On Interface Closeness and Problem Solving

Thomas J. Donahue[1], G. Michael Poor[2], Martez E. Mott[3], Laura Marie Leventhal[4], Guy Zimmerman[4], Dale Klopfer[4]

[1]Tufts University
Medford, MA, 02155, USA

[2]Baylor University
Waco, TX, 76798, USA

[3]University of Washington
Seattle, WA, 98195, USA

[4]Bowling Green State University
Bowling Green, OH, 43403, USA

thomas.donahue@tufts.edu, michael_poor@baylor.edu, memott@uw.edu, {leventha,gzimmer,klopfer}@bgsu.edu

ABSTRACT

Prior research suggests that "closer" interface styles, such as touch and tangible, would yield poorer performance on problem solving tasks as a result of their more natural interaction style. However, virtually no empirical investigations have been conducted to test this assumption. In this paper we describe an empirical study, comparing three interfaces, varying in closeness (mouse, touchscreen, and tangible) on a novel abstract problem solving task. We found that the tangible interface was significantly slower than both the mouse and touch interfaces. However, the touch and tangible interfaces were significantly more efficient than the mouse interface in problem solving across a number of measures. Overall, we found that the touch interface condition offered the best combination of speed and efficiency; in general, the closer interfaces offer significant benefit over the traditional mouse interface on abstract problem solving.

Author Keywords

Interface closeness, Tangible interface, Problem solving

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous. K.3.m Computers and education: Miscellaneous

General Terms

Measurement, Experimentation, Human Factors

INTRODUCTION

Computing systems are increasingly integrated into everyday human situations. Their uses range from straightforward use as tools to uses that involve both tools and augmentation for learning. As such systems become more pervasive in our lives, the nature of the interaction plays an enormous role in shaping the experience. For example, systems in the classroom environment would require an interface that is both engaging and fosters learning.

TEI 2013, Feb 10-13, 2013, Barcelona, Spain.

Some (e.g. [9, 17]) have suggested that interface styles which are "close" should promote these characteristics in a learning environment. Interface closeness is defined by the degree to which an action performed via the interface differs from the action performed to achieve the desired result in the natural world [6]. In terms of Norman's Seven Stages of Action [14], the closer an interface, the smaller the size of the gulf of execution (i.e. the disparity between the intended actions and the affordances) and the gulf of evaluation (i.e. the difficultly in determining how our intentions and expectations match our interpretation of the system). Two interface styles which offer a closer interaction when compared to the traditional mouse interface are touchscreen and tangible. These styles are closer and have smaller gulfs of execution and evaluation, as the user doesn't have to map their intentions into the system as they would with a mouse. While some believe that closer interfaces should yield a better, more robust learning experience (e.g. [9]), emerging research has found conflicting results [2, 6, 8, 10]. One aspect of the learning process in many learning tasks is that of problem solving. Unlike traditional learning tasks, closer interfaces are thought to actually hinder problem solving performance, as their manipulation styles are thought to decrease planning and reflection [9]. This notion, however, has never been empirically tested, a recurring problem in HCI research [1].

We present an investigation into the effects of interface closeness on a novel abstract problem solving task. The contribution of this investigation is multifaceted. First, a novel problem solving task is introduced, which focuses on deductive reasoning and hypothesis generation. Second, this investigation will empirically test the validity of assumptions which suggest that closer interfaces, such as touch and tangible, are detrimental to problem solving. Finally, if closer interfaces do offer significant benefit, this investigation will compare and contrast the learning benefits of an interaction which is simply closer (i.e. mouse vs. touch) against those of a closer interaction which also has physicality (i.e. mouse/touch vs. tangible).

RELATED WORK

Problem solving, at its core, is the act or process of applying rules, techniques, or strategies in order to find a solution to a given problem. The application of such techniques or strategies, however, is context dependent. Similarly, which strategies are chosen and how they are applied to a given situation vary greatly across people, and even within a single person. Given the fluid nature of problem solving, it is important to understand how problem solving strategies can be influenced, particularly by the interaction style.

