Students’ use of technological features while solving a mathematics problem

Hollylynne Stohl Lee*, Karen F. Hollebrands 1

326C Poe Hall, Campus Box 7801, North Carolina State University, Raleigh, NC 27695, United States

Available online 17 October 2006

Abstract

The design of technology tools has the potential to dramatically influence how students interact with tools, and these interactions, in turn, may influence students’ mathematical problem solving. To better understand these interactions, we analyzed eighth grade students’ problem solving as they used a java applet designed to specifically accompany a well-structured problem. Within a problem solving session, students’ goal-directed activity was used to achieve different types of goals: analysis, planning, implementation, assessment, verification, and organization. As we examined students’ goals, we coded instances where their use of a technology feature was supportive or not supportive in helping them meet their goal. We categorized features of this applet into four subcategories: (1) features over which a user does not have any control and remain static, (2) dynamic features that allow users to directly manipulate objects, (3) dynamic features that update to provide feedback to users during problem solving, and (4) features that activate parts of the applet. Overall, most features were found to be supportive of students’ problem solving, and patterns in the type of features used to support various problem solving goals were identified.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Applet; Technology; Mathematics; Design; Middle school; Problem solving; Research methodology

1. Introduction

In the past 20 years, a greater emphasis has been placed on students’ mathematical problem solving as reflected in standards for school mathematics (e.g., National Council of Teachers of Mathematics, 1989, 2000). Within the same time frame, there has been an increase in the development of technological tools that are used in mathematics classrooms for a variety of purposes, one of which is to facilitate students’ problem solving. In a recent review of literature, Clements (2000a,b) makes a clear case for how computers can, among other things, enhance students’ problem solving by providing an environment to engage in playful exploration, test ideas, receive feedback, and make their understanding public and visible.

What one considers to be a “problem” in mathematics includes, at one extreme, well-structured, less complex exercises of a routine nature often found at the end of a section in a textbook or, at the other extreme, ill-structured complex questions that are of an abstract or applied nature (Jonassen, 2004). The type of problem in which students engaged in our research study could be characterized as: well-structured with a clear set of constraints but non-routine
in nature, such that an immediate solution path or algorithm is not known to the solver (even though an algorithm may be available to a person with more mathematical knowledge), and includes more than one solution. Thus, the problem on which we focused could be described as existing between the two extremes described previously. We view problem solving as a dynamic process that involves the problem-solver in an iterative cycle of problem interpretation and/or problem posing with subsequent goal-oriented mental or physical activity. Throughout students’ problem solving processes, he or she may develop and work towards many goals that help reach the desired solution state.

The mathematics and mathematics education communities certainly have used the classical works of Polya (1957) to inform the teaching of problem solving processes and skills. Schoenfeld (1985, 1987, 1992) contributed a framework of different factors that affect students’ abilities to solve problems. In his framework, four components comprise the major aspects of students’ problem solving:

Resources: facts, definitions, procedures, rules, and intuitive understandings of mathematics.

Heuristics: strategies and techniques for approaching a problem.

Control: the ways in which one monitors their own problem solving process, uses their observations of partial results to guide future problem solving actions, and decides how and when to use available resources and heuristics.

Beliefs: what one believes about mathematics, mathematics tasks, and what it means to do mathematics. [Note: Although beliefs are an important component, for the purposes of this paper, we are focusing on the first three aspects.]

In a technological environment, the resources available to students also include those made possible by the particular technology tool being used. A student needs knowledge of how to use various features of a technology tool from an operational standpoint as well as knowledge of what the various objects and actions on those objects can afford for the problem solving process. Thus, within a technology environment, it is necessary to expand the definition of resources to include both mathematical resources and features of a technology tool available for students. As Healy and Hoyles (2001) found in the context of a dynamic geometry environment, the presence of particular technological resources (features) can also afford or constrain certain actions which can affect the available heuristics a student may use during problem solving.

As students are solving a problem, they not only need to implement heuristics and utilize resources, but they also need some mechanism to evaluate their progress so that they are aware of, and critically examining, their own decision-making. We believe mechanisms for control may be influenced by the features made available in technology tools such that students can use resources in the technology tool along with their mathematical resources as means for reflection that result in control activities.

2. Design of technological tool features

The design of technology tools has the potential to dramatically influence how students interact with tools and these interactions in turn may influence students’ mathematical understandings and problem solving. “The knowledge and the relation to knowledge which is allowed by a piece of software” has been characterized as “epistemological validity” (Balacheff & Sutherland, 1994, p. 1). For example, students may come to understand certain mathematical ideas differently in a dynamic geometry environment in which students are able to manipulate objects directly through dragging rather than through inputting commands (e.g., when programming in Logo). The construction of a square in Logo may focus the user’s attention on the fact that a square is a quadrilateral with four congruent angles and sides whereas a constructed square in a dynamic environment may focus the user’s attention on the fact that a square is a rectangle with two adjacent congruent sides.

