# Role of Interface Manipulation Style and Scaffolding on Cognition and Concept Learning in Learnware

KAMRAN SEDIG
The University of Western Ontario
and
MARIA KLAWE and MARVIN WESTROM
The University of British Columbia

This research investigates the role of interface manipulation style on reflective cognition and concept learning through a comparison of the effectiveness of three versions of a software application for learning two-dimensional transformation geometry. The three versions respectively utilize a Direct Object Manipulation (DOM) interface in which the user manipulates the visual representation of objects being transformed; a Direct Concept Manipulation (DCM) interface in which the user manipulates the visual representation of the transformation being applied to the object; and a Reflective Direct Concept Manipulation (RDCM) interface in which the DCM approach is extended with scaffolding. Empirical results of a study showed that grade-6 students using the RDCM version learned significantly more than those using the DCM version, who in turn learned significantly more than those using the DOM version. Students using the RDCM version had to process information consciously and think harder than those using the DCM and DOM versions. Despite the relative difficulty when using the RDCM interface style, all three groups expressed a similar (positive) level of liking for the software. This research suggests that some of the educational deficiencies of Direct Manipulation (DM) interfaces are not necessarily caused by their "directness," but by what they are directed at—in this case directness toward objects rather than embedded educational concepts being learned. This paper furthers our understanding of how the DM metaphor can be used in learning- and knowledge-centered software (i.e., learnware) by proposing a new DM metaphor (i.e., DCM), and the incorporation of scaffolding to enhance the DCM approach to promote reflective cognition and deep learning.

This research was part of the E-GEMS (Electronic Games for the Education of Math and Science) project funded by the Natural Sciences and Engineering Research Council of Canada, Electronic Arts Canada, and Apple Canada.
Authors' addresses: K. Sedig, Faculty of Information and Media Studies and Department of Computer Science, The University of Western Ontario, Middlesex College, Rm. 355, London N6A 5B7, Ontario, Canada; M. Klawe, Biological Sciences, The University of British Columbia, Dean of Science Rm. 1505, 6270 University Blvd., Vancouver V6T 1Z4, British Columbia, Canada; M. Westrom, Faculty of Education, The University of British Columbia, 2329 West Mall, Vancouver V6T 1Z4, British Columbia, Canada.

Categories and Subject Descriptors: G.4 [**Mathematics of Computing**]: Mathematical Software—*User interfaces*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*User-centered design*; *Interaction styles*; *Theory and methods*; K.3 [**Computing Milieux**]: Computers and Education

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Human-computer interaction, direct manipulation, reflection, education, learning, cognition, learnware, transformation geometry, problem solving

## 1. INTRODUCTION AND BACKGROUND

A great deal of interface design research has been devoted to determining mechanisms for making productivity tools (e.g., word processors and drawing tools) easy to use and intuitive so that users can perform a given task more smoothly and efficiently. In contrast, the aim of most educational software is not to optimize effort and speed up performance, but rather to engage the user in conscious attention to and reflection on the desired concepts being learned [Ormrod 1995]. The complexity involved in the design of interactive educational software is substantial. Not only should the user interface be easy to learn and work with, but the system must also engage the learner in conscious construction of knowledge. For these reasons, the suggested guidelines and objectives developed by HCI researchers for the design of user interfaces may not all be appropriate for the design of interactive educational environments [Rappin et al. 1997].

One of the prevalent modes of interaction with computer systems is through Direct Manipulation (DM) graphical interfaces. DM refers to systems which allow users to see graphical representations of objects and directly manipulate them on the computer screen with some kind of pointing device [Shneiderman 1988; 1993; Norman and Draper 1986]. Direct manipulation of graphical representations of objects is meant to allow users to replicate the behavior of the world literally. DM stands in contrast to command-based syntax for manipulating objects. As a technique for controlling computer systems, it is widely used in many application areas, including teaching/learning applications.

Norman and Draper [1986] and Norman [1991] present the notion of two gulfs (Gulf of Execution and Gulf of Evaluation) between the computer system and the user. The Gulf of Execution refers to "the difficulty of acting upon the environment," and the Gulf of Evaluation refers to "the difficulty of assessing the state of the environment" [Norman 1991, p. 23]. These gulfs can be bridged by bringing either the system closer to the user, or the user closer to the system. Hutchins et al. [1986 p. 95] state that "the feeling of directness is inversely proportional to the amount of cognitive effort it takes to manipulate and evaluate a system." Hutchins et al. [1986] outline different aspects of "directness." They state:

> The Gulf of Execution is bridged by making the commands and mechanisms of the system **match the thoughts and goals of the user as much as**

> **possible**. The Gulf of Evaluation is bridged by making the output displays present a good Conceptual Model of the system that is readily perceived, interpreted, and evaluated. **The goal in both cases is to minimize cognitive effort**....The more of the gulf spanned by the interface, the less distance need be bridged by the efforts of the user....Semantic distance in the Gulf of Execution reflects how much of the required structure is provided by the system and how much by the user. The more that the user must provide, the greater the distance to be bridged [Hutchins et al. 1986, pp. 94-101]. (bold added)

Another factor characterizing the directness of an interface is "direct engagement." Direct engagement refers to the feeling that results when the user is directly engaged with control of the interface objects. DM interfaces put users into direct contact with a world of objects, eliminating the need for communication through an intermediary, as in command-based interfaces. Because of this, users tend to find a DM interface more satisfying, whereas a command-based interface may be less motivating.

According to the above characterizations, to produce the feeling of directness in users, the challenge of DM interface designers is to design systems that maximize engagement and minimize distance—i.e., minimize the required cognitive effort. The results in the research literature, however, are not entirely clear as to whether the minimization of cognitive effort is only desirable for using a productivity tool, or whether this minimization is also desirable when designing instructional interfaces. Indeed, a number of HCI studies in problem solving and learning indicate that interfaces striving for ease of use and the lowest cognitive effort may not be effective for learning purposes [Svendsen 1991; Trudel and Payne 1995; Holst 1996; Golightly 1996; Rappin et al. 1997]. For instance, Svendsen reports that subjects who used a command-based interface learned more than those who used a DM interface. They made fewer errors, required fewer trials, and spent more time per trial.

We do not believe that DM is harmful, nor that we need to go back to using command-based interfaces to improve learning. However, to bring out the full potential of DM interfaces, as Hutchins et al. [1986, p. 118] suggest, the challenge of interface designers is to provide "new ways [to think of and interact with a domain] and creat[e] conditions that will make [users] feel direct and natural." This research is an attempt to conceptualize and test new ways of using the DM metaphor in concept-centered learnware, in particular, concepts which are encoded in visual structures.

One of the central issues in perception and learning is attention [Ormrod 1995]. It determines whether incoming sensory information is lost or processed. In educational software the main goal is to learn concepts in a particular domain of knowledge. One would ideally want to place users in direct engagement with the concepts in the domain (i.e., focus their attentions on interaction with visual representations of concepts) rather than the objects upon which the concepts act. Concepts can be represented visually. Visual representations of concepts encode knowledge in the visual structure [Card et al. 1999]. Unlike objects whose meaning is at the "surface" level, conceptual representations can embed knowledge at several levels, making these representations "highly abstract and with great interiority" of meaning (see Skemp [1979] and Norman [1991]). In this paper, we make a distinction between manipulation of objects and concepts. We call these Direct Object Manipulation (DOM) and Direct Concept Manipulation (DCM) respectively.

Once attention is directed toward a concept rather than objects, two other issues must be considered: cognitive processes and concept learning. Generally, cognitive processes can be differentiated in terms of whether they require conscious control or not—i.e., controlled versus automatic processing [Posner and Synder 1975]. Automatic processes are performed without conscious awareness, demand little or no effort, and are relatively fast. Norman [1993] refers to this as experiential cognition. Controlled processes, in contrast, take longer to perform, require conscious awareness and reflection, and involve making comparisons and decision making. Norman [1993] refers to this as reflective cognition. Whereas automatic, experiential cognition is valuable for performance-based tasks [Norman 1991], it is not suitable for knowledge acquisition. Deep, insightful learning takes place through reflective and effortful thought [Hayes and Broadbent 1988; Kolb 1984; Langer 1997; Norman 1993]. The more mental effort is exerted to "elaborate" a concept, the better it is learned [Salomon 1979; Ormrod 1995]. Inducement of "epistemic conflict" is one of the characteristics of the constructivist model of learning to promote reflective thought [Forman and Pufall 1988].

