CHAPTER

40 Learning to Think: Cognitive Mechanisms of Knowledge Transfer

Kenneth R. Koedinger *and* Ido Roll

Abstract

Learning to think is about transfer. The scope of transfer is essentially a knowledge representation question. Experiences during learning can lead to alternative latent representations of the acquired knowledge, not all of which are equally useful. Productive learning facilitates a general representation that yields accurate behavior in a large variety of new situations, thus enabling transfer. This chapter explores two hypotheses. First, learning to think happens in pieces and these pieces, or knowledge components, are the basis of a mechanistic explanation of transfer. This hypothesis yields an instructional engineering prescription: that scientific methods of cognitive task analysis can be used to discover these knowledge components, and the resulting cognitive models can be used to redesign instruction so as to foster better transfer. The second hypothesis is that symbolic languages act as agents of transfer by focusing learning on abstract knowledge components that can enhance thinking across a wide variety of situations. The language of algebra is a prime example and we use it to illustrate *(1)* that cognitive task analysis can reveal knowledge components hidden to educators; *(2)* that such components may be acquired, like first language grammar rules, implicitly through practice; *(3)* that these components may be "big ideas" not in their complexity but in terms of their usefulness as they produce transfer across contexts; and *(4)* that domain-specific knowledge analysis is critical to effective application of domain-general instructional strategies.

Key Words: computational modeling, language and math learning, educational technology, transfer, cognitive tutors, cognitive task analysis, in vivo experiments

"Learning to think" is different from "learning" in that it implies that a learner achieves an increase in more general intellectual capability, rather than just in more specific domain content. Learning to think implies more than learning English, learning math, learning history, or learning science. In other words, learning to think implies transfer (Barnett & Ceci, 2002; Gick & Holyoak, 1983; Singley & Anderson, 1989). Is it possible to "learn to think"; that is, is general transfer possible? A simple "yes or no" answer is not to be expected. A substantial amount of transfer can be achieved, especially with scientifically designed instruction, but full general transfer is a tempting dream, not a practical reality. It is not possible to teach children how to be generally intelligent, experts in all domains, without specific instruction in some domains. But dreams can help inspire action, and the action we need is research and development to understand human learning capacities and constraints and to design and evaluate instruction that achieves *as-general-as-possible* transfer. Progress is required not only in domain-general cognitive and learning science but also in domain-specific analysis of the content domains across and within which we hope to see general transfer.

Components of Learning to Think and How Symbolic Languages Help

One way to learn to think is learning languages with which to think more powerfully. The obvious case is learning a natural language, like English. It seems uncontroversial that a good part of what makes human thinking so powerful is our capability for language (cf. Tomasello, 2000; see Gleitman & Papafragou, Chapter 28). The spoken form of natural language not only facilitates communication and collaborative thinking but also provides a medium for reasoning and logical thinking (Polk & Newell, 1995). The written form further facilitates collaborative endeavors over wide stretches of time and space and is also arguably a vehicle for improving thought. Anyone who has written a manuscript, like this one, is likely to have had the experience that the writing process changes one's thinking and yields a better product than spontaneous speech would have. Clearly, language-learning activities are a huge part of learning to think and a major responsibility of our educational systems.

But natural languages like English are not the only languages we use to enhance our thinking. By "language," in a more general sense, we mean a culturally transmitted symbolic system including any commonly used form of external representation. Examples include symbolic algebra and other mathematical notation systems (e.g., calculus notation, probability and statistics notation), Cartesian graphing and other scientific visualization techniques, and computer programming languages, including the huge and growing number of end-user programming languages (e.g., functions in Excel or html). We create external symbols (including pictures and diagrams; see Hegarty & Stull, Chapter 31) to make abstract ideas more available to our brains' powerful perceptual processing and learning mechanisms (e.g., Goldstone, Landy, & Son, 2010; Koedinger & Anderson, 1990). These forms make the abstract concrete and leverage thinking by allowing easier processing of the abstract ideas (e.g., Larkin & Simon, 1987).

In this chapter we explore two themes. The first theme is the idea that external symbol systems (languages in the broad sense) greatly enhance the power of our thinking and learning. They organize the changes internal to the mind that are necessary to implement learning and transfer (cf. Goldstone et al., 2010; Novick, 1990).

The second theme is that learning grows in "pieces" and thinking involves using those pieces in new combinations. The amazing flexibility humans can exhibit in thinking and learning would not be possible if it were not for their extendable and reusable base of knowledge. Many pieces of knowledge have analogs or near analogs in external symbol systems, such as knowledge of English letters and words, but many do not. Such "knowledge components" (Koedinger, Corbett, & Perfetti, 2010) can include categories for which there is no word (e.g., a category of objects my 2-year-old calls "phones" but which includes remote controls and small blocks of wood). They can be "pragmatic reasoning schemas" (Cheng & Holyoak, 1985; see Evans, Chapter 8) that support correct reasoning about certain kinds of social rules, but not about abstract logical rules. They can be the "intuitions," heuristics, or abstract plans that guide our search, decision making, and discoveries. They can be metacognitive or learning strategies, like knowing to try to "self-explain" a worked example (Chi, Bassok, Lewis, Reimann, & Glaser, 1989) or to cover up one's notes and try to recall what's there while studying (cf. Pashler et al., 2007).

This knowledge component view is supported by researchers who have done detailed cognitive analysis of complex real-world learning domains, such as learning physics (diSessa, 1993; Minstrell, 2001; VanLehn, 1999), mathematics (Koedinger & Corbett, 2006), legal argumentation (Aleven, 2006), or programming (e.g., Pirolli & Anderson, 1985). The view is also supported by attempts to create large-scale computational models of complex human reasoning and problem solving (e.g., Anderson & Lebiere, 1998; Newell, 1990). With respect to transfer, this view echoes the classic "identical elements" conception of transfer (Thorndike & Woodworth, 1901) but is enhanced by advances in cognitive theory and computational modeling (cf. Singley & Anderson, 1989). The identical elements, or units of transfer, are no longer stimulus-response links but are latent representations of components of tasks or their underlying structure. It is not sufficient to provide behavioral task (stimulus-response) descriptions of such elements; rather, we need a language of abstraction for specifying cognitive representations that will often generalize over, or be applicable across, many situations. In computational modeling terms, there need to be "variables" (or some functional equivalent) in the formalism or language used (by cognitive scientists) to represent general cognitive elements (see Doumas & Hummel, Chapter 5).

In this chapter, we begin by elaborating the knowledge component view on transfer. We then provide examples of identifying and modeling knowledge components in algebra and show how these models can be tested in classroom studies. These studies have also identified the missing components—missing both in terms of students lacking the relevant knowledge, as well as cognitive scientists overlooking these pieces in their puzzle of the domain. We then argue that these missing components are intimately tied with the symbolic forms of the language itself. Next, we demonstrate how supporting language acquisition helps students achieve better transfer. Last, we discuss various aspects of applying laboratory-derived instructional principles in the classroom, and we review methods for knowledge component (or cognitive model) discovery from data. The conclusion raises open questions about the insufficiently understood interplay between language-mediated and non-language-mediated processes in transfer and learning to think.

Knowledge Components as Carriers of Transfer

Questions of transfer of learning have been addressed by a number of contemporary cognitive scientists (e.g., Barnett & Ceci, 2002; Singley & Anderson, 1989; see Bassok and Novick, Chapter 21). Much discussion has been around issues of what is "far" transfer, how much transfer instructional improvements can achieve, and what kind of instruction does so. A key observation is that when transfer occurs some change in the mind of the student is carrying that transfer from the instructional setting to the transfer setting. It might be a new skill, a general schema, a new mental model, better metacognition (see McGillivray et al., Chapter 33), better learning strategies, a change in epistemological stance, new motivation or disposition toward learning, or a change in self-beliefs about learning or social roles.[1] In the Knowledge-Learning-Instruction (KLI) Framework, Koedinger et al. (2010) use the term "knowledge component" to include all these possible variations and provide a taxonomy of kinds of knowledge components. Knowledge components are the carriers of transfer.

To better understand how knowledge components act as agents of transfer, one should identify the breadth or scope of applicability of those knowledge components in tasks, problems, or situations of interest. In other words, in how many different kinds of situations does the acquired knowledge apply, and what are its boundaries? Understanding the answer to this question allows us to design instruction that better supports transfer. For example, instruction on computer programming might yield knowledge that applies *(a)* only to programming tasks that are quite similar to those used in instruction, *(b)* to any programming task involving the same programming language, *(c)* to programming in other languages (e.g., Singley & Anderson, 1989), or *(d)* to reasoning tasks outside of programming, like trouble-shooting or "debugging" a set of directions (e.g., Klahr & Carver, 1988).[2]

These two questions suggest a scientific path toward achieving more general transfer or learning to think. A key step along that path is identifying those knowledge components that are as broad or as general as possible in their scope of application. Take, for example, the algebraic task of combining like terms (e.g., 3x + 4x). Students may learn this skill as a mental equivalent of something like "combine each number before each x." This encoding produces correct results in some situations, but not all. It is overly specific in that it does not apply to the cases like x + 5x where the coefficient of x (1) is not visually apparent. It might also yield –3x + 4x as –7x, if the mental skill encodes "number before" too specifically as a positive number rather than more generally as a signed number (e.g., Li et al., 2010). Acquired knowledge may also produce incorrect responses by being overly general. For example, if students have encoded "number before each x" too generally, they may convert 3(x + 2) + 4x to 7x + 2.

