J·T·L·A

The Journal of Technology, Learning, and Assessment

Volume 8, Number 5 · February 2010

Specificity of Structural Assessment of Knowledge

David L. Trumpower, Harold Sharara,
& Timothy E. Goldsmith

www.jtla.org

A publication of the Technology and Assessment Study Collaborative
Caroline A. & Peter S. Lynch School of Education, Boston College

JTLA

Volume 8, Number 5

Specificity of Structural Assessment of Knowledge

David L. Trumpower, Harold Sharara, & Timothy E. Goldsmith

Editor: Michael Russell
russelmh@bc.edu
Technology and Assessment Study Collaborative
Lynch School of Education, Boston College
Chestnut Hill, MA 02467

Copy Editor: Jennifer Higgins
Design: Thomas Hoffmann
Layout: Aimee Levy

JTLA is a free online journal, published by the Technology and Assessment Study Collaborative, Caroline A. & Peter S. Lynch School of Education, Boston College.

Preferred citation:

Trumpower, D.L., Sharara, H., & Goldsmith, T.E. (2010). Specificity of Structural Assessment of Knowledge. *Journal of Technology, Learning, and Assessment, 8*(5). Retrieved [date] from http://www.jtla.org.

Abstract:

This study examines the specificity of information provided by structural assessment of knowledge (SAK). SAK is a technique which uses the Pathfinder scaling algorithm to transform ratings of concept relatedness into network representations (PFnets) of individuals' knowledge. Inferences about individuals' overall domain knowledge based on the similarity between their PFnets and a referent PFnet have been shown to be valid. We investigate a more fine grained evaluation of *specific links* in individuals' PFnets for identifying particular strengths and weaknesses. Thirty-five undergraduates learned about a computer programming language and were then tested on their knowledge of the language with SAK and a problem solving task. The presence of two subsets of links in participants' PFnets differentially predicted performance on two types of problems, thereby providing evidence of the specificity of SAK. Implications for the formative use of SAK in the classroom and in computer-based environments are discussed.

J·T·L·A

Specificity of Structural Assessment of Knowledge

David L. Trumpower
Harold Sharara
University of Ottawa
Timothy E. Goldsmith
University of New Mexico

Introduction

The need for objective, easy to construct, and easy to score measures of deep-level understanding (e.g., conceptual knowledge) is well recognized by those in the field of educational assessment. In response to such needs, a procedure known as structural assessment of knowledge (SAK) has been developed. As it is commonly employed, SAK provides a general/overall measure of domain knowledge, best suited to summative assessment (Goldsmith & Johnson, 1990; Goldsmith, Johnson, & Acton, 1991). In order to be useful for formative purposes, however, an assessment tool must provide more detailed information about students' strengths and weaknesses. That is, a formative assessment tool must be able to (a) identify students' precise knowledge gaps and/or misunderstandings, and (b) provide feedback that can be used to fill the gaps and remediate misunderstandings (Earl, 2003; McManus, 2006). In this paper we investigate the specificity of information provided by SAK. More specifically, we examine the relationship between the presence of specific subsets of links in students' knowledge structures derived via SAK and their performance on different types of problems in a computer programming domain. As such, this study explores whether or not SAK meets the first requirement of a formative assessment tool[1]—the ability to identify specific areas of strength and weakness.

We begin by describing the general SAK procedure, followed by a review of evidence for the validity of inferences that can be made from SAK regarding overall domain knowledge. We then discuss some preliminary indications of the specificity of information provided by SAK and describe the present study.

Structural Assessment of Knowledge (SAK)

SAK refers to a procedure for evaluating the organization of an individual's knowledge within a particular domain. SAK is based on the premise that knowledge requires not only acquiring facts, procedures, and concepts, but also having an understanding of the *interrelationships* among those facts, procedures and concepts—i.e., the structure of a domain's content (Goldsmith & Johnson, 1990). This notion is consistent with the volumes of expert-novice research results, which show that experts possess more knowledge and, perhaps more importantly, better organize knowledge than novices. Expert knowledge is stored in the form of schemas that are organized around higher-level domain principles, whereas novice knowledge is often organized around superficial domain features (e.g., Chi, Feltovich, & Glaser, 1981; Larkin, McDermott, Simon, & Simon, 1980; Schoenfeld & Herrmann, 1982; Weiser & Shertz, 1983). Accordingly, SAK evaluates the *structure* of an individual's knowledge.

Recent theories of learning and cognition stress the importance of knowledge organization in the development of expertise (e.g., Anderson, 1995; Marshall, 1995). The prevailing view of cognitivists today is that humans store knowledge as associative networks of ideas, concepts, procedures, and other forms of knowledge. During learning, new knowledge is integrated into the network by linking it to semantically related prior knowledge. The structure of one's knowledge has been implicated in recall, inferencing, comprehension, and problem solving (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004; Baxter, Elder, & Glaser, 1996; Trumpower & Goldsmith, 2004).

Consequently, knowledge organization has been recognized as important in the fields of education and educational assessment. In their exploration of recent research on the science of learning and its link to classroom practice, the Committee on Developments in the Science of Learning and the Committee on Learning Research and Educational Practice concluded that "Effective comprehension and thinking require a coherent understanding of the organizing principles in any subject matter…" and that "Transfer and wide application of learning are most likely to occur when learners achieve an organized and coherent understanding of the material…" (National Research Council, 2000, pp. 238–239). Similarly, the National Research Council recommends that "Assessments should evaluate what schemas an individual has…" and that "This evaluation should include how a person organizes acquired information…" (National Research Council, 2001, p. 102). Although traditional assessment techniques may allow knowledge organization to be indirectly inferred, SAK does so more directly. In this respect, SAK is similar to concept maps, although there are some critical differences which we will discuss later.

Generally, SAK involves three phases: 1) knowledge elicitation, 2) knowledge representation, and 3) knowledge evaluation. Following is a description of each of these three phases.

In the *knowledge elicitation* phase, an individual uses a rating scale to judge the relatedness of all pairwise combinations of a set of concepts taken from the domain of interest (Figure 1, next page). Typically, a domain expert or group of experts will determine the most critical concepts in the domain to be assessed, either by generating a list of key concepts or by listing the steps required to solve a problem or complete some process (i.e., task analysis). The number of concepts chosen, n, determines the number of concept pairs to be rated in accordance with the equation, $n(n-1)/2$. For example, a set of 12 concepts would result in the need to collect 66 relatedness ratings. Although Goldsmith, Johnson, and Acton (1991) showed that the predictive validity of SAK increases with larger numbers of concepts for sets ranging from 5 to 30, it is expected that sets much larger than 30 will result in decreased validity due to student fatigue. Also, due to time constraints, classroom applications of SAK likely cannot exceed about 20 concepts.