In terms of concept learning, Novak and Gowin [1984] prescribe that concepts have to be understood in ever greater degrees in a "progressively differentiated" manner—i.e., in terms of detail and specificity. Likewise Skemp [1986] argues for the need for a gradual transition from an intuitive stage of understanding concepts to a reflective stage. The process of supporting this transition is referred to as "scaffolding" [Brown et al. 1989; Jackson et al. 1998; Rogoff 1990; Perkins 1992] where the scaffolds are temporary supports on which users can rely during initial stages of learning a concept.

The research presented in this paper[1] addresses the following questions arising in the preceding discussion:

(1) Is a shift from DOM to DCM conducive to effective learning?

(2) Does DCM afford more reflective cognition and conscious processing of concepts?

(3) How can the interface support reflective cognition, and are there scaffolding strategies that would support and enhance the DCM metaphor?

---

[1]A preliminary report of this study was reported at the Premier European Conference on Human-Computer Interaction [Sedigand Westrom 1997] while the research was in progress. This paper provides a complete report of the study including a more coherent theoretical framework, a more detailed discussion and conclusions sections, and an in-depth analysis of the results.

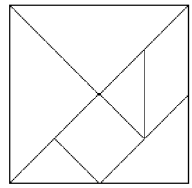

Fig. 1. Square from which tangrams pieces are created.

(4) In general, what are some characteristics of interfaces intended for concept learning?

Our research addresses these questions by comparing the effectiveness of three versions of an educational software prototype for learning two-dimensional transformation geometry. Each version implements the traditional tangrams puzzle-solving activity using a different interface style: DOM, DCM, and RDCM (the Reflective Direct Concept Manipulation interface which extends DCM with scaffolding strategies). These three versions are described in Section 2. Section 3 describes the methodology of an empirical study evaluating the different versions. The results of the study are presented in Section 4, followed by discussion (Section 5) and conclusions (Section 6).

## 2. DESCRIPTION OF THE THREE PROTOTYPES

A tangrams puzzle consists of a set of geometrically shaped pieces and a target outline, where the pieces must be arranged to form the given target outline. In the most common version, Chinese Tangrams (see [Read 1965]), there are seven pieces (two small triangles, a medium triangle, two large triangles, a square, and a parallelogram) that can create hundreds of possible outlines when arranged differently. These pieces can be cut from a single large square as shown in Figure 1. Figure 2 shows a possible target outline, and a solution to this puzzle is shown in Figure 3.

We have created an interactive, computer-based version of tangrams puzzles allowing users to solve the puzzles by manipulating on-screen images of the given geometric pieces. In each version of the software the user progresses through a linear sequence of puzzles. While users can replay any puzzle already completed, they cannot attempt a puzzle without completing all puzzles that precede it in the sequence.

Each puzzle is presented as a target outline with the geometric shapes (pieces) placed around it, as in Figure 4. The speckled area represents the outline. The black triangle, in contrast to the gray shapes, is the selected piece. Once a shape is selected, the user must specify a transformation operation (translation, rotation, or reflection) to perform on the shape. After selection of the operation, the parameters for the operator must be determined (e.g., how many degrees to rotate), either implicitly through direct manipulation of the selected shape, or explicitly through setting those parameters and then applying them to the selected shape.

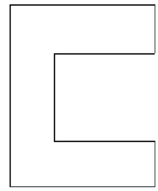

Fig. 2. Target outline of a tangrams puzzle.

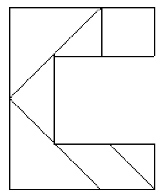

Fig. 3. Solution to a tangrams puzzle.

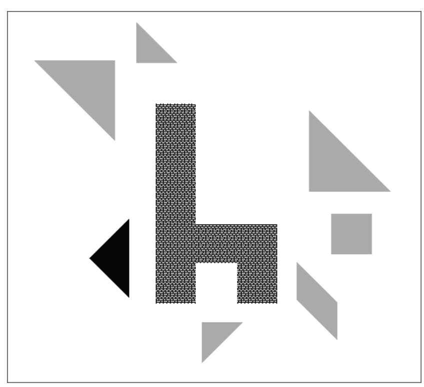

Fig. 4. Presentation of the tangrams activity on the screen.

## 2.1 The DOM Interface

In the DOM version, the user manipulates the geometric shapes directly. Buttons on the side allow users to select drag, clockwise rotate, counterclockwise rotate, horizontal flip, or vertical flip mode (see Figure 5). In drag mode, the user drags the shape to the desired location. In rotate mode, clicking on a shape causes the shape to turn 22.5 degrees (clockwise or counterclockwise). The resolution of the rotation, 22.5 degrees, was chosen because this is the largest setting that permits all possible solutions of the given puzzles. In flip mode, clicking on any shape causes it to flip over (horizontally or vertically). This manipulation style is both natural and intuitive, since the main goal of the activity is to move the pieces to the desired target locations. Indeed, there are a number of tangrams programs on the market which provide this type of interface manipulation style.

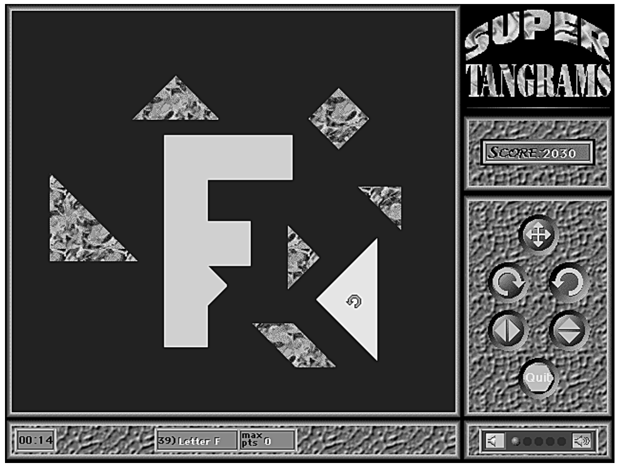


Fig. 5. DOM screen showing a counterclockwise rotation.

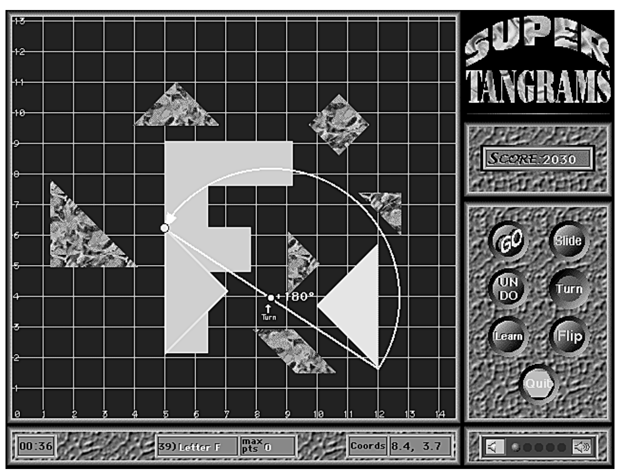


Fig. 6. DCM screen showing arc of rotation.