Big Ideas and Useful Ideas

In the search for transfer-enabling knowledge components, special attention has been given to the notion of "big ideas," which has been a rallying cry in much educational reform, particularly in mathematics. It is worth reflecting on what the "big" in big idea means. It is often used in contrast with facts, procedures, or skills and associated with concepts, conceptual structures, or mental models. For instance, Schoenfeld (2007, p. 548) makes a contrast between "long lists of skills" and "big ideas," and Papert (2000, p. 721) characterizes school as "a bias against ideas in favor of skills and facts."

A particularly tempting example of this is the general problem-solving strategies of the sort mathematician George Polya (1957) identified in his reflections on his own mathematics thinking and teaching. Experimental efforts to investigate the effect of instruction designed to teach such general problem-solving strategies have met with limited success. Post and Brennan (1976), for instance, found no improvement from teaching general problem-solving heuristics such as "determine what is given" and "check your results." Schoenfeld (1985) developed more specific versions of Polya's heuristics, and Singley and Anderson's (1989, p. 231) analysis of that study suggests an important reason for caution in pursuing the "big idea" approach. They noted that Schoenfeld's heuristics that led to transfer were ones that indicated when the heuristic should be applied, such as "If there is an integer parameter, look for an inductive argument." Other heuristics, such as "Draw a diagram if at all possible," did not indicate conditions of applicability and did not lead to transfer. Students can learn such heuristics, in the sense that they can repeat them back, but they do not get any use out of them because it is not clear when they apply. So a first caution is that sometimes an apparently as-general-as-possible knowledge component may not lead to broad transfer because it is *too vague to be useful*.

Just because a student knows general strategies for working backward, or problem decomposition, does not mean that he or she can successfully execute those strategies in a specific context. Haverty, Koedinger, Klahr, and Alibali (2000) provide an informative example. They investigated college students ability to induce functions, like "y = x*(x – 1)/2," from tables of x-y pairs, like {(2, 1) (3, 3) (4, 6) (5, 10)}. They found that all students engaged in working backward by performing operations on the y values, such as dividing each by the corresponding x value to produce {.5 1 3/2 2}. However, those who succeeded were differentiated from those that did not by recognizing that this pattern is linear (increasing by ½). In other words, it was specific fluency in number sense that distinguished students, not general problem-solving skills that all students manifest. Thus, a second caution regarding the search for big ideas to yield far transfer is that many general concepts or strategies require the learner to obtain domain-specific knowledge in order to apply those general strategies effectively.

A third caution is that some general problem-solving or critical-thinking skills may be relatively easy to learn, in the sense that they are generally acquired without any formal schooling. Lehrer, Guckenberg, and Sancilio (1988) taught third graders a general "problem decomposition heuristic" as part of an intensive 12-week curriculum surrounding the LOGO programming language. Although they found evidence of transfer of some big ideas, they found no evidence of improvement in general problem decomposition as measured by puzzle tasks. It may be that many children acquire the relevant problem decomposition skills through prior experiences. Another example can be found in the self-explanation literature. One consistent and surprising result is that merely prompting students to self-explain improves learning, even without teaching students how to self-explain productively (Chi et al., 1989; Siegler, 2002). While improving the dispositions toward self-explanation is an important goal, it seems that the skill of knowing how to self-explain does not need much support.

The temptation to seek really big ideas is strong and these cautions are nuanced. Statements like the following are indicative: A recent National Academy of Education paper (2009) called for better assessment of "skills such as adapting one's knowledge to answer new and unfamiliar questions." This statement and the surrounding text, which is a call for assessments to measure "higher order, problem-solving skills," implies that there is a general skill of "adapting one's knowledge" that can be acquired, measured, and applied generally. It is quite unlikely, however, that there is a single, general skill of adapting knowledge. Adapting one's knowledge is not a single skill, but, more likely, a complex set of skills that have domain-specific ties—some adaptations come easily, when the domain knowledge is in place, and others do not. There *may* be general skills that students can acquire for better adapting knowledge, but until we have identified assessments that can measure them, we should not assume that they exist.

The search for big ideas assumes that some ideas are sophisticated enough to be applied across a wide range of domains and tasks. But focusing on the complexity or sophistication of the idea itself is not sufficient if we are aiming for more effective transfer and learning to think. The issue is not the size of the idea itself, but rather the size of what the idea opens up for a learner. More precisely, it is about the size of the productive *reuse* of the idea. Some sophisticated concepts indeed get a lot of reuse, but so do some simpler facts, associations, or procedural knowledge components. For instance, knowing the phoneme associated with the letter "s" is not big in the sense of the idea being big—this fact is a small and simple one. However, it is big in the sense of its reuse. As children acquire it and the 36 or so phonemes associated with the 26 letters,[3] a whole new world opens up for them. They can use and reuse this relatively small set of knowledge components to identify (decode and pronounce) a much larger set of words in text. They can thus read words that they have not read before. Many will be words they already know from spoken language and some will be new words, which they can begin to acquire in context. Furthermore, this decoding capability (made possible by this small set of grapheme->phoneme knowledge components) greatly increases the learning capabilities of the child—he or she can now learn by reading in addition to listening, watching, and doing.[4] Thus, rather then simply search for *big* ideas, we should be searching for *useful* ideas.

Small-big ideas are knowledge components that may be quite specific but are big in the scope of their use—they get reused in many contexts. The notion is similar to Gagne's "vertical transfer" in its emphasis on components that can combine with others to produce broader competency. However, unlike vertical transfer, small-big ideas are not limited to within-domain transfer. Small-big ideas may extend beyond their nominal domain to improve performance or learning in other domains. Phonemes are nominally in the reading domain, but acquisition of them improves learning in history, science, math, and so on—all domains in which reading is part of learning. Similarly, the distributive property is nominally in the algebra domain, but acquisition of it (and the cluster of skills associated with it) supports performance and learning in physics, chemistry, engineering, statistics, computer science, and so on—in the STEM disciplines generally.

Testing Knowledge Component Models in Instructional Contexts

Beginning in the 1980s, John Anderson and colleagues have put the series of ACT theories of cognition (e.g., Anderson, 1983) to test through the development of a brand of intelligent tutoring systems, called Cognitive Tutors (Anderson, Corbett, Koedinger, & Pelletier, 1995). Since then the work has greatly expanded in its dissemination—over 500,000 students a year use the Cognitive Tutor Algebra course (e.g., Ritter, Anderson, Koedinger, & Corbett, 2007)—and in its scope—Cognitive Tutors and variations thereof have been created for a wide variety of content, including intercultural competence (Ogan, Aleven, & Jones, 2010), statistics (Lovett, Meyer, & Thille, 2008), chemistry (Yaron et al., 2010), and genetics (Corbett, Kauffman, MacLaren, Wagner, & Jones, 2010). Many large-scale evaluations of these courses have demonstrated their effectiveness (Ritter, Kulikowich, Lei, McGuire, & Morgan, 2007). More important for this chapter, tutoring systems (and online courses more generally) have become platforms for advancing research on thinking and learning in the field and in the context of substantial knowledge-based academic content. This makes it possible to test and advance theories of learning that may be overgeneralized or otherwise inaccurate given their origins in laboratory settings and typically knowledge-lean content. Such technology allows us to carry out well-controlled studies in the classroom environment and to collect detailed moment-by-moment data. Many of the research findings of such research can be found in the open research wiki of the Pittsburgh Science of Learning Center at http://www.learnlab.org/research/wiki.

A key tenet of the ACT-R theory is that human knowledge is modular—it is acquired and employed in relatively small pieces (Anderson & Lebiere, 1998). Although such pieces can be recombined in many different ways, they are not completely abstracted from context. Indeed, a second key tenet is that knowledge is context specific. A procedural form of knowledge (implicit knowledge for doing) is characterized by an if-then production rule notation (see Doumas & Hummel, Chapter 5), whereby the context in which the production applies is specified in the if-part and an associated physical action, subgoal, or knowledge retrieval request is specified in the then-part. Similarly, a declarative form of knowledge (explicit or directly accessible knowledge that can be visualized or verbalized) has retrieval characteristics that depend, in part, on the *context* of other active knowledge (the more that related knowledge is active, the more likely to retrieve the target knowledge).