Figure 1: Example Relatedness Rating Task with Experimental Design Concepts

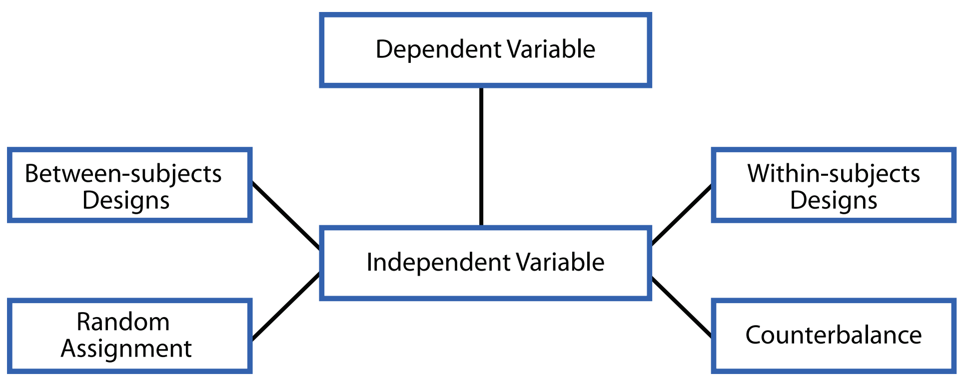
Directions: Please rate the relatedness of the terms below. Terms can be related in many ways—they can be in the same category, used in a similar way, or even related by time. We would say that "bird" and "nest" were highly related as well as "hurt" and "ambulance", "early" and "morning", and so forth.

For each of the pairs of terms listed below, select a number from 1 to 5 to indicate how related you think the terms are. Smaller numbers mean less related and larger numbers mean more related. Use what you have learned about the terms to make your ratings. Try not to spend more than 10 to 15 seconds to decide how related a pair is. We are interested in your first impressions. Once you have selected a rating, circle the corresponding number on your answer sheet. Please work quickly, but accurately.

		Less Related				More Related
counterbalance	random assignment	1	2	3	4	5
within-subjects design	between-subjects design	1	2	3	4	5
between-subjects design	dependent variable	1	2	3	4	5
independent variable	counterbalance	1	2	3	4	5
random assignment	independent variable	1	2	3	4	5
independent variable	within-subjects design	1	2	3	4	5
random assignment	between-subjects design	1	2	3	4	5
dependent variable	independent variable	1	2	3	4	5
between-subjects design	counterbalance	1	2	3	4	5
within-subjects design	random assignment	1	2	3	4	5
dependent variable	random assignment	1	2	3	4	5
counterbalance	within-subjects design	1	2	3	4	5
independent variable	between-subjects design	1	2	3	4	5
counterbalance	dependent variable	1	2	3	4	5
dependent variable	within-subjects design	1	2	3	4	5

In the *knowledge representation* phase, relatedness ratings are transformed via the Pathfinder scaling algorithm into a structural representation of the individual's knowledge. The Pathfinder algorithm is available in the Knowledge Network Organizing Tools (KNOT) software (Schvaneveldt, Sitze, & McDonald, 1989; available at http://interlinkinc.net/). The resulting structural representation is referred to as a Pathfinder network, or PFnet for short. A PFnet is a network comprised of nodes and links. Nodes represent each of the rated concepts, whereas links represent relatively strongly perceived relationships between concepts. Pathfinder treats relatedness ratings as proximities. The Pathfinder algorithm works by searching for the shortest indirect path between each pair of concepts. A direct link between two concepts is included in the PFnet only if the shortest indirect path between those two concepts is greater than the direct path (see Schvaneveldt, 1990, for a more complete description of Pathfinder; available for download at http://interlinkinc.net/Ordering.html). Thus, it is not the absolute magnitude of a rating that determines whether or not a link between the rated concepts will occur in the PFnet. Rather, it is the relative magnitude of the rating in comparison with all other ratings. In this way, there is no "right" or "wrong" rating. Figure 2 shows an example of a PFnet.

Figure 2: Example PFnet of Experimental Design Concepts



In the *knowledge evaluation* phase, the individual's PFnet is evaluated by comparing it to a referent PFnet. The referent PFnet is typically derived from the averaged ratings of a set of instructors and/or other domain experts. Acton, Johnson, and Goldsmith (1994) have shown that the averaged ratings of multiple experts provide a better referent than any individual expert or instructor. They suggest that although different experts may show variability in their judgements of concept relations, this variability often appears to be the result of random error rather than systematic differences in conceptual thinking. Similarity between an individual and referent PFnet can be quantified as the number of links shared by the two networks (in graph theoretic terms, the "intersection") divided by the number of links found in either of the two networks (in graph theoretic terms, the "union"). This network similarity measure, which we will refer to as PFSIM, ranges from 0 to 1, with values closer to 1 indicating greater similarity to the referent PFnet and, hence, better conceptual knowledge.

Each one of these phases of the general SAK procedure can be conducted in several ways. For example, in the knowledge elicitation phase, one might determine the most important concepts to be assessed by applying automatic text analysis techniques to large corpuses of text (Montemurro & Zanette, 2009) or by simply examining chapter titles and headings from textbooks (Cooke, 1987), and proximities may be generated from the co-occurrence of concepts in a student-written text rather than obtaining relatedness ratings (Clariana & Wallace, 2007). In the knowledge representation phase, one could use Multidimensional Scaling rather than Pathfinder to transform the proximities into a visual representation (Goldsmith & Johnson, 1990). And in the knowledge evaluation phase, a referent-free measure of internal coherence could be used instead of PFSIM to evaluate the PFnets (Acton, 1991). Consideration of the strengths and weaknesses of all of the many different possibilities is beyond the scope of this paper, but is examined in some depth by Schvaneveldt (1990). The particular method described above is perhaps the most extensively studied and so was used in the present study. The evaluation phase, however, was extended to include comparison of specific subsets of links in addition to the overall evaluation provided by the PFSIM measure.

At this point, it may be realized that PFnets appear very similar to concept maps. Therefore, SAK is similar to the use of concept maps for evaluative purposes (cf. Novak & Gowin, 1984). Both techniques evaluate the "goodness" of a student's visually-displayed knowledge representation. In both techniques, the visual representation that is evaluated is a set of linked concepts. However, SAK differs from concept mapping in several ways. First, because concept maps are directly constructed by the students themselves, they require student training. SAK, instead, simply requires students to make judgments of concept relatedness. These judgments require minimal instruction of what is meant by "relatedness" which can usually be achieved through presentation of some everyday examples. Second, concept mapping typically requires students to label or describe links that represent what are believed to be the relatively more important concept relationships in a domain. SAK, on the other hand, does not require labeling/describing links; again, it only necessitates that students make numerical judgements of concept relatedness. Therefore, SAK may be less dependent on language abilities than concept mapping (Schau, Mattern, Weber, Minnick, & Witt, 1997). Third, students are fully aware of the structure of concept maps as they are constructing them. Thus, they may be constrained by their own biases to construct maps that are visually or structurally appealing (e.g., hierarchical, symmetrical, and/or uncluttered by lots of links, especially cross links). PFnets, on the other hand, are determined from students' relatedness ratings. Because it is very unlikely that one could mentally translate raw relatedness ratings into the corresponding PFnet, SAK is not likely to be affected by any such self-imposed constraints. The point of this discussion is to highlight the features of SAK that distinguish it from concept mapping (e.g., lesser training requirements, unlabeled links, implicit elicitation of structure) and why we think that they may be relevant. Whether or not these differences have any effect on the validity of inferences drawn from the two approaches remains to be tested empirically.

Validity of Inferences Based on SAK

An increasing number of studies demonstrate the validity of inferences made from SAK when used for the purpose of measuring overall domain knowledge. For example, evidence based on relations to other variables has been obtained by showing that the similarity between student and expert PFnets was positively related to course grades in a teacher education course in elementary mathematics (Gomez, Hadfield, & Housner, 1996), to course points earned in a research techniques course (Goldsmith, et al., 1991), to scores on an essay exam covering the topic of evolution (d'Appolonia, Charles, & Boyd, 2004), and to other performance measures (Day, Arthur, & Gettman, 2001; Kraiger, 1993; Trumpower & Goldsmith, 2004). In addition, studies have shown that the similarity between student and expert PFnets increases following instruction in a variety of domains and situations, including a human resources management course (Acton, 1991), a computer programming training program (Davis & Curtis, 1996), and a naval decision making task (Kraiger, Salas, & Cannon-Bowers, 1995). Collectively, these studies suggest that SAK allows valid inferences to be made about overall domain knowledge across a diverse array of domains, ranging from those that are more procedural (e.g., computer programming) to those that are more conceptual (e.g., evolution).