## 2.2 The DCM Interface

Computers can be used to help learners move from intuitive to explicit knowledge acquisition [Forman 1988]. A shift from DOM to DCM is intended to help learners move from an intuitive manipulation of concepts to an explicit one, thereby promoting formal knowledge construction. In the DCM version, the user directly manipulates explicit, visual representations of the transformation concepts rather than the shapes themselves. The user selects a piece to be transformed. The selected piece is indicated with a different color and pattern. The transformation then is selected by clicking on one of the Slide, Turn, or Flip buttons on the right. This causes the transformation representation to appear, as well as a "ghost" image of where the piece will move under the chosen transformation. Figure 6 shows an example with the rotation (turn) transformation. The user manipulates

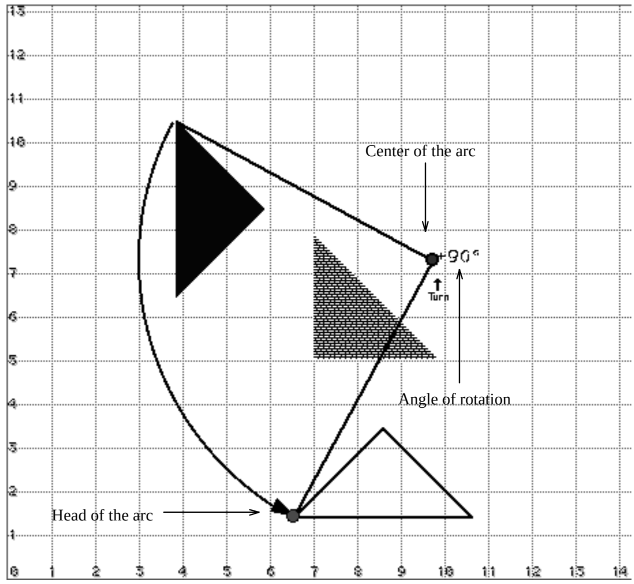


Fig. 7. Representation of rotation in DCM.

the transformation via handles (controls) on its visual representation, and when satisfied with the result, clicks the GO button. An animation of the selected shape moving into the ghost image spot occurs. A coarse grid is shown in the background, and the coordinates of the cursor location are shown at the bottom of the screen on the right. Since reversing the transformations is not as easy to do as in the DOM version, an UNDO button is also included in the DCM version. Furthermore, a Learn button provides access to a help module with explanations of the transformations and their representations, and opportunities to practice manipulating the transformation representations. The representations and interfaces for two of the transformations (rotation and reflection) are described below.

2.2.1 *Rotation*. Rotation is represented using an arc, where the center signifies the center of the rotation and the angle shows the direction and magnitude of the rotation (see Figure 7). The direction of a rotation is either clockwise or counterclockwise, and its magnitude is correspondingly shown in either negative or positive degrees. Changing the center or angle of rotation affects the destination of the rotation image. The black triangle is the image to be moved, and the gray, speckled triangle is the target area. The ghost image shows the destination that would result if the current rotation arc was applied.

The arc has two mouse-sensitive handles: the head of the arc and the center of the arc. The handle at the head of the arc allows users to set the angle of rotation, dragging it in either a clockwise or a counterclockwise direction. Changing the angle of rotation does not affect the position of the center of rotation, but changes the orientation and position of the ghost

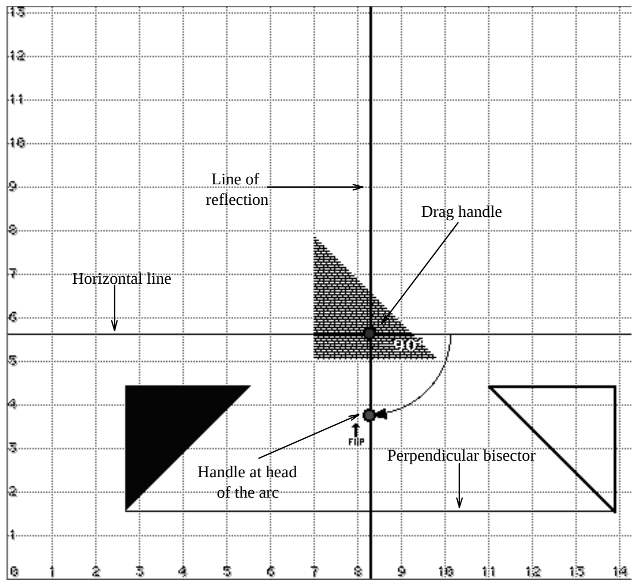


Fig. 8. Representation of reflection in DCM.

image. The handle at the center of the arc allows users to adjust the position of the center of rotation. Dragging the center of the arc does not affect the orientation of the ghost image, but changes the position of the ghost image and shrinks or enlarges the arc. The value next to the center of the arc shows the magnitude of the angle of rotation in either negative or positive numbers, depending on whether the direction of the rotation is clockwise or counterclockwise, respectively. Application of the rotation by clicking the GO button causes a smooth motion of the black piece along the perimeter of the arc into its ghost image, with the vertex of the piece that was connected to the tail of the arc ending up at the head of the arc.

2.2.2 *Reflection*. Reflection is represented by a line (the line of reflection) as illustrated in Figure 8. Changing the position or orientation of this line affects the destination of the reflected image. As in the previous cases, the black triangle is the image to be moved; the gray, speckled triangle is the target area; and the ghost image shows the destination that would result if the current reflection line was applied. The thick vertical line shows the reflection line, and the line connecting the lower vertex of the black image to its corresponding ghost image is the perpendicular bisector. The arc shows the angle of the reflection line with respect to the horizontal line.

The reflection line has two mouse-sensitive handles. The drag handle, which in this figure is seen in the target area, allows users to drag the line of reflection in any direction on the screen plane. Dragging this handle does not affect the orientation of the ghost image, but it changes the position of the ghost image moving it closer or farther from the black triangle. The

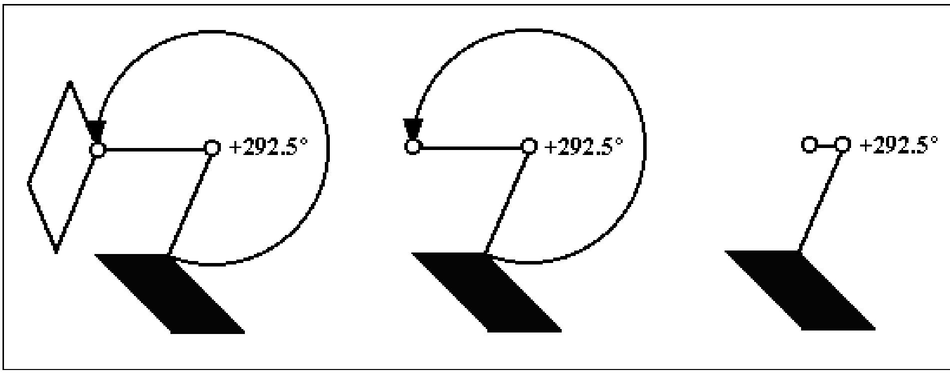


Fig. 9. Removing visual feedback in the representations of rotation.

handle at the head of the arc allows users to change the angle of the line of reflection by dragging the handle in either a clockwise or a counterclockwise direction. Changing the angle of reflection does not affect the position of the drag handle, but changes the orientation of the line of reflection, as well as the orientation and position of the ghost image. The numerical value next to the center of the arc shows the magnitude of the angle of the line of reflection with respect to the horizontal line in either negative or positive numbers, depending on whether the direction of turning the line of reflection is clockwise or counterclockwise, respectively. Application of the reflection by clicking the GO button causes a smooth motion of the black piece flipping over the line and landing on its ghost image.

### 2.3 The RDCM Interface

In the RDCM version, the user manipulates transformation representations as in DCM, but the interface incorporates a scaffolding process of gradual reduction of visual feedback in the transformation representations. That is, the scaffolding supports a gradual fading process (see Rogoff [1990] and Jackson et al. [1998]). This process is intended to serve two interrelated purposes that promote more insightful learning: (1) induce epistemic conflict to promote reflection, and (2) slowly shift the learner away from direct visual strategies (i.e., relying primarily on the ghost image) to strategies that require a more conscious processing of transformation geometry concepts.

In RDCM, the puzzles are grouped into three levels. In level one (puzzles 1 to 14) the representation interfaces are those used in DCM. In level two, (puzzles 15 to 24) the representations are the same except that there is no longer any ghost image. In the final level (puzzles 25 to 43), the representation is modified to disable visual alignment of the representation to match up a vertex of the piece to be transformed with the desired vertex of the target outline. Figure 9 demonstrates these three levels as applied to the visual representation of the concept of rotation.

