SOFTWARE MAPPING ASSESSMENT TOOL DOCUMENTING BEHAVIORAL CONTENT IN COMPUTER INTERACTION: EXAMPLES OF MAPPED PROBLEMS WITH *KID PIX* PROGRAM

Associate Prof. Dr. Servet BAYRAM
Marmara University

THE NEED

There is a growing consensus that traditional methods, such as standardized testing, criterion-referenced tests, and teacher-constructed tests fail to measure important learning outcomes (Shepard, 1989; Anderson & Bachor, 1998; Shavelson & Ruiz-Primo, 2000; Koker, 2001; Davies, 2001). Such tests provide little to indicate either the level at which a student understands or the quality of individual thinking (Nickerson, 1989; Slack, 1993; Raychaudhuri, 1998; Lee, 2002). They emphasize homogenized recall of memorized factual knowledge and procedures rather than unique, and highly differentiated reflection. Because external criteria, they typically emphasize standards which can be applied to typical students. Changing the way we assess will inevitable change how teachers teach and how students learn. Today new ways of thinking about learning call for new ways for monitoring learning.

Reform in school assessment builds from the vision that assessment can become the bridge for instructional activity, accountability, and teacher development. Romberg (1995) stated that if assessment results are used by the learners or teachers, then the assessment tools must be available in the classroom on a regular basis, weaving together instruction and assessment (p. 29). The content of tests influences teaching and learning processes. Teachers often "teach to the test" rather than emphasizing underlying concepts.

Skills are thought in the manner measured on tests rather than how they are used in everyday contexts. When tests require the recall of memorized information, students develop memorization strategies that tend to de-contextualize their knowledge, promoting compliant cognition (McCaslin & Good, 1992; Pettig, 2000; Dolan & Hall, 2001). In order to become capable, learners need experience in solving real problems and understanding complex tasks (Duffy, 1997; Linn et al., 2000). Shepard (1989) stated that assessments need to approximate real-life tasks and to reflect multiple perspectives and diversity-versus-singularity of problem solutions.

Another problem with traditional testing is that it tends to emphasize evaluation, or classification, as a primary goal (Hart, 1994; Wilson, 1995; Ayala et al., 2002; Yin et al., 2004). Since a primary goal of education is to promote students' thoughtfulness, the basic of concept of testing needs to change, not just the structure of the tests (Brown, 1989; Koretz, 1998; Sizer, 2001). Also, McLellan (1993) pointed out that assessment needs to be dynamic, and reflect every-emerging samples of the learner's progress. As a consequently, traditional testing strategies are often counter-productive for the solving of real-world problems (Collins, 1990; Yin & Shavelson, 2004).

Choi & Hannafin (1995) and Reese (2003) stressed that in order to be useful in promoting higher thinking skills, testing needs to shift from domain-referenced evaluations to student-center assessments. Student-centered assessment emphasizes the ability to diagnose and manage cognitive growth rather than to evaluate student achievement. They said that since assessment in situated learning environments emphasizes cognitive and learning processes, improvement of learning strategies, and higher-order thinking skills, assessment alternatives typically require varied evidence (Pettig, 2000). As a programmatic change is occurring, there is a need to align student assessment practices with curricular aims, instructional practice and performance standards (Black &Wiliam, 1998; Their & Daviss, 2001).

The development of problem solving involves students' efforts to overcome obstacles and attain goals (Stecher, 1998). It involves the orchestration of a large number of other processes toward this end (Siegel & Thier, 2002). How well students encode to form mental modes are among the key determinants of their success on many problems. As Siegler (1991) stated that their success also depends critically on the ability to integrate general and specific knowledge, and on their selection of the right process in the right situation. Choi and Hannafin (1995) pointed out that constructivists will have to develop ways of expressing what is to be accomplished that do not constrain learning outcomes as they feel specific objectives would. Without some idea regarding student outcomes, evaluation would be an empty exercise (Pisha & Coyne, 2001; Sizer, 2001; Reese, 2003).

LITERATURE

Traditional assessment is the process of gathering information about students- what they know and can do. In fact, assessment data simply mirrors what is going on in the classroom. This information become meaningful only when we decide that it reflect something that we value, such as how well a student has mastered long division. Authentic assessment emphasizes the development of assessment tools that more accurately mirror and measure what we value in education (Hart, 1994). An assessment is authentic when it involves students in tasks that are worthwhile, significant, and meaningful (Siegel et al., 2002). The computer can provide a further perspective on the learner. It can effectively track the process of learning as well as a learner's response to feedback. It can also "simulate realistic situations in the classroom" (Lajoie, 1995, p.28).

The computer provides opportunities for assessing the dynamic nature of problem solving and opportunities to systematically vary the instructional environment on feedback dimensions and observe the effects on learning outcomes (Rose & Meyer, 2002). Computers make possible the dynamic assessment of relevant criteria (Newby et al., 1996). Most of the evaluation procedures involve having evaluators use a rating form (e.g., Litchfield, 1992; Tolhurst, 1992; Voogt, 1990; Robert & Wilson, 1998; Thier et al., 1999; Wilson & Sloane, 2000) to evaluate each of a variety of features of a piece of educational software for classrooms. An overview of the components, functions and limitations of the human cognitive system provides a framework for understanding why some educational software that "looks good" fail to produce positive student's outcomes (Siegel et al., 2002). Unfortunately, nearly all software evaluation systems are heavily weighted on computer-related dimensions of error-handling and aesthetic considerations, such as the quality of screen displays, sound, touch, and content related issues of scope, sequence, and accuracy. Although important, these characteristics do not address the consistency of our knowledge with how students learn (Charleston, Villagomez, & Shaffer, 1989; Lounge et al, 1986; Wilson et al., 1996; Thier et al., 1999; Thier & Daviss, 2001).