Specificity of SAK

We use the term "specificity" to indicate the ability of an assessment tool to identify specific areas of strength and weakness, as opposed to indicating overall level of competence. In each of the above mentioned studies, SAK was used to produce overall measures of network similarity (e.g., PFSIM) which were compared to overall measures of performance (e.g., course grades, exam scores, etc). Although overall measures such as PFSIM are useful for summative assessment, they are less useful for providing specific feedback to students and teachers that may be used formatively to help focus instruction. That is, network similarity measures only indicate *how much* student structures differ from referent structures; they do not indicate specifically *in what ways* structures differ. As an illustration of this point, consider two students of Introductory Research Design whose knowledge of several basic research design concepts is assessed by SAK. Suppose that a referent PFnet contains the links shown in Figure 2 (page 8). Further, suppose that Student X's PFnet contains the exact same links as the referent except that it is missing the link between *counterbalance* and *within subjects design*, while Student Y's PFnet contains the exact same links as the referent except that it is missing the link between *random assignment* and *between subjects design*. Under this scenario, both Student X and Student Y would have identical PFSIM values (intersection = 6, union = 7, PFSIM = 6/7 = .86) indicating that they possess the same *amount* of overall knowledge but they have different missing links from their PFnets. If we assume that these links are missing due to a lack of understanding of the specific relationship between the two associated concepts, then we might expect Student X and Student Y to make very different types of errors when designing experiments—Student X could be expected to design poor *within subjects* designs whereas Student Y could be expected to design poor *between subjects* designs. Differentially identifying these weaknesses could not be accomplished on the basis of PFSIM (or other overall measures of similarity) alone, as both students had identical PFSIM values. A central goal of the current study is to use a more fine grained analysis of specific links in students' PFnets.

There is some evidence from prior studies that alludes to the specificity of information captured by links in structural knowledge representations. For instance, Dayton, Durso, and Shepard (1990) showed participants the following riddle: "A man walks into a bar and asks for a glass of water. The bartender pulls a shotgun on the man. The man says, "thank you" and walks out. What missing piece of information would cause the puzzle to make sense?" Later, participants' structural knowledge of the riddle was assessed with SAK. The relatedness ratings of 14 concepts relevant to the riddle were obtained and used to generate a PFnet for each participant.

Rather than evaluate the PFnets with one of the commonly used overall measures such as PFSIM, the authors examined specific links between concepts that they deemed crucial for solving the riddle. The resulting PFnets of those who solved the riddle and those who did not solve the riddle were compared. The PFnet of Solvers contained a link between the concepts *remedy* and *glass of water*, a link between *remedy* and *surprise*, and a link between *surprise* and *shotgun*. The PFnet of Non-solvers, on the other hand, did not contain these three links. Therefore, the presence of a specific subset of links in the PFnets was able to predict whether or not participants solved the riddle. Apparently, Solvers realized that the glass of water asked for by the man, and the surprise caused by the bartender's shotgun, were both remedies for the man's hiccups. This study illustrates the capacity of specific subsets of links, rather than the overall PFnet, to predict performance on a cognitive task.

In a study of statistics problem solving, Trumpower, Guynn, and Goldsmith (2004) found that different types of practice problems led to the acquisition of a specifically hypothesized subset of links in participants' PFnets. It was predicted that traditional types of problems, in which students are given values for certain variables and are then asked to solve for a specific unknown goal, would lead to acquisition of links between the goal concept and other irrelevant concepts due to the strong focus on the goal. It was further predicted that goal-free problems would shift focus away from a single goal, thereby allowing acquisition of more pedagogically relevant links. Results supported these predictions—those in the goal free condition possessed more relevant links (as determined by statistics experts) and fewer irrelevant links with the goal concept. Further, those in the goal free condition displayed better problem solving performance than those in the standard goal condition. These results show that different experiences can lead to different links in one's PFnet, and that individuals who possess different links in their PFnets perform differently on related problem solving tasks. Thus, an analysis of specific links in PFnets may be used to identify deficiencies in prior learning (i.e., acquisition of missing and misdirected relational knowledge) and to predict future problem solving performance.

In order to better assess the specificity of PFnets derived from SAK, a task domain is needed with multiple types of problems, each of which can be associated with a different subset of links. With this sort of problem domain, both convergent and divergent evidence regarding the specificity of links in PFnets derived from SAK can be assessed. That is, the absence of one subset of links could be used to identify a particular weakness as indicated by poor performance on a related type of problem, whereas the absence of a second subset of links could be used to identify a different weakness as indicated by poor performance on a different type of problem.

This strategy for assessing convergent and divergent validity evidence is much like the multi-trait multi-method strategy (Campbell & Fiske, 1959). Link subset A and problem type A are multiple methods for measuring trait A (in this case knowledge of the relationships amongst a specific set of concepts), whereas link subset B and problem type B are multiple methods for measuring trait B (knowledge of the relationships amongst a different set of concepts). Link subset A should be related to performance on problem type A but not to performance on problem type B, whereas link subset B should be related to performance on problem type B but not to performance on problem type A.

Present Study

In the present study, participants were provided information to be learned about a computer programming language. Following a period of study, participants were asked to solve a series of problems requiring knowledge of the programming language. Two different types of problems were included, each determined by a pair of subject matter experts to require understanding of a different set of concept relations. Participants were also asked to rate the relatedness of pairs of concepts from the programming language, so that PFnets could be derived. It was hypothesized that the presence of a specific subset of links in participants' PFnets would be related to their performance on the first type of problem but not the second, and that, conversely, the presence of a different subset of links would be related to performance on the second type of problem but not the first, thereby providing convergent and divergent evidence for the specificity of SAK.

Method

Participants

Participants were 35 undergraduate psychology students who participated for partial course credit. None of the participants had ever taken a course in computer science, nor had any computer programming experience.

Problem Solving Domain

The domain used was a simple programming language that was custom designed for use in an earlier series of studies (see, e.g., Trumpower & Goldsmith, 2004). The language was modeled after the Pascal programming language and was limited to the implementation of sorting algorithms. Sorting algorithms take a random array of objects, for example letters, and arrange them in some predefined way, such as alphabetical order. The language contained both data structures (e.g., lists, elements of a list, indices to designate list elements) and control structures (e.g., go-to, if-then statements). Although the language was limited in scope (consisting of just 12 key concepts) so that naïve students could learn much about the language in a relatively short amount of time, it contained programming concepts found in more general languages. Hence, it was complex enough to construct a variety of challenging programming problems. For definitions of the 12 concepts which comprised the language, see Appendix A in Trumpower and Goldsmith (2004).

Instrument Development

Problem Solving Task

Eight selected response problems were constructed to assess participants' understanding of the programming language. The problems presented lists of letters arranged in a particular order, along with pointers used to reference the letters. Beneath the lists were several lines of programming code. Problems asked participants to determine how the code would change the list of letters or move the pointers, or to determine what missing lines of code would transform the list from one order to another.

