AuthorIT & TutorIT: Attacking Bloom’s 2-Sigma Problem from a Different Perspective

JOSEPH M. SCANDURA1* AND ELENA NOVAK2

1Director of Research, TutorITweb, Emeritus Professor, University of Pennsylvania, Editor in Chief: Technology Instruction Cognition & Learning (TICL)
2Assistant Professor of Instructional Technology, Kent State University, Managing Editor: Technology Instruction Cognition & Learning (TICL), e-mail: elannovak@gmail.com

AuthorIT and TutorIT represent 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.

Keywords: Structural Learning Theory, Intelligent Tutoring Systems, Adaptive Learning, BIG DATA, AuthorIT, TutorIT, TutorITweb.com

I want to thank Elena for the tremendous effort she has put into organizing the AERA 2017 TICL symposium on Intelligent Tutoring Systems, BIG DATA–Learning Analytics, and Automated Humanlike Tutoring. The first two sessions (Intelligent Tutoring Systems (ITS) by Robert Sottilare, and BIG DATA–Learning Analytics by John Dexter Fletcher) introduced systems that were motivated by research on ITS and Big Data, respectively. I will be talking about our AuthorIT authoring and TutorIT delivery systems. As will become clear in in the

*Corresponding author: scandura@scandura.com
following session, AuthorIT and TutorIT rest on a fundamentally different theoretical foundation than either ITS or Big Data. But, more on that later.

Let me begin by recalling the gold standard for CBI set by Ben Bloom many years ago (Bloom, 1984). Bloom found that students aided by good human tutors performed at a level two-standard deviations above those taught in classrooms – in the top 2%. Today, I want to introduce you to an automated solution that has an unprecedented chance of meeting that standard in predefined content domains.

Specifically, I’d like to introduce you to our AuthorIT authoring and TutorIT delivery systems – more specifically, how these systems bring Bloom’s goal within reach. Equally perhaps more important, these systems enable instructional designers to build dynamically adaptive (ASA intelligent) tutoring systems in their own areas of expertise. These systems also will enable instructional researchers to conduct more definitive research on the efficacy of different methods of instruction.

As will become clear in my paper on Modeling Human Tutors (next TICL issue), TutorIT and AuthorIT build directly on the Structural Learning Theory (SLT) in turn on decades of basic research in math education, problem solving, artificial intelligence (AI), cognitive psychology, software engineering and instructional design. Under development for some time, I’m very happy to say TutorIT and AuthorIT are now ready for use. You can now go directly to www.TutorITweb.com and sign in as an author.

In what follows, we will see that AuthorIT and TutorIT are designed to work like a human tutor. We’ve looked at the problem from a different perspective than most. What does a good tutor need to know? Whether human or automated, a good tutor must address four basic questions.

First, instead of worrying about how students learn, a good tutor needs to know what a student needs to know to be successful. This fact is the conclusion of my very first piece of serious research (Scandura, 1964a, b). During the heyday of the new math, a major assumption was that students learned better when they discovered something. Deeper analysis revealed that what really matters is what a student knows when he or she receives instruction. Too soon and it falls on deaf ears, too late and it didn’t matter. We later found that one could directly teach by exposition what was learned in discovery – and this could be done more efficiently (e.g., Roughead & Scandura, 1968; Scandura & Scandura, 1980).

Second, an effective tutor also needs a way to determine what any given student does and does not know – at each point in time. We conducted a series of experiments ranging from externalizing student thought processes to rigorous experiments (e.g., Scandura, 1964a,b, 1970, 1971, 1973, 1977; Durnin & Scandura, 1973) confirming feasibility.
Third, a good human tutor needs to know how students learn. Starting with intuition (Scandura, 1964, 1970) followed by exploratory research (e.g., Scandura 1970, 1971) culminating in definitive research (e.g., Scandura, 1973, 1974, 1977; Scandura & Voorhies, 1977), we found that higher order knowledge enables subjects to acquire new knowledge.

The big question came next. How do good human tutors use the above to decide what to do next (e.g., Scandura, 1977, 2001, 2007; Wulfeck & Scandura, 1977)? Let’s take a deeper look! Here is a quick overview of how we got a bit closer to the truth. We had to answer four basic questions:

1. How to represent knowledge – we found that hierarchical representation solved a lot of problems.
2. How to measure what individual students know.
3. How people learn.
4. How tutors work.

These ideas are summarized in Figure 1.

Figure 2 shows this in more detail, pulling it all together, and summarizing how TutorIT works. The Knowledge Representation at the top represents what

![TutorIT Tutorials](image)

**FIGURE 1**
TutorIT tutorials ask and answer four basic questions.
students are to learn. TutorIT interacts with the Learner (student) through the Blackboard Interface. TutorIT displays problems and/or partially solved problems on the Blackboard Interface. The Learner responds. TutorIT evaluates each Learner’s response and reacts accordingly.