The design of a technology tool also influences how problem solving can be “supported” through the use of such tools. Indeed, the design and use of tools can be considered as an enactment of one’s epistemological and pedagogical beliefs about mathematics, problem solving, and teaching. How one perceives what it means to know mathematics certainly affects the interpretation of what constitutes a problem, what it means to solve a problem, and what mathematical knowledge can be developed through problem solving. This in turn, affects the pedagogical approaches one takes toward developing students’ problem solving heuristics and the way in which technology is perceived as facilitating problem solving and the construction of mathematical knowledge (Schoenfeld, 1992).
A few researchers have suggested principles to guide the development of technological tools to support students’ learning of mathematics and students’ mathematical problem solving (e.g., Clements, 2000a; Underwood et al., 2005; Yerushalmy, 2005). Some of the principles Clements has suggested that are relevant to our work include recommendations that technology should provide tools for students that allow them to test ideas and receive feedback, engage in playful mathematical exploration, and directly manipulate objects. Underwood et al. (2005) agree with these suggestions and also recommend that technological tools should support multiple solution strategies and approaches, employ multiple representations, and make links between representations obvious. In working with interactive diagrams (applets) that are embedded in digital textbooks, Yerushalmy (2005) suggests different functions of such diagrams — which typically include multiple representations of a concept — to narrate, complement or elaborate text found in the digital textbooks. Within each of these types of diagrams, users have opportunities to interact with the diagrams in ways that promote problem-solving activities, even if the purpose of the diagram is not specifically to solve a particular problem. In addition, Ainsworth (1999) indicates that when multiple representations are available for students to use, they can make choices based on their own familiarity and comfort. Also, multiple representations may serve complimentary functions by providing different information or the same information in different ways.

The principles that guide technological tool designers vary and are apparent in the features and interactivity that are provided in the subsequent tools they design. Thus, although there may be general categories of the types of features in various technology tools, the specific enactment of particular features will likely be aligned with the mathematical domain of interest and intended learning or problem solving goals of the designers. The intent of our study is to examine how students interact with particular features in a software tool while solving a mathematical problem. From this, we hope to glean some insight about when and how students use various technology features and whether or not these uses appear to be supporting their problem solving goals.

3. Methods

3.1. The technological tool

The problem (Fig. 1) used in this study is designed with an accompanying java applet to give students access to different possible solution strategies and representations for making sense of the problem2. Although students can solve this problem using manipulatives or paper-and-pencil, the intent of using the problem situation and applet is to allow students to enact different strategies and solution paths beyond a numerical or symbolic approach. In addition, the well-structured nature of the applet requires students to only drag and drop fish icons rather than knowing specific technology skills in more generic software tools (e.g., graphing calculator, spreadsheets). Thus, we felt a context requiring little technology skills would be fruitful for analyzing students’ problem-solving processes and the extent to which the students used features in the applet.

The Fish Farm problem is well-structured and has more than one solution. Considering the problem context as situated in the real world where every child will want some fish in their pond, there are two different solutions with positive integer values. To help facilitate students’ work on the problem, the applet has a tank on its left side with 26 fish (13 males, 13 females) and three ponds representing each of the children’s backyard ponds described in the problem.

There are two static features in the applet that we included in our analysis. First, the goal ratio to the right of the ponds indicates the ratio of red (male) and yellow (female) fish each child wants in his or her pond. The second feature is the color coordination of the fish, goal ratio, status ratio, and the pie graph. The status ratio and pie graph are dynamic features that provide feedback to the user about the number of red and yellow fish in each pond. The status ratio displays the ratio as a part-part representation (e.g., a pond with three red fish and one yellow fish, 3:1) while the pie graph provides a part-whole representation of the contents of the ponds (e.g., a 3:1 ratio would be represented as 75% red and 25% yellow).

---

2 During 1999–2001, middle school mathematics problems and accompanying java applets were designed and implemented as part of a grant funded by the National Science Foundation (REC #9804930) Educational software components of tomorrow: A Testbed for sustainable development of interoperable objects for middle school mathematics (ESCOT). The problems and applets are hosted at the MathForum at Drexel at mathforum.org/escotpow.
A Fishy Family

For their birthday, the Carp triplets received 26 tropical fish: 13 females and 13 males. They discussed ways to divide the fish among their three tiny backyard ponds.

Angel said, "I want the same number of male and female fish in my pond."
"Okay," said Molly. "I want three times as many males as females in my pond."
"Then I want twice as many females as males in my pond," Gar replied.

Is there a way to put all 26 fish into those three ponds, while giving each triplet what he or she wants? Use the applet to explore this question.

![Fish Farm problem and applet](http://mathforum.org/escotpow/solutions/solution.ehtml?puzzle=40)

The user has many ways to directly manipulate the objects in the applet. They can “drag and drop” fish from the tank on the left into any of the three ponds to its right (fish tank-to-pond). As a fish is dropped into a pond, they begin to “swim” within the boundaries of the pond (fish animation) and the status ratio and pie graph are updated. The fish can also be moved from the ponds back to the tank (fish pond-to-tank), between ponds (fish pond-to-pond), and within the tank itself (fish tank-to-tank). In addition, fish can also be moved onto the green areas of the applet representing grass and on top of the red bricks (fish to non-water). Two other objects within the applet can also be moved: the water and the red bricks outlining the pond. These features appear to be an artifact of the software used to create the applet (Agent Sheets, Repenning, 2002) such that every object in the applet is actually an “agent” that has capabilities. In the case of the squares of water and rectangular bricks, they were not programmed to be stationary background objects, as one might expect. Instead, clicking and dragging on each of these objects allows one to move it to a variety of locations within the applet. When the problem is solved (i.e., all fish removed from tank and distributed to ponds in way that all three ratios are correct), the bricks surrounding the tank turn green, indicating to the student that they have a correct solution (solution status).

