IDEA: Identifying Design Principles in Educational Applets

☐ Jody S. Underwood
Christopher Hoadley
Hollylynne Stohl Lee
Karen Hollebrands
Chris DiGiano
K. Ann Renninger

The Internet is increasingly being used as a medium for educational software in the form of miniature applications (e.g., applets) to explore concepts in a domain. One such effort in mathematics education, the Educational Software Components of Tomorrow (ESCOT) project, created 42 miniature applications each consisting of a context, a set of questions, and one or more interactive applets to help students explore a mathematical concept. They were designed by experts in interface design, educational technology, and classroom teaching. However, some applications were more successful for fostering student problem-solving than others. This article describes the method used to mine a subset (25) of these applets for design principles that describe successful learner-centered design by drawing on such data as videos of students using the software and summaries of written student work. Twenty-one design principles were identified, falling into the categories of motivation, presentation, and support for problem solving. The main purpose of this article is to operationalize a method for post hoc extraction of design principles from an existing library of educational software, although readers may also find the design principles themselves to be useful.

☐ The Internet is increasingly becoming a vehicle to design, develop, and publish educational software in the form of miniature applications (e.g., applets) that can be used to explore specific problem contexts or concepts in a domain. The design of technology tools has the potential to dramatically influence how students interact with tools, and these interactions in turn may influence students' content area understanding and problem solving. However, the rapid development and dissemination of such tools in many cases occur without an explicit set of design principles in place. The purpose of this article is to describe the process that IDEA (Identifying Design principles in Educational Applets) participants used to cull design principles from a library of applets developed for mathematics education. We view the process of extracting these design principles as a first step in a broader methodology of proposing, refining, vetting, and using design principles.

Design principles published for educational software range in their specificity depending on the tools being analyzed. For example, Sinclaire (2003) studied student interactions with java-based dynamic geometry sketches that are accompanied by a set of questions to guide student exploration of the objects in the sketches. From her analysis of student work with the sketches, she has extracted a set of guiding principles that should inform future development (e.g., questions should aim to focus student attention on aspects of the sketch, whereas the

ETR&D, Vol. 53, No. 2, 2005, pp. 99–112 ISSN 1042–1629

sketch must provide the visual stimulus to draw attention through color, motion and markings).

One example of a general list of design principles is published in Clements's (2000) review of research of the use of computers in mathematical problem solving, in which he suggests several contributions that the use of computers can make to facilitate students' problem solving. As another general example, Schoenfeld (1985) contributed a framework of factors that affect student abilities to solve problems, which in turn could inform the design of educational software for problem solving. For example, as students are solving a problem, they need to implement strategies, use resources, and evaluate their progress so that they are aware of and critically examining their own decision making. In a technological environment, the resources available to students include their knowledge of concepts, facts, and procedures, as well as those offered by the technology. Students need knowledge of how to use the technology, and how the various objects and actions on those objects can aid their problem solving. The presence of a particular technology tool also affects the available strategies a student may use during problem solving, since a tool may afford or constrain certain actions, which make some strategies more accessible than others.

The National Science Foundation-funded Educational Software Components of Tomorrow (ESCOT, Roschelle et al., 1999) project produced a library of educational software. There were no preexisting design principles that ESCOT designers knowingly adopted. Instead, they drew on the expertise of people in a variety of complementary fields. Over the course of two school years (1999–2001), integrated design teams consisting of professional programmers, teachers, mathematics education researchers, and educational technologists, produced 42 problem contexts with supporting applets that were intended to facilitate mathematical problem solving for middle school students. These problems and applets were published and disseminated through the Math Forum's existing "Problem of the Week" structure (see http://www.mathforum.org /escotpow/). Students from around the world used these problems and applets, leaving a large database of students' problem-solving work.

The ESCOT project resulted in valuable lessons learned regarding problem context, questions, and applet design, as well as the interactions among these features. The design principles published prior to 1999 did not address these features and interactions. The IDEA project was undertaken as a follow-up project to ESCOT with the goal of identifying principles that could guide the design of effective problem-solving technologies. It focused on the analysis of the ESCOT products and the large database of student work with these products. Five members of the IDEA team were part of the ESCOT project, each with different areas of expertise—middle school teacher, software developer, educational technologist, mathematics educator, and project evaluator. The remaining IDEA team members, not having been part of ESCOT, brought less subjective views about the software we set out to evaluate, and had similarly broad areas of expertise—teacher, mathematics educator, and technology designer.

In a mature field such as architecture, a specific template for design is followed: State the recurring problem that needs to be solved, present the design pattern that addresses the problem, enumerate examples and varieties of solutions that meet the design pattern (Alexander et al., 1977). In order to create effective educational software, a similar template could be followed: (a) Define educational problem, (b) find recognizable solutions, (c) consider pedagogical implications of possible solutions (misconceptions, tradeoffs, intended effects, etc.), (d) identify principles that are appropriate for the educational setting and learning goals, (e) craft the features of the software that would enact those principles. However, in this early stage of educational design, until we identify the recognizable solutions, the rest of the process is somewhat halted. By identifying what worked and what did not work from the ESCOT experience, we are generating hypotheses about useful design principles that can be generalized beyond the specific context of interactive problems of the week.