In the leftmost representation (Level 1) the user encounters the notion of rotation without needing to worry about the structural subtleties and details of the concept. That is, they can adjust the angle of rotation, experientially observing the change in the position of the ghost image until it assumes the desired orientation; afterward, they can adjust the center of rotation until the ghost image reaches the desired target location. At this level, the interface is intended to help users develop a notion of the existence of the two subconcepts of angle and center of rotation that are involved in rotating a shape.[2] However, using this interface, users need not consciously know or reflect on what the angle of rotation should be, or how to determine the location of the center of rotation. Rotating a shape with the aid of the ghost image does not require the investment of much mental effort to "elaborate" and understand the meanings embedded in this complex and abstract mathematical concept at any depth.

In the middle representation (Level 2) the ghost image has disappeared. Since users can no longer rely on the visual feedback of the ghost image, in order to rotate a shape to a desired location they must reflectively and consciously visualize and/or determine the required angle of rotation. Therefore, they must pay attention to the numerical value of the angle of rotation—i.e., the number displayed near the center of rotation. Once this angle is determined and set, the proper setting of the center of rotation is not very difficult. Upon rotation, the shape's vertex, which is connected to the tail of the arc, is transformed to where the head of the arc is. Users can use this visual information to experientially drag the center of rotation until the head of the arc reaches the desired target position.

In the rightmost representation in Figure 9 (Level 3), not only has the ghost image disappeared, but the arc of rotation itself has changed. Now, users have to determine both the angle and center of rotation consciously. In this figure, two pieces of information still remain: a numerical feedback indicating the current angle of the arc, and a line connecting a vertex of the selected shape to the center of rotation. However, the position of the handle for adjusting the angle of rotation has been moved closer to the center so as not to easily reveal the position of the head of the arc of rotation. Determining where a selected shape is supposed to move, what the angle of rotation should be, and where the center of rotation should be placed are not trivial tasks; they require hard work in the form of reflective mathematical thinking and means-ends analysis reasoning. Rotating shapes to desired target positions is very difficult at this stage, and even people with a good knowledge of transformation geometry may find it mentally taxing and challenging.

2.3.1 *Hint and Snap-in-Place Features.* The reduction in visual feedback substantially increases the difficulty of completing the puzzles (in comparison to the DCM version). To allay this situation and free mental resources so that users can pay more attention to interaction with the concepts, RDCM has several added features. A Hint button is added which,

[2]It is assumed that the user does not know anything about the concept of rotation and its visual representation in mathematics.

when invoked, indicates where the selected shape fits in the outline. In addition, unlike the other two versions, RDCM has a Snap-in-Place feature. This feature allows the program to automatically compensate for slightly inaccurate placements of shapes. When a piece is moved sufficiently close to a correct location (i.e., a location that is part of a correct solution), the piece snaps into the location, displays a happy-face, and a click (snap) auditory feedback sound is provided.

## 3. METHODOLOGY

A multimethod (quantitative and qualitative) research design was used including a number of types of data-collection instruments and procedures.

### 3.1 Subjects

Forty-four grade-6 students (11- and 12-year olds; 21 females, 23 males) from two classes in an elementary school located in an upper-middle-class neighborhood of Vancouver, British Columbia, participated in the study. One of the classes was a mixed-grade class (grade-6/7), and all 15 grade-6 students participated. The other class was a straight grade-6 class, and all 29 students participated. None of these students had used the software before.

In addition a control group was included to determine if a repetition of the measures would influence students' knowledge of transformation geometry, and to gauge the stability of the transformation geometry test (described later in this section). The control group was a grade-6 class. The size of the control group was 20 students (8 females and 12 males).

### 3.2 Research Setting

The research was conducted in a temporary computer room set up in a small room adjacent to a classroom at the school. A partition wall separated the research room from the classroom. Eight Macintosh Performa computers were arranged on three long tables, with two or three computers on each table. All sessions of the study for all the groups as well as the interviews were conducted in this computer room.

### 3.3 Sources of Data

Five sources of data were used to evaluate the effectiveness of different treatments and design features: a transformation geometry achievement test, a design questionnaire, direct observations, videotaping, and individual interviews with subjects. The different sources were intended to help cross-validate each other.

3.3.1 *Transformation Geometry Test (TGT).* A paper-and-pencil transformation geometry test (TGT) was constructed. The purpose of TGT was to provide a comparative measure of students' overall understanding of transformation geometry, thereby permitting comparison of the relative effectiveness of the different design prototypes. The questions in TGT were designed and presented in a format that would be understandable to anyone learning transformation geometry, not just to the users of the different versions of the tangrams software. TGT's content and construct were validated and approved by two mathematics education experts. Furthermore, to assess TGT's reliability, a coefficient of stability was calculated by applying the Pearson product-moment correlation coefficient to the pretest and posttest scores of the control group. A high reliability coefficient was obtained with $r = 0.88$ (see Schumacher and McMillan [1993]). TGT contained 51 questions with varying degrees of difficulty, ranging from easy to very difficult (compared to the existing standardized tests).

3.3.2 *Design Questionnaire (DQ).* The Design Questionnaire (DQ) consisted of Likert scale and Semantic-Differential scale questions (see Schumacher and McMillan [1993]). The questions required a range of responses from factual (e.g., "Which level did you reach?") to perceptual (e.g., "I think that not having the ghost image made me frustrated and confused."). DQ was designed to collect both quantitative and qualitative data. The quantitative and qualitative data were intended to cross-validate each other. Many scaled questions terminated with an "explain why" prompt inviting open-ended written comments. This was the main source of qualitative data, which captured students' opinions and feelings and added more depth to their quantitative responses.

3.3.3 *Videotape Recordings of Interviews.* Videotape recordings were made of memory-recall interviews with 20% of the students from each group. The purpose of these interviews was to provide further information about responses on the test and questionnaire, and to explain the quantitative data.

3.3.4 *Direct Observations.* As students interacted with the different versions, they were videotaped and written notes were made of their overall patterns of use and verbal comments. Attention was paid to students' interaction with the different interface styles, how hard they had to work to solve the puzzles, what design features attracted their attention, and whether, at any stage during the study, students lost interest in the activity.

3.3.5 *Log Files.* Logs of students' interaction with the software were stored by each version of the software. The programs stored data log files of the number of moves made to solve each puzzle, the number of puzzles solved, the transformations selected, and the time taken to solve each puzzle.

3.3.6 *Design of the Quantitative Study.* A quasi experimental nonequivalent pretest-posttest group design [Schumacher and McMillan 1993] was used. Subjects were divided into three groups for the DOM, DCM, and RDCM treatments. The DCM and DOM groups were established by randomly dividing the grade-6 class, and the RDCM group was drawn from the grade-6 students of the grade-6/7 class. The DOM, DCM, and RDCM groups contained 15, 14, and 15 students each respectively. All subjects were given the TGT as pretest, one of three different treatments, and the TGT as posttest.

3.3.7 *Procedure*. Students were given consent forms describing the research project and the software. Students were told that they were not obliged to participate in the investigation, and they had the option to drop out any time they wished during the course of the study. All participating teachers (including the control group) agreed not to teach any topics related to geometry or transformation geometry concepts for a period of six weeks after the administration of the pretest. Students and teachers were told that different groups would use different prototypes of an educational software. To prevent possible biases toward the versions, neither teachers nor students were informed about how the prototypes differed.

A researcher visited each classroom and, in the presence of their teacher, asked students to write TGT, giving them 40 minutes to complete the task. Students were told that the results of the test would not affect their mathematics grade at school. Students were explicitly asked not to guess answers to questions. The participants were told that the purpose of the study was to assist researchers understand how to design better educational software.

Students were assigned to work on the computers in pairs. These groupings were made by the students' teachers. Each group worked on the software for 10 40-minute sessions, held on consecutive school days. Students were given a brief overview during the first session but thereafter were not given any adult help.

Immediately after the final session of their treatment, each group of students completed TGT and DQ. A few days later, clinical interviews were conducted with three students from each group to provide further information about their responses on the tests and questionnaires. The students were selected for their perceived thoughtfulness and ability to communicate. The interviews were 20 to 30 minutes long. Students were shown the other versions of the program, given an explanation, and asked to comment on the differences.

## 4. RESULTS

The results are divided into three sections: analysis of overall achievement results in TGT, affordance of reflective cognition, and attitude.