Svedsen [18] was one of the first empirical investigations into the influence of the interface on problem solving. The study was built upon the work of Hayes and Broadbent [5], which stated that there are two modes of learning, S-mode and U-mode. In S-mode, "learning takes place by means of abstract working memory and is selective and reportable." In contrast, U-Mode "learning occurs outside abstract working memory and is unselective and unavailable for verbal report." Svedsen, however, simplified S-mode to be synonymous with higher order "insight learning," and relegated U-mode to mere trial and error. Applying these ideas to HCI, [18] conducted two experiments in which participants were required to solve the Towers of Hanoi puzzle in one of two interfaces: command line or direct manipulation. [18] found across both experiments that command-line interaction induced S-mode learning, with participants being less error-prone and more efficient, while direct manipulation induced U-mode learning, engaging more trial and error. Perhaps unsurprisingly, across both experiments, participants overwhelmingly preferred direct manipulation interaction due to increased ease of use.

Though the results of Svedsen's experiments are compelling, the underlying reasons for the results are unclear. However, O'Hara and Payne [16] offer a strong explanation which focused on the underlying mechanisms that may lead to the significant differences between command-line and direct manipulation in problem solving tasks. Their analysis centered on the cost of performing an action, and how that affected "planfulness"—a term they introduce to describe the level of planning during problem solving. Along these same lines, they introduced the notion of implementation cost, which is "the cost associated with bringing about the effects of a particular operator in the world" and included factors such as time, as well as physical and mental effort.

Within the framework of implementation cost, O'Hare and Payne explained that the differences Svedsen found between the direct manipulation condition and the command-line condition were a result of differential levels of operator implementation cost. More specifically, the direct manipulation interface has a lower implementation cost than the command-line interaction, as it required less time and mental effort. They conclude that Svedsen's results should be expected as the higher implementation cost of the command-line interaction prompts the user to think harder, make fewer errors and thus learn more efficiently. Following in the vein of Svedsen's comparison between command-line (high implementation cost) and direct manipulation (low implementation cost), O'Hare and Payne carried out a similar comparison which utilized an 8-puzzle. The 8-puzzle consisted of eight numbered tiles arranged in a 3x3 matrix, with one cell left empty. The goal of the puzzle was to rearrange the tiles, utilizing the empty cell, until the goal configuration was reached. The results corroborated and validated their cost/benefit analysis, with the high cost group making significantly fewer moves to reach the goal configuration.

Further, Noyes and Garland [15] conducted a number of experiments which investigated the effect of presentation on solving the Tower of Hanoi puzzle. Participants were to solve the puzzle in three interaction conditions: mental, computer, and physical. In the mental condition, participants had no physical aids at their disposal, forcing them to mentally solve the task. In the computer condition, participants were afforded a direct manipulation interface, whereby they could 'drag and drop' the disks of the puzzles and see the effect of each action. Finally, in the physical condition, participants were given a paper model of the task which they could manipulate freely. Between the three conditions, participants in the mental condition were found to be significantly more efficient in completing the puzzle, though they also had significantly longer completion times, as well as a greater probability of failure. Participants in the computer condition had a significantly higher success rate, and shorter completion times, yet they generated more moves in order to solve the puzzle. Finally, participants in the physical condition had the highest rate of unsuccessful completions and, similar to their computer condition counterparts, generated significantly more moves to solve the puzzle and had shorter average time per move.

Summary

Together these studies suggest that in order for an interface or interaction to increase problem solving ability, it is imperative the interface foster increased reflection and planning. Adopting the verbiage of O'Hare and Payne, an interface which has a higher operator implementation cost will foster such problem solving characteristics. As a result, direct manipulation and physical interfaces, which have lower operator implementation cost, have yielded less efficient, though quick, problem solving. This would imply that such interfaces promote U-mode learning, or trial and error. In contrast, command-line interfaces and mental interactions, which have higher implementation costs, have yielded the most efficient, though slow problem solving. These results seem to suggest that TUIs, with their lower implementation cost, would not be beneficial in promoting efficient problem solving. Indeed, Marshall [9] argues that "it is possible that if tangible interfaces support easy manipulation of concrete objects, that they could in turn lead to decreased reflection, planning and learning." Such claims, however, require empirical investigation. Thus far, command-line and direct manipulation interfaces have garnered significant research, while closer interfaces, such as touch and TUIs, have been virtually ignored, despite previous research (e.g. [6, 3, 15]), which suggest closer interactions can foster more efficient and faster performance over traditional interfaces.