The problems were intended to be complex enough so that the solution depended on integration of several interrelated concepts. Performance on one of the problems, however, was perfect and, thus, was not included in any of the following statistical analyses. To solve five of the remaining problems, participants needed to know the relationships between the concepts P*osition, Pointer, Assign*, and *Increment*. That is, they must know that

Assign is used to place a *Pointer* in a specific *Position* and that *Increment* is used in conjunction with a specific *Pointer* to increase the *Position* of that *Pointer* by one. These problems will be referred to as "Pointer-type problems." Three of the problems required knowledge of the relationships between the concepts *If-Then*, *Go-To*, and *Step*. More specifically, these three problems required knowing that *Go-To* is used to change program control to a specific *Step* and that the *Go-To* procedure can be used in conjunction with *If-Then* to change program control only under certain circumstances. These problems will be referred to as "Go-To-type problems." It should be noted that two of the problems can be classified as both a Pointer and Go-To-type problem. Figure 3 (next page) shows an example of each problem type.

Confirmation that the problems did, indeed, require the relational knowledge described above was provided by the two developers of the programming language who were utilized in the current study as subject matter experts. One of the subject matter experts noted that a distinction is made in teaching computer programming languages between data structures and control structures, and verified that the simple programming language used in the current study required students to understand both of these ideas. Both subject matter experts agreed that the Pointer-type problems require understanding of how data structures work together, whereas the Go-To-type problems require understanding of how control structures work together.

Figure 3: Example Problems Used in this Study

Example Pointer-type problem:

Begin State:

List:	A	B	C	D	
Positions:	1	2	3	4	
Pointers:					* #

End State:

List:	A	B	C	D
Positions:	1	2	3	4
Pointers:		*		#

Step	Instruction
1	Assign Pointer * to 1
2	______________________________
3	______________________________

-Increment Pointer #
-Go-To Step 2
-Assign Pointer # to 2
-If Pointer * is less than Pointer #, Then Increment Pointer *

Example Go-To-type problem:

List:	E	D	C	A	B	
Positions:	1	2	3	4	5	
Pointers:						* #

Step	Instruction
1	Assign Pointer * to ______
2	Assign Pointer # to ______
3	If Letters indicated by Pointers * and # are Ordered, Then Go-To Step 5
4	Stop
5	Increment Pointer *

Step 5 will only be executed if Pointer * is Assigned to ____ and Pointer # is Assigned to ____?

1 and 3
2 and 3
2 and 5
3 and 5
4 and 5

Relatedness Rating Task

The set of concepts chosen for inclusion in the ratings task were the 12 concepts that comprised the programming language. This yielded 66 pairwise combinations of concepts to be rated. All 66 combinations were presented in random order. Concept pairs were presented side by side, with the left-right ordering of concepts randomly determined. Next to each concept pair was a 5-point rating scale (1=Not at all related, 5=Very related). Although the Pathfinder algorithm is not limited to 5-point scales, we have found that the 5-point scale allows for acceptable variation in responses, without creating too heavy of a cognitive load. It also contains a midpoint for students who are unsure whether a pair of concepts is related or not.

Instructions provided an explanation of what is meant by relatedness. They also asked participants to complete the task quickly, but accurately. The same format and instructions were used as those displayed in the example relatedness ratings task in Figure 1 (page 7).

Procedure

Participants were allowed 15 minutes to study the material describing the programming language. Next, participants rated the relatedness of all pairwise combinations of the 12 key concepts of the programming language. Upon completion of the rating task, participants attempted to solve the eight programming problems. Both the problem solving and relatedness ratings task were completed using paper and pencil. Participants were given as much time as required to complete the ratings and problem solving tasks, but most took no more than approximately 15 minutes on each task.

Analysis and Hypotheses

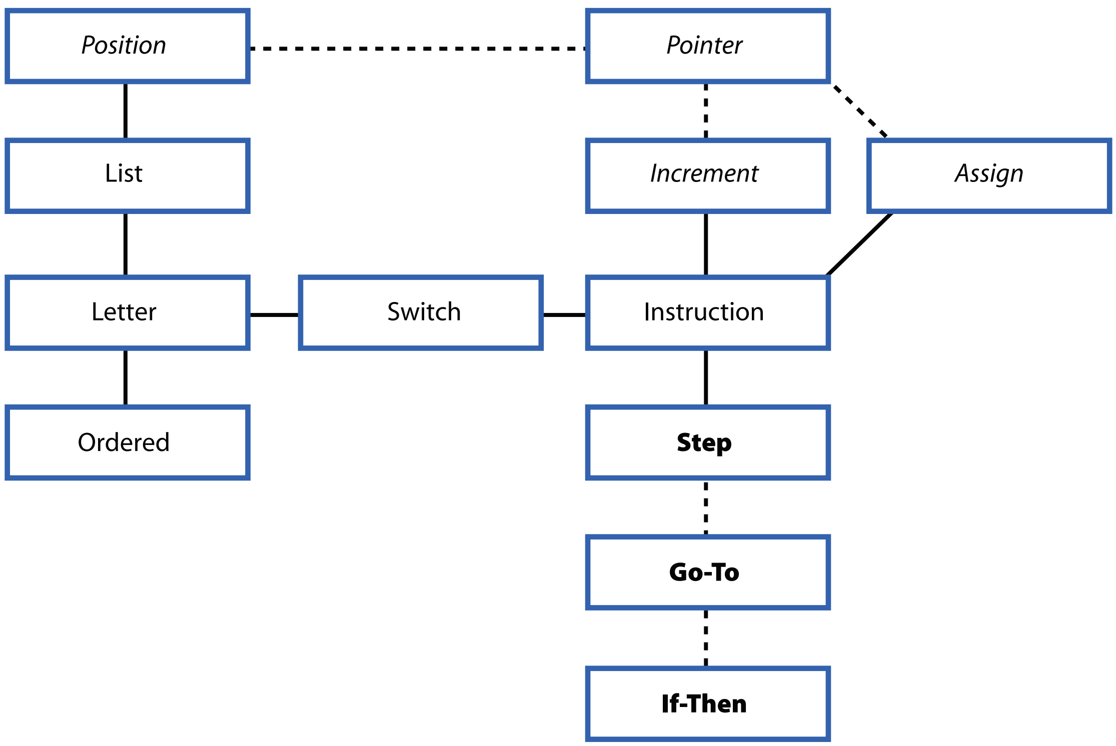
Solutions to each problem on the problem solving task were scored as 0 or 1. A score of 1 was obtained if the correct choice was selected. Some of the problems required participants to select more than one option for solution (see, e.g., the Pointer-type problem in Figure 3, page 16). In these problems, a score of 1 was obtained only if *all* of the correct options were selected. Partial credit was not considered appropriate because the choice for one blank could only be considered correct relative to the choice for other blanks. Stated differently, a correct line of code in one blank coupled with an incorrect line of code in another blank did not seem to indicate partial knowledge, as such a solution would not move the program any closer to the end state than would incorrect lines of code in both blanks.