A Brief Background of ESCOT

ESCOT tested and disseminated its educational software via the Math Forum, a large mathematics education resource portal. In their well-established Problems of the Week (PoWs; http://mathforum.org/pow/), students read the problem, work on a solution either individually or with a peer or group, and write an explanation of how to obtain a solution. Students submit their solutions with explanations online, receive feedback about their work via e-mail, and are encouraged to submit a revised solution if there are areas that can be improved. ESCOT created a number of PoWs (ESCOT PoW, or EPoW) that followed this same arrangement. Teachers used the EPoWs in many different ways: for example, as class requirements, as extra credit, in small groups, and also individually. An example EPoW can be found in a later section of this article.

Findings from Renninger et al.'s (2003) study of student work with the EPoWs suggests that these software-enhanced problems were motivating for students. They also found that the EPoW problem-solving environment appeared to override differences that would typically be found as a function of interest and self-efficacy with respect to student ability to connect to, generate strategies for, and be autonomous in problem solving. These differences may be owing to the design of the EPoWs. Thus, it was a natural extension of ESCOT to investigate the student data post hoc to hypothesize critical elements in the design of those EPoWs that seemed to support students' problem solving.

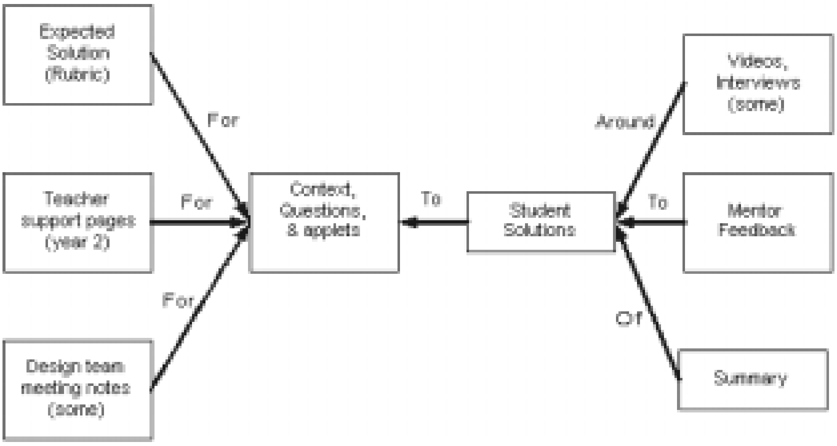
The data collected as a part of the ESCOT project include hundreds of student solutions to the EPoWs, summaries of feedback given to students, rubrics, teacher support pages, videotaped sessions of some students using the EPoWs, and documented design decisions made by the teams who created the EPoWs (Figure 1 shows the relations between these various pieces of data). These data informed the effort of the IDEA group.

We next present an example EPoW and then describe the mining method and the resulting design principles in detail.

AN EXAMPLE EPoW: FISH FARM

To situate our discussion, consider the Fish Farm EPoW (see Table 1 and Figure 2, available online at http://mathforum.org/escotpow/solutions/solution.ehtml?puzzle=40) in which students solve a ratio problem. The Fish Farm problem is flexible in the sense that there are multiple strategies that can be used to find three possible correct solutions. The applet was also designed to give students access to different representations for making sense of the problem. Although this problem can be solved using algebraic techniques, the intent of using the problem situation and tools in the Java applet was to engage stu-

Figure 1 ☐ Educational Software Components of Tomorrow (ESCOT) problem-of-the-week data.



dents in thinking about different strategies and solution paths, as well as part–part and part–whole reasoning and equivalent ratios. The bonus question was designed to induce a perturbation for students about the relation between a part–part and a part–whole representation of a ratio. The students are asked to compare the part–part ratio of 1:2 to a pie graph showing a part–whole ⅓:⅔ representation. Many students intuitively think about a 1:2 ratio as representing a one-half situation and do not easily make the transition to a ⅓:⅔ representation.

The applet was created with a tank on its left side with 26 fish (13 males, 13 females) that the user could drag and drop into one of the three ponds to its right. As a fish is dropped into a pond, a numerical count and pie graph are updated to keep tally of the number of females and males and the percent of females and males in each pond. Once a fish is dropped into a pond, it will swim within its boundaries. Although a user can move the fish without hitting the RUN button at the bottom of the screen, the RUN button is used to activate the applet so that the updates and swimming occur when a fish is dropped in a pond. The STOP button deactivates the update and swimming features. The CLEAR button will erase all fish from the tank on the left and the three ponds, whereas the RESET button will place all 26 fish back into the tank.

The complete materials associated with this EPoW include: (a) the problem situation and questions, (b) an interactive applet, (c) a teacher support page with suggestions for pre and post activities, and (d) expected solutions. The solutions were prepared by the ESCOT design team and used by mentors who provided feedback in response to student solutions. After all student work was submitted and scored for each EPoW, student solutions and mentor feedback and scores were archived along with comments from a lead mentor summarizing students' solution strategies and difficulties, and sample student responses.

Below is part of one 13-year-old girl's solution to the first question in the Fish Farm EPoW. Notice that the student described her strategy in terms of ratios, and used resources in the applet to help her reflect on her problem solving.