These tenets lead to two important ideas for instructional design. First, it is possible to create specifically targeted instruction that isolates the learning of a particular knowledge component. Second, it is critical to design instruction so that knowledge components are acquired with appropriate context cues or features so that they generalize or transfer broadly, but accurately. Thus, isolated practice cannot be too decontextualized, otherwise inert or shallow knowledge acquisition may result.

In the process of applying the ACT intelligent tutoring systems to support learning of programming and mathematics, eight principles of instructional design were formulated to be consistent with ACT and with experience in developing, deploying, and evaluating these systems (Anderson, Corbett, Koedinger, & Pelletier, 1995). One of these principles is that tutor design should be based on a knowledge component analysis of the target domain.[5] This principle emphasizes the importance of the modular nature of human knowledge and the great value of domain-specific cognitive task analysis for producing effective instruction. Clark et al. (2007) describe a meta-analysis of seven studies comparing existing instruction with redesigned instruction based on a cognitive task analysis, which yielded an average effect size of 1.7 (i.e., students who received the redesigned instruction scored 1.7 standard deviations better on posttests than did students who received normal instruction).

The theory of transfer, as briefly outlined earlier, makes specific predictions about students' learning. For example, one prediction is that knowledge of a specific component can manifest itself in many different contexts. Students who acquire multiplication knowledge (with general and accurate retrieval features) can apply it to different numbers (e.g., 2*3=?; 6*5=?), to more complex symbolic problems (e.g., 2*3+5=?), and to problems presented as or emerging in a broader situation (Danny bought two pens for $3 each). Another useful hypothesis is that performance on each component improves with practice. Cognitive Tutors and paper-based assessments allow us to put these hypotheses and analysis of the domain to test. By analyzing students' behavior on problems that share knowledge components, we can evaluate whether our analysis of knowledge components is accurate.

This idea can be illustrated with a story about algebra story problems. The cognitive science literature includes statements about how students "find word problems . . . more difficult to solve than problems presented in symbolic format (e.g., algebraic equations)" (Cummins et al., 1988, p. 405). When asked to predict student performance, teachers and educators indicate that algebra story problems are harder for students to solve than matched symbolic equations, since students need to first translate these word problems into symbolic notation (Nathan & Koedinger, 2000).

Koedinger and Nathan (2004) compared students' performance on story problems and matched equations, and discovered that the assumed knowledge component analysis (e.g., that equations are needed to solve story problems) was incorrect. They found that beginning algebra students are actually better able to solve introductory story and word problems than matched equations. For instance, students were 62% correct on word problems such as, "Starting with some number, if I multiply it by 6 and then add 66, I get 81.9. What number did I start with?" but only 43% were correct on matched equations, such as "$x \times 6 + 66 = 81.90$."

One striking fact regarding these studies is the contrast between beliefs of researchers and educators on the one hand and actual student performance on the other. In this example, students' actual performance is at odds with the predictions of scientists and teachers alike. Koedinger and Nathan's (2004) domain analysis revealed previously unrecognized knowledge demands in acquiring symbolic skills. This analysis pinpoints knowledge components for symbolic language comprehension.

Learning to Think by Learning Languages to Think With

The fact that students are better at solving word problems than solving equations may sound counter to our point that learning symbolic languages, symbolic algebra in this case, facilitates thinking. However, our point is not that being "given" a symbolic language suddenly makes one a better thinker, but that the payoff for *learning* a symbolic language is more powerful thinking. That students are still struggling with the language of algebra many months into a course is surprising to many and, ironically, more so to those who have succeeded in doing so. For instance, high school algebra teachers are more likely to make the wrong prediction (equations are easier) than elementary or middle school teachers (Nathan & Koedinger, 2000). It seems that many successful algebra learners do not have good explicit memory for, and perhaps did not have much explicit awareness of, all the work they did (or their brains did) while learning algebra. This observation is consistent with the hypothesis that much (not all!) of algebra learning is done with little awareness of many of the mental changes that are taking place. Although algebra textbooks and classes include lots of verbal instruction, much of the process of learning appears to occur while students are studying examples and practicing on problems (cf. Matsuda et al., 2008; Zhu & Simon, 1987), and neither examples nor problems contain verbal descriptions of the to-be-learned patterns or rules. Many of the pattern, rule, or schema induction processes that carry out this learning are implicit (see Evans, Chapter 8), that is, are not mediated by (nor involve the subvocalization of) the verbal rules read in the text or heard in class.[6]

Although our empirical and theoretical support for this claim comes largely from research in the algebra domain, this substantial nonverbal character of learning may extend beyond algebra. That is, much of our learning even in the academic context may be implicit (nonverbally mediated), particularly so in the STEM disciplines where symbolic notations are so common (e.g., chemical symbols, genetics notations, process diagrams). This claim is less surprising when you consider that the human brain has an amazing capability for learning language (without already having language available). Does this capability stop working once we have learned our first language, so that subsequently we only learn through language and through conscious awareness? That seems unlikely. It seems more likely that the ability to learn languages without language continues to operate even after students begin using language-mediated learning strategies. The hypothesis that nonverbal learning mechanisms are critical to learning formal symbolic "languages," like algebra, is at least worth pursuing.

Learning a symbolic language is hard, not in the sense that it feels hard (though it might), but in the sense that it takes a long time, many months or years, to reach proficiency. Children are acquiring their first spoken language, like English, during the first 4 or 5 years of life. It typically takes a few more years to learn the written form of English, that is, reading and writing. It takes at least a year for most students to (begin to) learn algebra, and most only become fluent as they continue to use (and improve) their algebra skills in downstream STEM courses, such as Algebra II, Calculus, Chemistry, Physics, Statistics, and Programming.

Through work on Cognitive Tutor math projects, it became increasingly apparent to the first author that students' struggles were as or more often with the specifics of the math content than with general strategies for employing it effectively in problem solving. Some targeted instruction on strategies for general problem solving may be effective, but a major challenge for students is learning specific symbolic material (e.g., vocabulary, notational tools, principles) and specific symbolic processing machinery (interpretive procedures) to fuel those general problem-solving strategies.

The Koedinger and McLaughlin (2010) study summarized later in this chapter provides evidence of grammar learning being important to progress in algebra. Computational modeling by Li, Cohen, and Koedinger (2010) indicates that probabilistic grammar learning mechanisms are not only capable of acquiring key aspects of algebra but appear to provide a candidate answer to a mystery in expertise development. Such mechanisms provide an explanation for how learners achieve representational (or "conceptual") changes along the path from novice to expert, accounting not only for their improved performance (accuracy and speed) but also for acquisition of deep features (e.g., Chi, Feltovich, & Glaser, 1981) and perceptual chunks (e.g., Gobet, 2005; Koedinger & Anderson, 1990). Similar learning mechanisms (Bannard, Lieven, & Tomasello, 2009) have been demonstrated to characterize children's language acquisition.

Non-language-mediated learning mechanisms may also be a key part of learning in other STEM domains, which involve specialized symbol systems (e.g., chemistry equations, physics principles, genetics notations) and associated semantics, new vocabulary, and problem-solving processes. These *physical* symbol systems[7] allow our powerful perceptual pattern-finding and structure-inducing learning mechanisms to operate in new abstract worlds that are designed as *visible* metaphors of hidden scientific phenomenon. Learning to see complex scientific ideas in symbolic forms allows experts to off-load long chains of abstract reasoning into physical space (on paper or computer screens). Such chains of reasoning are difficult to do in one's head, without the external memory and perceptual processing support of symbol systems (cf. Goldstone et al., 2010). Experts are certainly capable of chains of mental reasoning without external symbolic support. This reasoning may often be done through subvocalization or subvisualization (in our "mind's eye"), whereby (with experience) we can simulate in our minds what we might have previously done with the external support of a symbolic language (cf. Stigler, 1984).

Improving Transfer With Domain-General and Domain-Specific Approaches

So far we have focused on the role of *domain-specific* symbols and modular knowledge acquisition in facilitating transfer. However, differences in *domain-general* instructional and learning strategies (e.g., spacing practice, comparison, self-explanation) also influence transfer (cf., Koedinger et al, 2010; Pashler et al., 2007). Can such strategies be effectively and straightforwardly applied across domains?

We first illustrate how applying domain-general instructional strategies necessitates addressing the domain-specific question of what are the as-general-as-possible knowledge components. Next, we illustrate how the nature of knowledge components in a domain may change regardless of whether one instructional strategy produces more learning and transfer than another.