Additionally, participants' relatedness ratings were submitted to the Pathfinder scaling algorithm[2]. The resultant PFnets were then analyzed for the presence of specific links which were hypothesized to represent the structural knowledge necessary for solving each of the specific types of problems. Recall that in order to successfully solve Pointer-type problems, one must know how the concepts *Assign, Pointer, Position*, and *Increment* are interrelated. In a previous study using the same programming domain, Trumpower and Goldsmith (2004) determined the interrelationships among these concepts by asking a set of experts (the developers of the computer programming language) to complete the relatedness ratings task and then submitted the averaged experts' ratings to Pathfinder to derive a referent PFnet[3]. According to this referent PFnet, the concepts *Assign, Position,* and *Increment* are all linked to the concept *Pointer* (Figure 4, next page). Therefore, individuals whose PFnets contain these three links should be more likely to successfully solve Pointer-type problems than those whose PFnets do not contain these three critical links. From this point forward we will refer to these three critical links as constituting the "Pointer link subset."[4]

Similarly, recall that in order to successfully solve Go-To-type problems, one must know how the concepts *If-Then, Go-To,* and *Step* are interrelated. According to the referent PFnet from the Trumpower and Goldsmith (2004) study shown in Figure 4, the concepts *If-Then* and *Step* are both linked to the concept *Go-To*. Therefore, individuals whose PFnets contain these two links should be more likely to successfully solve Go-To-type problems than those whose PFnets do not contain these two critical links. From this point forward we will refer to these two critical links as constituting the "Go-To link subset."

Due to small sample sizes and the ordinal nature of our outcome variables, we used the non-parametric Mann-Whitney U test to evaluate our hypotheses (Hollander & Wolfe, 1999). Specifically, the sum of the total number of Pointer-type problems solved correctly by each participant was calculated and ranked. This was also done for the non-Pointer-type problems. Although individual problems may vary with respect to difficulty, each is considered an ordinal measure in which a score of one indicates greater knowledge than does a score of zero. Cliff and Keats (2003) demonstrate that for such dichotomously scored items, "there is a theoretical justification for simply adding item scores of zero and one" (p. 60) and then treating the resulting sum as an ordinal-level variable. Mann-Whitney U tests were then utilized to compare the distribution of ranks for those participants who did and did not possess the Pointer link subset, separately for the Pointer-type and non-Pointer-type problems. Likewise, Mann-Whitney U tests were conducted to compare the distribution of ranks for those participants who did and did not possess the Go-To link subset, separately for the Go-To-type and non-Go-To-type problems. It was hypothesized that the ranks of the participants who possessed the Pointer link subset would be higher than those who did not possess the Pointer link subset for Pointer-type problems, but not for non-Pointer-type problems, and that the ranks of the participants who possessed the Go-To link subset would be higher than those who did not possess the Go-To link subset for Go-To-type problems, but not for non-Go-To-type problems.

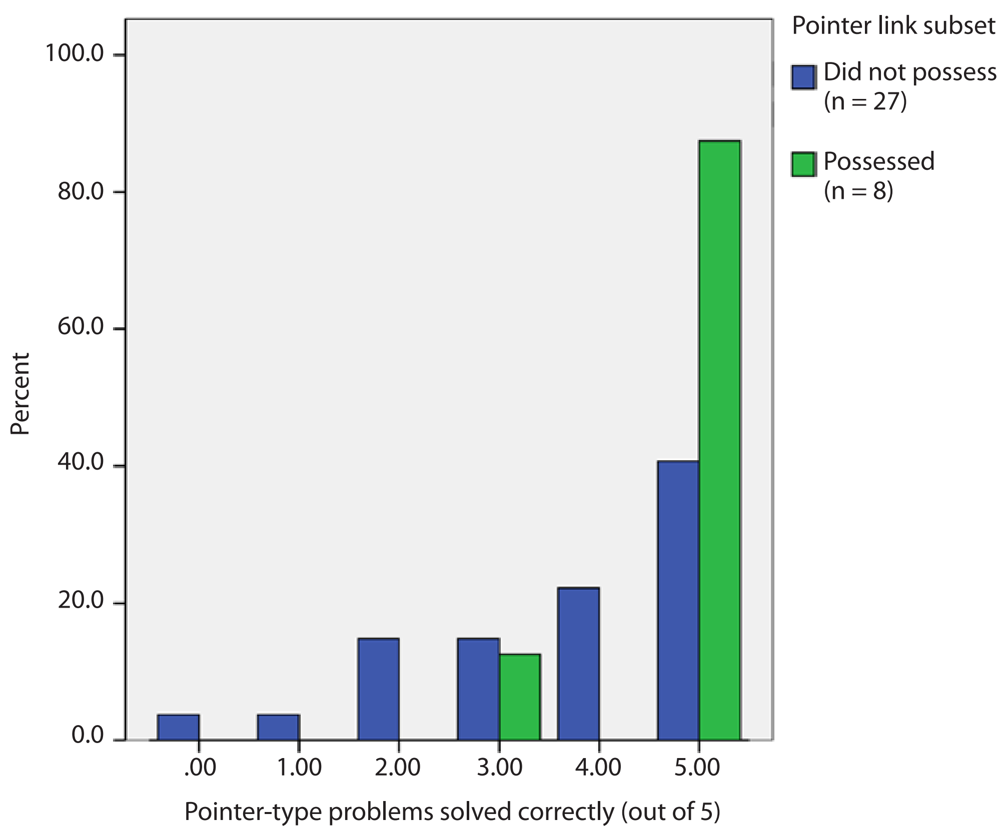
Figure 4: Referent PFnet of Computer Programming Concepts (*Pointer* Link Subset is Shown in italics and *Go-To* Link Subset is Shown in Bold)



Results

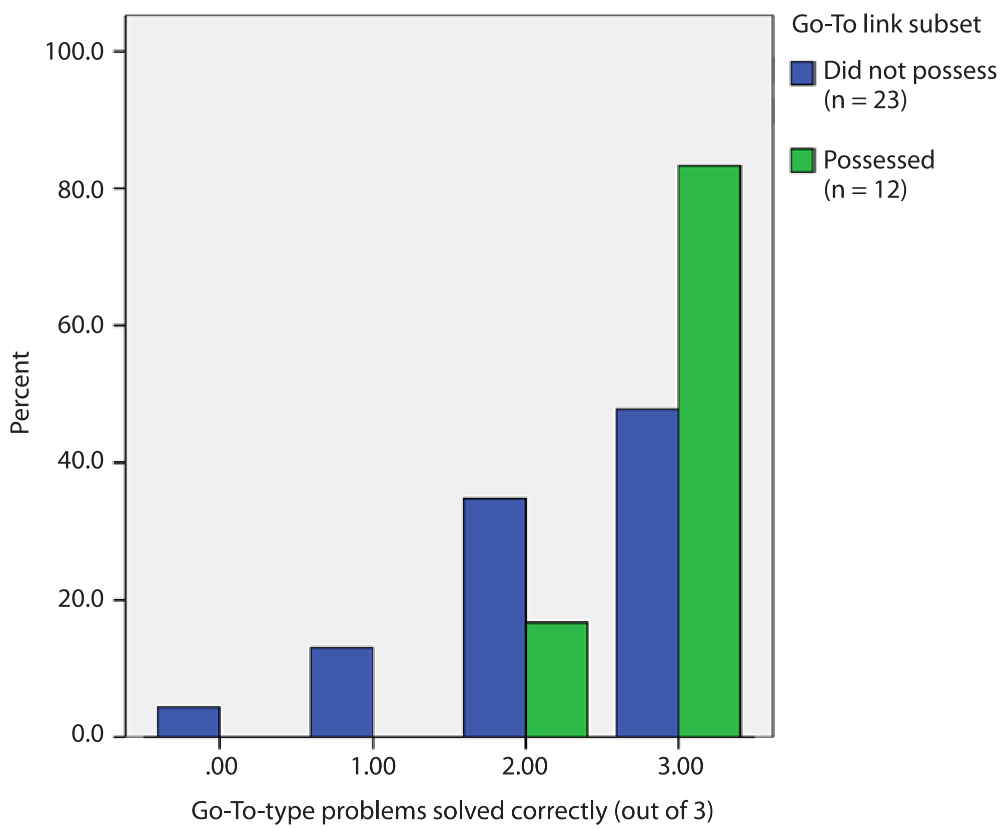
Eight of the 35 participants' PFnets possessed all of the links comprising the Pointer link subset. As predicted, a Mann-Whitney U test indicated that those who possessed the Pointer link subset performed statistically significantly better on the Pointer-type problems than those who did not possess the Pointer link subset ($U = 57.50$, $p = .032$). The average rank of participants who did and did not possess the Pointer link subset was 24.31 and 16.13, respectively (Figure 5 shows the distributions of number of Pointer-type problems solved correctly by those with and without the Pointer link subset). There was, however, no statistically significant difference in performance on non-Pointer-type problems between those who did and did not possess the Pointer link subset ($U = 84.00$, $p = .234$); the average ranks for the two groups were 21.00 and 17.11, respectively.