> Angel had 8 male fish and 8 female fish in her pond. Molly had 3 male fish and 1 female fish in her pond. Gar had 2 male fish and 4 female fish in her pond. I first put one male and one female in Angel's pond, 3 male and 1 female in Molly's pond, and 2 males and 4 females in Gar's pond. I thought I could put the rest into Angel's pond, but I noticed that there was unequal

Table 1 ☐ Fish Farm Educational Software Components of Tomorrow problem of the week (EPoW) text of the problem.

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.

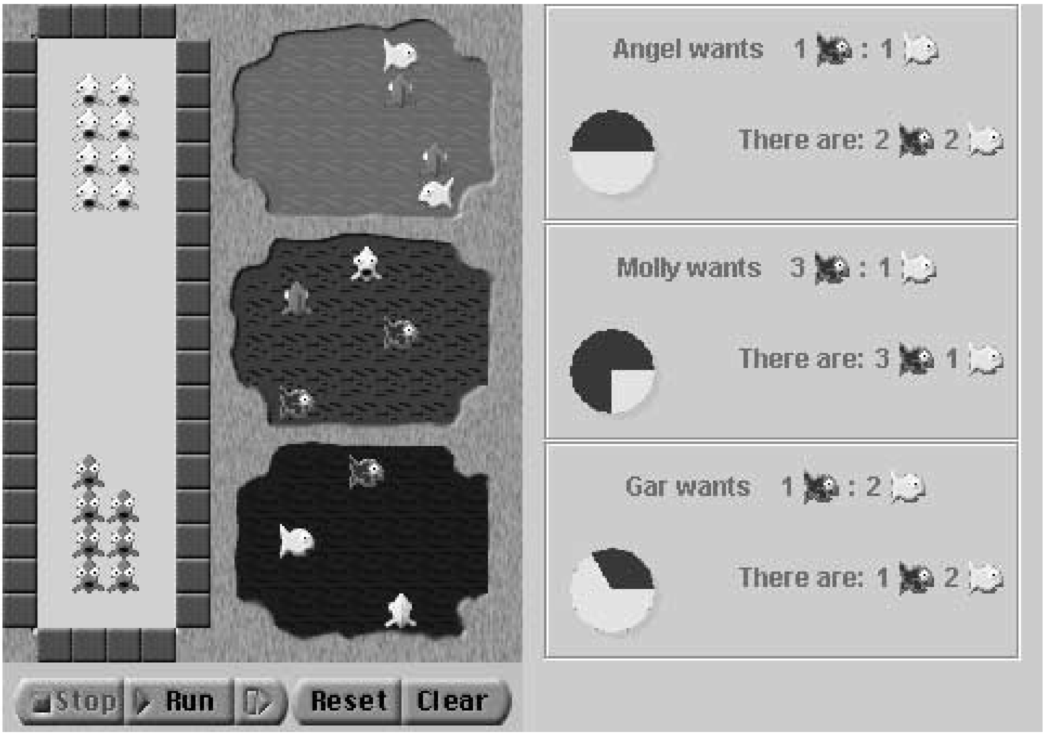
Is there a way to put all 26 fish into those three ponds, while giving each triplet what he or she wants? Use the applet to explore this question.

Questions

1. How many male fish and female fish does each triplet get in his or her pond? Describe the work you did to find the solution. (Sample questions you can answer: Into which pond did you put fish first? How many fish of each kind went into that pond? Why? What was your next step? How were you sure a pond had the correct ratio?)
2. Given the 13 males and 13 females, what are ALL the possible numbers of male and female fish that would satisfy the ratio of 1 male to 2 female fish in Gar's pond? Explain why these different amounts are equivalent to the ratio 1:2.

Bonus: Explain why all possible answers in question 2 result in the same pie graph for Gar's pond.

Figure 2 ☐ Fish Farm Educational Software Components of Tomorrow problem of the week (EPoW) applet.



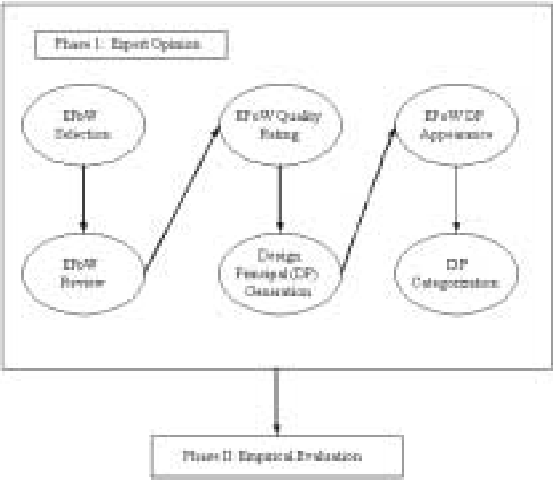
> amounts of males and females. So to make it equal, I put one more male fish and 2 more female fish in Gar's pond, that would still be the same as 1:2. That left me with the same amount of male fish and female fish, so they could all go into Angel's pond (that would still equal 1:1). Since I did everything slowly, I made sure that my amounts of fish were equal to the ratios. All I did was get the total amounts and then reduce them, and the reduced number should've equaled the ratio. I knew I got everything right when the bricks turned green.

From this student's description, it appeared that several of the design elements in the applet provided tools for her to complete the task. Specifically, the displayed ratios allowed her to compare a pond's male-to-female ratio with the desired ratio so she could check if one ratio reduced to the other. It appeared that having the bricks turn green provided closure and confirmation that her solution was correct. We considered these types of interaction and response as we mined the EPoW data to identify design principles and intended effects of those principles.