Reiser and Kegelmann (1994) stated that there is no evaluation methodology which is equally applicable to the service of the strategic intentions of administrators, the empirical requirements of academics and the needs of classroom practitioners for information in support of tactical decision-making. Then they pointed out that it is important that software evaluation organizations incorporate examinations into their software evaluation processes. By doing so, these organizations will take a significant step forward toward accomplishing their primary mission—assisting educators in identifying software that will truly enhance student learning. In light of these facts, what can organizations responsible for the evaluation of software do to improve their evaluation methods? They also stated that in order to overcome the problems associated with subjective evaluations, those who have critiqued software evaluations techniques often suggest that an examination of student use should be an important part of the process. Most researchers suggest that evaluators collect attitude data from the students who have worked through the program (Schuecker & Shuell, 1989; Jolicoeur & Berget, 1989, Barab et al., 1996; Davies, 2001; Yin et al., 2004). Scholars point out that portfolio; protocol analysis (think a loud) performance assessment and concept maps are the most popular alternative, authentic and meaningful methods of the assessment of situated learning (Collins et al. 1993; Wolf, 1989, 1998; Wolf et al. 1991; Lajoie, 1995; Ayala et al., 2002; Yin et al., 2004). Wilson & Sloana (2000) said that a clear vision of the overall framework can be constructed with a coherent authentic assessment system.

FROM SOFTWARE ASSESSMENT TO SOFTWARE MAPPING

Science and mathematics educators are enjoined to assess what students know and come to understand through knowledge representation (Harlen, 1985; Gentner & Markman, 1997). Representing such knowledge can be accomplished by the use of technology that provides, not only a vehicle for their storage, display, and active presentation but also a "malleable and interchangeable" electronic forum for products of the mind. Over the past decade, research on human cognition has revealed insights into the mental processes involved in learning, remembering, reasoning, and problem solving. These findings also have implications for the *design* and *evaluation* of instructional software (Rose & Meyer, 2000a, 2000b and 2002).

The process of learning with computers is influenced by the ability of the medium to dynamically represent formal constructs and instantiate procedural relationships under the learner's control (Kozma, 1991). The ability to control the flow of action and events in the software allows the instructor to tailor a lesson to the specific needs of the learner. Menu systems, design components (such as diagnostic routines, unit planning assistance, flexible lesson sequencing, multi-level lesson sequences, flexibility to easily branch, random use and so on), interaction with user, and management services are the components of functional design. Thus, software design must allow for smooth and rapid exit from one menu to another with minimum effort. Menus should provide a non-destructive means to cancel or abort the action selected to allow the learner to escape from an unintended choice. It is only in this concern over components of functional design that menu, sub-menu, and tool variety become explicit.

One study that prompted further thinking of *software mapping* was that of Schuerman and Peck (1991) who studied the effects of pull-down menu (PDM), returning to pull-down menus or the sub-menu (RSM) options, and return to main menu (RMM) options as exercised by users of the graphic-user interface. PDMs did not necessarily encourage random, as opposed to sequential, access of lesson activities. The RSM condition promoted significant grouping of options. The PDM condition did not produce weaker groupings than RMM condition. PDMs yielded significantly higher menu inspection; i.e. looked but did not "leap." Ergonomics of the PDM does not diminish deliberation on the part of the user. The RSM option facilitates return to sub-menu item 80% of the time whereas RMM does not. And the "compact menu system provided by pull down sub-menus and a menu bar offered full functionality of a two-level, tree-structured menu system without the formality of full-screen static menus (pp. 97- 98).

It has been stated that "courseware is often instructionally sound but fails because it lacks the touches of the creative mind-spontaneity, humor, variety, and pizzazz" (Kearsley, 1985, p. 217). Effective software design permits curriculum content to be delivered in a manner most appropriate for the target group of learners (Hernandez & Reese, 2004). Appropriate software design allows the learner to obtain the maximum benefits from the material being presented (Fernandez & Body, 1997; Fisher, 2000). The amount of effort required of the learner to interact with the software will be rewarded by a sense of satisfaction. Effective learner's interaction with the software will result in immediate delivery of desired, expected, and relevant information from the software (Barab et al., 1996; Draney & Wilson, 1997, Kurtz et al., 2001). The concentration of the learner should be focused on the curriculum material and content and not on the process required to interact with the software. Interaction with the software needs to be as natural and intuitive as possible (Charleston, Villagomez, and Shaffer, 1989; Kumar, 1994). Interaction between teacher, student and computer may lead to significant changes in the processes rather than the measurable products of education (Barab et al., 1998; Shavelson & Ruiz-Primo, 2000; Reese & Hergert, 2004). Evaluation must therefore be sensitive to classroom process, and must be capable of providing information about these processes.

Without a doubt the graphic-user interface or usability engineering has achieved a favorable status in educational computing. We now thought of having a vehicle, with such an interface, to explore the behavioral content (actions) of students as they "navigate" on screen and through the software. Before developing software-teaching training programs, we need to have a more through understanding of what particular aspects of 'good practice' with computers lead to what specific gains in the quality of pupil learning. It is at this point that *software mapping* is proposed to match the curriculum with student behavior.