TASK

In order to investigate the effects of interface closeness on abstract problem solving, a task was required that fulfilled three main constraints, 1) the task should be engaging, 2) the task should lend itself to being implemented as both a GUI and a TUI, with little lost between each implementation, and 3) the task should be less familiar than the classic tasks used in in previous research, such as Towers of Hanoi or 8-puzzles. The board game Mastermind™, manufactured by Hasbro, a two-player code-breaking game that relies heavily on deductive reasoning as well as hypothesis generation and modification, worked as an ideal starting point. We modified the Mastermind™ game in order to eliminate any advantage to users who were familiar with the original.

Description

Though our task retained the discovery element and underlying feedback structure of Mastermind™ that fostered deductive reasoning and hypothesis generation, it is distinctly different. In our task, each trial had a contiguous arrangement of blocks that needed to be discovered. In order to discover this arrangement, participants added, removed and rearranged colored blocks on a 4x4 grid. The position of the arrangement of blocks on the grid was not important, only how the blocks were arranged relative to one another. Each trial's arrangement had 3 core features; dimension, number of blocks present, and number of colors present.

Dimension referred to how many dimensions the trial's arrangement of blocks existed in, one or two. If the arrangement was horizontal or vertical (i.e. it took place all in one row or all in one column), it was one-dimensional. If the arrangement were both horizontal and vertical (i.e. it took place in two or more rows and columns simultaneously), it was two-dimensional. The number of blocks present referred to how many blocks each trial's arrangement contained. This ranged from 2 to 4 blocks and 3 to 5 blocks, for one- and two-dimensional configurations, respectively. The number of colors present referred to how many colors each trial's correct arrangement contained. This ranged from 1 to 4 colors and 2 to 5 colors, for one- and two-dimensional configurations, respectively. Overall, there were 5 colors available: white, blue, red, yellow and green. A trial's arrangement can be thought of as a number of pairs put together in a specific way. Thus, for each arrangement, there are a set of pairs that it can be decomposed into. There were two types of pairs: horizontal and vertical. An example of a horizontal pair was a blue block to the left of a red block. However, the reverse of this pair, a red block to the left of a blue block, is a different pair. The same would be true for vertical pairs. It should be noted that two or more pairs can share a common piece (see Figure 1a for an example).

During each trial, the arrangement's dimension, number of blocks and number of colors were constantly available to the participant. Given only the core features of the arrangement, the participant began each trial by making an initial submission. The system analyzed each submission and gave two types of feedback to the participant. First, how many colors the participant's submission had in common with the trial's arrangement. The count is context-free, location and frequency of the color block on the grid had no bearing on the feedback given. Second, how many pairs the participant's submission had in common with the trial's arrangement. Like the color feedback, where the pairs were positioned on the grid was irrelevant. Furthermore, multiple instances of the same pair were also ignored by the system, unless the trial's arrangement also had multiple instances of that specific pair.

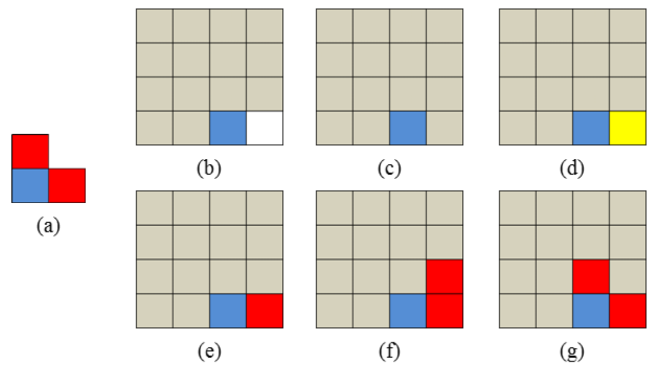

Figure 1. (a) Example trial arrangement from the experimental task (b - g) Example solution path taken by a participant