Finding the Right Level of Generality to Apply an Instructional Strategy

Consider the Gick and Holyoak (1983) studies that compared the effects on an analogical transfer task of different instructional strategies (see Holyoak, Chapter 13). The domain involves "convergence" tasks whereby a story lays out a problem (e.g., about the need for radiation treatment of high intensity that reaches a brain area, but without damaging the skull and tissue surrounding it), to which the solution involves a dividing of forces along multiple paths that then converge together on a target. Gick and Hoyoak found that the best transfer was achieved by instruction that asked students to compare two examples (or analogs) of a general solution schema in the context of a symbolic abstraction (a diagram) representing the general solution schema. Other studies have also demonstrated the effectiveness of prompting for example comparisons (Gentner et al., 2009; Rittle-Johnson & Star, 2009), or for providing an abstraction of a general rule, pattern, or theory behind solutions (e.g., Judd, 1908; Holland, Holyoak, Nisbett, & Thagard, 1986).

How might we best map laboratory results like these onto educational practice? One challenge is determining the general schema that is the target of instruction or, to put it in more concrete terms, the scope of tasks, examples, or analogs from which to draw for use in instruction and in assessing transfer. It is necessary to have a clear definition of the target knowledge, or, in other words, the as-general-as-possible knowledge components.

Consider the goal of applying these results to the teaching of algebra symbolization, that is, translating story problems to algebraic expressions.[8] Figure 40.1 illustrates the potential complexity of this question (data from Koedinger & McLaughlin, 2010). Is there a general schema that covers all algebra problems (or even broader, covering all math problems or all problems including convergence problems)? Or is the schema something more narrow, like all problems whose solution is a linear expression of the form mx + b? Gick and Holyoak (1983) observed that the level of similarity or dissimilarity of the analogs may play an important role in how much learning and transfer occurs. Analogs with higher similarity have the advantage that students may be more likely to make a reasonable mapping between them and induce a general schema. They have the disadvantage that the schema that is induced may not be as general as it could be.

Here are two symbolization problems that are quite close analogs:

1) Sue is a plumber and gets \$30 for showing up to a job plus \$60 per hour she works. Write an expression for how much Sue makes if she works for x hours. Answer: 60x + 30

2) Rob is an electrician. He gets paid \$50 per hour and also gets \$35 for every job. Write an expression for how much Rob makes if he works for h hours on a job. Answer: 50h + 35

They are both in the form Mx + N, where M and N are small positive integers. They have some different features, including the different numbers (values for M and N) and different cover story features (e.g., "plumber" in one but "electrician" in the other). Analogs that are even closer in similarity are possible, for example, where the cover story is the same, but only the numbers change; or where the cover story changes, but the numbers do not. Analogs that are more dissimilar can also be produced, for example, by changing the type of quantities from money to distances. Structural changes are also possible, for example, introducing problems of the Mx – N form (i.e., story problems with solutions like 5x – 10). As shown in Figure 40.1, student performance on Mx + N and Mx – N forms is quite similar, which suggests that such variation is not too much—does not cross into the disadvantage side of dissimilarity—and thus transfer between such problems may well be achieved. But what about more dissimilar problems?

Would it be better to include an analog with a negative slope, that is, story problems with a solution like 800 – 40x (31% correct)? Might that foster more generalization such that students would be more likely to transfer their learning experience not only to other positive slope problems but also to all problems of the form mx + b, where m and b can be positive or negative? Might we go even further to foster even greater generalization and transfer? The generalization hierarchy of problem types (potential general schemas) in Figure 40.1 illustrates

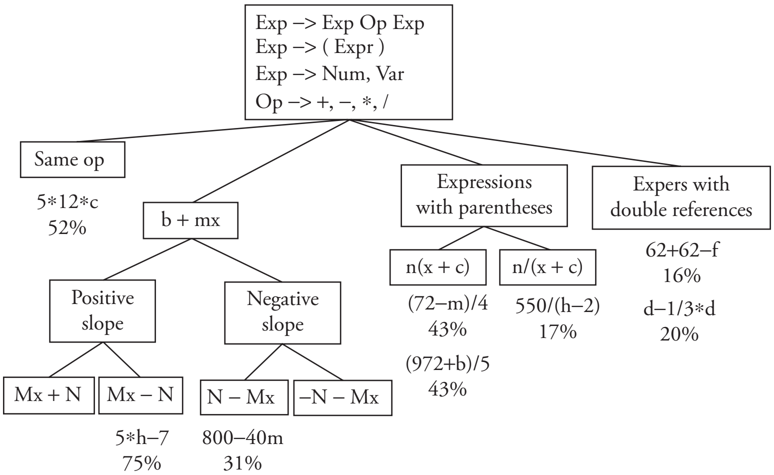

Fig. 40.1 What is (are) the general schema(s) for algebra symbolization? More broadly defined schemas (at the top) increase the *potential* for transfer, but more narrowly defined schemas (at the bottom) increase the probability that a general schema is induced and some transfer is achieved.

a challenge for instructional designers: Even if the instructional principle is clear and well documented, applying it to a domain of interest is not obvious. As shown next, the challenge is greater in cases where different instructional principles suggest competing paths to learning.

Knowledge-Based Dependencies in Applying Instructional Principles

Two competing instructional principles are *learning by doing* and *learning from worked-out examples*. Learning by doing (e.g., Dewey, 1916) essentially suggests that ideas or skills that we are told or shown do not stick, are not robustly learned, unless we use them. A version of this principle is another one of the Cognitive Tutor principles: Provide instruction in a problem-solving context (Anderson et al., 1995). It is related to the "testing effect" (e.g., Roediger & Karpicke, 2006), in that problem solving requires or "tests" recall in targeted contexts and so strengthens the mental link between context and appropriate action.

However, cognitive load theory (e.g., Sweller, 1988; Van Merriënboer & Sweller, 2005) suggests that premature problem-solving practice (e.g., before enough study of worked examples) produces *extraneous* cognitive load. One unfortunate outcome of intuitive instructional design is that it often introduces activities requiring more cognitive processing than necessary, which distracts students from processes relevant to learning (cf. Clark & Mayer, 2003). Indeed, many studies in science, math, and technical domains have demonstrated the "worked example effect," whereby replacing many problems with worked examples enhances learning (see reviews by Renkl & Atkinson, 2010; Pashler et al., 2007). In an apparent contrast with the "testing effect," the worked example principle implies more study (of examples) and less testing (fewer problems to solve). Both principles reject the extremes of all learn by being told or all learn by doing. However, it is unclear whether the greater student assistance recommended by the worked example principle is actually, or just apparently, contradictory to the greater student challenge recommended by the testing effect.

One possible account of the apparent contradiction focuses on differences in the knowledge content involved in the corresponding instructional experiments. Experiments on the testing effect have targeted specific facts (e.g., in language learning), simpler knowledge components, and the corresponding learning theory emphasizes memory processes. In contrast, experiments on the worked example effect have targeted general schemas (e.g., in math and science learning), more complex knowledge components, and the corresponding learning theory emphasizes the induction of schemas (see Holyoak, Chapter 13). Koedinger et al.'s (2010) KLI Framework provides a principled distinction between simpler and more complex knowledge components. Schema induction may be more optimally supported by increased study of examples before turning to practice (e.g., Gentner et al., 2009; Gick & Holyoak, 1983), whereas fact memory may benefit from a faster transition from study to recall practice. However, as far as we know, no one has attempted to test this content-by-principle interaction hypothesis, namely, that a higher ratio of study/examples to test/problems is appropriate for more complex knowledge (schema induction), whereas a lower ratio is appropriate for simpler knowledge (fact memory).

Translating general instructional approaches, such as comparison of examples, to real instructional problems may seem relatively straightforward, but this discussion suggests that it requires the answer to what can be a difficult question: What is the target schema that should guide which analogs are selected, how general should it be, and, correspondingly, how much transfer can be achieved? Typically, instructional designers make a decision based on intuition, but a more scientific approach is possible. The next section provides an example of such an approach.

Discovering a Small-Big Idea: Algebra as Language Learning

Koedinger and McLaughlin (2010) performed an experiment targeting the highest level in Figure 40.1. They identified a general schema (or knowledge component) common to all problems in this broad category. Prior empirical cognitive task analysis (Heffernan & Koedinger, 1998) had demonstrated that students' difficulties in translating story problems to algebra were not so much in understanding the English of the story but primarily in producing algebraic expressions. We hypothesized that a general knowledge component that many students were missing was (probably implicit) knowledge of the grammar of algebraic expressions involving more than one operator (see the top box in Fig. 40.1). Such students can accurately produce algebraic expressions of the form "number operator number" (e.g., 800 – y or 40x), but have trouble producing expressions involving a subexpression like "number operator expression" (e.g., 800 – 40x), that is, learning recursive grammatical patterns. We hypothesized that we could support students in learning of such recursive grammar patterns through exercises isolating the production of two-operator expressions, namely, substitution problems like "Substitute 40x for y in 800 – y." We found that tutored practice on such substitution problems led to greater transfer to performance on translating two-operator stories than did tutored practice on translating one-operator stories (Koedinger & McLaughlin, 2010).

