Tech., Inst., Cognition and Learning, Vol. 0, pp. 00–00
Reprints available directly from the publisher
Photocopying permitted by license only

The Role of Automation in Instruction: Recent Advances in AuthorIT and TutorIT Solve Fundamental Problems in Developing Intelligent Tutoring Systems

Joseph M. Scandura

Research Director, Intelligent Micro Systems, Ltd. (DBA Scandura)
Emeritus Professor, University of Pennsylvania

More and more things that humans used to do can now be automated on computers. In each case, complex tasks have been automated – not to the extent that they can be done as well as humans, but better. In this introduction, I summarize important parallels in education – showing how and why advances in the Structural Learning Theory (SLT) and the AuthorIT development and TutorIT delivery technologies based thereon make it possible not only to duplicate many of the things that human math tutors can do but to do them better.

The use of computers for instructional purposes is much more diverse now than even a few years ago when Macromedia's (now Adobe's) Authorware was dominant in computer based instruction (CBI)[1]. Computers are used as a medium to support various forms of eLearning, ranging from communicating with human

---

* Editorial Note: This article on the role of automation in instruction has important implications for the practice of tutor development. My commentary at the end of this issue proposes directions for future research.

[1] For the historical record, we developed the SURPAS authoring system for the old Apple II computer (the very first Apple authoring system) back in 1979-80. This was several years before Author Ware came on the scene. SURPAS was first used to develop a series of MicroSystem 80 CBI systems for Borg Warner Educational Systems (sold to Josten's later bought by Compass Learning). We also developed about 200 CBI titles which we licensed to Queue, Inc. under the MicroTutor II label. In turn, Queue used SURPAS to develop several 100 addtional titles, about 1000 in all. Robust sales led to Queue being selected as one of Inc. Magazine's fastest growing companies. MicroTutor II sales continued throughout the 1980s and 90s finally petering out at the turn of the century.

tutors over the web to sophisticated simulations. Lacking major advances, CBI has taken a back seat. With its focus on "one size fits all", traditional CBI does not lend itself to truly adaptive instruction.

To be sure, a few commercial CBI marketers purport to offer individualized instruction. Under the hood, however, these systems are little different from what has gone for decades before. Essentially, individualized instruction in CBI consists of a diagnostic test followed by one size fits all instruction directed at areas of weakness. What these systems lack is the ability to adjust instruction dynamically during the course of learning – as would a skilled human tutor.

To date, Carnegie Learning's "Cognitive Tutors" (e.g., Ritter, 2005), are the only Intelligent Tutoring Systems (ITS) that have made significant commercial inroads. These tutors are based on the biologically inspired theories of Anderson (1993) and his colleagues (Anderson, Koedinger et al, 1995; Koedinger, 2007). Development methods and the resulting tutorials have evolved gradually over a period of many years of research.

This approach, however, has major limitations. For one thing they are incomplete. Only certain kinds of content can be handled in traditional ITS. Intelligent Tutoring Systems (ITS) have traditionally been based on a biological metaphor. Focus has been on what is going on in student brains (e.g., Anderson, 1993, Koedinger, 2006, 2009). This requires making fundamental assumptions about the cognitive elements ("productions") associated with the knowledge to be acquired – and the learning mechanisms governing their use. Choosing, arranging and/or adjusting the right productions can be a very complex task. Having to make instructional decisions as to what hints and other instruction to give at the right time is especially difficult. Variations such as using constraints instead of productions (Ohlsson & Mitrovic, 2007) help in some ways but they have been challenged both by ITS purists (e.g., Koedinger, 2009; Weitz et al, in press) and by other promising approaches (cf. Scandura, 2007; Scandura, Koedinger, Ohlsson & Paquette, 2009). Recent attempts in ITS to work at higher levels of abstraction (e.g., Ritter et al, 2006, Paik et al, 2010) appear to be in the right direction but are at best incomplete.

Representing knowledge in terms of relational networks (e.g., as in ALEKS or Paquette, 2007, 2009), also has inherent limitations. First, relationships between concepts do not lend themselves to arbitrary refinement. As shown by Scandura (2005, 2007, 2009), arbitrary refinement is essential to ensure that sufficient information is available for knowledge transfer to the weakest learners in a target population. Second, the number of relationships goes up exponentially with content complexity (Scandura, 2005). As with biologically inspired approaches a major complication is that instruction is inextricably related to the semantics (meaning) of the content. Each tutoring system requires its own unique pedagogical logic, which adds significantly to development costs. As a consequence, the effectiveness of ITS and instructional systems based on relational models can only be determined through costly experiments.

