This issue of Technology, Instruction, Cognition & Learning (TICL) further advances the state of the art internationally in this growing interdisciplinary field. Issue 1 of TICL included articles ranging from formalized methods for representing complex structured knowledge (Scandura) to analysis of alternative approaches to model-based learning and instruction (Seel) and a critique of constructivism based on genetic algorithms in modeling complex learning and behavior (Winn). Issue 2 featured articles on various approaches and emphases in instructional design (Lowyck, Poysa & vanMerrienboer) and distributed learning (Paquette & Rosca). Issue 2 also features Compendium I in which various members of the Editorial Board outline their research objectives, achievements and current activities.

Compendium II in this issue (3) adds contributions by Dijkstra, Durnin, Gibbons, Scandura and Seel. Not surprisingly, given the diversity of views and creativity of the TICL board, these contributions further define and extend the boundaries introduced in Compendium I. Dijkstra traces his research over the years and concludes with a call for more research on teaching problem solving, representing reality and signs denoting that reality, the role of multiple representations and integrating new information with existing knowledge. Durnin presents cogent thoughts both on discernment as a guiding framework for evaluating more specific educational goals and the importance of higher order rules and processes (often called domain independent knowledge) in problem solving. Gibbons emphasizes that instructional designers should become more consciously aware of their role as designers, seek guidance in other design disciplines and use such guidance in developing methods and frameworks in instructional design. Scandura stresses the need for deeper level of understanding in instructional theory. After summarizing classical results in knowledge representation, knowledge assessment, cognition and instruction (e.g., prediction of problem solving results based on higher order knowledge with 98-99% accuracy), he emphasizes a major advance in knowledge representation. First used in software engineering research and later in knowledge representation, Abstract Syntax Trees play a central role in Scandura's current research on AuthorIT. Scandura also identifies a number of fundamental issues in the Structural Learning Theory requiring research. Seel contribution centers on model-centered learning. He traces his early work beginning with representing models in terms of homomorphisms (structural similarities). Seel's research places considerable emphasis on detecting such similarities between one domain and another (e.g., the real world and a model thereof), and the processes for teaching same. Emphasis is placed on the role of instruction on the progression of model learning over time. Related concerns include the question of assessment and the effects of different kinds of instruction.

The first article in Issue 3 by Achtenhagen raises the important issue of authenticity -- making sure that an instructional system meets real world requirements. Issue 3 also includes thoughtful commentaries by Bruder & Hesse and Gibbons, respectively, on the Winn and Seel articles in Issue 1. Each contribution addresses important issues: Without authenticity, any instructional system is of questionable value. This is especially important in corporate training, a central focus in Actenhaen's research, but it also is important in education, especially in the U.S. with its increased emphasis on accountability. Bruder and Hesse raise an equally important issue: They cogently argue that no one philosophical framework has the inside track on "truth": In most cases, they all make similar arguments so it is difficult in practice to distinguish alternatives. It is not so much a question of classical cognitive theory or constructivism or embodied cognition, or genetic algorithms as proposed by Winn, but rather
that each perspective has particular benefits and limitations. Gibbons commentary further extends the scope of model-centered learning. His arguments stress the complexity of the problem, ranging from "We don't really know what a mental model is" to "Mental models are idiosyncratic to their owners".

Although this editorial is not the place for detailed solutions, I feel obliged to raise an issue that is fundamental to further advancement in the field: The large gap between theorizing about learning and instruction and the realization of those theories in automated systems.

The focus in this issue on alternative theoretical frameworks and model-based learning offers a good opportunity to confront this issue directly. In the former context Buder and Hesse emphasize that most frameworks (classical information processing, constructivism, embedded cognition as well as genetic algorithms) make very similar predictions. This raises the question of why it is so difficult to distinguish them in practice. One possible answer is that most such frameworks are formulated so generally that they can accommodate almost anything.¹

Those who design systems, based on theoretical prescriptions, are invariably captives of the software tools and/or tutoring systems currently available. Conversely, programmers and software engineers who assist in software implementation all too frequently lack sufficient conceptual background. Even where this is not the case, the theories themselves fail to provide detailed guidance -- detail that is at once crucial in development and beyond the scope of existent theoretical frameworks. As a result it is often very difficult to distinguish among tutoring systems, irrespective of the theory purportedly used in guiding development.