We did not provide any direct instruction on the algebraic grammar, but nevertheless students improved in their algebraic language production. The transfer observed is consistent with the hypothesis that implicit (non-language-based) symbolic language-learning mechanisms are operative even for algebra students.

While the difference in transfer was statistically reliable, it was not large. The large differences in error rates for problems in Figure 40.1 suggest that learning recursive grammar rules is not be the only challenge for students. Problems whose solutions include parentheses are likely harder than otherwise similar problems. In fact, producing the associated expressions requires new grammar rules. It also appears that additional knowledge is needed to address problems in which a quantity serves in multiple roles, like the "d" in "d – 1/3d." These observations about problem difficulty suggest hypotheses for alternative instructional design. Students may benefit, for instance, from more focused instruction on the use of parentheses in algebraic expressions.

The earlier example is illustrative of using student performance data to drive cognitive task analysis, search for as-general-as-possible knowledge components, and correspondingly improve instructional design. The next section describes some strategies for fostering the process of discovering such general knowledge components.

Methods for Discovering As-General-As-Possible Elements of Transfer

We review a number of empirical methodologies for discovering transfer-enabling knowledge components or, in different terms, to perform cognitive task analysis to aid designing effective instruction for transfer. Because so much of learning is not language based and not available to our intuitions, we need techniques that bring theory and data to bear on questions of what constitute ideal instructional objectives, what are the big ideas instruction should target, and how general instructional principles can be best applied in specific content domains to produce transfer.

THINK ALOUD: EMPIRICAL ANALYSIS OF EXPERTS AND NOVICES

Having experts or novices think aloud as they solve tasks in a target domain (Ericsson & Simon, 1984) is a powerful tool for aiding in identifying the knowledge they employ. Chi et al. (1989) used think-aloud methods with physics learners to identify a potential "big-big" idea, *self-explanation*. Designing instruction that prompts students to self-explain has been demonstrated to greatly enhance student learning in a variety of domains (see recommendation #7 in Pashler et al., 2007). As mentioned earlier, Haverty et al. (2000) performed a think-aloud procedure with students engaged in inductive pattern discovery and were surprised to find that success was differentiated not by big-big ideas like general problem-solving strategies, but by small-big ideas that produce fluency with number patterns.

One too-rarely-employed strategy for identifying as-general-as-possible elements of transfer is to use the think-aloud method on tasks for which experts lack domain-specific knowledge. For example, Schunn and Anderson (1999) asked *social* psychologists to design experiments to address a *cognitive* psychology question so as to isolate domain-general scientific inquiry skills from domain-specific experience.

Difficulty Factors Assessment: Experimental Analysis of Task Factors that Reduce Learners' Performance

Wanting to preserve the value of getting empirical data on student task performance yet reduce the costs of think-aloud data collection and analysis, the first author began a strategy of placing item-based experimental designs into classroom quizzes. We have called this approach Difficulty Factors Assessment (DFA), and the Koedinger and Nathan (2004) story problem data described earlier is an example of this approach. A DFA is a factorial design of matched tasks or problems that vary in a multidimensional matrix of factors, for instance, whether the problem is presented in a story, in words, or in an equation, whether the unknown is in the result (4 * 25 + 10 = x) or start (x * 25 + 10 = 110) position, or whether the numbers involved are whole numbers or rational numbers. These items are distributed on multiple forms and administered to students as a quiz. We have run DFA studies in many domains, including algebra problem solving, algebra symbolization, negative numbers, fractions, data display interpretation and production, and finding areas (e.g., Heffernan & Koedinger, 1998; Koedinger & Nathan, 2004). Baker, Corbett, and Koedinger (2007) discuss how DFA studies can be used in instructional design. The idea, described earlier, of using algebraic expression substitution exercises to improve transfer in algebra symbolization, was discovered from DFA studies. In general, identifying the task factors that cause students the most difficulty supports the instructional designer both in focusing their effort on the greatest need, and in testing whether purported forms of instructional assistance actually improve student performance.

Educational Data Mining: Discovering Cognitive Models from Student Performance and Learning Data

As we have fielded more interactive tutoring systems, they have increasingly become valuable sources of data to understand student learning (Cen, Koedinger, & Junker, 2006; Koedinger et al., 2010). In these tutoring systems, students typically solve a series of problems and the system evaluates student performance on a step-by-step basis such that error rate (did they get that step right on their own on the first try) can be logged for each task (each step in each problem). As with DFA studies, such student error data can be used to develop better cognitive models of the factors of problems or tasks that cause students difficulty. One current disadvantage of using tutor data relative to DFAs is that the set of problems given to students in the former type of study are typically not as systematically designed and administered, with matched sets of problems in a Latin square design, as they are in DFA studies.

However, there are also multiple advantages of tutor data over DFA data. First, fielded tutors allow for much more data to be naturally collected as a part of normal system use, that is, without the need to administer a special-purpose paper-based quiz. Second, student performance is automatically graded. Third, the data are more fine grained: whether the student got each step right, rather than just whether the student got the whole problem right. Even if students show their work on paper, if an error is made on an early step in a problem, the rest of the steps are typically absent or suspect. Tutoring systems, on the other hand, provide data on every step because they give students assistance so that early steps are eventually performed correctly and thus the student can attempt every step on his or her own. The fourth, and most important, advantage of tutoring systems over DFAs is that tutor data are longitudinal, providing an indication of changes in student performance over time. Seeing change in performance over time (sequence data) is of critical importance for understanding transfer.

Let us illustrate this point by contrasting how a question of knowledge decomposition (what are the elements of transfer) can sometimes be better addressed using sequence data (from tutor log data) rather than factorial design data (from a DFA). The question in the abstract is whether there is transfer between two tasks that have some core similarity, that is, they share a deep structure (or key aspects of a deep structure) but have some significant dissimilarity, which may be either or both substantial (but solution-irrelevant) surface differences or nonshared aspects of the deep structure. Call these tasks "dissimilar analogs." The knowledge component question concerns the components that are in common in these dissimilar analogs and what components, if any, are specific to each analog.

Consider task A and task B as dissimilar analogs where the average success rate on A is 43% and B is 27%. Table 40.1 shows an example of two such tasks.

One of the simplest knowledge component models to characterize this situation is that the harder task B requires two knowledge components, say K1 and K2, whereas the easier task A, requires just K1. This "*overlap*" knowledge component model predicts transfer between instruction on one of these tasks and performance on another. This situation is illustrated in Table 40.1 where task A (in the first row) is modeled by a single knowledge component representing algebra grammar knowledge for producing a recursive expression (an expression, like "72 – m," inside another expression, like "(72 – m)/4"). Task B is represented by two knowledge components, one for comprehending English sentences and translating them to math operations, like 72 – m and x/4, and the other is the overlap, the knowledge for producing a recursive expression.

A competing *nonoverlap* model corresponds to the idea that these dissimilar analogs are not (functionally) analogs at all, but they are different topics and draw on different skills or concepts. Each task involves a separate knowledge component, say Ka for task A and Kb for task B. This model predicts no transfer. How can we use data to determine which is the correct model?

According to the "identical knowledge components" theory of transfer, the overlap knowledge component model predicts that successful instruction (e.g., tutored practice) on task A will improve performance on task B, whereas the nonoverlap knowledge component model predicts no improvement. We can distinguish these models if we have sequence data that provide performance on task B after task A for some students and before task A for others. If performance on task B is better after task A than before it, then the overlap knowledge component model is the better model. For the tasks shown in Table 40.1, students who saw Task B after seeing two substitution problems (isomorphic to Task A) indeed achieved reliably greater success, 38% correct, as compared to 27% for students who had not seen substitution problems.[9]

More generally, computer-collected student performance and learning data have been used to evaluate cognitive models and to select among alternative models (e.g., Anderson, 1993; Ohlsson & Mitrovic, 2007). Automated methods have been developed to search for a best-fitting cognitive model either purely from performance data, collected at a single time (e.g., Falmagne, Koppen, Villano, Doignon, & Johannesen , 1990), or from learning data, collected across multiple times (e.g., Cen et al., 2006).

Table 40.1 Two Dissimilar "Analogs" With Different Problems but Similar Solutions

	Problem	Solution	% Correct	Knowledge Components Needed
A	Substitute 72 – m for d in d/4. Write the resulting expression.	(72 – m)/4	43%	RecExprProd[a]
B	Ann is in a rowboat on a lake. She is 800 yards from the dock. She then rows for *m minutes* back toward the dock. Ann rows at a speed of 40 yards per minute. Write an expression for Ann's distance from the dock.	800 – 40x	27%	EngToMathOps[a] + RecExprProd

[a]EngToMathOps = English comprehension and translation to math operations, such as 40x and 800 – y. RecExprProd = Recursive expression production, like 800 – 40x from 40x and 800 – y.