A complete example of a trial found in the experimental task, including the trial's arrangement and an actual solution path by a participant in the study is shown in Figure 1. Figure 1a shows the trial's arrangement to be discovered by the participant. The core information for this trial was: 2 Dimensions, 3 Blocks, 2 Colors. This arrangement was decomposed into two pairs: a red block on top of a blue block, and a blue block to the left of a red block. The initial submission is shown in Figure 1b. In response, the system reported: 1 correct colors and 0 correct pairs. This feedback was repeated for the following two submissions shown in Figure 1c and d. The submission shown in Figure 1e yielded the feedback: 2 correct colors and 1 correct pairs, as did the submission found in Figure 1f. Shown in Figure 3.1g, the participant had discovered the trial's arrangement and proceeded to the next trial.

Arrangement Creation

Each trial's arrangement was generated within a number of constraints. First, as the participant progressed, the complexity should gradually increase in order to minimize the risk of a participant getting stuck on a difficult trial early on. Complexity refers to how difficult a given trial's arrangement was to discover and was defined as the product of the number of three core features of every arrangement (i.e. Dimensions x Number of blocks x Number of colors). A second constraint was the board size. Given the grid was of size 4x4, one-dimensional arrangements were automatically capped at a maximum of 4 blocks. As each trial's arrangement was a set of pairs put together in a specific way, the minimum number of blocks possible for an arrangement to a given trial at 2—a single pair. With the block range for one-dimensional solutions being 2 to 4, the only other parameter left to change was the number of colors. For all arrangements with more than two blocks, the one color arrangement was discarded because they offered little opportunity for problem solving. Finally, two-dimensional arrangements were capped at five blocks as a result of pilot testing, which revealed that few participants pushed beyond the five block trials within one hour.

Benefits
The experimental task was designed to test problem solving within a broader context of leveraging aspects of discovery learning [19]. There were two phases of the experimental task: discovery and synthesis. During the discovery phase, participants sought to establish both the colors and pairs that composed the trial's arrangement. The feedback given to the participant built a foundation of knowledge about the trial's arrangement that was constantly updated. The synthesis phase was characterized by the process of combining and testing the information acquired in the discovery phase. Hypothesis generation and modification were most present during this phase. As the participant worked, the number of correct pairs rose and fell. This forced constant revision and expansion upon a current hypothesis or abandon the current hypothesis and devise a completely new one. It is important to note that these two phases were not discrete (i.e. aspects of each phase overlapped somewhat) nor linear (i.e. the participant moved back and forth between the phases).

IMPLEMENTATION
We implemented the experimental task in Java—selected for the TopCodes [7] library. TopCodes are barcode like symbols detectable by a standard webcam, and were chosen to use in the implementation of the tangible interface condition. The software presented the user with a 4x4 grid (in the middle), 5 colored blocks (on the right side), trial information (above the grid) and a submission button (below the grid) (see Figure 2). In the mouse and touch conditions, the colored blocks on the side were interacted with by clicking/touching and dragging them onto and around the 4x4 grid. In order to make each submission, the user clicked/touched the submit button. To ensure parity between all conditions, blocks in the mouse and touch conditions were treated as physical blocks. Thus, participants could only place a block in the second row if there was a block directly below it in the first row (and so on); participants could not remove a block from the grid which had a block directly above it without first removing the block above it; and a block could not be placed in an occupied position without first removing the occupying block.

All participants viewed the software on a GVISION P15BX 15-inch XGA resolution touchscreen monitor. Participants in the mouse condition utilized a standard two-button mouse to interact with the software's GUI. Participants in the touchscreen condition utilized the GVISION monitor's touchscreen capability to interact with the software's GUI.