OVERVIEW SOFTWARE MAPPING

Software mapping is simple but proposed to be a dynamic method that facilitates access to general software program parameters and engenders a knowledge of powerful, but embedded and "deep-rooted" options allowing for increased inter-rater reliability in documenting or replicating richly constructed student responses. Rich and diverse student constructions can only be mediated by rich and diverse curriculum activities. Designing a sequenced set of student activities which use program menus, submenus, and palettes and constructing probable maps associated with each activity reveals divergence in thinking on the part of the student.

The purpose of the software mapping is to delineate a method for software menu, tool, and palette use in the construction of elementary school science and mathematics curriculum activities. With this method, software 'maps' were created for traversing science and math curriculum problems and activities using software. The other purpose of the software mapping efforts is to delineate a method for student assessment in the classroom. Software mapping is an approach to assessment documenting behavioral content in computer interaction. Examples of mapped problems are presented, analyzed, and discussed in this paper. These examples are derived from curriculum handbooks received from Apple Corporation through efforts of the National Council of Teachers of Mathematics (NCTM) and the Association for the Education of Teachers of Science (AETS) (Apple Computer, Inc., 1992; 1993).

THE SOFTWARE MAPPING METHOD WITH *KID PIX*

A unit of study focused on investigation entitled "Patterns, Patterns Everywhere, and Not a Drop of Ink" is to be implemented using documentation provided for the program *Kid Pix* from Brøderbund Software (Apple, 1993, pp. *119-50). Kid Pix* and *Kid Pix* Companion are authored by Craig Hickman (1991, 1992). The mathematical concepts and methods used are constructed for students in grades 2-4. NCTM Standards addressed are Mathematics as communication, Mathematics as reasoning Geometry and spatial sense, Measurement, and Patterns and relationships. Specifically "the explorations will enable students to:

- Describe, model, draw, and classify shapes
- Investigate and predict the results of combining subdividing and changing shapes.
- Relate geometric ideas to the concepts of numbers and measurement.
- Recognize and appreciate geometry in their world.
- Communicate their understanding of geometry both verbally and pictorially"

(Apple,1993, p.120).

Classroom management is highlighted early in the unit. Equipment requirements, software preparation, helpful hints, and an overview are provided for the teacher. Student prerequisites are: working comfortably with manipulativeness, distinguishing basic two-dimensional geometric shapes, making simple patterns, and the ability to count past twenty. They must also know how to point and click as well as how to drag lines. Sufficient exploration, of an open-ended nature, is recommended to acquaint students with *Kid Pix.* Another interesting recommendation is made to "try training a few *Kid* Pix 'experts' *so* they can assist their peers. This will free you up to work with other students and will go a long way toward improving students' self esteem" (Apple, 1993, p. 123).

Assuming students, aged 8 to 10, have sufficient access to computers and the *Kid Pix* program, exploration would entail use of 5 menu items, 33 submenu items, 13 tools, and 13 palette items, not to mention the 39 colors available with the color tool and 600 palette elements. The developmental level of the student, alone, would limit so-called "sufficient exploration." Figure 1 itemizes these elements, along with unique number designations for each (i.e. M2.1, or M5.3, T1, or P4), and constitutes the first step in what it is called "software mapping". Text has been used to represent graphics that normally appear on the screen. Numbers in parentheses represent the elements available when the tool or palette item is chosen.

M1 Apple	M2 File	M3 Edit	M4 Goodies	M5 Companion
M1.1 **About KidPix**	M2.1 **New**	M3.1 **Undo**	M4.1 **Small KidsMode**	M5.1 **Swap stamps**
M1.2 **Help**	M2.2 **Open**	M3.2 **Cut**	M4.2 **Edit Stamp**	M5.2 **Swap Hidden Pictures**
M1.3 ... M1.n are machine dependent	M2.3 **Close**	M3.3 **Copy**	M4.3 **Alphabet Text**	M5.3 **DrawMe**
	M2.4 **Save**	M3.4 **Paste**	M4.4 **Tool Sounds**	M5.4 **ColorMe...**
	M2.5 **Save as**	M3.5 **Clear**	M4.5 **Record Sound**	
	M2.6 **Page Setup**		M4.6 **Play Sound**	
	M2.7 **Print**		M4.7 **Switch to Spanish**	
	M2.8 **Quit**			

Tools	allow access to a ⟷	Palette
T1 **Pencil**		P1 **Sharpness-Dots** (1... 14)
T2 **Line**		P2 **Line** (1...6) , **Patterns** (7...14)
T3 **Rectangle**		P3 **Rectangle** (1...14)
T4 **Circle**		P4 **Eliptics** (1...14)
T5 **Brush**		P5 **Brush** (1...28)
T6 **Mixer**		P6 **Mixer** (1...14)
T7 **Fill Bucket**		P7 **Fill Bucket** (1...14)
T8 **Eraser**		P8 **Eraser** (1...14)
T9 **Alphabet**		P9 **Character** (1...42)
T10 **Stamp**		P10 **Stamp** (1...112)
T11 **Truck**		P11 **Truck** (1...14)
T12 **Face**		P12 **Correction** (1, 2; tool dependent)
T13 **Colors** (39)		P13 **Colors** (1...300)

Figure1. Generic elements of the desktop presented in Kid Pix

Detailed explanations about, and illustration of, Kid Pix Tools, Menus, icons, the Gallery, sound recording, and the Small Kids Mode are provided in a user's guide and not recreated here (Brøderbund, 1991). The generic elements identified in Figure 1 add only letter and number designations to better catalogue student actions, or behavioral content, that accumulate on the screen (desktop).