In this context, general purpose authoring and delivery systems applicable to more than a very limited range of content have been unthinkable, much less tutorials that can in principal guarantee learning. **Recent advances in AuthorIT & TutorIT not only support development of ITS for entire curricula (e.g., algebra word problems) but dramatically reduce the cost of developing such systems. Order of magnitude improvements in productivity over traditional ITS have already been demonstrated** (see *www.tutoritmath.com*).

AuthorIT/TutorIT is by far the most efficient system for developing and delivering highly adaptive tutoring systems developed to date. TutorIT is a general purpose tutor that takes a representation of what is to be learned as input and makes all diagnostic and remedial decisions automatically and dynamically at each point in time. **Pedagogical decision making is independent of content semantics, based entirely on the structure of what is to be learned**. TutorIT does this in a way that not only performs as might a skilled human tutor, but in principle can exceed that level of performance. TutorIT tutorials are easily developed and naturally lend themselves to incremental improvement.

As an added bonus, TutorIT can easily be configured to deliver tutoring in any number of ways[2]. TutorIT systems can be developed once, and easily configured to serve multiple purposes. They can serve equally well as dynamically adaptive tutoring systems, adaptive diagnostic instruments or simple performance aids, guiding learners step by step as to the actions and decisions that must be made to solve given problems. They also can be configured to support pedagogical preferences ranging from expository to discovery learning (cf. Scandura, 1964a,b)

Each **TutorIT** tutorial is based on a precise representation of what needs to be learned for success. Patented technologies make it easy to pinpoint what any given student already knows at each point in time as well as the information that student needs to progress. Unlike ITSs, which are primarily directed at procedural knowledge, all knowledge in the Structural Learning Theory

[2] To be sure a variety of shell-based authoring systems have been developed (e.g., XAIDA, Merrill's ID2). Traditionally, these offer different instructional strategies for different categories of learning (e.g., XAIDA was directly motivated by Gagne's, 1985, taxonomy of learning categories). Unlike TutorIT, however, these shells are severely limited in both scope and interactivity.

(SLT, e.g., Scandura, 2007, 2011) on which TutorIT is based necessarily includes varying degrees of both declarative and procedural knowledge. Authors also have the option of representing to-be-learned knowledge with arbitrary degrees of precision. Full analysis ensures contact with prerequisites available to all students in a given population.

In this case, it is possible to guarantee mastery. Any student who enters with pre-specified prerequisites and who complete a given tutorial will by definition have mastered the content. Even approximations are typically more than sufficient to obtain results with the potential to exceed classroom instruction.

The precision with which to-be-acquired knowledge can be represented, together with available delivery options, makes it possible in principle to **guarantee** learning. Any student who enters a TutorIT tutorial with predefined prerequisites and who completes that tutorials will by definition have mastered the skill to the specified level of assurance. The empirical question is not whether a student will succeed, but rather how long it will take and the extent to which the student is motivated to complete any given tutorial.

Continuing improvements have and are continually being made to both AuthorIT and TutorIT. Development times (and costs) have been a small fraction of that in ITS development. AuthorIT and TutorIT have successfully been used over the past year to build highly adaptive and configurable tutorials for a significant body of content, primarily in mathematics but not exclusively (e.g., critical reading). As this is being written, the following are ready for field testing: Basic Facts, Whole Number Algorithms, Fractions, Decimals, Signed Numbers, Complex Expressions, Basic Math Processes, Pre-algebra, Essential Topics in Algebra 1 & 2 and Logical thinking. A comprehensive tutorial on algebra word problems is under development.

AuthorIT and TutorIT can now be used to develop and deliver highly adaptive and configurable tutorials for essentially any content, whether in mathematics, reading, other school subjects or business training. In the process, development times and costs have been further reduced -- in some cases to as little as a day. To further leverage these capabilities, we also have made a start in developing TutorIT tutorials to help subject matter experts create TutorIT tutorials in their own areas of expertise.

To summarize, the **AuthorIT** authoring system makes it possible to develop dynamically adaptive intelligent tutoring systems very cost effectively, at a small fraction of traditional costs. AuthorIT is used to create a precise representation of what must be learned to master a given body of content, and to layout displays through which TutorIT and students interact. **TutorIT** is a general purpose delivery system that takes the output of AuthorIT as input. It makes all diagnostic and instructional decisions automatically during the course of instruction as might a skilled human tutor.

See Scandura (2007) to learn more about the theory (SLT) guiding the development of AuthorIT and TutorIT. More detailed descriptions of these technologies can be found in Scandura (2005, 2011). The latter publication, entitled "What TutorIT Can Do Better Than a Human and Why: Now and in the Future", is based on an invited address to the Technology, Instruction, Cognition & Learning (TICL) SIG of the AERA. A copy of the published article is available at http://www.scandura.com/tutoritmath/Articles/191-What_TutorIT.pdf.