Figure 5: Distributions of Pointer-type Problems Solved Correctly by those With and Without the Pointer Link Subset



Twelve of the 35 participants' PFnets possessed all of the links comprising the Go-To link subset. Again as predicted, a Mann-Whitney U test confirmed that those who possessed the Go-To link subset performed statistically significantly better on the Go-To-type problems than those who did not possess the Go-To link subset (U = 85.00, p = .035). The average rank of participants who did and did not possess the Go-To link subset was 22.42 and 15.70, respectively (Figure 6 shows the distributions of number of Go-To-type problems solved correctly by those with and without the Go-To link subset). The difference in performance of those who did and did not possess the Go-To link subset on non-Go-To-type problems was not statistically significant (U = 134.50, p = .892); the average ranks for the two groups were 18.29 and 17.85, respectively.

Figure 6: Distributions of Go-To-type Problems Solved Correctly by those With and Without the Go-To Link Subset



Discussion

The present study extends our understanding of SAK – what it measures and how it can be applied to classroom assessment. Previously, inferences drawn from SAK for the purpose of indicating a learner's overall structural knowledge of a domain were shown to be valid. As such, its use in research and the classroom has been primarily summative in nature, or what Earl (2003) refers to as assessment *of* learning. Our findings, however, show that a more fine-grained evaluation of PFnets derived from SAK can be used to identify learners' specific strengths and weaknesses. The presence of particular links in students' PFnets was associated with their performance on related types of problems. Thus, evaluation of specific links in a student's PFnet may be used to locate areas in need of further instruction. As such, our findings suggest that SAK also has potential to be used as assessment for learning (Earl, 2003).

In general, students with poor structural knowledge of a domain as assessed by SAK perform poorly on tasks within that domain (Day, Arthur, & Gettman, 2001; Kraiger, 1993; Trumpower & Goldsmith, 2004), thereby indicating the predictive ability of SAK and the importance of structural knowledge. However, structural knowledge of a domain is comprised of many conceptual relations. Therefore, a student could have poor structural knowledge due to a failure to understand any of a number of important relations. In order to efficiently and effectively improve students' structural knowledge, instruction must be able to target specific missing or misunderstood relations. This, in turn, requires an assessment tool that allows identification of such missing and misunderstood relations. Our findings indicate that subsets of links in PFnets can, indeed, identify specific strengths/weaknesses. In particular, evidence for the convergent and divergent validity of two subsets of links in discerning performance on particular types of problems was obtained. Participants who possessed the Pointer link subset performed better than those who did not have these links on Pointer-type problems, but no differently on other types of problems. Conversely, participants who possessed the Go-To link subset performed better than those who did not possess these links on Go-To-type problems, but no differently on other types of problems. These findings indicate that links in PFnets represent specific bits of structural knowledge that have particular consequences when attempting to apply one's knowledge. Thus, it would appear that a fine-grained evaluation of links within students' PFnets can be used to identify specific areas of weakness to be targeted in further instruction, thereby providing the basis for applying SAK to formative assessment.

Although the findings in the present study are correlational, there is some experimental evidence to support the formative application of SAK.

In a recent study, Trumpower and Sarwar (in press) used SAK to provide individualized feedback to students in a high school physics class. Students were shown both their PFnet and a referent PFnet and were asked to reflect upon the differences. They were also given individual problems to solve and examples to study which were developed by a physics instructor to highlight the concept relationships indicated by links that were present in the referent PFnet, but that were missing from the student's own PFnet. Following this formative feedback and instruction, students' structural knowledge was re-assessed. Structural knowledge of the concept relations targeted by the formative instruction improved, whereas structural knowledge of a control set of concepts did not improve significantly.

This recent study illustrates the process that would be required for teachers to use SAK in a formative capacity. The five steps are summarized below:

Step 1: Identify the key concepts to be assessed. This can be accomplished through a task analysis, perusal of curriculum documents, and/or simple consideration of the core concepts of a domain. As with the construction of any classroom assessment, the set of concepts chosen should provide adequate coverage of the content to-be-assessed. However, due to time constraints within the classroom, the number of concepts should probably not exceed twenty.

Step 2: Obtain referent structure. The teacher (and/or other domain experts) must rate the relatedness of all pairwise combinations of the identified concepts for the purpose of deriving a referent PFnet. Using the averaged ratings of a group of experts to derive the referent structure has been shown to improve validity (Acton, et al., 1994) and is, therefore, recommended.

In the future, it is possible that repositories of referent PFnets for various domains could be created which would eliminate the need for teachers to perform the ratings task themselves. Similar repositories of knowledge structures in the form of concept maps have been created (see, e.g., Cañas, Hill, Carff, Suri, Lott, Eskridge, et al., 2004).

Step 3: Obtain student structures. The students must rate the relatedness of all pairwise combinations of the identified concepts. The KNOT software can be used to automatically collect the requisite relatedness ratings and convert them into PFnets. Alternatively, a paper and pencil version of the relatedness rating task may be created, in which case the teacher would need to enter the relatedness ratings into a text file and submit them to the KNOT software for conversion into PFnets.

Step 4: Evaluate student PFnets. The KNOT software can be used to display, print, and save the resulting PFnets. Evaluation involves comparing the student and referent PFnets to determine which referent links (or subset of links) are missing from the student's PFnet. Teachers may evaluate each individual link or they may choose to focus on certain subsets of links determined to represent an important principle. Although the KNOT software will compare the overall similarity of student and referent PFnets, it does not presently perform a comparison of subsets of links as required for the more fine-grained use of SAK described here. However, we are currently beginning to develop a computer application that will perform this type of analysis.

Step 5: Provide feedback and instruction to students. We suggest several ways that PFnets can be used for learning. Students may be shown both their PFnet and the referent PFnet and asked to reflect on the similarities and differences. In addition, they may be asked to solve problems or review examples intended to illustrate missing or misunderstood concept relations as indicated in their PFnets. Finally, they may be asked to find or create examples that illustrate missing or misunderstood relations. As previously mentioned, Trumpower and Sarwar (in press) have recently implemented such a SAK based formative assessment process in a high school physics classroom with positive results. Further investigations will attempt to determine how much and what type of remedial instruction is sufficient for improving weaknesses in student's structural knowledge as identified by SAK. We are also beginning to develop a computer application that will link problems, examples, and other instructional content with specific links in referent PFnets. Based on referent links that are absent in a student's PFnet, the application will present an individualized set of learning activities to the student.

