Comments on Ansari and Sykes, and Gogus
Suggestions for Future Research

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

TICL has a long term commitment to promoting a deeper understanding of the teaching and learning process, and to encouraging the development of automated tutoring systems based on such understanding. These concerns were the major focus of a special issue of TICL in 2007 (Volume 5, Nos. 2-3-4). The first three articles in that issue (Ohlsson & Mitrovic, 2007; Paquette, 2007 and Scandura, 2007) have important implications for Ansari and Sykes (this issue) and Gogus (this issue). These three articles described three fundamentally different approaches to developing advanced tutoring systems. Each rests on a rigorous, but very different theoretical foundation.

The first of the three articles in that 2007 issue (Ohlsson & Mitrovic, 2007) retained the biologically inspired perspective of traditional production-based ITS systems. At the same time, it emphasized the complexity of production systems and offered a less prescriptive Constraint Based Modeling (CBM) alternative within the same biologically oriented tradition.

The Ansari and Sykes article (this issue) falls in the same category. While accepting basic ITS premises, Ansari and Sykes identify another ITS limitation, namely the inability of ITS systems to support algorithm development. They propose a way to address that limitation based on enthymemes (tacit assumptions). I return to algorithm construction below.

The second article in the 2007 issue (Paquette, 2007) was exemplary of the relational network paradigm, wherein knowledge is represented in terms of relationships (and relationships between relationships). Gogus (this issue) similarly focuses on relational models, albeit in the context of model building. Again, I return to below to the role of relational networks in model building.
The third article in the 2007 issue (Scandura, 2007) details the latest iteration of the Structural Learning Theory (SLT) in which SLT rules based on Abstract Syntax Trees (ASTs) play a central role. Directed graphs (equivalent to Flow Charts) were originally used for this purpose (Scandura, 1971, 1973, 1977). In fact, my original expositions of SLT downplayed differences between the two. Like production systems, directed graphs serve quite well for representing procedural knowledge. They both, however, force one to choose some predetermined level of analysis. AST-based SLT rules are fundamentally different. SLT rules represent both structural/declarative knowledge and procedural knowledge simultaneously at all levels of abstraction. Unlike productions, which purport to have a biological basis, SLT rules are cognitive (theoretical) constructs. They are operationally (behaviorally) defined theoretical constructs that have played a central role in the development of AuthorIT and TutorIT.

Formalisms detailed in the three 2007 articles, production systems, relational networks and AST-based SLT rules, were the subject of published commentary in TICL (Koedinger, Ohlsson, Mitrovic, Scandura & Paquette, 2009). In this commentary, the authors compared, contrasted and debated the three basic approaches. All three formalisms, variants on production systems, relational networks and Abstract Syntax Tree AST-based SLT rules are all computationally equivalent. Anything that can be represented using any one formalism can equally well be represented using any other. As emphasized in Scandura (2005), and confirmed in the above commentary (Koedinger et al, 2009), however, each formalism has different characteristics, benefits and limitations insofar as developing adaptive (intelligent) tutoring systems is concerned.

Biologically inspired approaches (production systems, CBR, etc.) are precise. They are designed to represent what is known about how the mind works and are closely tied to behavior. They face considerable difficulties in ITS development, however, when used to develop ITS systems — precisely because of their complexity. Among the more onerous tasks is devising, implementing, testing and revising appropriate pedagogical strategies based on production systems. It is questionable whether any ITS would have reached practice at all if it hadn’t been for substantial and sustained federal support over a period of decades.

Relational models by way of contrast are broadly applicable and inherently more intuitive. Moreover, they can be made more or less precise, depending on the goals and resources available to developers. Nonetheless, they share a common limitation with traditional ITS systems. Defining, implementing, testing and refining appropriate pedagogical strategies remains a major challenge when knowledge is represented in terms of relational networks. Each body of content
must be considered anew. As with traditional ITS, pedagogical decision making depends inextricably on content semantics.

In short, formal equivalence is not the same as practical viability. The AuthorIT authoring and TutorIT delivery systems offer a unique solution in this context. As detailed in (Scandura, 2011), summarized and updated in the first article (Scandura, this issue), eliminating the need to program pedagogical decision making dramatically reduces development times and costs.