We should also be alert for decomposition opportunities, where the transfer may not be at the whole problem level (problem schemas), but at the level of intermediate steps (step schemas or knowledge components). The concept of a problem schema, which is emphasized in most studies both in the psychology (e.g., Gentner et al., 2009; Gick & Holyoak, 1983) and educational psychology (e.g., Sweller & Cooper, 1985) literature, mostly ignores the important componential character of the *construction of* human intelligence. Transfer of knowledge to novel situations comes as much or more from the reconfiguration or recombination of smaller pieces of knowledge into new wholes than from the analogical application of larger pieces of knowledge.

Conclusions and Future Directions

Learning With and Without Language

We have focused on an aspect of learning to think that involves the leverage of symbolic systems or languages to enhance thinking and learning (see Gleitman & Papafragou, Chapter 28). This kind of learning to think is rather direct because before one can make use of a new language to aid thinking, one must first learn that language. We have construed learning languages broadly to include learning symbolic patterns (syntax, perceptual chunks) and semantic interpretations of those patterns, as is done in many STEM disciplines. A child learning his or her first natural language is the prime example of the fact that the human mind can learn language without knowing any language to support that learning. Let us call the learning processes used in this case *non-language-mediated (NLM) learning.*[10] NLM learning is responsible not only for language acquisition but also for learning of other knowledge components (e.g., How do you know how much salt to put on your food?). Such processes include perceptual learning (e.g., Gobet, 2005), unsupervised statistical learning (e.g., Blum & Mitchell, 1998), and some supervised learning in which the learner imitates or induces from correct examples (Gentner et al., 2009; Hummel & Holyoak, 2003) or gets nonverbal negative feedback from incorrect actions (e.g., Matsuda et al., 2008; VanLehn, 1987).

On the other hand, it seems clear enough, given how much instructors talk, that language is an important part of the academic learning process (cf., Michaels, O'Connor, & Resnick, 2008). Let us call the learning processes used in this case *language-mediated (LM) learning*.

Here are some related questions for future cognitive science research on learning and thinking:

1) *NLM learning*: What are the learning mechanisms that humans use to learn a language (at least their first language) without using language? What are the mechanisms behind learning by watching, by doing after watching, or after nonverbal feedback?

2) *LM learning*: What are the learning mechanisms that humans use to make use of language in learning other things? What are the mechanisms of learning by listening, by reading, by writing, by doing after verbal explanation, or after verbal feedback?

3) *Continued use of NLM learning*: How do NLM learning mechanisms continue to be used by learners after they have acquired the relevant language?

4) *Separate or subservient*: Do NLM and LM processes operate independently or do LM learning processes work by calling upon NLM learning processes?

LANGUAGE IMPROVES THINKING

We have evidence for the "language improves thinking" claim in the domain of algebra (Koedinger et al., 2008). We see that students who have learned the language of algebra are much more likely to solve a particular class of complex story problems (which do not absolutely require equations) than students who have not learned the language of algebra.[11]

Some research has shown that prompting students to think with a symbolic language can enhance learning (Roll, Aleven, & Koedinger, 2009; Schwartz, Martin, & Pfaffman, 2005). In both of these studies, students who were asked to reason using mathematical symbols acquired a more general, transferable representation knowledge than students who were not instructed to use mathematical notations. Other studies have shown that prompting students to explain their reasoning (in English) as they solve problems or study examples helps them acquire deeper understanding of the target knowledge components (Aleven & Koedinger, 2002; Chi, De Leeuw, Chiu, & LaVancher, 1994).

Experimental evidence that language improves thinking has been collected in other domains as well. For example, 3-year-old children given instruction on relational language (e.g., *big*, *little*, *tiny*) make better abstract inferences than children without such symbolic support (Gentner, 2003). Beyond algebra, others have also argued that human-invented symbol systems, like computer modeling languages, are "literally languages and accordingly offer new cognitive tools" (Goldstone & Wilensky, 2008).

Language-Mediated Versus Non-Language-Mediated Learning, Expert Blind Spot, and Effective Educational Design

A key point of this chapter is that too much instructional design is suboptimal because it is driven by memories of LM learning experiences. It is not driven by memories of NLM learning processes because these processes and the resulting tacit changes in knowledge are hard for learners to reflect upon. In essence, because of NLM learning processes, experts have worked harder and know more than they realize. Indeed, experts often have difficulty describing what they know (e.g., Biederman & Shiffrar, 1987). As illustrated earlier, instructors and educators can have *expert blind spots* whereby their own expertise may lead them to overestimate students' abilities with the normative problem-solving strategies (e.g., use of algebraic equations). In general, our intuitions about what and how to teach are underinformed. Thus, cognitive task analysis can provide a powerful tool for improving our intuitions and producing more effective instruction (cf. Clark et al., 2007).

More effective educational practice could be achieved through a better understanding of the role of NLM learning in academic learning. The popular (and appropriate) rejection of the "transmission model" of instruction is a step in the right direction. Students usually do not learn simply by being told. Unfortunately, the constructivist alternative (cf., Tobias & Duffy, 2009) is sometimes *misinterpreted* to essentially mean students must teach themselves and instructors need not teach at all! A more sophisticated interpretation suggests that it is the students who should be primarily doing the talking, and the teachers' role is to get them talking (Michaels, O'Connor, & Resnick, 2008). While there is much merit in this idea, it is driven by intuitions that LM learning is where all the action is. The merits of classroom dialog can be bolstered by a complementary emphasis on the role of NLM learning (example induction and repeated practice with feedback) and its interplay with LM learning.

Knowledge Representation and Transfer

SMALL-BIG IDEAS

The focus on big ideas tends to ignore the importance of *small-big* ideas, that is, *specific* facts, skills, or concepts that have a very high frequency of reuse. Furthermore, the learning of big-big ideas is often mediated by the learning of external symbols—new ideas are often associated with new vocabulary. For example, new mathematical ideas like the distributive property have concise symbolic descriptions, and learning the syntax of those symbolic descriptions strengthens and may even seed the semantics of the underlying ideas. How can we identify the small-big and big-big ideas that are most valuable to pursue in improving educational improvement?

SEARCH FOR APPROPRIATE GRAIN SIZE OF TRANSFER

A related issue is the instructional decision of what level of analog to choose to target. This decision is a nontrivial part of testing the generality of basic cognitive science research in educational contexts. Can we develop empirical and theoretical approaches to identify the ideal level of generality that instructional design should target to best enhance learning and transfer?

DOES UNDERSTANDING (VIA LANGUAGE-MEDIATED LEARNING) OCCUR BEFORE OR AFTER PRACTICE (NON-LANGUAGE-MEDIATED LEARNING)?

Is understanding necessary for transfer, that is, does understanding lead transfer? Or is understanding emergent from transfer, that is, does "understanding follow transfer"? To the extent that much of learning to think is about learning specialized languages, it may be that what it means to "understand" is to develop a "metalanguage" (e.g., words like "term" and "coefficient" in algebra or "conjugation" in second language learning) that one can use to describe and reflect on the language that has been learned, as well as give words and symbols to knowledge components that were acquired via MLM learning mechanisms. This kind of understanding, that is, the development of such metalanguage, may as often be a consequence of NLM learning, rather than a source of it. Here the idea/skill is first acquired in nonverbal form (e.g., students who have learned to remove a "coefficient" in an algebra equation but don't know the term), and later the student may learn the language to describe what he or she learned. However, other examples suggest the reverse route. For example, asking students to reason with data prior to giving them instruction may facilitate mental representations that support the subsequent acquisition of procedural competencies (Roll, Aleven, & Koedinger, 2009; 2011; Schwartz & Martin, 2004; Schwartz, Martin, & Pfaffman, 2005).

A Literally Physical Symbol System

Thinking of these pieces of mental function or "knowledge components" as "symbols," as proposed in Newell and Simon's (1976) physical symbol system hypothesis, is both helpful and potentially misleading. Cognitive scientists can certainly describe these pieces in symbols, whether we use English or a computational modeling language. However, the cognitive scientist's symbolic representation of a piece of mental function, like the code that implements a neural network model, is not the mental function itself. Whether the mental pieces *are* symbols we will leave to others (e.g., see Nilsson, 2007). The important claim here is that representing mental pieces or knowledge components in symbolic form, in some language, facilitates thinking on the part of the cognitive scientist or analyst.

Closing

The notion of cultural transmission is suggestive of language, but, while quite powerful, language is not the sole contributor to learning to think. The human brain is built on an animal brain that has perceptually based forms of learning that are effective despite functioning without the use of language (see Penn & Povinelli, Chapter 27). These learning mechanisms, we have argued, are still in use by adult learners with language abilities. It is an interesting and important scientific goal to either disprove this hypothesis or to better identify and understand these NLM learning processes and how they interact with LM learning. Such an endeavor will also have important practical consequences for educational improvement.