METHOD:
MINING DATA FOR DESIGN PRINCIPLES

The types of design principles that the IDEA team sought were those that successfully support problem solving and learner-centered design issues. Given the large amount of data and diverse backgrounds of the researchers on the IDEA team, we decided to use a methodology inspired by Erickson's (1986) analytic induction. We first explored the EPoWs and reviewed summaries of student work (not raw student data) to hypothesize design principles, some of which would finally emerge as a convergence of the opinions and theoretical underpinnings from our diverse research and experiential backgrounds (Phase I). Once these hypothesized

Figure 3 ☐ Design principle mining process.



design principles emerged, we combed through the student data to look for confirming or disconfirming evidence that the given design principle applied across EPoWs, including videos of students using the EPoWs to empirically validate our design principles (Phase II). The process we employed is depicted in Figure 3 and described below.

Phase I: Expert Opinion

Design principles were identified using a six-step process.

Step 1: EPoW selection. Of the 42 EPoWs that were developed for the ESCOT project, we selected a subset of 25 for the purpose of mining design principles. Two key points were considered in selecting an EPoW: (a) There were at least 20 submitted student solutions, or (b) the EPoW had been revised over the course of the two-year period, and student data collected for both versions.

Step 2: EPoW review. Taking a bottom-up approach, we, as experts from a variety of disciplines and perspectives, individually reviewed each selected EPoW and noted the characteristics we found strong and weak with respect to its mathematical purpose. The diversity of opinions helped us overcome preconceived notions about what makes good applets and activities. We also reviewed the design rationales from the ESCOT design teams and summaries of student solutions and strategies for each EPoW. These were the building blocks from which the design principles were generated.

Step 3: Overall quality ranking. We individually ranked the overall quality of the selected EPoWs. Individuals used their own, sometimes unnamed, criteria to select the five highest and five lowest ranked EPoWs. The ranking exercise helped make these differences of opinion and criteria selection explicit. The activity also initiated a valuable discussion on how individual applet features combined to give an overall impression. After a lengthy discussion, during which some researchers changed their rankings, we created an aggregated ranking as if to vote for the highest and lowest valued EPoWs.

Step 4: Design principle generation. As a whole group, we took the single highest ranked EPoW, chose one of the three that were tied for lowest rank, and generated design principles and associated intended effects based on the characteristics we had noted about them during our earlier analysis in Step 2. We then moved to small groups containing people with complementary expertise to continue this process for the remaining EPoWs. We then combined the lists of design principles and modified them until we came to consensus on definitions and behaviors. Finally, we reduced the list of design principles to those that were relevant to multiple EPoWs.

Step 5: Design principle appearance in the EPoWs. Small groups consisting of 2–3 people with complementary expertise took each of the design principles and a subset of the EPoWs (with some EPoWs common to all the groups) and noted if features in each EPoW suggested whether the principle was followed, violated, or irrelevant, while the definitions of the design principles continued to evolve through discussion in the larger group.

Step 6: Design principle categorization. Finally, we clustered the resulting principles into four categories: (a) ease of applet use, (b) motivation, (c) presentation, and (d) support for problem solving. We tabled any further refinement of principles in the ease-of-use category, because we realized our own nascent principles had already been well justified and articulated in the relatively large body of literature on general usability engineering (e.g., see Nielsen's 1994 key usability principles). Thus, we focused our efforts on the latter three categories, each of which is described below. The categorized list, along with intended effects on students' problem solving, is located in Appendix A.

Motivation (four design principles): These design principles promote motivation, including staying on task, showing excitement about the process, and so forth. They include such principles as "provide a familiar problem context" and "enable a reward for students early in the problem-solving process."

Presentation (seven design principles): The simplest way to think about these design principles is in terms of appropriateness for the intended learner audience. Some principles relate to the clarity of the problem context or question, or to the use of professional conventions. Some principles get at low-level interface issues when they are the *meaning* of the objects in the applet and not about the *use* of the applet. For example, some principles suggest that linked representations need to be obvious, or that attention should be drawn only to things that support the problem solving.

Support for problem solving (10 design principles): A plurality of the principles falls into this category. All these design principles are intended to facilitate problem solving, including principles such as allowing multiple solution paths, multiple entry points, appropriate feedback, and rewarding strategic thought. Because the general goal of EPoWs is to promote problem solving in a variety of mathematical contexts, these design principles are at the heart of helping characterize ways in which problem and applet design may enhance students' mathematical problem solving.

Phase II: Empirical Validation

Once design principles and intended effects were finalized, to the extent that they could be in our process, videos of students interacting with some of the EPoWs were examined. The videos available to us included the following:

1. Two high ability seventh-grade boys using Fish Farm 1.
2. Two low ability seventh-grade girls using Fish Farm 1.
3. Two low ability eighth-grade students (one boy and one girl) using Fish Farm 1 while working with a preservice mathematics teacher.
4. Two low ability seventh-grade girls using Scale n Pop.

The videos were collected for other studies (Renninger et al, 2003; Stohl, 2003, in press), and served the purpose of allowing the IDEA team to validate whether the design principles and the intended effects for students were observable from student actions with the applet and their social interactions as they solved the problem.