### 4.1 Analysis of Overall Achievement Results

Table I shows the measures of variability and central tendency for each of the different treatment groups as well as the control group. To determine if repeating TGT influenced students' knowledge of transformation geometry on the test, a paired samples $t$-test (two-tailed) was used to compare the pre- and posttest scores for the control group. No statistically significant differences were found ($t\ (19) = 0.62$; $p > 0.05$). Since the control group sample came from another school (i.e., another population), to avoid confounding

Table I. Measures of Central Tendency and Variability by Group for Pre- and Posttest Change Scores

| | | | Measures | | | |
|---|---|---|---|---|---|---|
| | | | Central Tendency | | Variability | |
| Group | N | Tests (Pre-Post Change) | Mean% | Median% | SD | Range |
| | | pretest | 25.0 | 25.2 | 15.0 | 0.1–59.8 |
| RDCM | 15 | posttest | 76.1 | 86.2 | 20.6 | 0.2–100.0 |
| | | (Δ) | 0.1 | 61.0 | 23.1 | 0.2–89.6 |
| | | pretest | 22.3 | 16.7 | 16.8 | 0.0–45.8 |
| DCM | 14 | posttest | 38.4 | 37.3 | 15.3 | 0.1–67.2 |
| | | (Δ) | 0.1 | 20.6 | 19.3 | 0.3–46.3 |
| | | pretest | 22.5 | 22.0 | 10.0 | 0.5–40.1 |
| DOM | 15 | posttest | 18.9 | 18.0 | 10.3 | 0.5–39.5 |
| | | (Δ) | 0.7 | −4.0 | 11.1 | 0.1–13.6 |
| | | pretest | 0.1 | 41.8 | 0.3 | 26.0–76.8 |
| Control | 20 | posttest | 0.0 | 39.3 | 0.9 | 23.7–80.8 |
| | | (Δ) | −1.1 | −2.6 | 0.0 | −14.2–10.2 |

effects the control group was not used in comparisons with the other conditions.

At a descriptive level, Figure 10 shows the pre- and posttest TGT scores for each of the three treatment groups. Table I lists the data points used to plot Figure 10.

Table I shows mean pretest scores were at about the same level for all the groups. However, the results (Table I and Figure 10) show large gains for the RDCM treatment group. The DCM group shows some improvement from pretests to posttests. The DOM group does not show improvement from pretests to posttests. In fact, DOM group's posttest mean decreased relative to their pretest mean.

The results in Table I indicate, that whereas the distribution of scores for the DOM and DCM groups on the posttest are relatively normal (i.e., mean median), the distribution of scores for the RDCM group is negatively skewed (i.e., mean $<$ median). Thus, in the RDCM group, a majority of the students showed superior knowledge rather than a few students improving dramatically.

At an inferential level, an analysis of covariance (ANCOVA) was performed using the pretest scores as the covariate. As displayed in Table II, there was a significant main effect among the groups ($F(2, 40) = 49.78$; $p < 0.025$). Tukey HSD post hoc tests were performed to determine where these differences occurred. There was a significant difference between the DOM and DCM groups ($q(2, 40) = 4.75$, $p < 0.01$), between the DCM and RDCM groups ($q(2, 40) = 8.75$, $p < 0.001$), and between the DOM and RDCM groups ($q(2, 40) = 13.48$, $p < 0.001$).

### 4.2 Affordance of Reflective Cognition

To assess the extent to which the three versions afforded reflective cognition, Norman's [1993] description of this psychological construct was used;

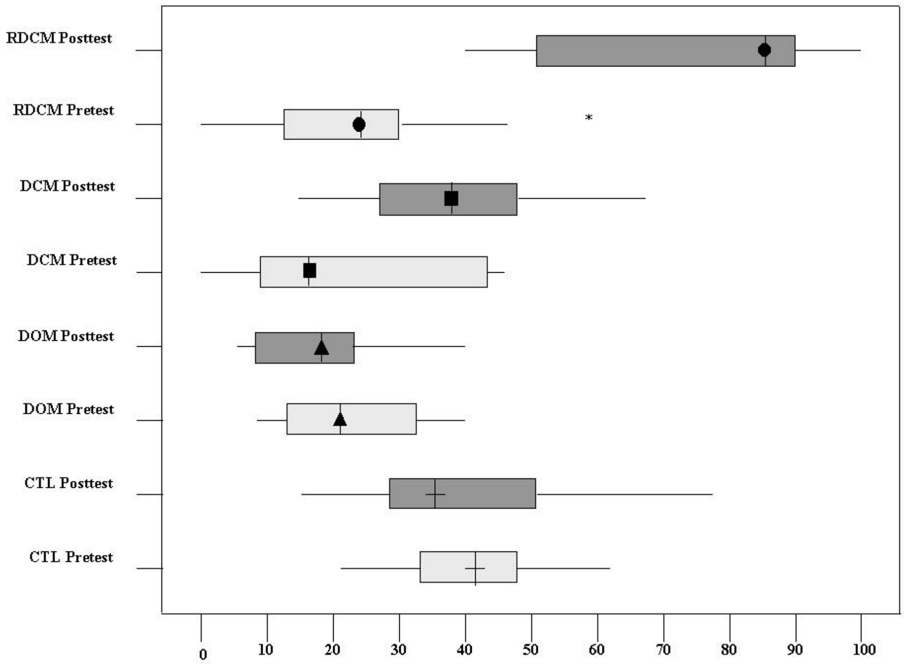


Fig. 10. Boxplots for pretest and posttest scores by group.

Table II. *F* Table Comparing Different Treatment Groups

| Source | SS | DF | MS | F | p |
|---|---|---|---|---|---|
| Covariates | 634.42 | 1 | 634.42 | 2.58 | 0.116 |
| Main Effects | 14482.49 | 2 | 12241.24 | 49.78 | 0.000 |
| Residual | 9835.65 | 40 | 245.89 | | |
| Total | 3576.64 | 43 | 831.64 | | |

namely, reflective cognition refers to conscious, purposeful thought that is directed at a problem in order to understand it and form integrated conceptual structures. Using responses (both quantitative judgments and qualitative comments) to questions in DQ as well as the program's log files, two indicators of reflective cognition are reported here: students' own perception of how hard they had to think to solve the puzzles; and, number of puzzles solved by each treatment group. On the questionnaire, students were asked if they had to think hard before moving a piece, and were also asked if it was easier and faster to guess. The number of puzzles solved by each treatment group was used as an indicator. Solving fewer puzzles was interpreted as a need for more time to reflect in order to solve the puzzles. The latter indicator was deemed appropriate, since during the entire length of the study, students were never observed to be off task. During all 10 sessions, students were always engaged in solving the puzzles.

Table III. Group Perception of Thinking versus Guessing and Average Number of Puzzles Completed

| Group | N | Thinking (%) | | | Guessing(%) | | | No. of Puzzles Solved |
|---|---|---|---|---|---|---|---|---|
| | | Agree | Undecided | Disagree | Agree | Undecided | Disagree | |
| RDCM | 15 | 93 | 6 | 0 | 6.5 | 6.5 | 87 | 31 |
| DCM | 14 | 57 | 21.5 | 21.5 | 21 | 43 | 36 | 68 |
| DOM | 15 | 53 | 27 | 20 | 27 | 40 | 33 | 78 |

Table III compares the three groups' perceptions of how hard they had to think to make each move. Students were asked two questions. These questions were intended to cross-validate each other in case the students did not understand them properly. The first question was phrased as: "Most of the time, I had to really think about how to move a piece. I couldn't just move a piece without much thinking." The second question was phrased as: "It was easier and faster to solve puzzles if I guessed, instead of calculated, each move." The numbers show the percentage of the students who agreed, disagreed, or were undecided with regard to the two questions. As can be seen, 93% of the students in the RDCM group thought that they could not move a piece without much thinking. Supporting this figure, 87% of these same students did not think that guessing was helpful in making their moves

Table III shows the average number of puzzles solved by each of the three groups, as recorded in the log files. The RDCM students solved fewer than half the number of puzzles solved by the DCM and DOM students.

### 4.3 Attitude

One of the questions in DQ asked the students how much they liked the software, compared to other educational games. Table IV shows the attitude of the different groups.