Considerations: One considering the use of SAK might be concerned that the validity of inferences drawn from the technique may be affected by the appropriateness of the set of concepts chosen to assess and by the referent structure derived. This comes from a concern that teachers/experts may disagree about what are the most important concepts in a domain and about the relationships between those concepts. We believe that this concern is more justified in some domains than others. For example, Biglan (1973) defined the "hardness" of a domain as the extent to which its central body of theory is universally agreed upon. Therefore, teacher disagreement is more likely in "softer" domains than in "harder" ones. Consequently, Keppens and Hay (2008) have suggested that the use of a referent-based SAK is more suitable for hard domains, while referent-free assessment (e.g., SAK using coherence as a measure of the quality of student PFnets; see Acton, 1991) is more suitable for soft domains. Regardless, we believe that the best way to minimize this concern is to gather input from multiple teachers/experts when developing SAK.

In the initial stage of developing SAK for application in the classroom, we recommend that a team of teachers begin by individually generating a list of what they believe to be the most important concepts in the domain/unit of instruction to be assessed. As with any assessment, the concepts chosen to include must adequately cover the intended target. We have suggested careful task analysis or consideration of curriculum documents, textbook content, and other pedagogical material as a starting point. After generating their individual lists, we recommend that the team of teachers then meet as a group to discuss any discrepancies in the concepts that they chose. The objective of this discussion is to come to consensus on a final list of concepts to be assessed. If perfect agreement is not achieved, then concepts that are suggested by some, but not all, team members could be considered for inclusion in the final list as long as the concepts have been addressed during instruction and the total number of concepts does not exceed about twenty. Although larger sets of concepts allow for greater content coverage and have been shown to provide more valid inferences about students' level of understanding (Goldsmith & Johnson, 1990), any more than twenty concepts would require over 200 ratings to be made by each student. This number of ratings likely could not be completed by most students in a typical length class.

In the next phase of SAK, deriving a referent structure, we have recommended that the averaged ratings of a group of experts be used. Here, each member of the team of teachers would individually rate the relatedness of the concepts chosen for assessment. Correlations between the ratings of each team member can be calculated to determine level of agreement before averaging the ratings. In situations where a particular team member disagrees substantially with others about the concept relationships being assessed, that team member's ratings could be excluded from the averaged ratings. The rationale for such a decision is based on the assumption that if the other team members' ratings are relatively more strongly correlated with one another, then: (a) there does appear to be some general agreement about the conceptual relations within the domain, and (b) that particular team member may not be as knowledgeable as the others. In situations where many of the team members disagree substantially, the use of a referent-free SAK may be warranted.

However, it should be recognized that what constitutes substantial disagreement is somewhat subjective. Acton et al. (1994) showed that even when ratings varied considerably from one expert to another (with correlations as low as .31 between expert's ratings), the averaged ratings provided a referent PFnet that was used to validly predict students' performance in a university course. Furthermore, the referent structure based on the averaged ratings generated better predictions than referent structures based on any single expert's ratings. Nonetheless, more research investigating the acceptable level of variability among experts and the optimal number of experts to be included in the SAK process may help to further address such concerns.

One considering the use of SAK might also wonder if the method used to assess the relatively simple, proscribed computer programming language in the present study can be applied more generally to assess more sophisticated knowledge. We believe that it can. Our conclusion is based on the fact that much of the previous research on SAK has been conducted in larger, more sophisticated knowledge domains, including complete university courses (e.g., a human resources management course, Acton, 1991; a teacher education course in mathematics, Gomez et al., 1996; a research techniques course, Goldsmith et al., 1991).

Although the above mentioned issues deserve consideration, our present findings, as well as those of Trumpower and Sarwar (in press), indicate that SAK holds the potential for filling the identified needs for new formative (Earl, 2003) and structural (National Research Council, 2000, 2001) assessment tools.

Endnotes

1. There is often much confusion when using terms like "formative" and "summative" assessment. For the purpose of clarification, we adopt the Council of Chief State School Officers' definition of formative assessment as "…a process used by teachers and students during instruction that provides feedback to adjust ongoing teaching and learning to improve students' achievement of intended instructional outcomes" (as cited in McManus, 2006). Further, it should be noted that a given assessment tool cannot be said to be "formative" in and of itself. A tool can only be said to be formative when it is being used in the process of formative assessment. And, it can only be used in the process of formative assessment if it can (a) identify students' specific strengths and weaknesses and (b) provide feedback that helps remediate the weaknesses. Therefore, a given assessment tool could be both a formative assessment tool and a summative assessment tool at different times. SAK has traditionally been used for summative purposes, but we begin to investigate its appropriateness for formative purposes. This study addresses the first criteria for a formative application—the ability to identify students' specific strengths and weaknesses. Determining whether or not it meets the second criteria—the ability to provide feedback that can help remediate identified weaknesses—is left for future study (but see Trumpower & Sarwar, 2009 for preliminary results of such an investigation).
2. Parameter values of $r = \infty$ and $q = n$-1 (where n = the number of concept nodes) were used to generate the PFnets. Schvaneveldt, et al., (1989) recommend using the parameter value of $r = \infty$ for ordinal data. The parameter value q determines the number of indirect proximities that the KNOT software evaluates when generating the PFnets. The maximum value for the q parameter is n-1, which results in PFnets with the fewest number of (but, relatively most related) links.
3. As mentioned earlier, Acton, et al. (1994) have shown that the averaged ratings of multiple experts provide a better referent than any individual expert. This procedure for determining a referent from averaged ratings is further justified by the relatively high inter-rater reliability (r = .83) of the pair of experts used by Trumpower and Goldsmith (2004) to derive the referent network. Further, both of the experts verified that the referent network was an accurate representation of their knowledge of the relationships among the concepts, with a clear delineation between data structures and control structures.
4. For this analysis, we have decided to use an all-or-none approach to identify those who possess the Go-to and Pointer link subsets. The links within the analyzed subsets were chosen because they were all deemed critical for successfully solving the associated problems. For example, to successfully solve Pointer-type problems, it was believed that one must know the relationships between *Assign* and *Pointer*, *Position* and *Pointer*, and *Increment* and *Pointer*; failure to understand any of these relations would likely lead to solution failure. It is possible, however, that structural knowledge develops gradually such that one may have partial knowledge concerning the relationships among concepts within the Pointer link subset without possessing all of the critical links. It is also possible for one to possess all of the critical links in addition to some other extraneous links. These extraneous links may represent misconceptions that can get in the way of successful problem solving, too. Including an even more detailed evaluation of the number of critical and extraneous links within each subset may provide a more powerful diagnostic tool. See Trumpower and Sarwar (in press) for an example of formative structural assessment using this type of evaluation.

References

Acton, W.H. (1991). Comparison of criterion referenced and criterion free measures of cognitive structure. Unpublished doctoral dissertation, University of New Mexico, Albuquerque.

Acton, W.H., Johnson, P.J., & Goldsmith, T.E. (1994). Structural knowledge assessment: Comparison of referent structures. *Journal of Educational Psychology, 86*(2), 303–311.

Anderson, J.R. (1995). ACT-R: A simple theory of complex cognition. *American Psychologist, 51*(4), 355–365.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036–1060.

Baxter, G.P., Elder, A.D., & Glaser, R. (1996). Knowledge-based cognition and performance assessment in the science classroom. *Educational Psychologist, 31*, 133–140.