There are several activation features along the bottom of the applet. The Run button at the bottom of the screen is used to activate the applet so that the “updates” and “swimming” occur when a fish is dropped in a pond; however, a user can move the fish without hitting the Run button. The Stop button deactivates the “updating” and “swimming” fish features. For each time a user presses the Step button (depicted in Fig. 1 as an arrow to the right of the Run button), the fish will only move one step in their swimming motion in the ponds. In addition, if a user drags a fish in or out of a pond while the Step button is activated, the status ratio and pie graph do not update with the correct number of fish in the ponds. The status ratio and pie graph only accurately update once the Run button has been activated again. The
Table 1
Participants in study

<table>
<thead>
<tr>
<th>Pair</th>
<th>Description of students and teacher</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One African American female (SF1)</td>
</tr>
<tr>
<td></td>
<td>One Caucasian male (SM1)</td>
</tr>
<tr>
<td></td>
<td>Caucasian female preservice teacher</td>
</tr>
<tr>
<td>2</td>
<td>One African American female (SF2)</td>
</tr>
<tr>
<td></td>
<td>One Caucasian male (SM2)</td>
</tr>
<tr>
<td></td>
<td>Caucasian male preservice teacher</td>
</tr>
<tr>
<td>3</td>
<td>Two African American males (SM3, SM4)</td>
</tr>
<tr>
<td></td>
<td>Caucasian male preservice teacher</td>
</tr>
<tr>
<td>4</td>
<td>Two African American females (SF3, SF4)</td>
</tr>
<tr>
<td></td>
<td>Caucasian male preservice teacher</td>
</tr>
</tbody>
</table>

Reset button will place all 26 fish back into the tank on the left and updates the status ratio and pie graph accordingly, while the Clear button will erase all fish from the applet but does not update the representations for each pond. These activation features are the default buttons created when rendering an applet using Agent Sheets (Repenning, 2002).

We categorized the features into four subcategories: (1) features over which a user does not have any control and remain static, (2) dynamic features that allow users to directly manipulate objects, (3) dynamic features that update to provide feedback to users during problem solving, and (4) features that activate parts of the applet. To determine if these features do in fact assist students in their problem-solving activities, we needed to examine how students interacted with the tool.

3.2. Data sources

The data for the current project were collected as part of a larger study on preservice teachers’ learning to facilitate students’ problem solving of the Fish Farm problem (Lee, 2005). Each preservice teacher worked with a pair of eighth-grade students (13–14-year-old) from below average ability “Math 8” classes (not algebra or pre-algebra) that include students from diverse backgrounds. The preservice teachers were juniors in an undergraduate secondary mathematics education class focused on learning to teach mathematics with technology. The preservice teachers and students were videotaped so as to capture the computer screen (PC-to-TV converter) as well as social interactions (webcam). For the current study, our purpose was to do a fine-grained analysis of students’ problem solving and the features of the applet that they utilized during their work. Thus, we chose to limit the scope of the study to four pairs of students working with a preservice teacher on the first question of the Fish Farm problem (see Table 1). These pairs were selected based on the high quality of sound and video images in order to allow the most careful analysis of students’ interactions with the technology tool. For the purposes of this paper, we treated the students and the preservice teacher as a unit for the analysis. A more detailed analysis that focuses on the role of the preservice teachers appears in a separate paper (Lee, 2005).

3.3. Analysis techniques

Each videotape was transcribed and annotated with screenshots to depict students’ work with the applet, descriptions of what they did with objects in the applet, where they may have pointed (physical gestures as well as mouse movements) within the applet, and what was being written on paper. Researchers viewed each videotape and its corresponding transcript to segment the transcript according to goals towards which students were working as they solved the problem. We define a goal as: an objective or purpose towards which effort is directed. Thus, a problem solving goal was explicitly or implicitly implied in the work students put forth in an effort to resolve a problem. For the purpose of analysis, a goal was identified when a student or teacher either:

(1) Explicitly stated a new goal (e.g., “Molly wants 3 to every 1 so put 3 of these red ones here and put 1 [yellow]”), or
(2) Began exerting activity (verbal or physical) towards a goal (e.g., student moves 1 red fish, 2 yellow fish, then 2 red fish and 1 yellow fish from Gar’s pond over to tank, leaving 1 red and 2 yellow fish in Gar’s pond).
In the latter case, the intention of the goal was inferred by the researchers from their activities (e.g., move fish out of Gar’s pond into tank to still meet Gar’s 1:2 ratio). Within a problem solving session, students’ goal-directed activity was used to achieve different types of goals. Thus, we felt it was important to classify each goal according to its overall purpose in students’ work. Adapting from Schoenfeld’s (1985) stages of problem solving, we considered six types of goals: analysis, planning, implementation, assessment, verification, and organization. We have defined these different goals as follows:

- **Analysis**: Students attempt to understand the problem, its constraints, and consider useful perspectives for the problem.
- **Planning**: Students work towards devising a sequence of actions or formulate a strategy to pursue a solution.
- **Implementation**: Students carry out a pre-planned strategy, which may or may not be explicitly stated.
- **Assessment**: Students take explicit actions to monitor their progress towards a solution.
- **Verification**: Students believe they have a solution and either evaluate whether or not it is correct, or engage in activities to explain or justify their solution.
- **Organization**: Students take actions to arrange their environment (either with paper or in the technology tool) in a way that they perceive will be beneficial in working towards a solution.

In our coding, each goal-directed segment in the transcript was identified as being in one of the above six categories. We created each goal segment at a “grain size” level such that every problem solving goal would be classified as only one of the six types. This resulted in some of the problem solving episodes having a large number of goal segments (e.g., Pair 3 had 69 segments). Within each goal segment, we determined which technology features students were employing and whether students’ use of these features was supporting (S) or not supporting (NS) their goal-directed activity. A feature is supporting if its use assists a student in making progress toward his or her goal. A feature is not supporting if its use distracts students from their work toward a goal or does not allow them to make progress toward their goal.