As above, paralleling what a human tutor might do, it marks nodes in each hierarchical AST display as known or unknown as the case may be. Furthermore, hierarchical relationships between the nodes allow TutorIT to infer status on untested nodes. This is essentially what human tutors do instinctively – this type of inferencing is what makes it possible for human tutors to quickly home in on student needs. The main difference is that TutorIT does this both automatically and more systematically.

There is obviously a lot more to say about how learners learn, how TutorIT makes its decisions and the relationships between them. There also is a lot to say about our authoring processes. Toward this end, I recommend that you review the articles referenced under “GET AuthorIT/TutorIT Publications” at www.TutorITweb.com. Key ideas are unique and recently patented.

FIGURE 2
Here is how TutorIT works. It takes a representation of the knowledge to be learned as input. TutorIT then interacts with individual student via a blackboard interface – to determine what the learner (student) does and does not know at each point in time, and provides precisely the information the student needs to progress.
Let’s take a peek at www.TutorITweb.com, showing how TutorIT works. Figure 3 shows the entry screen. Students go one place, Teachers, Instructional Designers, Subject Matter Experts, all authors go to another. Authors can see everything students can see, but they also can do much more.

Figure 4 shows a sample TutorIT tutorial in action. These screen shots illustrate the processes students need to perform in solving column subtraction problems. The right shows a sample screen where TutorIT is interacting with a student. Notice that this is not a full subtraction problem. While TutorIT can certainly do this, it can also as shown here present partially solved problems and efficiently determine like a good human tutor if and where a student is having problems and directing the student through to full mastery. The Harder (and/or Easier as appropriate) button gives the student the option of moving faster or slower if more help...
is needed. Based on the student’s answers to such probes, TutorIT automatically adjusts what comes next. How does it do this? I’ll explain all this in my keynote to be published in the next TICL issue.

TutorIT automatically assures that each step and each decision necessary for success is covered. This is illustrated in Figure 4. The tree view on the right gives a hint at how TutorIT makes its decisions. The top two screen shots show successive steps in solving one problem. The three screen shots on the bottom show steps in a more complex problem—involving regrouping across zeros.

The question you might be asking yourself is: How does TutorIT make its decisions? If you think we programmed it into the tutorial, you are incorrect. “How does TutorIT Make Its Decisions?” is the subject of my paper in the next TICL issue.

You might be thinking that this is a nice example, but so what? There are thousands of tutorials of various kinds build by almost as many people. What’s different is that TutorIT makes ALL of its decisions automatically. All content treated in the same way.
FIGURE 5
Shows what happens when the student makes a mistake. He or she is immediately shown what he did wrong, and given instruction on what he or she should have done.

FIGURE 6
Shows what might happen when the student makes an error.
Here are sample screen shots from two other TutorIT tutorials (Figures 7 & 8). And it’s not all math. TutorIT can be used with essentially any content. These tutorials are incomplete but give a broader sense of what can be done. Figure 9 shows the Periodic Table in Chemistry. Here, students can explore properties of various elements before answering questions.

The other two screen shots on the right are from a sample TutorIT tutorial preparing lawyers for the law boards (Figure 9). The law boards are required to practice in most states. If an aspiring lawyer gets a question correct, he is congratulated and told why his answer is correct. If incorrect, he is told why.

These tutorials cover fundamentally different content and look very different. TutorIT’s pedagogical decision making, however, works the same way.

Figure 10 provides an overview of how TutorIT tutorials are constructed. Authors use a proprietary method of Structural Analysis to represent the knowledge or skill to be acquired. This knowledge representation is hierarchical in the case of structured knowledge—where knowledge is cumulative in nature (as in column subtraction, solving quadratic equations, etc.).
This figure goes further. It illustrates TutorIT’s ability to support alternative solutions. This screen shot shows screens addressing three different ways of solving quadratic equations: Factoring, using the Quadratic Formula and Completing the Square.

TutorIT tutorials for the periodic table in chemistry and one for the law boards, quite different.
When a task domain covers a broad range of content, individual items are more or less independent (as in the SATs or the law boards, or classroom examinations). In this case, TutorIT tutorials are especially easy to construct.

There are three increasingly sophisticated TutorIT authoring systems: EZauthor, Customizer and AuthorIT (itself), plus a Scope & Sequence tool for putting them together to create entire curricula (Figure 11).

There is a lot more to say about how AuthorIT and TutorIT systems work. Press GET INFO at www.TutorITweb.com for more information.

EZauthor authoring system allows developing tutorials covering various content areas (Figure 12). EZauthor provides an interface for entering various types of questions (e.g., multiple-choice, matching, picture recognition, etc.), including feedback and media (e.g., videos, animations, pictures) as well as recording voice instructions to accommodate needs of various students. EZauthor was used to create and field test a wide variety of tutorials in mathematics and statistics (e.g., basic math facts, math processes, statistics by Novak, 2014).

Figure 13 shows math tutorials created with EZauthor. The tutorials use a variety of test items, including multiple-choice answers, fill-in-the-blank, true/false,
FIGURE 11
This screen shows four different authoring systems for developing TutorIT tutorials: EZauthor, Customizer, AuthorIT, and Scope & Sequence.