Notes

1. That such changes are not directly observable poses a challenge to scientific advance. One way to gain leverage is to create scientific languages to describe these hidden changes, as has been frequently done, for instance, for genes, elements in chemical compounds, or atoms. That is why computational modeling is so important to advance cognitive science and fuel better educational applications.

2. Or, for another example, reading instruction could target learning to decode (sound out) a particular word list (e.g., cat, dog, run, etc.) and the scope of that knowledge, if acquired, might reasonably be reading of those words in the context of the many sentences in which they might appear. Or, in contrast, reading instruction might use related words (e.g., cat, fat, car, far, etc.) to target particular letter-to-sound mappings (e.g., c, f, a, t, and r), and the scope of that knowledge, if acquired, might reasonably be the reading of the many words that involve those letter-to-sound mappings in the many sentences in which they might appear.

3. Estimates of the number of phonemes vary. See http://www.spellingsociety.org/journals/j30/number.php#top.

4. Note that decoding is necessary but not sufficient for comprehension. Good decoders may not be good comprehenders, but bad decoders are bad comprehenders. Furthermore, children have a head start on comprehension through their spoken language experience and decoding opens doors to greater conceptual and vocabulary acquisition that can expand comprehension.

5. The original principle called for a "production rule" analysis, which focuses, in ACT-R terms, on procedural knowledge. We generalize to "knowledge component" analysis (Koedinger, et al., 2010) to include declarative knowledge and to account for declarative as well as procedural transfer (cf., Singley & Anderson, 1989).

6. The point is not that algebra students are unaware of learning algebra in the whole or in the strong sense of implicit learning used, for instance, in paradigms like Reber (1967) grammar tasks. Instead, we mean that there are many details being learned (like the grammar of algebra equations) that are not the result of verbally mediated reasoning. Students have difficulty explaining many of these details and, especially, how they came to know them.

7. We use "physical" here not in the sense of Newell and Simon (1976) of having a physical existence in human minds, but in the more literal sense that the symbols exist, on paper, white boards, computer screens, and so on, in perceptually available spaces.

8. Algebra symbolization is a particularly important area of algebra learning. Today computers can solve algebraic equations and such capabilities are increasingly available even on mobile phones. However, translating the semantics of a problem situation into abstract symbolic form will remain a task for humans for quite a while.

9. This statistically reliable difference ($\chi^2(1, N = 303) = 4.68$, $p = .03$) comes from alternative analysis of data in Koedinger and McLaughlin (2010).

10. Why have we used "non-language-mediated learning" instead of "implicit learning"? Implicit learning is learning that one is not aware of, but it may be possible that one is sometimes aware of the results of NLM learning and thus such learning would not qualify, at least empirically, as implicit learning. It is straightforward to tell when language was not used (by a teacher or in instructional materials) as part of instruction. That's clearly NLM *instruction*. Pinpointing NLM *learning* is harder as subvocalization and self-explanation may count as LM learning. One form of evidence is when we can create computational models that can learn from examples without being given any verbal instructions that behaviors of the model match those of human learners (e.g., Matsuda, Lee, Cohen, & Koedinger, 2009).

11. Data from Study 2 in Koedinger et al. (2008) show that when a student correctly solved a complex equation, the student was 82% correct on the matched complex story problem, but when the student failed on the complex equation, he or she was only 44% correct on the matched story problem. That is, symbolic competence enhances reasoning.

References

Aleven, V. (2006). An intelligent learning environment for case-based argumentation. *Technology, Instruction, Cognition, and Learning*, *4*(2), 191–241.

Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, *26*(2), 147–179.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*(2), 167–207.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahway, NJ: Erlbaum.

Baker, R. S. J. d., Corbett, A. T. & Koedinger, K. R. (2007). The difficulty factors approach to the design of lessons in intelligent tutor curricula. *International Journal of Artificial Intelligence in Education*, *17(4)*, 341–369.

Bannard, C., Lieven, E., & Tomasello, M. (2009). Modeling children's early grammatical knowledge. *Proceedings of the National Academy of Sciences*, *106*(41), 17284–17289.

Barnett, S. M., & Ceci, S. J. (2002). When and where do we apply what we learn? A taxonomy for far transfer. *Psychological Bulletin*, *128*(4), 612–637.

Biederman, I., & Shiffrar, M. M. (1987). Sexing day-old chicks: A case study and expert systems analysis of a difficult perceptual learning task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *13*(4), 640–645.

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of Eleventh Annual Conference on Computational Learning Theory (COLT)* (pp. 92–100). New York: ACM Press.

Cen, H., Koedinger, K. R., & Junker, B. (2006). Learning factors analysis: A general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, T-W. Chan (Eds.), *Proceedings of the 8th International Conference on Intelligent Tutoring Systems* (pp. 164–175). Berlin, Germany: Springer-Verlag.

Cheng, P. W., & Holyoak, K. J. (1985). Pragmatic reasoning schemas. *Cognitive Psychology*, *17*, 391–416.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, *13*, 145-182.

Chi, M. T. H., de Leeuw, N., Chiu, M. H., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Science*, *18*(3), 439–477.

Chi, M. T. H., Feltovich, P. J., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*, 121–152.

Clark, R. E., Feldon, D., van Merriënboer, J., Yates, K., & Early, S. (2007). Cognitive task analysis. In J. M. Spector, M. D. Merrill, J. J. G. van Merriënboer, & M. P. Driscoll (Eds.), *Handbook of research on educational communications and technology* (3rd ed., pp. 577–593). Mahwah, NJ: Erlbaum.

Clark, R. C., & Mayer, R. E. (2003). *e-Learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning*. San Francisco, CA: Jossey-Bass.

Corbett, A., Kauffman, L., MacLaren, B., Wagner, A., & Jones, E. (2010). A cognitive tutor for genetics problem solving: Learning gains and student modeling. *Journal of Educational Computing Research*, *42*(2), 219–239.

Cummins, D., Kintsch, W., Reusser, K., & Weimer, R. (1988). The role of understanding in solving word problems. *Cognitive Psychology, 20,* 405–438.

Dewey, J. (1916). *Democracy and education*. New York: Macmillan.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and Instruction*, *10*(2 & 3), 105–225.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. Cambridge, MA: The MIT Press.

Falmagne, J-C., Koppen, M., Villano, M., Doignon, J-P., & Johannesen, L. (1990). Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, *97*, 201–224.

Gentner, D. (2003) Why we're so smart. In D. Gentner, & S. Goldin-Meadow (Eds)., Language in mind: Advances in the study of language and thought (pp. 195–236). Cambridge, MA: MIT Press.

Gentner, D., Loewenstein, J., Thompson, L., & Forbus, K. D. (2009). Reviving inert knowledge: Analogical abstraction supports relational retrieval of past events. *Cognitive Science*, *33*, 1343–1382.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, *15*, 1–38.

Gobet, F. (2005). Chunking models of expertise: Implications for education. *Applied Cognitive Psychology*, *19*, 183–204.

Goldstone, R. L., Landy, D. H., & Son, J. Y. (2010). The education of perception. *Topics of Cognitive Science*, *2*(2), 265–284.

Goldstone, R. L., & Wilensky, U. (2008). Promoting transfer by grounding complex systems principles. *Journal of the Learning Sciences*, *17*(4), 465–516.

Haverty, L. A., Koedinger, K. R., Klahr, D., & Alibali, M. W. (2000). Solving induction problems in mathematics: Not-so-trivial pursuit. *Cognitive Science*, *24*(2), 249–298.

Heffernan, N. & Koedinger, K. R. (1998). A developmental model for algebra symbolization: The results of a difficulty factors assessment. In M. A. Gernsbacher & S. J. Derry (Eds.), *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society* (pp. 484–489). Mahwah, NJ: Erlbaum.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*, *110*, 220–264.

Judd, C. H. (1908). The relation of special trai[illegible] general intelligence. *Educational Review*, *36*, 28–42.

Koedinger, K. R., & Anderson, J. R. (1990). Abstract planning and perceptual chunks: Elements of expertise in geometry. *Cognitive Science*, *14*, 511–550.

Koedinger, K. R., & Corbett, A. T. (2006). Cognitive tutors: Technology bringing learning science to the classroom. In K. Sawyer (Ed.), *The Cambridge handbook of the learning sciences* (pp. 61–78). New York: Cambridge University Press.

Koedinger, K. R., Corbett, A. C., & Perfetti, C. (2010). The Knowledge-Learning-Instruction (KLI) framework: Toward bridging the science-practice chasm to enhance robust student learning. CMU-HCII Tech Report 10–102. Accessible via http://reports-archive.adm.cs.cmu.edu/hcii.html.