It has been known now to begin the excursion into the curriculum activity as it is "played out" on the screen. The following worksheets items are listed in the curriculum text (Apple, 1993, pp. 129-50) with spaces eliminated for brevity and are presented here to illustrate the mapping scheme. A worksheet direction requiring the use of the *Kid Pix* program is presented. Other directions not requiring use of the program are not listed. Page numbers are provided for referencing (Apple, 1993). Objects are drawn while menu bar, tool and palette selections are mapped and written in parentheses. Use of the asterisk (*) with a number indicates n-repetitions of the tool, palette, or menu bar item. The key to letters and numbers used in the following software maps can be found in Figure1.

In presenting the worksheet (pp. 129-130) for activity one, the students "exploring" Kid Pix displayed three different pictorial responses or approaches to the direction, "Build a triangle tower that is five rows high." The Figure 2 shows different student's responses.

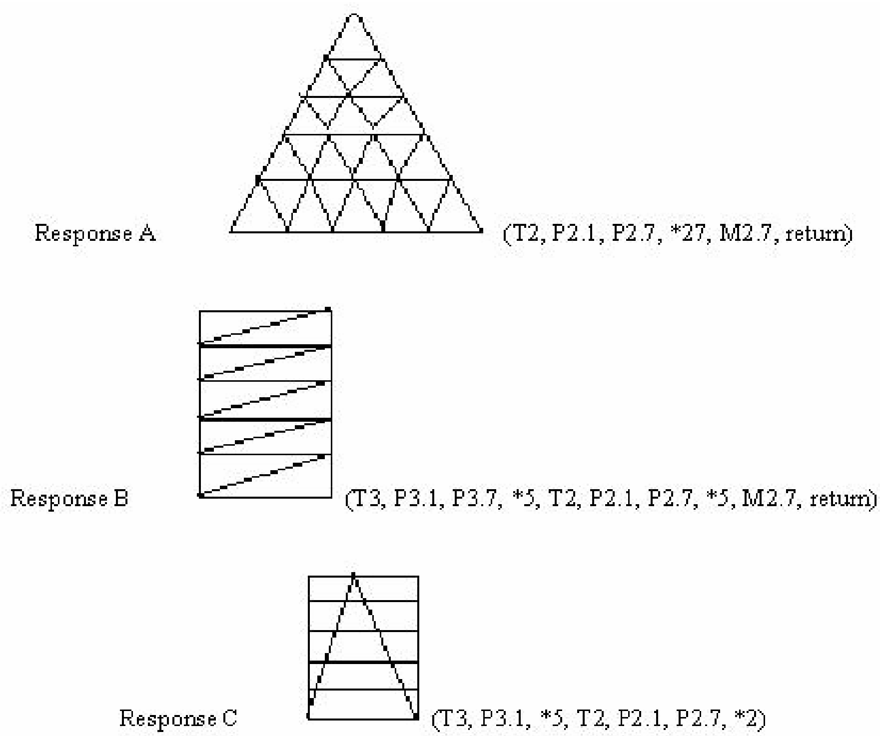
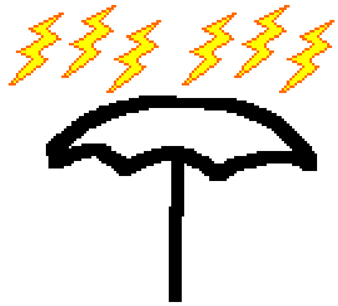


Figure 2. I Spy a Triangle elements of the desktop presented in Kid Pix.

The software maps of the responses are then juxtaposed to identify the similarity of selections and differences in order. All responses are similar in that they use the line tool (T2) and 1st and 7th line and pattern palettes, respectively. Response A repeatedly uses the tool-palette combination 27 times; response B, 5 times; and response C, 2 times. Clearly repetitive tool use is minimized in responses B and C, but with the rectangle tool (T3) and rectangle palettes (P3.1 and P3.7) used to set the unique outer border of their representations of triangle. Responses A and B were printed (M2.7) as hard copy.

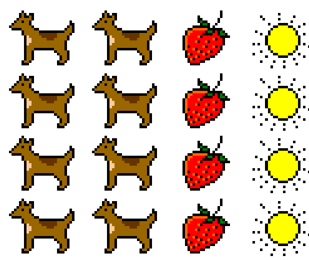
In presenting the worksheet for activity two, (p. 135): What Am I ?, with the direction, "Draw a shape using *Kid Pix* Paste your shape in the space below," a student constructed Figure 3 using the pencil (T1) tool with varying sharpness of dots (P1.3, P1.7), then using the stamp (T10) tool to access a lightning bolt (P10.110) and stamped three "bolts" (*3) above the umbrella. The truck (T11) tool and palette (P11.14) was used to copy (M3.3) and paste (M3.4) the next set of three bolts above the umbrella. After printing (M2.7) hard copy the student erased the drawing with the eraser (T8) tool and palette (P8.5). Both activities above ask for a drawing but the response to activity two illustrates a lengthier response map using additional activity.



(T1, P1.3, P1.7, draw, T10, P10.110, *3, T11, P11.14, M3.3, M3.4, M2.7, T8, P8.

Figure 3. What Am I?

