive Instructional System (AIS). In the process, he identifies important limitations of traditional Intelligent Tutoring Systems (ITS) and proposes ways GIFT currently attempts to address those limitations.
Whereas traditional ITS focus on given domains, AIS may include links to external aids that may not be adaptive. Among other things, these aids may incorporate traditional instructional design principles (e.g., Merrill, 1983). All are fair game as long as these aids may benefit the learning process.

Whereas the resulting AIS may be very complex, an equally perhaps even more important goal of GIFT is to make authoring more efficient. Whereas traditional ITS are time consuming and expensive to build, a parallel goal of GIFT is to make authoring more efficient.

Maximizing learning and reducing development costs are highly laudable goals. In an important sense, however, these goals conflict with one another. Expensive as traditional ITS have been, building tutorials that meet ALL GIFT goals may be even more so. The number of variables and conditions that can affect earning, not to mention their interactions, is large indeed.

Hence, we have a major dilemma. How can we best maximize learning but at the same time reduce development costs.

Rather than focusing on the HOW, the wide range of variables that may affect learning, AuthorIT and TutorIT focus on WHAT needs to be learned for success. This change in focus puts the emphasis on mastery. A student either knows something or does not. How he or she got there is of secondary importance.

Looked at from this perspective, the purpose of instruction can be summed in one word. That word, however, remained out of mind until Paul Shorter (US Army Research Laboratory) made an insightful remark after he had spent a day with us reviewing AuthorIT and TutorIT. Paul used the term “deterministic” in describing how they worked. Not surprisingly in retrospect, this term reminded me of my first paper introducing the Structural Learning Theory (SLT) in 1971 “Deterministic Theorizing in Structural Learning: Three Levels of Empiricism”.

The word “deterministic” singularly best describes a fundamental difference between SLT guiding AuthorIT and TutorIT and essentially every other theory guiding development of intelligent tutoring (or adaptive learning) systems. All ITS theories, including those within the GIFT framework, are inherently stochastic in nature (e.g., ACT-R by Anderson, 1995).

Given an instructional domain, methods based on SLT have made it possible to identify WHAT needs to be learned for success in that domain, and to do so simultaneously at ALL levels of expertise. These patented methods focus on the WHAT versus the HOW (Scandura, 2014) and greatly simplify AIS development. “What” represents the core.

Focusing on what completely eliminates the need to program pedagogical decision making. All else required or useful in facilitating mastery is optional. As detailed in my second paper (this issue), using Structural Analysis to represent
to-be-learned knowledge is fundamental. This focus on “what” has made it possible to develop scores of AIS at a fraction of traditional costs.

I would be remiss not to also mention Fletcher’s (this issue) excellent summary of research over the years in computer based learning. In this summary, Fletcher cites early work by Pat Suppes (1974) and Suppes, Fletcher & Zanotti (1975, 1976). This work elaborates and extends Suppes & Groen’s (1967) earlier work reported in an NCTM monograph (Henderson, Scandura & Trimble, Editors, 1967). All involve analysis of what needs to be learned for success in basic arithmetic – in this case counting up to add small numbers.

This work is very similar in nature to basic research on which the Structural Learning Theory is based (e.g., Scandura, 1971, 1973, 2001, 2007). The focus in both is on what needs to be learned for success. There is, however, one big difference. Like most ITS theories, Suppes et al “adding by counting up” theories were tested under average conditions -- average behavior under real (uncontrolled) conditions. Perhaps sensing the overwhelming complexity of dealing with more complex knowledge in the same vein, Fletcher focused on the strictly empirical approach adopted by DARPA. One can sometimes get good results if one does enough work.

As we will see in my keynote article, foundational experiments motivated by the Structural Learning Theory* were designed first and foremost to test fundamental assumptions under idealized conditions. Like experiments in classical Newtonian physics, early experiments in SLT were designed to answer questions like what will be learned if students have sufficient time and access to supporting information. In short, the goal of this research was to identify critical factors (analogous in physics to rates of fall in frictionless environments).

In short, SLT as implemented in AuthorIT and TutorIT offers an alternative route to achieving both GIFT’s broad as yet unrealized goals and DARPA’s highly pragmatic, some might say resource heavy, brute force approach to tutoring. My second paper in this issue describes evolution of the underlying SLT, how it was developed, the problems faced and their solutions. Irrespective of complexity, the focus is on what needs to be learned for success. All other factors affecting learning play a distinctly secondary role.

* In retrospect, SLT is better viewed as a theory of teaching-learning, not just or primarily a learning theory.

REFERENCES

Scandura, J.M. and Novak, E. AuthorIT & TutorIT: Attacking Bloom’s 2-Sigma Problem from a Different Perspective”. Technology, Instruction, Cognition & Learning (TICL), 2016, 10, pp. 267–288.
Scandura, J.M. Introduction to Compendium I. Technology, Instruction, cognition & Learning (TICL), 2003, 1, 2, 115–117.