## 5. DISCUSSION

The results of this study showed that the RDCM interface style was more effective than the DCM style. Moreover, the DCM style was more effective than the DOM style. On the transformation geometry test, the mean score of the RDCM group markedly improved by 51.1%; the mean score of the DCM group improved by 16.1%; and the mean score of the DOM dropped slightly by 3.7%.

These findings were further cross-validated and clarified through the students' DQ responses, the post hoc interviews, and on-site observations. While observing the different groups and children's conversations, we noticed that the different interface styles had an instant and tangible influence on students' conception of transformation geometry. The DOM and DCM interfaces had different operational and perceptual affordances, two of which included: (1) how explicitly the interface represented the embedded conceptual knowledge, and (2) the immediacy of interaction. For

Table IV. Attitude of Students

| Group | Loved (%) | Liked (%) | So-so (%) | Disliked (%) | Hated (%) |
|---|---|---|---|---|---|
| RDCM | 40 | 53 | 7 | 0 | 0 |
| DCM | 43 | 43 | 14 | 0 | 0 |
| DOM | 40 | 60 | 0 | 0 | 0 |

instance, with respect to the first factor, based on their comments and strategies for solving the puzzles, students using the DCM style seemed to clearly understand that translation happens along a straight line. In contrast, some students who used the DOM interface style seemed to conceive of translation as a “drag” motion, moving curvilinearly in a two-dimensional plane—i.e., the DOM interface allowed students to drag an object in any direction they desired. This was one of the reasons that the DOM students did worse on their post tests—the game had constrained their mind to think of transformations in terms of the operations afforded by the interface. With respect to the second factor, the DCM style afforded a lag between the time students adjusted the parameters of a transformation and the time they clicked on the GO button. This lag time provided students with the opportunity to observe and discuss (i.e., become more conscious of) what was taking place on the screen. However, the DOM interface allowed students to focus on the task at hand, quickly moving the puzzle pieces. These differences, both in terms of interface affordances and student perceptions, were also observed with respect to the concepts of rotation and reflection.

Addition of gradual visual feedback reduction to the DCM interface style, resulting in the RDCM style, significantly increased learning. In their DQs, some students commented that not having the “ghost image” made the game “harder” and “challenging,” so they had to “make the image in [their] head,” and “learn where the shadows were”—remarks much in accordance with what Salomon [1992] refers to as “cognitive residue.” More than 80% of the students stated that they learned most in Levels 2 and 3, where parts of the transformation representations were gradually removed. Students stated that inclusion of the ghost image in Level 1, however, “helped [them] learn the basic concepts.”

When in the post hoc interviews the students were shown all the different prototypes. The DOM students found the RDCM version too complex and had difficulty even understanding this version. In addition, they felt that DCM was more difficult than DOM. The DCM students found RDCM more difficult, and DOM easier. In contrast, the RDCM students who were shown the DCM prototype immediately grasped how the prototype worked. They observed that the DCM activities seemed much easier than their own tasks and even had a somewhat condescending reaction to the DOM prototype. These students stated that DOM was very easy compared with RDCM. These findings suggest that the knowledge of students who used the more concept-focused interface styles encompassed the knowledge of students who used the less concept-focused interface styles.

In terms of reflection, the DOM and DCM students did not show a significant difference in the amount of reflection required to progress through the game. In contrast, the RDCM group had perceived their task to require hard thinking. RDCM students solved fewer than half of the number of puzzles solved by the other two groups. The extent to which RDCM afforded reflective cognition and required investment of mental effort was assessed using two indicators: students' own perception of how hard they had to think to solve the puzzles, and the number of puzzles that each group solved. The results of both indicators suggested that RDCM afforded a high degree of reflective cognition. Moreover, relative to DOM and DCM, the RDCM version demanded the investment of a higher degree of mental effort.

These findings were corroborated by direct observations and post hoc interviews. During the study it was observed that the DOM and DCM groups' activities were characterized by considerable moving of the puzzle shapes. The RDCM group's activities were characterized by more pausing, consultation with peers, reflecting, and thinking. As students moved to Levels 2 and 3, the progressive differentiation of the concepts, as promoted by the gradual removal of visual feedback, required students to engage in progressively greater degrees of reflection and reasoning. Students had to pay more attention to details and specifics of the concepts they were manipulating. In their interviews, some students reported that in Levels 2 and 3 they had to think hard to "picture the ghost image in [their] mind" to be able to perform the transformations.

All three groups were motivated to solve the puzzles. A difference between DCM and RDCM was the inclusion of the Hint and the Snap-in-Place features in RDCM. Tangram puzzles can be quite difficult to solve, requiring significant cognitive effort. These two features helped students focus more on doing the mathematical transformations rather than constantly being engaged in trying to figure out how to solve the puzzles themselves. Based on the log files, the RDCM students used the Hint feature extensively. Using the Hint feature in RDCM, the students often knew where a shape should be placed; the difficult task was to figure out the settings on the visual transformation representations to move the shape to the desired location. Considering the availability of the Hint feature in the RDCM version and the overall achievement results of the groups, it is safe to state that the relatively low number of puzzles that the RDCM students solved can not be contributed to their lack of ability to solve these puzzles, but rather the degree of mental effort that was required to figure out the transformations needed to solve the puzzles. In light of this, the tallies, reporting the three groups' perceptions of how much they had to think to move a piece, take on greater significance—for the RDCM students solving the puzzles was secondary to solving the interface transformations.

Finally, the change in interface style did not seem to have a measurable impact on the students' attitude and perception of the activity as a whole. Generally, one would expect, that since students want to get the task accomplished as fast as they can (i.e., solve the tangrams puzzles), RDCM's nonintuitive and cognitively demanding interaction would have a negative effect on students' liking of this version compared to the other two. However, their liking of all three prototypes seemed quite uniform across these styles. Indeed, the post hoc interviewees considered the need for the investment of mental effort, necessitated by the interface, as challenging, and therefore fun.

## 6. CONCLUSIONS

The research findings in this paper have implications for the design of effective human-computer interaction for learning and knowledge technologies. The conclusions are divided into three sections and are presented below.

### 6.1 Interface Style in Learnware

At a general level, this study demonstrates that interface style in learnware plays a crucial role in how learners interact with the educational content, and consequently how they acquire knowledge, and what knowledge they acquire.

This study confirms research in cognitive performance and distributed cognitions that tools can influence human cognition to think along a particular path [Salomon 1993]. HCI artifacts can extend or, inadvertently, limit human cognition and thought processes. While the RDCM and DCM interface styles aided students to think about and visualize abstract and formal transformation geometry concepts, the DOM interface style did not afford this. In fact, DOM seemed to support and amplify students' naive and intuitive understandings of transformation geometry concepts.

This study highlights the need for making a distinction between content-related versus non-content-related aspects of the user interface—e.g., in this study, interaction with the visual representations of the transformation geometry concepts versus interaction with the buttons, respectively. This poses a difficult trade-off in terms of usability and ease of use of the system, as discovered by Rappin et al. [1997] as well. In the tangrams activity, the educational activity is embedded in the task. Conventional interface design guideline would suggest minimization of cognitive load to make interaction with the task easier. This assumption seems to be well-suited for productivity tools in which the main goal is to produce an external product or outcome—e.g., use a word processor to easily and speedily create a nice document (see Malone [1981] for a discussion on the distinction between tools and toys). However, it may not extend completely to learnware artifacts in which the main goal may not be the speedy or easy creation of any external product (learning being an intrinsic outcome). In the design prototypes in this study, the DOM interface style supported the task much better than the other two prototypes did. Both in terms of its conceptual model and interaction, the DOM style was easier for students to understand and manipulate than the DCM and RDCM styles. Nonetheless, in terms of educational effectiveness, DCM and RDCM were superior to DOM.

In learnware, HCI designs should aim at reducing learners' cognitive load for performing non-content-related tasks so as to enable learners to allocate more cognitive resources to understand the educational content. When knowledge is implicitly embedded in the task (as is the case in the tangrams activity), interaction with the system should not necessarily be made easier. Indeed, ease of interaction may unintentionally communicate to the user the message that they need not invest much mental effort to consciously process the interaction and plan their actions with care (see Salomon [1979; 1981] for a discussion on the reciprocal-interactionist view of communication and learning).