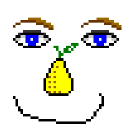
Worksheets A (p.141), B (p.143), and C (p.145) in activity three direct students to use the stamp tool to create and manipulate patterns. When directed to "Use the stamp tool in Kid Pix to make these patterns. Print the patterns. Dog, dog, strawberry, sun (Repeat this pattern four times.)," Figure 4 illustrates stamping two dogs and a strawberry (T10, P10.3, *2, P10.4, *l) then using the truck tool (T11) and "grab" palette (P11.14) to select the drawing, copies and pastes three more sets (M3.3, M3.4, *3). Recognizing that the sun is missing from the drawing the student returns to the truck tool, with "grab" palette already highlighted, selects the 3 element sets, cuts three sets from the desktop and adds the sun element (error recognition, T11, select, M3.2, T10, P10.13, *11). The truck tool copying and pasting sequence is repeated (T11, P11.14, select, M3.3, M3.4, *3).



(T10, P10.3, *2, P10.4, *1, T11, P11.14, select, M3.3, M3.4, *3, [error recognition, T11, select, M3.2, T10, P10.13, *1] T11, P11.14, select, M3.3, M3.4, *3)

Figure 4. Worksheet A

Worksheet B directs students to create a pattern (T10, P10.21, *2, P10.24, T1, P1.1, P1.7, draw), print it out (M2.7, return), add sound to the pattern-using "Record" under the Goodies menu-, to speak into the microphone, and describe the pattern (M4.5, record, return, M4.6). Figure 5 illustrates the output and map. Our fourth activity direction is "Use the stamps in *Kid Pix* to create a pattern of your own. Print it out. Paste it in the space below. Add sound to your pattern. Use 'Record' under the Goodies menu. Speak into the microphone and describe your pattern." (p. 143).



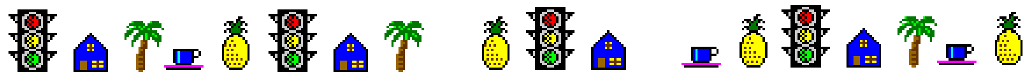
(T10, P10.21, *2, P10.24, T1, P1.1, P1.7, draw, M2.7, return, M4.5, record, return, M4.6)

Figure 5. Worksheet B

Worksheet C is similar to A in creating the elements (T10, P10.1, P10.14, P10.15, P10.28, P10.29, P10.42), copying and pasting the patterns (T11, P11.14, select, M3.3, M3.4, *3), then uses the eraser tool to remove elements from the pattern (T8, P8.1, select). Finally the pattern is printed out (M2.7) as hard copy and given to a partner to fill in the missing parts. The Figure 6 shows the implication of the following activity directions (p.145):

1. Use the stamps in *Kid Pix* to make a pattern. Be sure to repeat the pattern at least two times.

2. Leave one or two blank spots in your pattern, or use the eraser tool to remove one or two stamps from your pattern.
3. Print out your incomplete pattern and give it to your partner. Ask your partner to fill in the missing parts.



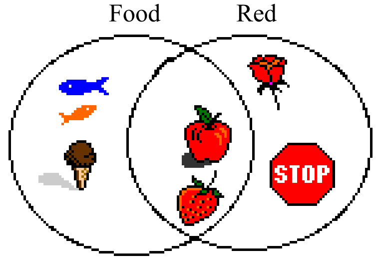
(T10, P10.1, P10.14, P10.15, P10.28, P10.29, P10.42, T11, P11.14, select, M3.3, M3.4, *3, T8, P8.1, select, M2.7)

Figure 6. Worksheet C

While the drawn images may appear similar in presentation, the construction using menu bar, tool, and palette options are again highlighted through use of the software map.

In the final activity (Venn Diagrams, pp. 149-150) students use Kid Pix to draw two large overlapping circles on the screen (T4, P4.1, draw). Then the student must enter names of two groups of objects above the left and right circles, respectively (T9, P9.6, P9.15, *2, P9.4, P9.18, P9.5, P9.4). Then the student stamps images-in the correct circle, as well as the overlapping segments (T10, P10.4, P10.5, P10.54, P10.59, P10.84, P10.109) .The result is printed as hard copy (M2.7). Figure 7 shows the implication of the following directions (pp. 149-150):

1. Use Kid Pix to draw two large circles on the screen. The circle tool will make it easy. Draw the circles so they come together in the middle like the ones you see below.
2. Name the group of objects that you will stamp in the left circle. Write that name above the left circle.
3. Name the group of objects that you will stamp in the right circle. Write that name above the right circle.
4. Add at least five stamps to your drawing. Put each stamp in the correct circle. If it fits in both circles, put it where the circles come together.



(T4, P4.1, draw, T9, P9.6, P9.15, *2, P9.4, P9.18, P9.5, P9.4, T10, P10.4, P10.5, P10.54, P10.59, P10.84, P10.109, M2.7)

Figure 7. Venn Diagrams

IMPLICATIONS

Software mapping is a simple but dynamic method that facilitates access to general software program parameters and engenders knowledge of powerful but embedded and "deep-rooted" options allowing for increased inter-rater reliability in documenting or replicating richly constructed student responses. Rich and diverse student constructions can only be mediated by rich and diverse curriculum activities. Designing a sequenced set of student activities which use program menus, submenus, and palettes is constructing probable maps associated with each activity assures divergence in thinking on the part of the student. Since the software is rich with options, there will assuredly be many different responses that can be categorized.