A broad base of TutorIT tutoring systems has been developed in the last year with minimal resources. We now have a broad range of math tutorials ranging from basic facts through algebra 2. A comprehensive tutorial on algebra word problems is in process. Included as well is a series of TutorIT tutorials based on a critical reading workbook series developed back in the 1970s (Lowerre & Scandura, 1974).

We converted this workbook series into a series of adaptive TutorIT tutorials for one reason only. We wanted to demonstrate just how quickly existing (in this case paper) material can be converted into dynamically adaptive (and configurable) TutorIT tutorials. Current iterations of all of these tutorials ready for field testing are available at www.tutoritmath.com.

Let us now return to the Ansari and Sykes (this issue) and Gogus (this issue) articles. These articles in important respects parallel the Ohlsson & Mitrovic (2007) and Paquette (2007) articles, in many respects, they share the same benefits and limitations.

Ohlsson & Mitrovic (2007) as well as Ansari and Sykes (this issue) are addressing what they see as important, albeit different limitations of traditional ITS. Ohlsson & Mitrovic (2007) argue that traditional ITS are too difficult to develop. They propose a simpler but (they argue) equally valid approach called Constraint Based Modeling (CBM). Ansari and Sykes (this issue) identify another limitation of traditional ITS: There has been little if any work explicitly developing ITS for algorithm development.

Ironically, AuthorIT’s AutoBuilder environment (part of AuthorIT) has been supporting the systematic construction of algorithms for some years now (e.g., Scandura, 2003, 2007, 2011). It was designed from inceptions two decades back to support Structural Analysis (SA, Scandura, 2003). SA provides an explicit blueprint for identification and representation of to-be-acquired knowledge via successive and parallel refinement of data and process. SA is perfectly general and explicitly supports knowledge representation (what is to be learned for success) in any content area. It would be challenging but also extremely rewarding to use AuthorIT to develop and TutorIT to deliver a tutorial for tutoring students in
how to develop AST-based knowledge representations. Algorithms are but a special case of ASTs limited to one level of analysis (as was the case with directed graphs used in original formulations of SLT).

We have informally begun to sketch requirements for such a TutorIT tutorial. Flexforms are used in AuthorIT to represent what needs to be learned for success. Carrying this work to fruition, therefore, would require creating one or more AST-based Flexforms such as that shown below in Fig. 1 for solving Word Problems involving area.

In the case of a TutorIT algorithm tutor, the initial step would be to create a set of Flexforms representing what must be learned to build algorithms. The result would then be fed to TutorIT for delivery, exactly as with any other.

Figure 1 also is relevant to the Gogus (this issue) article. Gogus focuses on building cognitive models. From an SLT perspective, other than the way they are displayed, there is nothing unique about model building. Models, including how to build models, are simply one of several important things a student often needs to learn for success. In particular, building models is often very helpful in solving word problems.

---

**FIGURE 1**
Flexform representing what needs to be learned for solving word problems involving area.
For specifics, I refer to current work wherein we are developing prototypes for tutoring on algebra word problems (samples available at www.tutoritmath.com). Learning to solve word problems, includes identifying critical parts of problem descriptions, constructing mental models that represent those critical parts, converting those models into equations and finally solving those equations. These TutorIT tutors explicitly tutor students on how to solve word problems.

It is worth noting that while building mental models is important, it is only one step in the process. Furthermore, it is a step that experts often learn to automate/skip. Rather than having to first creating a model, they go directly to the equations (representing the meaning of word problems). Although still a work in progress, TutorIT word problem tutors are sufficiently well developed to illustrate the basic ideas.

To summarize, the AuthorIT authoring system now provides support for creating and TutorIT for delivering dynamically adaptive (and configurable) tutors in essentially any domain. Moreover, this can be done in a fraction of time normally needed with traditional ITS systems. The same principles and technologies are
used, whether this involves teaching students how to create algorithms or to create models to guide student behavior.

TutorIT is currently available for student learning and for research purposes. While not ready for prime time, we also are working on a process that will allow subject matter experts and educational technologists to use AuthorIT to develop their own TutorIT tutorials. Anyone interested in contributing should contract me at scandura@scandua.com. Please include an short outline of your ideas and/or plans.

REFERENCES