It is important to note the role of student video data in our design principle work. Because these data were not obtained from studies designed to validate design principles, the contribution of the data was to give examples, both pro and con, for specific principles. Our specific goal in examining the videos was to locate evidence to support whether or not a design principle was followed, and whether or not an intended effect occurred. Thinking about these options as a 2 × 2 matrix allowed us to see four possible outcomes: (a) followed with effect (FE), (b) followed no effect (FNE), (c) violated with effect (VE), and (d) violated no effect (VNE). The design principles are intricately related to the intended effects. For example, providing a history of actions could be useful or it could inhibit thinking. Viewing how students interact with the EPoWs gave an indication of the limitations and considerations of some principles.

These four cases (FE, FNE, VE, and VNE) were used to code video segments. Each researcher had a chart that listed each design principle, intended effect, and space for recording descriptions of segments from the video (including timestamps) that provided evidence supporting (FE and VNE) and evidence against (FNE) a design principle (see Table 2 for examples). For the fourth case, when a design principle was violated but there was evidence that an intended effect was achieved (VE), no conclusion could be made about whether the design principle caused or prevented the behavior. Thus, a segment coded as VE was inconclusive and not used to support or to refute a design principle and intended effect, but allowed other hypotheses to be made.

The videotape data were first reviewed in a whole group setting. Coding was done in successive 3-min segments. We paused to record relevant codes and notes after each segment and to discuss our observations. Following this segmented analysis of each tape, summaries were compiled as we compared the codes and evidence generated by the five members of the research team in order to evaluate the correspondence between assessed intended effects and evidence of these effects based on student activity (see examples in Table 2). For each design principle, the codes were shared, compared, and discussed until consensus was reached. Within each video analyzed, evidence was provided for almost all design principles. In addition, all four codes (FE, FNE, VE, VNE) were evident in the analysis of the video segments.

Table 2 ☐ Examples of evidence of design principles in several video sessions.

Category	*Design Principle*	*Effect*	*Intended Evidence*	*Video Session*
Motivation	Enable early reward for students.	Get involved in the problem that leads toward producing a solution.	FE (Min 5) They were happy when the balloon released. FE (Min 17–18) They were happy when the balloon enlarged for the improper fraction booth.	Scale n pop, two girls
Presentation	Links between representations should be obvious and warranted.	Less division of attention, understanding relationships.	FE (Min 11–13) One girl knew to use the sums instead of counting the fish. FNE (Min 11–13) The other girl didn't know to use the sums. VNE (Min 3–6) The girls expected the other representations to be updated when they did something.	Fish 1, two girls
Support for problem solving	Thoughtful strategic use of the tool should be rewarded more than random use.	Less try-and-trash, more thinking.	FE (Min 6–9) Student adds particular amounts of fish to ponds as they talked about maintaining the ratio. FE (Min 10-end) They got the pool to light up three times using various strategic approaches.	Fish 1, two boys

An Illustration of Validating Design Principles for Intended Effects

Four videotapes of eighth-grade students working in pairs with a preservice teacher on the Fish Farm EPoW (Stohl, 2003, in press) were also coded from the perspective of observing students' problem solving (Hollebrands & Stohl, 2004). The hypothesized design principles that support problem solving and their intended effects were used to gauge whether observed actions and effects aligned with or contradicted the hypothesized intended effects. The designers of the Fish Farm EPoW (ranked the fourth highest EPoW) seemed to follow many of the design principles related to problem solving (e.g., allows multiple entry points, supports multiple approaches and solution strategies, uses dynamic multiple representations); however, several design principles were not followed (e.g., history of actions, programming of applet supports level of accuracy necessary). To illustrate the four coding categories (FE, FNE, VE, VNE), consider the followed design principle of "uses dynamic multiple representations" and the violated design principle "history of actions."

Design principle, support for problem solving (PS)4: Use dynamic multiple representations appropriately. Many have suggested that multiple representations may enable students to focus on different aspects of a mathematical idea (DuFour-Janvier, Bednarz, & Belanger, 1987; Kaput, 1992). Fish Farm uses multiple representations to provide a visual display of male and female fish: ratio counts, pie graphs, and the fish themselves. The general intended effect for this design principle is to help students: (a) develop representational fluency, (b) facilitate better understanding of the problem, and (c) be engaged in mathematical thinking. The ratio counts and pie graph are intended to facilitate a better understanding of the problem and engage students in thinking about how to adjust their strategy for distributing fish. In addition, it is intended that the ratio counts can alleviate having students count fish in the ponds, and promote a transition between reasoning part–part and part–whole about the ratios.

Across these four videos, there were examples where students' observed actions and effects of these actions were aligned (FE) and not aligned (FNE) with the intended effect for multiple representations. One pair of students established the link between the representations and the number of fish in a pond early on, and subsequently did not have to count the number of fish in the pond (FE). With prompting from the preservice teacher, these same students also made connections between the ratio and the pie graph, and were able to connect the part–part ratio to the idea of a fraction in the pie graph (FE). Another pair of students mistakenly reversed the ratio of 1 male to 2 females for one pond and added 2 males and 1 female to this pond (see Figure 2) without apparently using the ratio count or pie graph to notice that the 3:3 was incorrect (FNE). However, because many students initially anticipated a ratio of 1:2 to result in a pie graph that is half red and half yellow, it is possible they used the pie graph according to that expectation. It also appears that they did not notice the difference in the pie graphs when they changed the state of the pond.