### 6.2 Implications of Direct Concept Manipulation

Results from this study suggest that DM graphical interfaces should be used with care in the context of learnware. The DM construct can unintentionally be misapplied or misused. In applications that are concept-centered and involve comprehension and manipulation of conceptual abstractions, there is need for new models and techniques of interaction which are conducive to better learning while maintaining the benefits of DM—i.e., direct engagement with visual representations. In particular, since the DM interface engages the user's attention with objects on the screen, the abstract knowledge embedded in the objects must be explicit, manipulable, and engaging in order to fully support learning and knowledge sharing.

This study shows that in learning transformation geometry concepts, direct manipulation of objects (or shapes) was the main educational deficiency of the DOM prototype. In contrast, the DCM interface style helped learners explicitly and directly interact with formal interface representations of the embedded concepts and acquire more transformation geometry knowledge. These findings imply, that in a concept-centered environment which uses a DM interface style, manipulation should be directed toward concept representations rather than objects (Figure 11).

In the DCM construct, the two parameters of DM interfaces (directness and engagement) are still present, however, with some modifications in their conceptualization and characterization. In DCM interfaces, it is still desirable to produce a feeling of direct engagement for users. However, the user is in direct control of graphical representations of concepts rather than objects. In terms of spanning the gulfs of execution and evaluation, it is still desirable to keep the distance across the Gulf of Evaluation to a minimum so that learners can readily perceive and interpret the results of their actions. However, it does not seem desirable for the interface to fully span the Gulf of Execution for the learner. In a concept-centered learning environment, the DCM interface must place a semantic distance across the

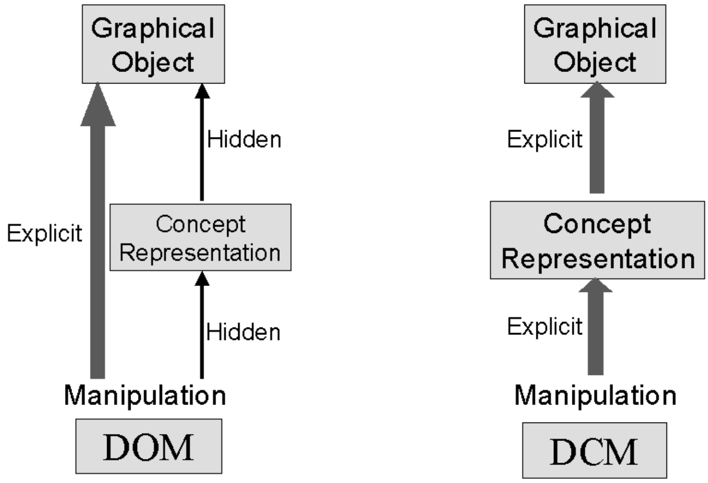


Fig. 11. DOM versus DCM.

Gulf of Execution for the learner to bridge. That is, the learner must exert effort to learn the conceptual semantics of the interface. Rather than the system conforming to and supporting how the learner already thinks, the learner must conform to the conceptual model of the system. Knowing that they are involved in a learning activity, learners already expect to encounter this distance. Consequently, the learning distance does not diminish the feeling of "directness" for the learner.

The way DCM interfaces are operationalized plays an important role in their instructional effectiveness. DCM designers must carefully examine and determine

(1) what visual representations to use to facilitate development of proper conceptual models;

(2) what elements of the representation of a concept should be allowed to be manipulated—i.e., what controls (or handles) to manipulate, how to embed these controls in the visual structure, and in what order, if necessary, they should be manipulated; and,

(3) what type of mouse interaction protocol to implement to direct learners' attention toward the essential aspects of a concept.

This study has explored an instance of what DCM interfaces may offer in helping learners interact with visual concepts. Other possible concepts that would lend themselves well to this type of manipulation include topics in physics (such as forces and momentum), abstractions that encode causal, functional, and semantic knowledge in a visual structure (such as scientific and information visualizations [Card et al. 1999]), and other mathematical concepts such as integrals. Further research is needed to gain a better understanding of how to design effective DCM interfaces.

### 6.3 Reflective Direct Concept Manipulation

As stated before, there are two poles of cognitive processing: controlled (reflective) and automatic (experiential). One of the problems with the DCM interface style is "habituation," a state in which the user becomes accustomed to a stimulus or interaction and gradually pays less and less attention to it, resulting gradually in automatic or experiential processing of interaction with the interface (see Sternberg [1999] for a discussion of habituation and dishabituation). However, since learning multilayered concepts requires deeper understanding of their details and specifics in a progressively differentiated manner, the DCM interface needs to be extended beyond mere interaction with explicit visual representations. In order to dishabituate the user, the interface can bring the user out of the habituated state by a slight change (see Sternberg [1999]).

RDCM is an extension of DCM. It uses gradual reduction of visual feedback as a dishabituation mechanism to accomplish several interrelated goals: to induce epistemic conflict, to cause controlled processing of the interaction, to promote reflective cognition, to scaffold and allow for the progressive differentiation of the visually represented concepts.

Once the DCM interface is in place, it must draw the learner's attention toward a deeper understanding of a concept by making the representation of the concept more abstract. A well-designed system must constantly and in a stepwise manner readjust the distance across the Gulf of Execution. However, it is important for the distance created in each step to be manageable for the learner to span. The reduction of feedback must be such that learners can use their current knowledge to discover and determine how to bridge the new Gulf of Execution at each step. Stated differently, as the visual scaffolds are removed, the learner should be presented with new representations communicating a sense of unfamiliarity and a need for a greater "amount of invested mental effort" (AIME) or "mental elaboration" (see Salomon [1979; 1981]). This process, however, should be performed in a careful manner so that by further elaboration the learner is able to discover the progressive differentiation of a concept in terms of its specifics and details. This process can be depicted as in Figure 12: the user starts interacting with an unfamiliar visual representation; initially the interaction is reflective; as the user becomes habituated, the interaction becomes experiential or automatic; the interface removes a visual component to dishabituate the user; the change makes interaction reflective again; and the cycle repeats itself.

An instance of interfaces that gradually reduce visual feedback can be visual-to-algebraic ones—i.e., interfaces that start by being visual and gradually become more abstract, eventually ending up as algebraic or command-based. An effective visual-to-algebraic interface style may well provide a timely solution for DM's current inappropriateness for some educational topics. In RDCM, the last stage of abstraction (Level 3) still included pictorial representations of transformation concepts. For instance, in the case of the rotation concept, after Level 3, there could exist a Level 4

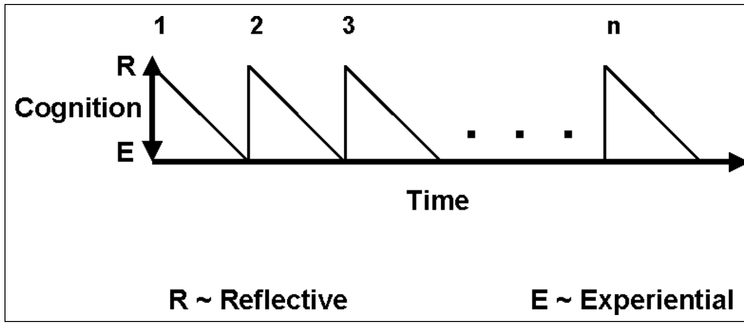


Fig. 12. Progressive alternation between cognition poles, Or attentive poles of habituation/dishabituation.