Biglan, A. (1973). The characteristics of subject matter in different academic areas. *Journal of Applied Psychology, 57*, 204–213.

Brown, M.B. & Forsythe, A.B. (1974). The ANOVA and multiple comparisons for data with heterogeneous variances. *Biometrics, 30*, 719–724.

Campbell, D.T., & FiskeD.W. (1959) Convergent and discriminant validation by the multitrait-multimethod matrix. *Psychological Bulletin, 56*, 81–105.

Cañas, A.J., Hill, G., Carff, R., Suri, N., Lott, J., Eskridge, T., et al. (2004). CmapTools: A knowledge modeling and sharing environment. In A. J. Cañas, J. D. Novak & F. M. González (Eds.), *Concept maps: Theory, methodology, technology. Proceedings of the first international conference on concept mapping* (Vol. I, pp. 125–133). Pamplona, Spain: Universidad Pública de Navarra.

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.

Clariana, R.B., & Wallace, P. E. (2007). A computer-based approach for deriving and measuring individual and team knowledge structure from essay questions. *Journal of Educational Computing Research, 37*(3), 209–225.

Cliff, N. & Keats, J.A. (2003). *Ordinal Measurement in the Behavioral Sciences*. Mahwah, NJ: Erlbaum.

Cooke, N.M. (1987). *The elicitation of units of knowledge and relations: Enhancing empirically-derived semantic networks*. Unpublished doctoral dissertation, New Mexico State University.

d'Appolonia, S.T., Charles, E.S., & Boyd, G.M. (2004). Acquisition of complex systemic thinking: Mental models of evolution. *Educational Research and Evaluation, 10*, 499–521.

Davis, M.A. & Curtis, M.B. (1996). Assessing structural knowledge in management education. Paper presented at the meetings of the Academy of Management, Cincinnati, Ohio.

Day, E.A., Arthur, W. Jr., & Gettman, D. (2001). Knowledge structures and the acquisition of a complex skill. *Journal of Applied Psychology, 86*, 1022–1033.

Dayton, T., Durso, F.T., & Shepard, J.D. (1990). A measure of the knowledge reorganization underlying insight. In R.W. Schvaneveldt (Ed.), *Pathfinder associative networks: Studies in knowledge organization* (pp. 241–254). Norwood, NJ: Ablex.

Earl, L. (2003). *Assessment as Learning: Using Classroom Assessment to Maximise Student Learning*. Thousand Oaks: Corwin Press.

Goldsmith, T.E. & Johnson, P.J. (1990). A structural assessment of classroom learning. In R.W. Schvaneveldt (Ed.), *Pathfinder associative networks: Studies in knowledge organization* (pp. 241–254). Norwood, NJ: Ablex.

Goldsmith, T.E., Johnson, P.J., & Acton, W.H. (1991). Assessing structural knowledge. *Journal of Educational Psychology, 83*, 88–96.

Gomez, R.L., Hadfield, O.D., & Housner, L.D. (1996). Conceptual maps and simulated teaching episodes as indicators of competence in teaching elementary mathematics. *Journal of Educational Psychology, 88*, 572–585.

Hollander, M. & Wolfe, D.A. (1999). *Nonparametric Statistical Methods* (2nd Ed.). New York: Wiley.

Keppens, J. & Hay, D. (2008). Concept map assessment for teaching computer programming. *Computer Science Education, 18*(1), 31–42.

Kraiger, K. (1993). Further support for structural assessment as a method of training evaluation. Paper presented at the annual meeting of the American Psychological Association, Toronto, ON.

Kraiger, K., Salas, E., & Cannon-Bowers, J.A. (1995). Measuring knowledge organization as a method for assessing learning during training. *Human Factors, 37*, 804–816.

Larkin, J.H., McDermott, J., Simon, D.P., & Simon, H.A. (1980). Expert and novice performance in solving physics problems. *Science, 208*, 1335–1342.

Marshall, S.P. (1995). *Schemas in problem solving*. New York: Cambridge University Press.

McClure, J.R., Sonak, B., & Suen, H.K. (1999). Concept map assessment of classroom learning: Reliability, validity, and logistical practicality. *Journal of Research in Science Teaching, 36*, 475–492.

McManus, S. (2006). Attributes of Effective Formative Assessment. Retrieved July 31, 2009 from The Council of Chief State School Officers website: http://www.ncpublicschools.org/docs/accountability/educators/fastattributes04081.pdf.

National Research Council (2000). *How people learn: Brain, mind, experience, and school*. Washington, D.C.: National Academy Press.

National Research Council (2001). *Knowing what students know: The science and design of educational assessment*. Washington, D.C.: National Academy Press.

Novak, J.D., & Gowin, D.B. (1984). *Learning how to learn*. New York: Cambridge University Press.

Schau, C., Mattern, N.,Weber, R.W., Minnick, K., & Witt, C. (1997, April). *Use of fill-in concept maps to assess middle school students' connected understanding of science*. Paper presented at the annual meeting of the American Educational Research Association, Chicago.

Schoenfeld, A.H. & Herrmann, D.J. (1982). Problem perception and knowledge structure in expert and novice mathematical problem solvers. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 8*, 484–494.

Schvaneveldt, R.W. (1990). *Pathfinder associative networks: Studies in knowledge organization*. Norwood, NJ: Ablex.

Schvaneveldt, R.W., Sitze, K., & McDonald, J. (1989). Knowledge Network Organizing Tools (KNOT 4.3) [Computer software for Pathfinder Network analysis]. Gilbert, AZ: Interlink. Retrieved October 31, 2007. Available from http://interlinkinc.net/.

Trumpower, D.L. & Goldsmith, T.E. (2004). Structural enhancement of learning. *Contemporary Educational Psychology, 29*, 426–446.

Trumpower, D.L., Guynn, M.J., & Goldsmith, T.E. (2004). Goal specificity and knowledge acquisition in statistics problem solving: Evidence for attentional focus. *Memory & Cognition, 32*(8), 1379–1388.

Trumpower, D.L., & Sarwar, G.S. (in press). Effectiveness of structural feedback provided by Pathfinder networks. *Journal of Educational Computing Research*.

Weiser, M. & Shertz, J. (1983). Programming problem representation in novice and expert programmers. *International Journal of Man-Machine Studies*, *19*, 391–201.

Author Note

A preliminary analysis of this study was presented at the 29th Annual Conference of the Cognitive Science Society (Nashville, TN).

Author Biographies

David L. Trumpower is an Assistant Professor of Teaching, Learning, and Evaluation at the University of Ottawa, Faculty of Education. His current research interests include: assessment of structural knowledge for the purposes of formative feedback and instructional design; conceptual understanding of statistics in naïve and experienced students; teachers' perceptions and practices regarding assessment and evaluation. He can be contacted at david.trumpower@uottawa.ca.

Harold Sharara is an MA student in the Teaching, Learning, and Evaluation concentration at the University of Ottawa, Faculty of Education. His research involves exploring the diagnostic capability of the structural assessment of knowledge technique.

Timothy E. Goldsmith is Associate Professor of Psychology at the University of New Mexico. His current research efforts are aimed at deriving and validating methods of eliciting, representing, and evaluating human knowledge and skill. This work is being performed in both academic and applied settings. He can be contacted at: gold@unm.edu.

J·T·L·A

The Journal of Technology, Learning, and Assessment

www.jtla.org

Technology and Assessment Study Collaborative
Caroline A. & Peter S. Lynch School of Education, Boston College