REFERENCES

Gagne, R. M. Conditions of Learning. N. Y.: Holt, Rinehart & Winston, 1965.

Ritter, S. Authoring Model-Tracing Tutors. Technology, Instruction, Cognition & Learning (TICL), 2005, 2, 3, pp. 231–247.

Anderson, J. R. Rules of the Mind. Hillsdale, NJ: Elbaum, 1993.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., Pelletier, R. Cognitive Tutors: Lessons Learned, Journal of the Learning Sciences, 1995, Vol. 4, No. 2, pp. 167–207.

Koedinger, K. Comparing Knowledge Representations & methods for creating KR in advanced learning & tutorial systems: Production Systems, TICL Symposium, April 10 2006.

Koedinger, K, Mitrovic, T, Ohlsson, S, Scandura, J. M. & Paquette, G. Knowledge Representation, Associated Theories and Implications for instructional Systems: Dialog on Deep Infrastructures, Technology, Instruction, Cognition & Learning (TICL), 2009.

Ohlsson, S. & Mitrovic, A. Fidelity and efficiency of knowledge representations of intelligent tutoring systems. Technology, Instruction, Cognition & Learning (TICL), 2007a, Vol. 5, pp. 101–132.

Paquette, G. Graphical ontology modeling language for learning environments. Technology, Instruction, Cognition & Learning (TICL), 2007a, Vol. 5, pp. 133–168.

Ritter, F. E., S. R. Haynes, et al. (2006). High-level behavior representation languages revisited. Seventh International Conference on Computational Cognitive Modeling (ICCM-2006). Trieste, Italy.

Scandura, J. M. Abstract card tasks for use in problem solving research. Journal of Experimental Education, 1964, 33, pp. 145–148. (a)

Scandura, J. M. An analysis of exposition and discovery modes of problem solving instruction. Journal of Experimental Education, 1964, 33, pp. 149–159. (b)

Scandura, J. M. Structural learning theory in the year 2000. Special Monograph of Journal of Structural Learning and Intelligent Systems (a special monograph), 2001, 14, 4, pp. 271–305. (Similar version sans Appendices in Scandura, J. M. Structural Learning Theory: Current Status and Recent Developments. Instructional Science, 2001, 29, 4, pp. 311–336.) On-Line at: http://www.scandura.com under Knowledge Base

Scandura, J. M. Domain Specific Structural Analysis for Intelligent Tutoring Systems: Automatable Representation of Declarative, Procedural and Model-Based Knowledge With Relationships to Software Engineering. Technology, Instruction, Cognition & Learning (TICL), 2003, 1, 1, pp. 7–57. On-Line at: On-Line at: http://www.scandura.com under Knowledge Base

Scandura, J. M. AuthorIT: Breakthrough in authoring adaptive and configurable tutoring systems. Technology, Instruction, Cognition & Learning (TICL), 2005, 2, 3, pp. 185–230. On-Line at: On-Line at: http://www.scandura.com under Knowledge Base

Scandura, J. M. Abstract syntax tree (AST) infrastructure in problem solving research.. Technology, Instruction, Cognition & Learning (TICL), 2006, 3, 1–2, pp. 1–14. On-Line at: On-Line at: http://www.scandura.com under Knowledge Base

Scandura, J. M. Knowledge Representation in Structural Learning Theory and Relationships to Adaptive Learning and Tutoring Systems. Technology, Instruction, Cognition & Learning (TICL), 2007a, Vol. 5, pp. 169–271. On-Line at: http://www.scandura.com under Knowledge Base

Koedinger, K, Mitrovic, T, Ohlsson, S, Scandura, J. M. & Paquette, G. Knowledge Representation, Associated Theories and Implications for instructional Systems: Dialog on Deep Infrastructures, Technology, Instruction, Cognition & Learning (TICL), 2009, Vol. 6.3, pp. 125–150. On line at http://www.oldcitypublishing.com/

Scandura, J. M. What TutorIT Can Do Better Than a Human and Why: Now and in the Future. Technology, Instruction, Cognition & Learning (TICL), 2011, 8, 53 pp. 175–227. On line at http://www.scandura.com/tutoritmath/Articles/191-What_TutorIT.pdf

Weitz, R., Kodaganallur, V. & Roaenthall, D. Intelligent Tutoring Systems – If You Build (and Open) Them, The Answers Will Come. Technology, Instruction, Cognition & Learning (TICL), 2010, 8–1, pp. 1–10.