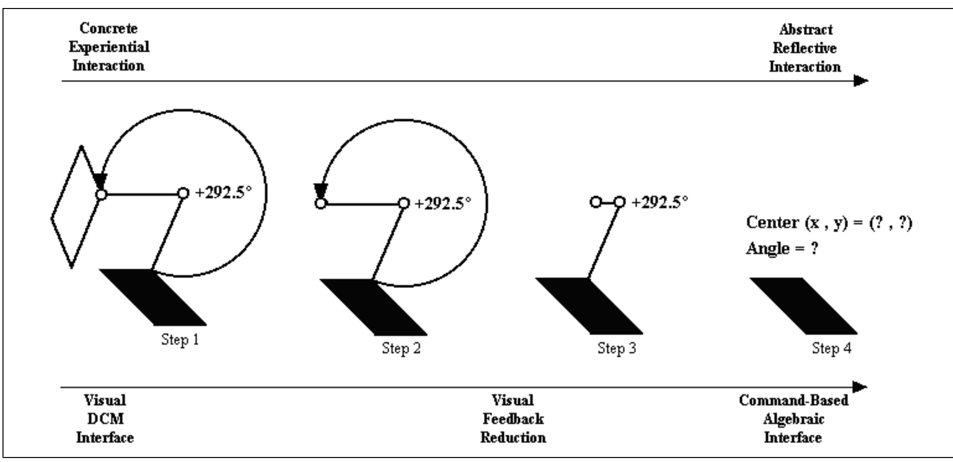


Fig. 13. Gradual conversion from a DCM to a command-based interface style.

in which the arc of rotation disappears altogether. In Level 4, learners would have to specify the parameters of the arc of rotation by typing in an (x, y) pair as the coordinates of the center of rotation and a number as the angle of rotation (see Figure 13). Based on the study presented here, it does not seem likely that learners would react to such an interface negatively, since they might view a command-based interface in Level 4 as part of the challenge of the game or learning. A visual DCM interface that gradually converts to an algebraic notation may satisfy the educational belief that algebraic knowledge should be built on the foundation of visual knowledge. Additionally, since feedback reduction from highly visual to algebraic abstraction is gradual, the command-based stage of the interface may not have a negative affective result on users.

## REFERENCES

Brown, J. S., Collins, A., and Duguid, P. 1989. Situated cognition and the culture of learning. *Educ. Res. 18*, 32–42.

Card, S., Mackinlay, J., and Shneiderman, B. 1999. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Forman, G. 1988. Making intuitive knowledge explicit through future technology. In *Constructivism in the Computer Age*, G. Forman and P. B. Pufall, Eds. Lawrence Erlbaum Associates Inc., Hillsdale, NJ.

Forman, G. and Pufall, P. B., Eds. 1988. *Constructivism in the Computer Age*. Lawrence Erlbaum Associates Inc., Hillsdale, NJ.

Golightly, D. 1996. Harnessing the interface for domain learning. In *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground* (CHI '96, Vancouver, British Columbia, Canada, Apr. 13–18), M. J. Tauber, Ed. ACM Press, New York, NY, 37–38.

Hayes, N. A. and Broadbent, D. E. 1988. Two models of learning of interactive tasks. *Cognition 22*, 249–275.

Holst, S. J. 1996. Directing learner attention with manipulation styles. In *Proceedings of the CHI '96 Conference Companion on Human Factors in Computing Systems: Common Ground* (CHI '96, Vancouver, British Columbia, Canada, Apr. 13–18), M. J. Tauber, Ed. ACM Press, New York, NY, 43–44.

Hutchins, E. L., Hollan, J. D., and Norman, D. A. 1987. Direct manipulation interfaces (excerpt). In *Human-Computer Interaction: A Multidisciplinary Approach*, W. A. S. Buxton. Morgan Kaufmann Publishers Inc., San Francisco, CA, 468–470.

Jackson, S. L., Krajcik, J., and Soloway, E. 1998. The design of guided learner-adaptable scaffolding in interactive learning environments. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '98, Los Angeles, CA, Apr. 18–23), C.-M. Karat, A. Lund, J. Coutaz, and J. Karat, Eds. ACM Press/Addison-Wesley Publ. Co., New York, NY, 187–194.

Kolb, D. A. 1984. *Experiential Learning: Experience as the Source of Learning and Development*. Prentice Hall Press, Upper Saddle River, NJ.

Langer, E. J. 1997. *The Power of Mindful Learning*. Addison-Wesley, Reading, MA.

Malone, T. W. 1981. Towards a theory of intrinsically motivating instruction. *Cogn. Sci. 5*, 333–370.

Norman, D. A. 1991. Cognitive artifacts. In *Designing Interaction: Psychology at the Human-Computer Interface*, J. M. Carroll, Ed. Cambridge Series on Human-Computer Interaction. Cambridge University Press, New York, NY, 17–38.

Norman, D. A. 1993. *Things That Make Us Smart: Defending Human Attributes in the Age of the Machine*. Addison-Wesley Longman Publ. Co., Inc., Reading, MA.

Norman, D. A. and Draper, S., Eds. 1986. User centered system design. In *New Perspectives in Human-Computer Interaction*, D. A. Norman and S. Draper, Eds. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Novak, J. D. and Gowin, D. B. 1984. *Learning How to Learn*. Cambridge University Press, New York, NY.

Ormrod, J. E. 1995. *Human Learning*. Prentice Hall Press, Upper Saddle River, NJ.

Perkins, D. N. 1992. What constructivism demands of the learner. In *Constructivism and the Technology of Instruction: A Conversation*, T. M. Duffy and D. H. Jonassen, Eds. Lawrence Erlbaum Associates Inc., Hillsdale, NJ.

Posner, M. I. and Snyder, C. R. R. 1975. *Attention and Cognitive Control: Information Processing and Cognition: The Loyola Symposium*. Lawrence Erlbaum Associates Inc., Hillsdale, NJ.

Rappin, N., Guzdial, M., Realff, M., and Ludovice, P. 1997. Balancing usability and learning in an interface. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '97, Atlanta, GA, Mar. 22–27), S. Pemberton, Ed. ACM Press, New York, NY, 479–486.

Read, R. C. 1965. *Tangrams-330 Puzzles*. Dover Publications, Inc., Mineola, NY.

Rogoff, B. 1990. *Apprenticeship in Thinking: Cognitive Development in Social Context*. Oxford University Press, NJ.

SALOMON, G. 1979. *Interaction of Media, Cognition, and Learning: An Exploration of How Symbolic Forms Cultivate Mental Skills and Affect Knowledge Acquisition*. Jossey-Bass Inc., Publishers, San Francisco, CA.

SALOMON, G. 1981. *Communication and Education: Social and Psychological Interactions*. Sage Publications, Inc., Thousand Oaks, CA.

SALOMON, G. 1992. Effects with and of computers and the study of computer-based learning environments. In *Computer-Based Learning Environments and Problem Solving*, E. D. Corte, M. C. Linn, H. Mandl, and L. Verschaffel, Eds. Springer-Verlag, Berlin, Germany.

SALOMON, G., ED. 1993. *Distributed Cognitions (Learning in Doing: Social, Cognitive, and Computational Perspectives)*. Cambridge University Press, New York, NY.

SCHUMACHER, S. AND MCMILLAN, J. H. 1993. *Research in Education: A Conceptual Introduction*. HarperCollins Publishers, New York, NY.

SHNEIDERMAN, B. 1988. We can design better user interfaces: A review of human-computer interaction styles. *Ergonomics 31*, 5, 699–710.

SHNEIDERMAN, B. 1993. Direct manipulation. In *Sparks of Innovation in Human-Computer Interaction*, B. Shneiderman. Ablex Publishing Corp., Norwood, NJ.

SEDIGHIAN, K. AND WESTROM, M. 1997. Direct object manipulaton vs. direct concept manipulation: Effect of interface style on reflection and domain learning. In *Proceedings of the European Conference on Human-Computer Interaction: People and Computers XII* (HCI '97, Bristol, UK). 337–358.

SKEMP, R. R. 1979. *Intelligence, Learning, and Action*. John Wiley and Sons Ltd., Chichester, UK.

SKEMP, R. R. 1986. *The Psychology of Learning Mathematics*. 2nd. Penguin Books, New York, NY.

STERNBERG, R. J. 1999. *Cognitive Psychology*. Harcourt Brace & Co., Orlando, FL.

SVENDSEN, G. B. 1991. The influence of interface style on problem solving. *Int. J. Man-Mach. Stud. 35*, 3 (Sept.), 379–397.

TRUDEL, C.-I. AND PAYNE, S. J. 1995. Reflection and goal management in exploratory learning. *Int. J. Hum.-Comput. Stud. 42*, 3 (Mar.), 307–339.