When coding for students’ use of a feature within a single goal segment, if a student made use of a particular feature more than once, but its use was always supportive, rather than count each individual use we considered the collective use of the feature as a single instance. For example, if a student has an implementation goal of moving fish from the tank to Angel’s pond to meet the one-to-one ratio and they move five yellow and five red fish, one at a time, then rather than counting ten instances of using the feature “moving fish from tank to pond” we considered this as one supportive instance of this feature. However, if a student utilized a particular feature more than once during a given goal segment and its use was sometimes supportive and sometimes not supportive then this goal segment was coded as one supportive and one non-supportive instance. An instance is not a use of a feature. It is a goal segment during which time students make use of a particular feature. Although there may have been other instances where features were supportive or not supportive to students’ work towards their goal, if we had no directly observable actions or words that indicated their use of that feature, we did not record it in our coding. Thus, our counts in each of the categories do not include times when the feature may implicitly appear to be supporting their work, or when the absence of a use of a feature seems to not be supporting students’ work, from the observer’s perspective. For example, students rarely pointed to or explicitly discussed the pie graph and although they may have been using it to monitor their work, it was not coded because there were no observable actions or words to support a claim that they were using this feature to assess their work.

4. Findings

The findings are organized into two parts. The first part discusses the types of problem solving goals students employed and whether or not they made explicit use of a technological feature to pursue that type of goal. The second part is a more detailed analysis of the ways in which the different types of technological features supported students’ work toward their various problem solving goals.

4.1. Problem solving goals

Across all four pairs of students’ work, the students engaged in all six types of problem solving goals as they pursued solutions to the Fish Farm problem. In all cases, the students were successful in finding two solutions to the problem.
Fig. 2. (a) Pair 1 students’ problem solving goals. (b) Pair 2 students’ problem solving goals. (c) Pair 3 students’ problem solving goals. Note: Due to the large number of goal segments, we are only displaying the first 41 (out of 69) goal segments, through their verification of the first solution. (d) Pair 4 students’ problem solving goals. Key: A black rectangle indicates students’ use of a technological feature during a goal segment while a gray rectangle indicates that students did not use any technological feature to work towards a goal in a given segment.

Fig. 2a–d provide a visual diagram that represents the order in which students engaged in different goal activities, not accounting for time elapsed, and whether or not they utilized a technological feature in the tool to assist in their work. These diagrams are adapted from a technique used by Schoenfeld (1985). There were a total of 140 goal segments across the four pairs: 54 (39%) implementation, 25 (18%) analysis, 24 (17%) planning, 21 (15%) assessment, 9 (6%) verification, 7 (5%) organization. Collectively, students used at least one technological feature during 90 of the 140 goal segments.

The number of goal segments and the pattern in student’s engagement in these six types of goal-directed activity varied (see Fig. 2a–d). For example, the students working in Pair 1 (see Fig. 2a) obtained two different solutions to the problem in the fewest number of goal segments (15), while the students working in Pair 3 engaged in 69 goal segments before reaching both solutions to the problem. The two problem solving sessions that had the greatest number of goal
segments (Pair 2 and 3), had frequent implementation–assessment–implementation sequences (see Fig. 2b and c). This back and forth movement between implementation and assessment activities was an indication that students were engaging in control activities to monitor their work. The sessions that had fewer number of goal segments (Pair 1 and 4) also appeared to engage in fewer monitoring behaviors with assessment or revisiting analysis and planning goals after implementation.

Of the 54 implementation goals across all four groups, 33 (61%) were immediately preceded by planning or analysis. Of the other 21 implementation goal segments, 10 were preceded by assessment goals. Students’ engagement in pursuing assessment, analysis, and planning types of goals, is an indicator that they were exerted control activities in their problem solving.

Students generally did not make use of a technological feature when they were engaged in an analysis or planning goal (32 non-technology out of 49, 65%). Students often stepped back from the technology and focused on the problem statement and utilized their mathematics resources to devise a plan that they later implemented with technology. The one exception occurred during the work of Pair 4. In this problem solving session, students used the technological features during all 7 of their analysis goal segments. It is interesting to note that during this same problem solving session there were only six implementation goals. The first five analysis goals were formulated by the preservice teacher while he and the students were making explicit use of the technological features (especially the goal and status ratios). As they worked toward these analysis goals, they were able transition to planning and this was followed by implementation. Perhaps because they spent an extended amount of goal-directed activity focused on the problem and the technology features during analysis, they needed fewer implementation goals to pursue their solution.

Of the 21 assessment goals, 15 did not involve the use of technology (71%). Those that did use technology features often referred to the goal ratio, and the various dynamic feedback features. During assessment, planning, and analysis (control type activities) the students made no explicit use of technological features about two-thirds of the time. Overall, students seemed to refer more often to their mathematical resources during these control activities.

Students did not pursue many verification (9) or organization (7) goals. Because there were two solutions to the problem, students generally pursued two verification goals using technological features, one after they obtained each solution. The exception occurred when students in Pair 2 employed two consecutive verification goals after finding the second solution. Of the seven organization goals, three of them occurred after the first solution was obtained and Verified and students Organized their environment to find the second solution. The other organization goals were interspersed with students’ work on the problem. Three of the organization goals did not involve technology.