Design principle PS1: Supply a history of actions. Fish Farm did not include a history-of-actions feature to keep track of students' correct or incorrect solutions and actions. It was hypothesized that a history of actions would encourage student reflection and strategy tuning, and reduce duplication of incorrect solution strategies. For three of the student pairs, the absence of a history of actions seemed to hinder their problem solving (VNE). All three pairs were able to find one solution to the problem, but when they were asked to find a different second solution they had difficulty remembering their first solution. Recalling the first solution was necessary in order for students to determine if their second solution was different. For example, one pair of students quickly found their first solution to the problem. When challenged to generate a second solution, these students repeated what they had done the first time. It was the preservice teacher, rather than the students, who recalled that their current strategy would result in the same solution. Nonetheless, for another pair of students, the absence of a history of actions did not seem to impede their work at all (VE). This pair used paper to record their first solution and was able to find a second solution without relying on the preservice teacher to recall what they did the first time. These examples could provide evidence to support the argument that memory aids (e.g., peer, preservice teacher, paper and pencil, history of actions) can assist students in solving problems.

DISCUSSION

We have attempted to provide a clear and detailed picture of the means by which we convinced ourselves that we had uncovered useful design principles, so that others could use them and improve upon them without needing to reinvent a mining and validation process. We do not believe that there is one optimal set of design principles that will completely specify what must be done to make an educational applet work. However, we are excited about our current method as a way to link craft and theoretical knowledge in educational software design. Below, we discuss how the IDEA principles link to other design principles in mathematics education, and how our method links to models of design knowledge.

Design Principles for Learning Mathematics

Because these design principles were compiled using the perspectives of more than one discipline, it is useful to compare them to a set of published design principles from a single discipline: a mathematics education perspective of utilizing technology tools to enhance problem solving. Clements's (2000) review of research of the use of computers in mathematical problem solving provides a nice parallel to the IDEA design principles. He suggested several ways in which computers can facilitate students' problem solving. They can:

1. Provide an environment to test ideas and receive feedback.
2. Provide a mirror to students' mathematical thinking where their understanding of a problem situation can be made public.
3. Encourage autonomy for making and testing conjectures and engaging in playful exploration of mathematical ideas.
4. Link the general and the specific by allowing direct manipulation of objects.
5. Link the symbolic to the visual.
6. Catalyze natural and mathematical language to communicate within a software environment and about software-generated results to teacher and peers.
7. Encourage more positive social interaction.

Items 1–5 above are highly related to many of the design principles in the Support for Problem Solving category (e.g., allow multiple entry points, multiple approaches, and multiple solution strategies, linked representations, history of actions, provide feedback). The sixth item above could fall into the category of Presentation, for example, aligning the text and applet, using professional conventions, and making links between representations warranted. Because we are uncertain how the EPoWs are used, we do not suggest design principles regarding social interaction, in relation to the seventh item above. However, based on the analyses of videotapes of pairs of students using the EPoWs, the problem contexts and applets did promote individual motivation, positive social interaction, and joint problem solving.

Our work uncovered valuable design principles in addition to those proposed by Clements (2000), including our entire category of Motivation (e.g., provide a familiar problem context, use a second person voice, provide early rewards for successful problem-solving, provide high quality interactivity and graphics), as well as additional principles in the categories of Support for Problem Solving (e.g., reward thoughtful strategic use of the applet, require a level of accuracy necessary for the problem solving, make effort involved in an activity proportional to the importance of what is needed to solve the problem) and Presentation (e.g., verbal parts should be concise and clear, keep user's age in mind, links should be obvious, draw attention only to things that support the problem solving). Clearly, other issues in evaluating design principles need to be addressed, for example, examining tradeoffs between principles. However, as some mathematics educators have noted, these design principles hold promise for informing the development of mathematics education software as well as for evaluating the interplay between the presence of features informed by design principles and students' problem solving with technology tools (e.g., Hollebrands & Stohl, 2004).

Design Principles and Design Knowledge

We began the IDEA project with the goal of extracting valid design principles from the ESCOT experiences. As researchers in instructional systems design have noted, what *valid* means may vary widely depending on what the principles are *for*—are they for explanation and prediction of the success or failure of the learning tool, or are they supports for the work of expert designers (Reigeluth, 1999)? Should they be judged on their weaknesses (when they fail to work) or their strengths (when they are useful, regardless of whether they are comprehensive and true) (Snelbecker, 1999)? As the famous statistician George E. P. Box noted, "All models are wrong, but some are useful" (1979, p. 202). Our goal in undertaking this project was to focus on the utility of our design principles while insisting on the highest degree of truth we could adduce—specifically, the principles had to be consistent with all the data we had at hand.