Figure 2. Screenshot of Mouse/Touch condition software

Tangible Condition
In front of the user was a base created from LEGO™ blocks. The base was the physical representation of the 4x4 grid present in the software. To add and remove blocks from the grid, participants physically added and removed the colored blocks available to them. These blocks were also composed of LEGO™ blocks which allowed them to be stacked in a similar manner as the other conditions. The 4x4 grid in the software was constantly updated to reflect the blocks that were present on the base. In order for the software to register the presence and position of blocks on the base, each of the colored blocks had been augmented with TopCodes, which were detected using a common Logitech webcam which faced the back of the base. To make a submission, the participant placed the submission block in the 'Submit' position of the base and then removed it. The act of adding and removing the submission block was meant to replicate the submission button in the software (which was not present on the screen for the tangible condition, nor were the 5 colors on the side, which were also recreated in physical form, see Figure 3), without changing the mode of interaction for the participant. During the task, the user only needed to attend to the screen when they were reading the feedback for their latest submission, all other actions/information were carried out/presented with the physical blocks. There were a total of 6 TopCodes present in the system, one for each color and one for the submission block. The tangible condition, with an example set of blocks placed on the base is shown in Figure 3.

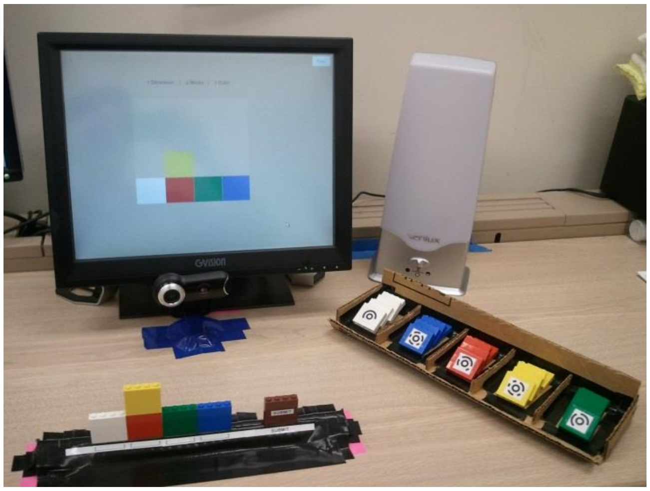

Figure 3. Setup of the tangible condition

METHOD

The primary goal of this study was to determine whether or not a closer interface (e.g. touch or tangible) lead to cognitive benefits over an interface that was less close (e.g. mouse). A secondary, more specific goal was to determine if the physicality of an interface, present in tangible UIs, would lead to cognitive benefits.

Hypotheses

There were several hypotheses: (H1) Participants in the tangible condition will have the lowest average number of submissions of the three interfaces, (H2) Participants in the tangible condition will have the longest average submission time, (H3) There will be no significant differences by condition in overall number of completed trials, (H4) There will be no significant differences between participants in the mouse and touch conditions across all performance measures.

Experimental Design

The study tested 3 conditions using a between-subjects design. The independent variable was the closeness of the UI. There were three levels of closeness: mouse, touch and tangible. Dependent variables included: number of trials completed overall, time taken per trial, time taken per submission within each trial, number of submissions made per trial, and number of blocks used per trial. All timing-related measures were recorded in millisecond time.

Participants

Forty-two college student participants (36 men, 6 women) with a mean age of 22.3 years old volunteered for this study. Participants were recruited from upper-level Computer Science classes. Extra credit was offered in return for their participation.

Procedure

Each participant completed a training session, which included thorough description of each aspect of the experiment as well as a set of training trials to be completed. During training, the experimenter was allowed to answer any questions that the participant may have had and offer help if needed. There were three training trials, chosen to represent each type of arrangement that would be found in the experimental trials (i.e. a 1-dimensional, 1 color arrangement; a 1-dimensional, multiple color arrangement; and a 2-dimensional, multiple color arrangement). Each participant was allowed to take all the time needed to complete the training trials.

The experimental trials directly followed. There were a total of 32 experimental trials, and participants had one-hour to complete as many of the trials as they could. During the experimental trials, the experimenter was only allowed to answer questions related to interacting with the system.

Participants were given the NASA Task Load Index (TLX) [13] questionnaire to report the workload placed upon them during the experiment.

RESULTS

The results are composed of 3 subsections: speed, efficiency and workload. Speed and efficiency both deal with objective performance measures of the experimental task, while workload covers all measures related to the self-report NASA TLX questionnaire. The analyses carried out on both types of measures included multiple analyses of variance (MANOVAs), and post-hoc pairwise comparison testing utilizing Tukey's-HSD.