To better understand the ways in which students’ work on this problem was supported by the features in this technological tool, we focused the next part of our analysis on the 90 goal segments in which students explicitly made use of a specific feature. During many of these 90 goal segments, students made use of more than one feature. Table 2 contains the tallies of the number of instances of a feature is supportive or non-supportive of students’ different types of problem solving goals. In the following sections, we provide a closer look at the ways students used static features, dynamic direct manipulation features, dynamic feedback features, and activation features.

4.2. Static features

Features described as “static” included the goal ratio and the color correspondence among fish, graph, goal ratio, and status ratio. There were a total of 17 instances of students’ explicit reference to or use of the goal ratio that were coded as supportive and 0 instances that were coded as non-supportive. There were also a total of seven instances of students’ explicit reference to or use of the color correspondence that were coded as supportive and four instances that were coded as non-supportive.

The goal ratio was used by students as they worked toward five of the six different types of problem solving goals: analysis, planning, implementation, assessment, and verification. The goal ratio was not used by students when they were working toward a goal of organization. The presence of the goal ratio served as a reminder to students about the ratio of red and yellow fish that were desired by Angel, Molly and Gar and they referred to it often when they were analyzing the problem. For example, at the beginning of one pairs’ work on the problem a student stated that, “Molly wants 3 to every 1 so put 3 of these red ones here and put 1 yeah” as she pointed to the goal ratio. In this case, the student pointed to the goal ratio as she was planning. This same pair of students referred to the goal and status ratio after they obtained a solution and were asked to Verify it.
Table 2
Number of instances when a feature is supportive (S) or non-supportive (NS) to students’ different types of problem solving goals

<table>
<thead>
<tr>
<th>Feature/categories</th>
<th>Feature Type</th>
<th>Type of problem solving goal</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AN</td>
<td>PL</td>
</tr>
<tr>
<td><strong>Static features</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Goal ratio</td>
<td>S</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Color coordination</td>
<td>S</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Fish tank-pond</td>
<td>S</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fish pond-pond</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fish pond-tank</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fish tank-tank</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fish animation</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fish to non-water</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Move water</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Move bricks</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pie graph</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Dynamic features: direct manipulation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status ratio</td>
<td>S</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Solution status</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stop</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Run</td>
<td>S</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Arrow</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Reset</td>
<td>S</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Clear</td>
<td>S</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>NS</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>All</td>
<td></td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>NS</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Key: analysis (AN), planning (PL), implementation (IM), assessment (AS), verification (VE), organization (OR), supportive (S) and non-supportive (NS).

Preservice teacher (PST): So if you were to tell somebody how to do this problem, how would you go about doing it? How would you tell them to start off? What would be their strategy?
SF2: I’d do it like we did it up there.
SM2: It’s like reducing...ah, using ratios.
SF2: It’s just like making everything equal, you like make everything equivalent to the ah, what’s given to you (the goal ratio), you’ve got to make it, you’ve got to simplify it, whatever you’ve got (the status ratio) you have to simplify it down to that one (the goal ratio). [Pair 2, 136–147].
The goal ratio seemed to serve an important role for students as they were solving problems, by maintaining an iconic representation of the goal they were trying to achieve for each pond in front of them at all times. This proved to be helpful when students were implementing a strategy or when students paused to assess if what they were doing was productive. Also, because the goal ratio was recorded in the applet for them, this was one less piece of information that they needed to remember and could focus more on how to solve the problem rather than on recalling what ratio of red and yellow fish were needed in each pond.

Another static feature of the applet involved the color coordination among fish, pie graph, goal ratio, and status ratio. The male fish were represented using red fish and the female fish were represented using yellow fish. The same red and yellow fish were used to describe the status and goal ratios and the same colors were used to represent the number of male and female fish in the pie graph. The feature was coded as supportive or not supportive when students made explicit reference to the coordination of the colors. Instances when students used a color name as a label for a particular fish (e.g., “move the red fish.”) were not coded. The consistency in the colors that were used helped to support students comparing the status ratio to the goal ratio and interpreting the pie graph when they were analyzing the problem, creating a plan, implementing a strategy, assessing their work, and verifying a solution. Cases when the color coordination was not supportive included instances when students had difficulty interpreting the status or goal ratio in terms of male and female fish; these were the labels that were used for the fish in the statement of the problem.

SM3: Which one is the red?
PST: The red is the males. So see where it says Molly’s...she wants 3 to 1, so three males for every one female [...]

SM4: [drags 3 red and 1 yellow fish into Molly’s pond]
PST: Okay you’ve got three females in there, so how many males are you going to need to make Molly happy?
SM4: One?
PST: The other way...the males are the red. See what I am saying. [Pair 3, 74–94].

Coordinating the problem statement with the applet required students to remember that red fish were male and yellow fish were female. However, a student could solve the problem by focusing only on red and yellow fish and many students chose to do this. Hence, the reference to fish by gender in the problem statement and by color in the applet seemed to increase the cognitive demands for some students during the four non-supporting instances of color coordination.

4.3. Dynamic features: direct manipulation

There are eight dynamic features in the category of direct manipulation. The direct manipulation in the applet is made possible through the use of the Agent Sheets (Repenning, 2002) software used to create the environment. Six of these features involve the movement of fish “agents” within the applet, and two concern the movement of background “agents” that one would not naturally expect to be able to manipulate (water and bricks around the tank). There were many goal segments during which students made use of these dynamic features during students’ problem solving, and most instances were supportive (68 supportive and 20 non-supportive). The vast majority of instances that involved the use of these features occurred when students were engaged in pursuing an implementation goal.