Taken to extreme, one could argue that theoretical frameworks are entirely useless. That, however, would be a mistake. Such frameworks are NOT useful in explaining, and even less in predicting important phenomena. Rather, they serve to emphasize important phenomena that any scientifically viable (e.g., unambiguous and operational) theory must address. Achtenahgen's analysis of authenticity serves a similar purpose. Instructional systems must address real world goals. If a theory cannot address such goals then it will be of limited value.

Irrespective of framework, the fundamental issue is that of knowledge representation. How are we to represent the knowledge to be acquired? Without a solid foundation to build on, where knowledge is represented in an unambiguous and precise manner, building adaptive learning and instructional systems will remain idiosyncratic and hard to evaluate. Lest I be misunderstood, let me put that differently. To the extent that we can represent to-be-acquired knowledge in an unambiguous and precise manner, and only to that extent, will it be possible to create adaptive learning systems that do exactly what they are supposed to do. Unless we know what a learner is to learn or do, we can never know whether we have achieved our goals.

Model-based learning provides an instructive example. Seel and Gibbons, for example, stress the importance of models in learning. Durnin extends this by emphasizing the importance of higher order (domain independent) knowledge. One point of view is that model-based learning, while critically important, is extremely complicated. According to Gibbons, for example, there is no agreement even on what a model is. Is it a mental model or image to be manipulated internally by the learner? Something requiring genetic algorithms for their representation? How

¹ Indeed, as noted by Gibbons, one can "explain" almost anything (even) using operant conditioning principles. No one today, of course, would seriously propose this as a foundation for model-based learning because any such explanation would be extremely cumbersome and impossible to test in practice.
do we determine the authenticity of a model? Without agreement on such issues, how can one hope to measure (assess) such knowledge? And how can one guide such learning?

These questions can only be answered to the extent that model-based knowledge can be represented in an unambiguous manner. Without a sound knowledge representation on which to build, we will continue to lack the strong guidance necessary for constructing sound learning and instructional systems. More specifically, can model-based knowledge be represented in an unambiguous manner? If so, does the representation support the efficient construction of learning systems? This commentary addresses only with the former issue -- although the latter is of equal importance.

Modern representation systems have generally been based on production systems or relational networks of one sort or another. Among the earliest are Anderson’s ACT-R theory (e.g., 1988) based on productions systems and Scandura’s Structural Learning Theory (SLT) and the closely associated method of Structural (Cognitive Task) Analysis (see Scandura, 2001a, for a recent summary & update). Both production systems and procedures in SLT are fully operational in the sense that they can be directly interpreted and executed on a computer. Other popular approaches are based on semantic/relational networks of one sort or another (e.g., knowledge spaces, conceptual graphs). Semantic networks represent structural knowledge (often hierarchically) in terms of nodes and links between them. Semantic networks have the benefit of representing knowledge in an intuitive fashion. Unlike production systems and SLT procedures, however, they are not easily interpreted (executed on a computer). In effect, each type of representation has important advantages and limitations: Production systems are operational but do not directly reflect structural characteristics of the knowledge they represent. Networks represent structure but are not directly operational. It also is worth noting that most approaches to knowledge representation make a sharp distinction in the way they treat declarative and procedural knowledge, on the one hand, and domain specific and domain independent knowledge, on the other.

A recent article by Scandura (Issue 1) shows that model based knowledge involves both declarative and procedural knowledge. It further shows how both declarative and procedural knowledge can be represented as Abstract Syntax Trees (ASTs). ASTs are fully operational and represent knowledge in an intuitive fashion at multiple levels of abstraction. A basic assumption is that any model can be represented as an interacting system. Such systems include both declarative knowledge, relationships between component operations/procedures in the system, and procedural knowledge, in which the component procedures interact with one another. Further, every model can be represented at multiple levels of abstraction as detailed in Scandura (Issue 1, pp. 25-27).

Reference in the article is made to a railroad crossing system (or model) represented in terms of relationships between train, signal and gate components. In order for a crossing to work properly, the signal must be red and the gate down whenever the train is in the crossing. When the train is not in (near) the crossing, the signal must be green and the gate up. In all other cases, the crossing is not operating properly.

These relationships represent declarative knowledge. Declarative knowledge can be made operational by testing parent-child mappings in ASTs. The analysis of any working crossing goes further. Components in a system normally represent (i.e., can be made more precise) by refining (i.e., defining) them more precisely as operations. Thus, the components “signal” and “gate” designate actions -- operations with their own inputs and outputs. For example, “signal” takes the train position as input and generates (i.e., turns) ”red” or green” as appropriate. “gate”
takes color as an input and generates (i.e., moves) "up" or "down". In effect, these operations interact with one another. Knowledge as to how a system works is hence testable. For example, when the signal operation receives a message (input) from the train that it is in the crossing, the signal operation sends a message to the gate to go down. In effect, actions in one part of the system may effect actions in other parts of the system. This dynamic character of systems is what gives so-called models their power.