FIGURE 12
Shows a process for building a TutorIT tutorial using EZauthor.
and self-evaluating questions. As a student interacts with a tutorial, it may become apparent that the student lacks some subordinate skills—e.g., multiplication facts as shown on the top right screen of Figure 13. In such cases, tutorial developers can link specific questions assessing the subordinate skill to other TutorIT tutorials (bottom right screen of Figure 13), thus creating non-linear tutorials that cover related content. Once the student shows mastery of the required subordinate skill, TutorIT will automatically redirect her/him back to the original tutorial.

Figure 14 shows how professionally developed TutorIT tutorials using AuthorIT (a powerful desktop application) can easily be customized. Customizer allows teachers as well as developers to easily adjust feedback, colors, media, and audio preferences. This option is particularly useful when teachers want to use an existing TutorIT tutorial but would like to add a personal touch to the tutorial, like graphics, colors, or record their own voice to provide instructions and feedback. In other cases, Customizer can be used to address special needs of the intended learners without making changes to the tutorial content. The changes can range from including narrations in addition to written instructions and feedback to supplementing existing tutorials with video recordings of classroom instruction. The
screen on the right of Figure 14 shows that users can also customize tutorial delivery options. Tutorials can be delivered as adaptive content, serve as a diagnostic instrument, or provide step-by-step instructions to support student learning.

Scope & Sequence allows putting multiple tutorials into a well-organized curriculum. Figure 15 shows a progression of easy to more difficult tutorials sequenced together using Scope & Sequence.

AuthorIT is the most powerful authoring system that allows full hierarchical representation of the content (Figure 16). This system was used by the original development team to develop the current array of TutorIT tutorials.

There is even more at TutorITweb.com. For example, you can get free electronic access to all Technology, Instruction, Cognition & Learning (TICL) articles courtesy of Old City Publishing. If your library does not currently subscribe to TICL, OCP would of course appreciate your requesting that they do.

AuthorIT and TutorIT are different from other ITS in the sense the patented methods used COMPLETELY ELIMINATE the need to program pedagogy. This holds no matter how different or complex the content associated with any given TutorIT tutorial might be. Our authoring platforms (actually
FIGURE 15
Shows an authoring process with Scope & Sequence

FIGURE 16
Shows an authoring process using AuthorIT.
four of them) dramatically reduce the time it takes to develop any given tutorial. In many ways this is the single most important benefit that derives from how we represent to-be-acquired knowledge.

ITS aim for something similar, but the control mechanisms used vary, order of productions can dramatically affect behavior, etc. In TutorIT one universal control mechanism works independently of the content, no matter what the content might be.

While our AuthorIT and TutorIT platforms have been built with our own resources, the resulting tutorials are completely open and can be adjusted by whoever has permission.

Let us know at scandura@scandura.com if you would be interested in using TutorIT in your research and/or using authoring platforms to develop TutorIT tutorials in your areas of expertise? It’s FREE.

Please send a short description of planned research: Include content area, whether funded or unfunded. How do you think TutorIT or AuthorIT might help?

Please include a short description of the TutorIT tutorial you would like to create, what you would like to do with it and your preference as to authoring tool: EZauthor, Customizer, AuthorIT and/or Scope & Sequence? Please include an assessment of your (or team’s) content, computer plus any programming skills, etc.
In the next TICL issue, I will have a lot more to say about how AuthorIT and TutorIT work. What makes them unique, how and how they open the door to a whole new class of tutoring systems that model how human tutors interact with students.

SELECTED REFERENCES BY CATEGORY

The following selected references are included 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. These 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 unified 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 are to be republished by Taylor and Francis, September 2017.
3. While structural learning research continuing 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 plus development of AuthorIT and TutorIT.

FORMATIVE RESEARCH – FROM ANALYSIS OF MATH EDUCATION TO EXPERIMENTAL PSYCHOLOGY TO RULE LEARNING


INTRODUCTION, REFINEMENT, TESTING AND APPLICATION OF THE STRUCTURAL LEARNING THEORY


http://ticl.coe.uh.edu/146_A%20structured%20approach%20to%20intelligent%20tutoring.pdf

http://ticl.coe.uh.edu/147_%20A%20cognitive%20approach%20to%20software%20development-the%20PRODOC%20...pdf

http://ticl.coe.uh.edu/148_Intelligent%20rule%20tutor.pdf

http://ticl.coe.uh.edu/149_Role%20of%20relativistic%20knowledge%20in%20intelligent%20tutoring_12.pdf

APPLICATION STRUCTURAL LEARNING THEORY IN SOFTWARE ENGINEERING AND MUTUAL ENRICHMENT


http://ticl.coe.uh.edu/154_Cognitive%20approach%20to%20systems.pdf


http://ticl.coe.uh.edu/159_A%20cognitive%20approach%20to%20software%20development%20-%20the%20PRODOC%20...pdf


http://ticl.coe.uh.edu/162_Converting%20legacy%20code%20into%20Ada-%20a%20cognitive%20approach.pdf

HISTORICAL ASIDE


RETURNING TO STRUCTURAL LEARNING THEORY