Students made substantial use of the feature to move fish from the tank into a pond (43 instances), and its use was always supportive for helping students meet their goals. Students also made use of the feature for moving fish between ponds (five instances) and the feature for moving fish from the pond back to the tank (19 instances) in supportive ways. Thus, these three features were always supportive of student’s work towards meeting their implementation, analysis and organization goals, and these three features also allowed them to use different strategies for distributing the fish and establishing correct ratios in the ponds. Consider students’ work towards an implementation goal of removing fish from Molly’s pond when they realized 6:3 does not meet the 3:1 ratio.

[Students had status ratios of 3:3 for Angel’s pond, 6:3 for Molly’s pond and 1:2 for Gar’s pond.]

SF2: Take one out [of Molly’s pond]. 3 to 1 would be 6 to 2.
SM2: Right here. Like that? [takes out one red fish from Molly’s pond and returns it to the tank.]
SF2: Now you’ve got five [red fish].
PST: So now the ratio is 5 to 3 and you want 3 to 1.
SF2: Let me see. [Takes the mouse from SM2 and moves same red fish from tank to Molly’s pond and moves a yellow fish from Molly’s pond back to the tank. Molly’s pond now has 6:2.] [Pair 2, 106–115].

The features of moving fish from tank to pond and pond back to tank in conjunction enable students more flexibility in their problem solving by allowing them to perform an operation and reverse it. Both of these features were supportive in helping students meet their goal. Without the feature of being able to move fish from the pond back to the tank, students would have been limited to a unidirectional movement. Thus, any change in their strategy or notice of an error would require students to reset the applet and start again. Thus, the combination of the two unidirectional features provided a bidirectional manipulation tool that provided flexibility in changing or altering strategies during problem solving.

The ability to rearrange fish within the tank was rarely used and was both supportive and non-supportive. In this supportive use, students rearranged the fish remaining in the tank to regroup by color and to line up in rows of two. This rearrangement was suggested by the teacher but helped the student visualize that they had enough fish to meet the 1:2 ratio (three red and six yellow) and that all the fish could be placed in Gar’s pond. The other instance when the tank to tank movement was non-supportive of students’ work towards their goals occurred in the implementation work of a single pair. One of the students had accidentally dragged and dropped a red fish in the tank on top of another red fish. The students did not seem to notice that there was one less red fish visible in the tank. This “phantom” red fish affected their problem solving in many subsequent goal-oriented actions. One of the most detrimental side-effects of this red fish being moved on top of another within the tank, was that it appeared to lose it’s visibility within the applet. That is, even when students eventually moved that fish into one of the ponds, the applet did not count it in the updates of the dynamic features of the pie graph and status ratio. Thus, these features did not provide accurate feedback to the students, and in turn, caused an extra cognitive load for them to remember that although they had a correct number of fish in Molly’s pond to meet the goal ratio, the status ratio and pie graph were not correct. This error resulted in this problem solving session having many more goals, several of which did not lead toward a solution to the problem because students were reasoning from an incorrect status ratio.

The animation of the fish was often relevant when students were attempting to directly manipulate fish to move one out of a pond. The process of needing to “catch” the fish in order to click and drag it became cumbersome and frustrating for some students when they were trying to Implement a strategy. There was one instance when students expressed frustration in verifying their work because they could not count the fish in the ponds because they were constantly in motion.

SF2: 8, 8. [pause] There’s eight yellow fishes up here?
PST: Mmhmm
SF2: I can’t see them, they are moving too fast.
SM2: You can stop them. Just look up there [points to status ratio]
SF2: I am. Don’t see 8. 1, 2, 3. I can’t count that fast. [SM2 presses the Stop button and SF2 confirms the count in Angel’s pond.] [Pair 2, 345–354.]

Thus, even though the animating fish adds a realistic feel to the applet, there were five instances when this feature was not supportive and no instances when there was evidence that it actually supported students’ problem solving.

The last three features in this category (ability to move fish to non-water locations, move squares of water and move bricks) were always non-supportive of student’s work in pursuing implementation goals, and once during an organization goal. The most critical ramification of being able to drag and drop fish onto non-water locations (e.g., the green areas surrounding the ponds or on top of a brick that outlines the pond) was that this action caused a fish agent to become non-recognizable by the applet. Thus, as in the context described when a fish was dropped on top of another fish within the tank, fish that have been dropped in non-water locations cause a mismatch between the number of fish present in a pond and the feedback provided by the status ratio, pie graph, and ultimately, the solution status (bricks turning green when solution is met).

The same phenomenon occurred five times when a square of water was dragged and dropped into a new location. Any fish that was then dropped on top of that relocated square of water was not counted by the applet and caused extra cognitive load and frustration for the students in their problem solving. There was even 1 instance when students spent a considerable amount of time moving water around within and between the ponds to try to organize the applet and
“fix” the counting error and this distracted students from their work on the mathematical problem. There were only three instances (all non-supportive) of students moving the bricks around the tank. This action was done accidentally as students were trying to drag and drop a fish that happened to be on or near a brick. However, it was clear that the fact that they could move a brick caused surprise and a short distraction from their problem solving.

4.4. Dynamic features: feedback

Overall, the status ratio feature seemed to be the most supportive of students’ problem solving. There were 13 instances of students’ explicit references to the status ratio that were coded as supportive and two instances that were coded as non-supportive. There was one instance of students’ explicit reference to or use of the pie graph that was coded as supportive and 1 instance that was coded as non-supportive. There were five instances of students’ explicit reference to the solution status feature that were coded as supportive and three that were coded as non-supportive.