The railroad crossing system may seem too simple. For one thing, nothing has been said about mental models in one's head. There is good reason for this. Trying to determine what is going on in a learner's brain is a job for brain researchers, not educators. Whatever the ultimate results of such research it will be very difficult, if not impossible, to link such information to knowledge of particular models one might want a learner might master. Practicality demands that the level of scientific explanation match the level of observation.

The crossing system also lacks the richness apparent in broader discussions of model-based learning, such as those by Seel and Gibbons. My original analysis of mathematical knowledge (Chapter 6 in Scandura, 1973), for example, details such complexities as syntactic and semantic relationships in models, multiple embodiments of systems, homomorphisms between embodiments, higher order rules, etc.

Global frameworks and analyses of this sort are useful in defining domain of applicability. They exemplify the kinds of models that a fully operational representation, such as the ASTs proposed in Scandura (2003), must (or at least should) accommodate. Specifically, frameworks may help identify the kinds of models needed in testing the limits of AST representations. Unlike ASTs, however, general frameworks lack the precision necessary to guide unambiguous implementation of specific instructional systems.

Although designed for general use, universality can only be determined by using ASTs to represent wide varieties of concrete models (e.g., those suggested by one or another global framework). For example, consider the relationship between computation with Arabic numerals (e.g., 435 - 176) and manipulations on corresponding models (e.g., take away using regrouping with Dienes blocks). These relationships may be represented as isomorphic systems. One isomorphic system involves column addition and another column subtraction. Each operation on numerals in column addition, for example, corresponds to a unique operation on (i.e., manipulation of) Dienes blocks, or on an equivalent embodiment (e.g., involving groups of sticks). The system also includes an isomorphism (i.e., operation) representing the equivalence between numerals (e.g., 435) and concrete representations thereof (e.g., 4 blocks representing a hundred, 3 representing ten and 5 representing one). "Knowing" such a system clearly provides the learner with considerable flexibility in behavior: A learner mastering this system should be able to do more than just perform column subtraction and manipulate Dienes blocks. For example, given a concrete embodiment of a subtraction problem, a learner should be able to output (e.g., write) the numeral representing the difference -- or, vice versa.

The second system involving column subtraction involves different operations but the same isomorphic mapping. The parallel between these two systems further illustrates the role of higher order domain independent knowledge. Addition (or combining) and subtraction (or take away) are clearly inverse operations. Given one binary operation, it is possible to construct an inverse operation by starting with the output of the composite operation and "undoing" one of the operations in the composite operation (e.g., constructing subtraction from addition, take away from combining). This process of constructing an inverse operation is a higher order operation, an operation that operates on other operations and/or generates new ones. The fact
that higher order inversion is potentially applicable to any number of binary operations offers the learner considerable leverage in problem solving.\(^2\)

In short, progress in TICL demands more attention to detailed analyses of model-based knowledge. Without analyses leading to intuitive, precise and operational knowledge representations, it will remain difficult to evaluate the relevance of alternative conceptual analyses and operational instructional systems based on such analyses. TICL contributors are encouraged to undertake and report such analyses. For example, one might select various kinds of models and determine whether and how they may be represented as ASTs, or in some alternative formalization. In doing so, one should be cognizant of a fundamental requirement: In order to be useful in building instructional systems, a knowledge representation should both be fully operational and directly reflect underlying semantics. Progress in this direction will raise the level of discourse, identify fundamental issues in knowledge representation and make it easier to communicate. Future progress in TICL lies in the balance.\(^3\)

Joseph M. Scandura, Senior Advisor  
July 16, 2003

\(^2\) The interested reader is referred to Scandura, J.M. Structural Learning Theory in the Year 2000. Journal of Structural Learning and Intelligent Systems (A Special Monograph), 2001, 4, 271-306. This article includes an recent overview of the issue along with an extensive set of detailed references on higher order rules. This article minus the historical overview also appears in Instructional Science, 29, 4, 311-336.

\(^3\) In concluding this editorial/commentary, I should mention that ASTs constitute the foundation upon which my small group is building a new kind of authoring system, called AuthorIT. For a brief overview, see my contribution in Compendium II. A brief demonstration of the prototype system was given at the TICL Master's Conference in April 2003 in Chicago. Interested readers may sign up for a free copy at www.scandura.com.