Koedinger, K. R., & McLaughlin, E. A. (2010). Seeing language learning inside the math: Cognitive analysis yields transfer. In S. Ohlsson & R. Catrambone (Eds.), *Proceedings of the 32nd Annual Conference of the Cognitive Science Society* (pp. 471–476). Austin, TX: Cognitive Science Society.

Koedinger, K. R., & Nathan, M. J. (2004). The real story behind story problems: Effects of representations on quantitative reasoning. *The Journal of the Learning Sciences*, *13*(2), 129–164.

Klahr, D., & Carver, S. M. (1988). Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. *Cognitive Psychology*, *20*, 362–404.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*(1), 65–99.

Lehrer, R., Guckenberg, T., & Sancilio, L. (1988). Influences of LOGO on children's intellectual development. In R. E. Mayer (Ed.), *Teaching and learning computer programming: Multiple research perspectives* (pp. 75–110). Hillsdale, NJ: Erlbaum.

Li, N., Cohen, W. W., & Koedinger, K. R. (2010). A computational model of accelerated future learning through feature recognition. In V. Aleven, J. Kay & J. Mostow (Eds.), *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 368–370). Heidelberg, Berlin: Springer.

Lovett, M., Meyer, O., & Thille, C. (2008). JIME - The Open Learning Initiative: Measuring the effectiveness of the OLI statistics course in accelerating student learning. *Journal Of Interactive Media In Education*, *2008*(1). Retrieved February 16, 2011, from http://jime.open.ac.uk/2008/14

Matsuda, N., Cohen, W., Sewall, J., Lacerda, G., & Koedinger, K. R. (2008). Why tutored problem solving may be better than example study: Theoretical implications from a simulated-student study. In A. Esma and B. Woolf (Eds.), *Proceedings of the 9th International Conference of Intelligent Tutoring Systems* (pp. 111–121), Berlin: Springer.

Matsuda, N., Lee, A., Cohen, W. W., & Koedinger, K. R. (2009). A computational model of how learner errors arise from weak prior knowledge. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society* (pp. 1288–1293). Cognitive Science Society, Amsterdam, Netherlands.

Michaels, S., O'Connor, C., & Resnick, L. B. (2008). Deliberative discourse idealized and realized: Accountable talk in the classroom and in civic life. *Studies in the Philosophy of Education*, *27*(4), 283–297.

Minstrell, J. (2001). Facets of students' thinking: Designing to cross the gap from research to standards-based practice. In K. Crowley, C. D. Schunn, & T. Okada (Eds.), *Designing for science: Implications for professional, instructional, and everyday science* (pp. 415–443). Mahwah: Erlbaum.

Nathan, M. J., & Koedinger, K. R. (2000). An investigation of teachers' beliefs of students' algebra development. *Cognition and Instruction*, *18*(2), 207–235.

National Academy of Education. (2009). *Standards, assessments, and accountability: Education policy white paper*. (L. Shepard, J. Hannaway, & E. Baker, Eds.). Washington, DC: Author.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, *19*.

Nilsson, N. (2007). The physical symbol system hypothesis: Status and prospects. In M. Lungarella (Ed.), *50 years of AI* (pp. 9–17) Berlin: Springer.

Novick, L. R. (1990). Representational transfer in problem solving. *Psychological Science*, *1*, 128–132.

Ogan, A., Aleven, V., & Jones, C. (2010). Advancing development of intercultural competence through supporting predictions in narrative video. *International Journal of Artificial Intelligence in Education*, *19*(3), 267–288.

Ohlsson, S., & Mitrovic, A. (2007). Fidelity and efficiency of knowledge representations for intelligent tutoring systems. *Technology, Instruction, Cognition and Learning*, *5*(2), 101–132.

Papert, S. (2000). What's the big idea: Towards a pedagogy of idea power. *IBM Systems Journal*, *39*, 3–4.

Pashler, H., Bain, P., Bottge, B., Graesser, A., Koedinger, K., McDaniel, M., & Metcalfe, J. (2007). Organizing instruction and study to improve student learning (NCER 2007–2004). Washington, DC: National Center for Education Research, Institute of Education Sciences, U.S. Department of Education.

Pirolli, P. L., & Anderson, J. R. (1985). The acquisition of skill in the domain of programming recursion. *Canadian Journal of Psychology*, *39*, 240–272.

Polk, T. A., & Newell, A. (1995). Deduction as verbal reasoning. *Psychological Review*, *102*(3), 533–566.

Polya, G. (1957). *How to solve it: A new aspect of mathematical method*. (2nd ed.). Princeton, NJ: Princeton University Press.

Post & Brennan (1976). An experimental study of the effectiveness of a formal versus an informal presentation of a general heuristic process on problem solving in tenth-grade geometry. *Journal for Research in Mathematics Education*. *7*(1), 59–64

Ritter, S., Kulikowich, J., Lei, P., McGuire, C. L., & Morgan, P. (2007). What evidence matters? A randomized field trial of cognitive tutor algebra I. In T. Hirashima, U. Hoppe, & S. S. Young (Eds.), *Supporting learning flow through integrative technologies* (Vol. 162, pp. 13–20). Amsterdam, Netherlands: IOS Press.

Ritter, S., Anderson, J. R., Koedinger, K. R., & Corbett, A. (2007). Cognitive tutor: Applied research in mathematics education. *Psychonomic Bulletin and Review*, *2*, 249–255.

Rittle-Johnson, B., & Star, J. (2009). Compared to what? The effects of different comparisons on conceptual knowledge and procedural flexibility for equation solving. *Journal of Educational Psychology*, *101*(3), 529–544.

Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Learning and Verbal Behavior*, *6*, 855–863.

Renkl, A., & Atkinson, R. K. (2010). Learning from worked-out examples and problem solving. In J. Plass, R. Moreno, & R. Brünken (Eds.), *Cognitive load theory and research in educational psychology* (pp. 91–108). New York: Cambridge University Press.

Roediger, H. L., & Karpicke, J. D. (2006). The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science*, *1*, 181–210.

Roll, I., Aleven, V., & Koedinger, K. R. (2009). Helping students know 'further' - increasing the flexibility of students' knowledge using symbolic invention tasks. In N. A. Taatgen & H. van Rijn (Eds.), *Proceedings of the 31st annual conference* *of the cognitive science society* (pp. 1169–1174). Austin, TX: Cognitive Science Society.

Roll, I., Aleven, V., & Koedinger, K. R. (2011). Outcomes and mechanisms of transfer in invention activities. In L. Carlson, C. Hölscher, & T. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 2824–2829). Austin, TX: Cognitive Science Society.

Schoenfeld, A. H. (1985). *Mathematical problem solving*. Orlando, FA: Academic Press.

Schoenfeld, A. H. (2007). Problem solving in the United States, 1970–2008: Research and theory, practice and politics. *ZDM Mathematics Education*, *39*, 537–551.

Schunn, C. D., & Anderson, J. R. (1999). The generality/specificity of expertise in scientific reasoning. *Cognitive Science*, *23*, 337–370.

Schwartz, D. L., & Martin, T. (2004). Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction. *Cognition and Instruction*, *22*(2), 129–184.

Schwartz, D. L., Martin, T., & Pfaffman, J. (2005). How mathematics propels the development of physical knowledge. *Journal of Cognition and Development*, *6*(1), 65–88.

Siegler, R. S. (2002). Microgenetic studies of self-explanation. In N. Granott & J. Parziale (Eds.), *Microdevelopment - transition processes in development and learning* (pp. 30–58). New York: Cambridge University Press.

Singley, K., & Anderson, J. R. (1989). *The transfer of cognitive skill*. Cambridge, MA: Harvard University Press.

Stigler, J. W. (1984). The effect of abacus training on Chinese children's mental calculation. *Cognitive Psychology*, *16*, 145–176.

Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, *12*, 257–285.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, *2*, 59–89.

Thorndike, E. L., & Woodworth, R. S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*, *8*, 247–261.

Tobias, S., & Duffy, T. M. (2009). *Constructivist theory applied to instruction: Success or failure?* New York: Routledge.

Tomasello, M. (2000). *The cultural origins of human cognition*. Cambridge, MA: Harvard University Press.

VanLehn, K. (1987). Learning one subprocedure per lesson. *Artificial Intelligence*, *31*, 1–40.

VanLehn, K. (1999). Rule learning events in the acquisition of a complex skill: An evaluation of Cascade. *Journal of the Learning Sciences*, *8*(1), 71–125.

Van Merriënboer, J. J. G., & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review*, *17*(1), 147–177.

Yaron, D., Karabinos, M., Evans, K., Davenport, J., Cuadros J., & Greeno, J. (2010). Learning chemistry: What, when, and how? In M. K. Stein & L. Kucan (Eds.), *Instructional explanations in the disciplines* (pp. 41–50). New York: Springer Science+Business Media.

Zhu, X., & Simon, H. A. (1987). Learning mathematics from examples and by doing. *Cognition and Instruction*, *4*(3), 137–166.