The status ratio was often supportive of students when they were engaged in working toward a verification goal. As an example, after SF1 and SM1 had placed a correct number of red and yellow fish in each of the ponds, the PST asked the students to explain why the status ratios were equivalent to the goal ratios.

PST: So Angel has 3 and 3 (referring to the status ratio), right? […] and do you understand why 3 to 3 is like 1 to 1 (referring to the goal ratio)?
SF1: because they divide each other.
PST: How about 6 and 2 (referring to the status ratio for Molly’s pond)?
SM1: I was thinking about that 3 times 2 is 6, and 1 times 2 is 2 [Pair 1, 91–98]

When students were verifying their solution they generally compared the ratio of fish that they had placed in each of the ponds, represented by the status ratio, with the ratio needed in each of the ponds, which was depicted by the goal ratio. The fact that the goal and status ratio were in close proximity to each other in the applet may have also supported these comparisons.

Students made limited use of the pie graph and it was both supportive and non-supportive as students assessed their work. For example, as students were moving fish from the tank to the ponds, they placed one too many yellow fish in Gar’s pond and this seemed to prompt one of the students to ask a question that focused the groups’ attention on the pie graphs.

SF2: Uh uh. Put one back. [SM puts one red fish into Gar’s pond. The status ratios display 1:1 for Angel’s pond, 3:1 for Molly’s pond, and 1:3 for Gar’s pond.]
SM2: What does the circle mean?
PST: Have you guys studied pie charts?
SF2: Kinda […]
PST: Well you see in Molly’s pond […] you have four fish, do you want to explain SM2?
SM2: I know how to do it […] there’s one out of three. One to three.
SF2: Gar wants one to two.
PST: So what do you think the ratio is there, or the percentage, what do you think the percentage is there?
SF2: Which one?
SM2 and SF2: Right here is 75, 25%, 75, that’s 50
PST: And which one’s 25 here [points to pie graph for Gar’s pond] and which one’s 25 here [points to pie graph for Molly’s pond] in Molly’s pond which is 25%.
SM2: The female. The female is 25%?
PST: Yeah, Good job. Take a look at Gar’s pond. Is that a 1 to 2 ratio?
SF2: No, that’s 1 to 3. [Pair 2, 49–73]

The pie graph was not supportive in that students were unsure at first how to interpret this representation and so it distracted them from the original goal of placing fish in each of the ponds. However, once students explicitly were asked to compare the pie graphs for Gar’s pond and Molly’s pond, they were able to recognize that the pie graph for Gar’s pond was showing a 1:3 ratio instead of the needed 1:2 ratio.
The solution status feature was both supportive and non-supportive when students implemented a strategy and assessed their problem-solving work. The feature was not supportive when a fish was not counted by the program and so, although the students had a correct solution, the applet did not recognize it and the bricks did not turn green. The absence of the change of color caused students to wonder whether they did indeed have a solution. The students then took on non-mathematical goal of reorganizing the fish by moving one from the pond, to the tank and back to the pond to try to correct this error and by doing so the fish was counted by the applet and the bricks turned green.

4.5. Activation features

The applet contains five buttons that serve as ways to activate different actions. Overall, these five features were supportive (18 supportive and 1 non-supportive) to students’ problem solving (Table 2). The most supportive features were the Run and Reset buttons that allowed students to activate the applet and to return all the fish from the ponds back to the tank. Students used the Run button to accomplish goals during the analysis, planning, and implementation phases of their problem solving. This often occurred because they recognized, or were reminded by the teacher, that the Run button was necessary to make the dynamic feedback features in the applet update accurately when fish were moved. Students used the Reset button when they either wanted to start an implementation sequence over, to re-analyze the problem constraints, or to re-organize the fish back into the tank to try to find a second solution. The one instance when the Run button was not supportive of students’ work occurred when students forgot to activate the button and had moved a considerable amount of fish from the tank to Angel’s pond before noticing that the feedback features were not updating. They then used the Reset and Run buttons in a supportive way to return the fish to the tank and begin again. Thus, their discovery that they had forgotten to activate the Run button was not supportive to their problem solving, while being able to Reset and Run again, allowed them to start over.

The Stop button was only used twice by the same pair of students. Once during an assessment goal, they used the Stop button because they were having difficulty keeping track of fish in the ponds and wanted to Stop the fish from swimming to count the fish. Thus, this supportive use of the Stop feature was connected to an effect of the fish animation feature that was non-supportive. The other instance of the use of the Stop button was when students wanted to pursue a second solution and the preservice teacher directed a sequence of actions (“you can press the Stop button, now Reset.”) for them to organize the fish so that they were returned to their initial state in the tank. Two of the activation features (Step and Clear) were neither referred to nor utilized by teachers or students. Thus, it is likely that no one perceived these features as potential supports for their problem solving.

5. Discussion

We take the approach that the features available in technology tools should give students and teachers opportunities to explore problems in a way that is not scripted or funneled, and where various features provide a support structure for active reflection on actions and results. By that we mean that the decisions students make about how to proceed with the solution process are not predetermined, yet there are features in the technology tool that provide feedback for student reflection. This is in contrast to software of the computer-aided instruction genre in which students’ problem solving processes are often structured with prompts and requested inputs. Aligned with our view, we believe a technology tool should add value to students’ problem solving by providing more control to the student. However, how one perceives the value added to a students’ problem solving is related to one’s epistemological and pedagogical beliefs about mathematics and problem solving (Schoenfeld, 1992).