For the analyses of speed and efficiency, all but one comparison was based on the data from the first 24 trials. Completion of all 32 trials was not guaranteed, and completion of 24 trials was established as the baseline for comparison. This resulted in 3 subjects (1 from each condition) being removed from the analyses, leaving 39 participants (13 in each condition). The comparison of the Number of Trials Completed used all 42 participants. For the analysis of workload, all 42 participants were present in the analysis.

Speed

There were two performance measures related to speed: Average Time Taken per Trial (measured in seconds) and the aforementioned Number of Trials Completed. There was a significant main effect of Average Time Taken per Trial, $F(2,36) = 5.07$, $p < .05$. For Average Time Taken per Trial, post-hoc comparisons revealed significant differences between the Tangible condition ($M = 107.02$, $SD = 21.91$) and both the Mouse condition ($M = 81.35$, $SD = 24.97$) and Touch condition ($M = 77.95$, $SD = 28.99$). For Number of Trials Completed, no significant differences were found between the Tangible ($M = 27.36$, $SD = 3.30$), Mouse ($M = 30.00$, $SD = 3.76$) and Touch conditions ($M = 29.86$, $SD = 3.94$).

Efficiency

There were 3 performance measures related to efficiency: Average Number of Blocks Used per Trial, Average Number of Submissions per Trial, and Average Time Taken per Submission (measured in seconds). There were significant main effects of Average Number of Blocks Used per Trial, $F(2,36) = 8.22$, $p < .01$; Average Number of Submissions per Trial, $F(2,36) = 5.45$, $p < .01$; and Average Time Taken per Submission, $F(2,36) = 19.38$, $p < .001$. For Average Number of Blocks Used per Trial, post-hoc comparisons revealed significant differences between the Tangible condition ($M = 34.96$, $SD = 6.73$) and the Mouse condition ($M = 62.13$, $SD = 26.14$), as well as between the Touch ($M = 43.91$, $SD = 13.45$) and Mouse conditions. For Average Number of Submissions per Trial, post-hoc comparisons revealed significant differences between the Tangible condition ($M = 13.12$, $SD = 1.52$) and the Mouse condition ($M = 18.76$, $SD = 6.99$), as well as between the Touch Condition ($M = 13.95$, $SD = 3.89$) and the Mouse condition. Finally, for the Average Time Taken per Submission, post-hoc comparisons revealed significant differences between the Tangible condition ($M = 8.15$, $SD = 1.27$) and both the Mouse condition ($M = 4.53$, $SD = 1.06$) and Touch condition ($M = 5.78$, $SD = 2.02$).

Workload

There was a significant main effect of the Frustration measure $F(2,39) = 3.78$, $p < .05$. A post-hoc comparison revealed a significant difference between the Tangible condition ($M = 7.93$, $SD = 1.35$) and the Touch condition ($M = 12.79$, $SD = 1.35$). Further, post-hoc comparisons revealed a significant difference between the Tangible condition ($M = 10.36$, $SD = 3.71$) and the Mouse condition ($M = 6.07$, $SD = 4.16$) for the measure of Temporal Demand.

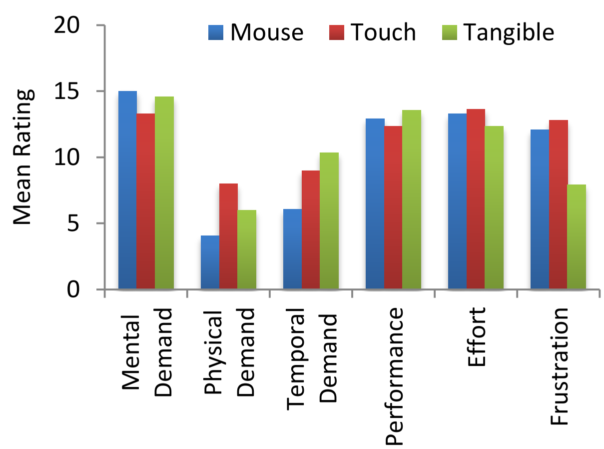

Figure 4. Summary of NASA TLX data

DISCUSSION

Reviewing each hypothesis, H1, was partially supported by the findings. Participants in the tangible condition averaged significantly fewer submissions than those in the mouse condition, and had the lowest number overall, but not significantly less than those in the touch condition. H2 was supported by the findings. Participants in the tangible condition averaged significantly longer time per submission than participants in both the mouse and touch conditions. H3 was also supported, as there were no significant differences in the overall number of trials completed by condition. Finally, H4 was not supported by the findings. On two performance measures, average number of blocks used per submission, and average submissions per trial, participants in the touch condition averaged significantly fewer than those in the mouse condition.