Principles should be grounded in both personal expertise and theory. Schön (1992) described the notion of a reflective practitioner, a designer who entered an interactive dialogue with the designed artifacts and their setting. We linked to this notion by explicitly including the expertise of ESCOT team leaders in our IDEA team, utilizing their expertise as reflective designers. Our work also linked to more theoretically driven cognitive psychology theories about interfaces, mental models, and affordances from the psychology and human-computer interaction research communities (Card, Newell, & Moran, 1983; Norman, 1983, 1992; Norman & Draper, 1986). Thus, one might view design knowledge here as the psychology of how a particular artifact is used, rather than imperative knowledge about how to design artifacts. Indeed, Simon (1969) and Carroll and Rosson (1992) each pointed out that it is difficult to provide generalized psychological properties of artifacts, because so much of their properties are equally dependent on the context of use. Simon (1969) defined *design knowledge* as a "science of the artificial" and highlighted that design science, in contrast to natural science, involves models that are relative not only to that which is designed (the artifact itself) but also, fundamentally, to the artifact's relationship to the setting and to external goals.

Our principles are our best attempt to derive the most supported hypotheses from our data and they deserve further examination by others in other settings. We propose them not as universal rules to be followed slavishly, but instead as possible, not necessary, techniques to achieve a desired aim. This is similar to the notion of pattern languages, a metaphor for design knowledge from architecture that has taken hold in the computer science community (Alexander et al., 1977). A pattern language describes a design problem and associated solution that is frequently reused in many situations, although the way in which the solution is manifested may vary widely (e.g., a central town square solves a particular architectural problem, but town squares vary widely from location to location). Our principles, like design patterns, are intended to inspire solutions to problems rather than describe immutable psychological truths.

There is more work to be done to reconcile our proposed principles with other principles emerging from key sources, such as the Center for Innovative Learning Technologies (CILT) Design Principles database (http://wise.berkeley.edu/design/), and similar efforts in other disciplines. We hope that other researchers can apply our techniques post hoc to other educational software collections, such as the Utah State National Library of Virtual Manipulatives (http://matti.usu.edu/nlvm/nav/index.html), Shodor Foundation Project Interactivate (http://shodor.org/interactivate/index.html), and the Math Forum Math Tools (http://mathforum.org/mathtools/). Studying a broad cross section of collections will allow the field to identify and codify important design strategies that enhance the success of educational software.

The proposed design principles were derived from a library of problem contexts and supporting applets designed with specific mathematical ideas in mind. We are confident that these principles will be helpful for the design of other mathematics problem-solving environments intended to enhance one's ability to approach and solve a particular mathematics problem or explore a specific concept. In addition, these principles may be useful when adapted to other contexts and subject-matter domains where one wishes to have focused problem-solving contexts enhanced by an interactive software environment. Generalizing in this way will require continued effort by others with different software, users, and contexts, but we believe the design principles themselves can be a springboard for discussion and future research. ☐

The affiliations of the authors are: Jody S. Underwood [junderwood@ets.org], Educational Testing Service; Christopher Hoadley, College of Education & School of Information Sciences and Technology, Penn State University; Hollylynne Stohl Lee, Department of Mathematics Education, North Carolina State University; Karen Hollebrands, Department of Mathematics Education, North Carolina State University; Chris DiGiano, Center for Technology in Learning, SRI International; and K. Ann Renninger, Swarthmore College.

We would like to thank Annie Fetter, Suzanne Alejandre, and Kristina Lasher for their contributions to this project as members of the IDEA team, as well as Ian Underwood and three anonymous reviewers for their very helpful comments.

This project was funded by a seed grant from the Center for Innovative Learning Technologies funded by the National Science Foundation (REC # 9720384), and was partially supported by Educational Testing Service. Any opinions expressed in this publication are those of the authors and not necessarily of Educational Testing Service.

Correspondence regarding this article should be addressed to Jody Underwood, Educational Testing Service, Rosedale Road, MS 13-E, Princeton, NJ 08541.

REFERENCES

Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A pattern language: Towns, buildings, construction.* New York: Oxford University Press.

Box, G. E. P. (1979). Robustness in scientific model building. In R. L. Launer & G. N. Wilkinson (Eds.), *Robustness in statistics* (pp. 201–236). NY: Academic Press.

Card, S. K., Newell, A., & Moran, T. P. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Carroll, J. M., & Rosson, M. B. (1992). Getting around the task-artifact cycle: How to make claims and design by scenario. *Transactions on Information Systems, 10*(2), 181–212

Clements, D. H. (2000). From exercises and tasks to problems and projects: Unique contributions of computers to innovative mathematics education. *Journal of Mathematical Behavior 19,* 9–47.

Dufour-Janvier, B., Bednarz, N, & Belanger, M. (1987). Pedagogical considerations concerning the problem of representation. In: C. Janvier (Ed.), *Problems of representation in the teaching and learning of mathematics* (pp 109–122). Mahwah, NJ: Lawrence Erlbaum Associates.

Erickson, F. (1986). Qualitative methods. In M. C. Wittrock (Ed.), *Handbook of research on teaching* (3rd. ed., pp. 119–161). New York: Macmillan.

Hollebrands, K. F. & Stohl, H. (2004). The interplay between technology design and students' control of problem solving. In D. McDougall (Ed.), *Proceedings of the Twenty-Sixth Annual Meeting of the Psychology of Mathematics Education—North American Chapter* (vol. 3, 1481–1487). Toronto.

Kaput, J. (1992). Technology and mathematics education. In D. Grouws (Ed.), *A handbook of research on mathematics teaching and learning* (pp. 515–556). NY: MacMillan.