In our analysis, we examined the ways in which features in the applet supported students’ problem solving. The students had many instances of using various features to pursue implementation goals, with 85% of those instances being supportive. Because implementation goals are action-oriented, one might expect students to make use of their technological resources in the applet to carry out these actions. The findings suggest that the features provided in the applet were appropriate for helping students implement a variety of heuristics. Students used the features less often in analysis, planning, and assessment, which are control-oriented activities. These control activities should engage students in employing their mathematical resources to consider different heuristics that are appropriate for the problem and monitor whether they are making progress toward a solution to the problem. About one-third of the control-oriented goals (analysis, planning, and assessment) involved explicit uses of features, mainly the static and dynamic feedback features, that were almost always supportive. During the remaining two-thirds of these control-oriented goals, students
did not make explicit reference to the technology; however, it is quite likely that students made use of features (e.g., viewing status ratio, comparing status and goal ratios) but did not specifically discuss or point to them.

There were a total of 16 occurrences of students following an implementation action goal with an assessment control goal. This sequencing of goals may be indicative of students who pause to reflect on their actions before considering future analysis, planning or implementation goals. As teachers, researchers, and designers think about how technology may encourage reflection on activities, encouraging students to assess what they have done with the technology may be an important goal to promote.

An analysis of the ways in which students made use of the technology tool provides insights that may be useful to technology designers about how features may support or not support students’ mathematical problem solving. For example, a design principle that is often suggested is that students should have access to multiple representations (e.g., Underwood et al., 2005) because these representations may enable students to focus on different aspects of a mathematical idea (DuFour-Janvier, Bednarz, & Belanger, 1987). The designers of the Fish Farm tool included the pie graph and ratio count in anticipation that students could use these representations in complimentary ways (Ainsworth, 1999) to reason about the part-whole or part-part relationships. However, we found that students did not choose to capitalize on the availability of these representations and use them in concert with each other. The representation that was used the least was the pie graph. It is possible that students were unfamiliar with pie graphs and therefore they chose to focus on the representations (e.g., goal and status ratios) with which they were most comfortable (Ainsworth, 1999). It is important to note, that simply making a representation (e.g., pie graph) available does not guarantee students will use it or that it is a meaningful representation for their mathematical understanding of the problem (Kaput, 1992).

It is possible that the absence of students’ self-initiated uses of multiple representations was influenced by the design of the tool. This applet had a single access point for manipulating representations. The only way students could interact with the applet was to directly manipulate the iconic fish. It might be that this design decision influenced the strategies students implemented, and the representations they employed as they worked toward problem solving goals. One might wonder how the ability to directly manipulate the status ratio or pie graphs and have the placement of the iconic fish update accordingly could affect students’ problem solving. However, the ability for students to manipulate the fish bi-directionally (from pond to tank and tank to pond) did seem to support their mathematical problem-solving because they could easily undo a move that they may have decided would not lead toward a solution to the problem.

In addition to the pie graph, other features that were not used by students included the Step and Clear activation buttons. It is possible that students did not use these features because they were unsure about what they did or because they did not think they would be helpful. If it was the former, then instructions about the function of each button should be provided to students and if it was the latter then it may be that these buttons could have been eliminated during the design of this particular technology tool. Because the tool was designed for solving a particular well-structured problem, non-essential features may only distract students from their work. There were instances when students’ ability to drag agents such as bricks and water within the applet distracted students from their problem solving goals. One might not anticipate that objects such as bricks and water, normally perceived as background objects, should be movable in an environment. The movements of these objects often resulted in an inaccurate count of fish displayed in the status ratio and pie graph, which in turn may have increased the cognitive demands placed on students. An additional feature that was entirely non-supportive in students’ work was the animation of the fish in the ponds. Although this feature may have provided a realistic appearance of how fish behave in water, it made it difficult for students to move fish out of the ponds and to count the fish as they moved. Designers need to carefully consider whether animation features and abilities to manipulate certain objects will contribute to and assist students’ mathematical thinking.

Our analysis focused on a small set of students working with a technology tool (e.g., java applet) designed to allow exploration of a particular problem. However, we feel the lessons learned from our research would also benefit designers of other types of open-ended software for mathematical problem solving. We do not believe that there is a single optimal set of features that will make an educational applet or software application “work” for every student. But our research results, combined with results from others, can provide insight into the nature and purpose of technology features that may lead to more successful problem solving.

We are encouraged about the possibility that our current process could prove to be a way to inform the design of software intended to promote problem solving, we are hopeful that other researchers will use our analytic techniques to analyze students’ problem solving with technology tools. Many researchers have called for the need for a more integrated approach to the design and research of educational software that includes early and systematic testing of software with the intended audience (e.g., Clements, 2000b; Clements & Battista, 2000; Roschelle & Jackiw, 2000).
Our fine-grained analytic techniques may be particularly useful when technology tool designers are field testing early versions of software with a small group of students. By focusing on how and when students use various features of a technology tool, designers can make decisions about the overall usefulness of the feature and whether other features may be more useful. For example, early field testing of students’ problem solving using the Fish Farm applet might have led designers to change certain features in the applet. Indeed, based on our research, we would suggest such minor changes as abandoning the animation feature of the fish swimming in the ponds, ensuring that background objects such as bricks and water are not moveable, and restating the problem in terms of red and yellow fish, rather than males and females. Such minor changes in the applet could reduce potential frustration and cognitive overload when dealing with the technological resources. These minor software changes may subsequently have major influences on promoting students’ use of their mathematical resources, use of various heuristics and control activities, and perhaps improvement in their overall problem solving processes.

References