Mapping provides a somewhat "standardized" approach for increasing teacher understanding of full-program options present in any software package although the constructed maps will vary in form, content, and screen representation. Teachers should not expect software to complement their transmission views of teaching but should be ready for expressions and representations of a unique nature.

Software, as we perceive it, enables us to see that the "...use made of a large collection of information depends on how it is organized. One of the values of computers is that they are able, on command, to reorganize information in a variety of ways, thereby enabling people to make more and better uses of the collection"(AAAS, 1993, p.202). However, software itself, mirrors or models information organization. But unfortunately this exploration of software deep structure, or concept mapping is compromised by strict reliance on mandates such as "Read and follow step by-step instructions in a calculator or computer manual when learning new procedures" (AAAS, 1993, p. 290).

It is proposed that if exploration, or free play, as it has been called, is insufficient to "map out the rich terrain" of software then it is up to the curriculum activity designer to ensure that activities are diverse rather than redundant in its use of the desktop comprised of menus, submenus and tools. Designed curriculum activities must be attended to carefully as they are "played out" on the screen to ensure existence of an adequate "window" into the psychomotor and cognitive functioning of the learner. The presence of a "window" into the mind of the student allows the teacher to form judgments as to how responses are constructed. Since the software is rich with options, there will assuredly be many different responses that can be categorized as appropriate for such worksheet activities. Authors of activities must ensure a "match" with the curriculum. The term "match" is used to ensure that curriculum activity is enhanced by the software, rather than being a burden, and does not mean getting an answer, constructed in the mind of the teacher, as the standard against which student responses are compared. The scores for software mapping might be derived from a total tally of menu, tool and palette options selected by a student. Software mapping can be used for analyzing the students' interaction with the software via process schema. Each interaction will be scored as one point in this prosess. A special note of gratitude must be extended to Dr. Albert P. Nous for his invaluable assistance and support during the pursuit of this method.

Software maps provide a unique opportunity to research how students construct new knowledge. This medium also enables researchers to record students' decision-making as they selectively integrate information into their conceptual framework. It stresses the key concepts and propositions that underlie what we label as 'cognition patterns' during software mapping. These maps show us how children's metacognition is organized and also how they organize information into a meaningful structure. This technique is also a helpful guide for (1) instructional software designers to improve and produce high quality software programs, (2) teachers and instructors to assess the performance of students as they work their way through their programs, and, (4) representing students' interactions with software.

While computer applications and the underlying assumptions of the role of computers in assessment open up doors of opportunity for the development of innovative computer based tools, they also raise serious issues. Some of the key issues relate to validity, gender, equity, instructional delivery, the mode of user interface, and responsibility to the public (Kumar, 2005). Therefore more research is needed in the development and evaluation of computer-based assessment applications that are valid on a large scale. Due to progress in computer technology, using virtual reality to stimulate hands-on assessment tasks may be useful for designing more effective computer-based performance assessment applications in terms of less obtrusive user interface and an increased sense of realism.

REFERENCES

American Association for the Advancement of Science (AAAS).(1993).Benchmarks for science literacy. New York: Oxford University Press.

Anderson, J. O., & Bachor, D. G.(1998). A Canadian perspective on portfolio use in student assessment. Assessment in Education : principles, policy & practice, vol. 5(3), 353-80

Apple Computer, Inc. (1992). Macintosh & curriculum handbook: Secondary mathematics. Cupertino, CA.

Apple Computer, Inc. (1993). Macintosh & curriculum handbook: Elementary mathematics and science. Cupertino, CA.

Ayala, C. C., Yin, Y., Vanides, J. & Shavelson, R. J. (2002, April). Investigating the cognitive validity of performance assessment with think alouds: Technical aspects. Paper presented at the annual meeting of the AERA, New Orleans, LA.

Barab, S. A., Hay, K. E., & Duffy, T. M. (1998, March). Grounded constructions and how Technology can help. The Association for Educational Communications & Technology, Techtrends, 15-25.

Barab, S. A., Fajan, B. R., Kulikowich, J. M., & Young, M. F. (1996).Assessing hypermedia navigation through Pathfinder: Prospects and limitations. Journal of Educational Computing Research, 15(3), 185-205.

Black, P., & Wiliam, D.(1998). Assessment and classroom learning. Assessment in Education: Principles, Policy &Practice, vol. 5 (1), 70-74.

Brown, R.(1989).Testing and thoughtfulness. Educational Leadership, 46(6), 31-33. Brøderbund Software. (1991). Kid Pix user's guide - Macintosh. Novato, CA.

Charleston, G. M., Villagomez, L., & Shaffer, L. B.(1989). A project to design an evaluation of the appropriateness and effectiveness of CAI packages used in remediation of basic skills. (ERIC Document Reproduction Service No. ED 321 088).

Choi, J. & Hannafin, M. (1995). Situated cognition and learning environments: Roles, structures, and implications for design. Educational Technology Research and Development, 43(2), 53-69.

Collins, A. S. (1990). Reforming testing to measure learning and thinking. In N. Frederiksen, R. Glaser, A. Lesgold, & M. G. Shafto (Eds.), Diagnostic monitoring of skill and knowledge acquisition (pp. 75-87). Hillsdale, NJ: Lawrence Erlbaum Associates.

Collins, A. S., Brown, J. S., & Holum, A. (1993). Cognitive apprenticeship: Making thinking visible. American Educator, 15(3), 6-11, 38-46.