Nielsen, J. (1994). *Usability engineering.* San Francisco, CA: Morgan Kaufmann.

Norman, D. A. (1983). Some observations on mental models. In D. Gentner & A. L. Stevens (Eds.), *Mental models* (pp. 348). Hillsdale, NJ: Lawrence Erlbaum Associates.

Norman, D. A. (1992). Design principles for cognitive artifacts. *Research in Engineering Design, 4*(1), 43–50.

Norman, D. A., & Draper, S. W. (1986). *User centered system design.* Hillsdale, NJ: Lawrence Erlbaum Associates

Reigeluth, C. M. (1999). What is instructional-design theory and how is it changing? In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A new paradigm of instructional theory* (Vol. II, pp. 5–29). Mahwah, NJ: Lawrence Erlbaum Associates.

Renninger, K. A., Sinclair, N., Hand, V. M., Alejandre, S., Stohl, H., & Underwood, J. S. (2003). *Interest and motivation for learning mathematics with online interactive challenge problems.* Paper presented as part of the symposium, Interest and Motivation for Computer-Based Learning, European Association for Research on Learning and Instruction. Padova, Italy.

Roschelle, J., DiGiano, C., Koutlis, M., Repenning, A., Jackiw, N., & Suthers, D. (1999). Developing educational software components. *IEEE Computer, 32*(9), 50–58.

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

Schön, D. A. (1992). Designing as Reflective Conversation with the Materials of a Design Situation. *Research in Engineering Design, 3*(3), 131–148.

Simon, H. A. (1969). *The sciences of the artificial.* Cambridge, MA: MIT Press.

Sinclaire, M. P. (2003). Some implications of the results of a case study for the design of pre-constructed, dynamic geometry sketches and accompanying materials. *Educational Studies in Mathematics 52*(3), 289–317.

Snelbecker, G. E. (1999). Some thoughts about theories, perfection, and instruction. In C. M. Reigeluth (Ed.), *Instructional design theories and models: A new paradigm of instructional theory* (Vol. II, pp. 31–47). Mahwah, NJ: Lawrence Erlbaum Associates.

Stohl, H. (2003). Preparing to teach mathematics with technology: Research implications for a learning trajectory. *Society for Information Technology and Teacher Education International Conference 2003*(1), 2961–2968.

Stohl, H. (in press). Facilitating students' problem solving: Prospective teachers' learning trajectory in a technological context. *Journal of Mathematics Teacher Education.*

Appendix A ☐ The categorized design principles with intended effects.

Category	*Design Principle*	*Intended Effect*
Motivation (M)	1. Provide a familiar problem context.	Motivation
	2. Use second person voice.	Immersive, motivating, creates ownership for student.
	3. Enable a reward for students early in the problem solving process (e.g. provide easy questions or activities they can do successfully)	Get involved in the problem that leads toward producing a solution.
	4. Match user expectations for playability for videogame-like activities, interactivity, high-quality graphics, etc.	Get game players to take seriously and students continue with the problem.
Presentation (P)	1. Question, cover story, and/or introduction should be clear, unwordy, unsuperfluous.	Students get started quickly because they know what to do.
	2. Proofread text, labels, etc., with target users and age range in mind.	Reduce distractions or snag, increased focus on learning issues.
	3. All other things being equal, use professional conventions for content domain.	Familiarity, enculturation.
	4. Make the links between representations obvious and warranted.	Less division of attention, understanding relationships.
	5. Use high-quality graphics and other media (e.g., still graphics, audio, animation).	Better understanding of the problem.
	6. Draw attention only to things that support the problem solving.	More on task, more focus on important issues to help student solve the problem.
	7. Make everything described in the question obvious in the applet; align interactive and noninteractive parts.	Students oriented more quickly. The applet supports student solutions to the questions.
Support for problem solving (PS)	1. Supply a history of actions. wasteful duplication.	Can lead to reflection, strategy tuning, and not
	2. Allow multiple entry points (e.g., ability, experiences, preferences, styles).	Students might have many ways to get started, get involved.
	3. Support multiple approaches and multiple solution strate gies (e.g., questions and/or applet).	Students can use different strategies to solve the problem—more students should be able engage in mathematical thinking.
	4. Use dynamic multiple representations appropriately (linked or not-linked, multiple or single sources of control).	Develop representational fluency. Facilitate movement toward better understanding of the problem. More students should be able to engage in mathematical thinking.

Table continues.

Appendix A ☐ *Continued.*

Category	*Design Principle*	*Intended Effect*
Support for problem solving (PS)	5. Give students opportunities to make predictions, commit to them, and examine outcomes.	Students may revise their solution strategies. Way to make learnable moment.
(continued)	6. Reward thoughtful strategic use of the tool more than random use.	Less try-and-trash, more thinking.
	7. Make a pedagogical decision about whether closure is needed.	Sense of accomplishment.
	8. Applet should give appropriate status feedback (say the right thing at the right time in the right way).	Appropriate challenge but doesn't get too far off track.
	9. Require a level of accuracy necessary for the problem solving.	Reduces wasteful hairsplitting.
	10. Make effort involved in an activity proportional to the importance of what is needed to solve a problem (aside from programming for accuracy).	More likely to stick with the problem. Students attend primarily on relevant factors. Less busywork in the student's mind.