Davies, P. (2001). 'Computer Aided Assessment MUST be more than multiple-choice tests for it to be academically credible?' In Danson M. & Eabry C. (eds) Conference proceedings, Fifth International Computer Assisted Assessment Conference, 2nd & 3rd July 2001, Loughborough University Loughborough: Learning & Teaching Development.

Dolan, R. P., & Hall, T. E., (2001). Universal design for learning: Implications for large scale assessment. IDA Perspectives, 27(4), 22-25.

Draney, K. & Wilson, M. (1997, July). Mapping student progress with embedded assessments: The challenge of making evaluation meaningful. Paper presented at the National Evaluation Institute Workshop, Indianapolis, IN.

Duffy, T. (1997). Strategic teaching framework: An instructional model for a constructivist learning environment. In C. Dills & A. Romiszowski (Ed.), Instructional development state of the art. Volume 3: Paradigms. Englewood NJ: Educational Technology Press.

Fernandez, C. & Body, K. (1997, March). Workplace simulations: The classroom Inc. design and implementation strategy. Paper presented at the annual meeting of the American Educational Research association. Chicago, IL.

Fisher, K. M. (2000). SemNet software as an assessment tool. In J. Mintzes, J. H. Wandersee, & J. D. Novak (Eds.), Assessing science understanding: A human constructivist view (pp. 197-221). New York: Academic Press.

Gentner, D., & Markman, A. B. (1997). Structure mapping in analogy and similarity. American Psychologist, 52(1), 45—56.

Harlen, W. (Ed.). (1985). Primary science: Taking the plunge. Oxford, England: Heinemann Educational.

Hart, D. (1994). Authentic assessment: A handbook for educators. Addison-Wesley Publishing Company. Menlo Park, California.

Hernandez, V., & Reese, D. D. (2004). Integrating strategies and technology in education practice: Conducting case studies — Guidelines for classroom observations. Wheeling, WV: Center for Educational Technologies, Wheeling Jesuit University.

Hickman. C.(1991). Kid Pix [Computer Program]. Novato, CA: Brøderbund Software, Inc.

Hickman. C. (1992). Kid Pix [Computer Program]. Novato, CA: Brøderbund Software, Inc.

Jolicoeur, K. & Berget, D. E. (1989). Implementing educational software and evaluating its academic effectiveness: Part I., Educational Technology, 28(9),7-13.

Kearsley, G. (1985). Microcomputer software: Design and development principles. Journal of Educational Computing Research. 1(2), 209-220.

Koker, M. (2001). What research says about SEPUP. Ronkonkoma, NY: Lab-Aids, Inc.

Koretz, D. (1998). Large-scale portfolio assessment in the US: evidence pertaining to be quality of measurement. Assessment in Education : principles, policy & practice, vol. 5 (3), 309-34

Kozma, R. B. (1991). Learning with media. Review of Educational Research. 61(2), 179-211.

Kumar, D. (2005). Computers and Assessment in Science Education, ERIC Digest (On line) Retrieved Feb. 25, 2005 From http://www.ericdigests.org/1997-1/science.html

Kumar, D. D. (1994). Hypermedia: A tool for alternative assessment? Educational & Training Technology International, 31 (3), 59-66.

Kurtz, K. J., Miao, C.-H., & Gentner, D. (2001). Learning by analogical bootstrapping. The Journal of Learning Sciences, 10(4), 417-446.

Lajoice, S. P. (1995). A framework for authentic assessment in mathematics. In Thomas A. Romberg (Ed.), Reform in School Mathematics and Authentic Assessment, State University of New York Press, Albany.

Lee, S.A. (2002). Planning curriculum in science. Wisconsin Department of Public Instruction. Milwaukee, WI. pp.127–129

Linn, M., Lee, K., Levenson, J., Spitulnik, M., & Slotta, J. (2000). Teaching and learning K-8 mathematics and science through inquiry: Program reviews and recommendations. North Central Regional Educational Laboratory.

Litchfield, B. C. (1992). Evaluation of inquiry-based science software and interactive multimedia programs. The Computing Teacher, 6, 41-43.
Lounge, J. P., Schneider, B., Ericson, F., Vank, J. A., & Cooney, J. B. 1986). Micros and Elementary Math Lesson Plans: A Directory of Software for Achieving Educational Objectives and Procedures for Evaluating Software (ERIC Documentation Reproduction Service No. ED 306139).
McCaslin, M. & Good, L. (1992). Compliant cognition: The misalliance of management and instructional goals in current school reform. Educational Researcher, 21(3), 4-17.
McLellan, H. (1993). Evaluation in a situated learning environment. Educational Technology, 33 (3), 39-45.
Newby, T. J., Stepich, D. A., Lehman, J. D., & Russell, J. D. (1996). Instructional technology for teaching and learning: designing instruction, integrating computers, and using media. Englewood cliffs, NJ: Prentice-Hall.
Nickerson, R. S. (1989). New directions in education assessment. Educational Researcher, 18 (9), 22-27.
Pettig, K. L., (2000). On the road to differentiated. Education Leadership, 8, 1, 14-18.
Pisha, B., & Coyne, P. (2001). Smart from the start: the promise of Universal Design for Learning. Remedial and Special Education, 22(4), 197-203.
Raychaudhuri, S. (1998). Self-assessment. Assessment in Education : principles, policy & practice, vol. 5 (1), 75-77.
Reese, D. D., & Hergert, T. R. (2004). Informal dissemination of visual representations: A function of nexus characteristics. In R. E. Griffin, V. S. Williams & J. Lee (Eds.), Changing tides: IVLA book of selected readings. Newport, RI: International Visual Literacy Association.
Reese, D. D. (2003). Trees of knowledge: Changing mental models through metaphorical episodes and concept maps. In R. E. Griffin, V. S. Williams, & J. Lee (Eds.), Turning trees: Selected readings. International Visual Literacy Association.
Reiser, R. A. & Kegelmann, H. W. (1994). Evaluating instructional software: A review and critique of current methods. Educational Technology Research and Development 42(3), 63- 69.
Roberts, L. & Wilson, M. (1998). Evaluating the effects of an integrated assessment system: Changing teachers' practices and improving student achievement in science (BEAR Report Series, SA-98-2) University of California, Berkeley.
Romberg, A. T. (1995). Reform in School Mathematics and Authentic Assessment, State University of New York Press, Albany
Rose, D., & Meyer, A., (2000a). Universal design for individual differences. Educational Leadership, 58(3), 39-43.
Rose, D., & Meyer, A., (2000b). Universal Design for Learning: Associate Editor Column. Journal of Special Education Technology, 15(1), 67-70.
Rose, D., & Meyer, A., (2002). Teaching Every Student in the Digital Age: Universal Design for Learning. Alexandria, VA: ASCD.
Schueckler, L. M. & Shuell, T. J. (1989). a comparison of software evaluation forms and reviews. Journal of Educational Research, 59 (1), 17-33.
Schuerman, R. L. & Peck, K. (1991). Pull-down menus, menu design, and usage patterns in computer-assisted instruction. Journal of Computer-Based Instruction. 18(3), 93-98.
Shavelson, R. J., & Ruiz-Primo, M. A. (2000). On the psychometrics of assessing science understanding. In J. Mintzes, J. H. Wandersee, & J. D. Novak (Eds.), Assessing science understanding: A human constructivist view (pp. 304-341). New York: Academic Press.
Shepard, L. A. (1989). Why we need better assessments, Educational Leadership, 46)6), 4-9.
Siegel, M.A., & Thier, M. (2002). Issue-oriented elementary science leadership. Paper presented at the annual meeting of the National Association for Research in Science Teaching, New Orleans, LA.
Siegel, M.A., Hynds, P., & Thier, M. (2002). Scaling our work: The ESTL Project as an example of a learning sciences reform. In Bell, P., Stevens, R., and Satwicz, T. (Eds.) Keeping Learning Complex: The Proceedings of the Fifth International Conference of the Learning Sciences. Mahwah, NJ: Erlbaum.
Siegler, R. S. (1991). Children's Thinking (2nd ed.). Prentice Hall Englewood Cliff, New Jersey.
Sizer, T. R., (2001). No two are quite alike: Personalized learning. Educational Leadership 57(1).
Slack, M. (1993). Alternative assessment: Can real-world skills be tested? ERIC Document Reproduction Service No. 3625750.
Stecher, B. (1998). The local benefits and burdens of large-scale portfolio assessment. Assessment in Education: principles, policy & practice, vol. 5 (3), 335-52
Thier, H. D., Seaver, D.B. & Walhof, L.K. (1999). Development and assessment of Science and Sustainability, the SEPUP issue-oriented high school science course. In Proceedings of the Ninth International Organization for science and Technology Education (IOSTE) Symposium.

Thier, H.D. & Daviss, B. (2001). Developing inquiry-based science materials: A guide for educators. Teachers College Press. New York, NY.
Tolhurst, D. (1992). A checklist for evaluating content-based hypertext computer software. Educational Technology, 33(3), 17-21.
Voogt, J. (1990). courseware evaluation by teachers: An implementation perspective. Computers and Education, 14, 299-307.
Wilson, M. (1995). Assessment Nets: An alternative approach to assessment in Mathematics achievement. In Thomas A. Romberg (Ed.), Reform in School Mathematics and Authentic Assessment, State University of New York Press, Albany.
Wilson, M., Thier, H., Sloane, K., & Nagle, B. (1996, April). What have we learned from developing an embedded assessment system? Paper presented at the annual meeting of the American Educational Research Association, New York.
Wilson, M., & Sloane, K. (2000). From principles to practice: An embedded assessment system. Applied Measurement in Education, 13(2), 181–208.
Wolf, A. (1998). Portfolio assessment as national policy: the National Council for Vocational Qualifications and its quest for a pedagogical revolution. Assessment in Education: principles, policy & practice, vol. 5 (3), 413-446
Wolf, D. P. (1989). Portfolio assessment: Sampling student work. Educational Leadership, 46(6), 35-39.
Wolf, D., Bixby, J., Glenn III, J., & Gardener, H. (1991). To use their minds well: Investigating new forms of student assessment. Review of Research in Education 17:31-7.
Yin, Y., Vanides, J., Ruiz-Primo, M. A., Ayala, C. C., & Shavelson, R. J. (2004, April). A comparison of two construct-a-concept-map science assessments: Created linking phrases and selected linking phrases. Paper presented at the annual meeting of the American Educational Research Association (AERA), San Diego, CA.
Yin, Y., & Shavelson, R. J. (2004, April). Generalizability analysis for concept mapping assessment of students' science achievement. Paper presented at the annual meeting of the AERA, San Diego, CA.
Yin, Y., Ayala, C. C., & Shavelson, R. J. (2002, April). Students' problem solving strategies in performance assessment: Hands on and minds on. Paper presented at the annual meeting of the AERA, New Orleans, LA.