Speed and Efficiency

Previous research [16] into the intersection of interfaces and problem solving have shown that increased implementation cost led to increased reflection and efficiency in problem solving, though speed of problem solving decreased. The significant differences between conditions on the experimental task, however, are contradictory to the previous research. For example, participants in the tangible condition had significantly longer average time taken per trial than both the mouse and touch conditions. In the context of previous research, this result would only make sense if the tangible condition had a higher implementation cost. However, in terms of interface closeness, the tangible condition should have the lowest implementation cost, as it is the closest of the three interface conditions. Furthermore, in terms of problem solving efficiency, participants in both the tangible and touch conditions were significantly more efficient than those in the mouse condition. Once again, this appears contradictory to the result that increased implementation cost should lead to more reflective and efficient problem solving.

Though the underlying causes are unclear, there are a number of possible explanations for the results of the tangible condition. One possible explanation could be that implementation cost and interface closeness are not strongly linked (i.e. a closer interface does not necessarily have a lower implementation cost). If not, then perhaps the physical nature of the tangible interface led to increased planning and reflection. Additionally, it's possible the effect of the closeness of the interface was canceled out by the novelty of the tangible interface. Though all participants were given an in-depth explanation and completed a set of training trials in their interface condition, for many users, the experimental task may have been their first experience with a tangible interface. Therefore, as most users have spent many years with mouse and touch interfaces, but only a few minutes with the tangible interface, some performance deficits should be expected. Another explanation could be found in previous research [17], which showed that physical actions leading to digital effects led to increased reflection. Finally, it was observed that some users utilized the physical nature of the blocks to organize their thoughts by working with the blocks off the grid and without submitting (though it is possible that this is a symptom of increased reflection time, and not the cause).

The results of the touch condition are less clear. The touch condition was as fast as the mouse condition and nearly as efficient as the tangible condition. Like the tangible condition, the touch condition is closer than the traditional mouse interface. According to the predictions of the work into implementation cost, the touch condition should be fast, though inefficient. Unlike the tangible condition, however, the idea that the effect of closeness was offset by the novelty of the touch interface does not apply, as touchscreens have seen widespread use with the explosive growth of the smartphone and tablet computer markets. One possible explanation for the increased reflection and efficiency may be a product of the touch condition's increased concreteness [11] over that of the mouse condition. In contrast to the tangible condition, which has physical manipulatives, the touch condition has virtual manipulatives [12]. Previous research [4] into virtual manipulatives has shown them to speed up the transfer from working with the manipulative to abstract mental representation. It is possible, therefore, that the touch condition's more concrete nature allowed participants to move more fluidly between what they were manipulating on the grid, and the abstract problem solving they were doing simultaneously.

Finally, the results of the mouse condition are less surprising. Like the results of the tangible and touch conditions, those of the mouse condition appear to contradict the predicted results when viewed through the lens of implementation cost. However, it is important to remember the overall context in which the three interfaces were tested. While the mouse condition is least close of the three interfaces used in this study and thus the highest implementation cost of the three, to the user, the mouse is the default manner by which they have interacted with computers for decades. As a result, the relative increase in implementation cost in comparison to the other two interfaces may be cancelled out due to the user's familiarity. If this were indeed the case, previous research [18] suggests that the performance data should reflect a trial and error approach. A trial and error approach would be characterized by shorter submission times, higher number of blocks used per submission on average, and higher number of submissions per trial on average—all of which match up to the performance results of the mouse condition. Therefore, the relative increase in implementation cost in comparison to the other two interfaces was offset by the users' familiarity.

Workload

By definition, interfaces which are closer are more natural, and should therefore be less frustrating [6]. Tangible interfaces are inherently closer than touchscreen interactions. As a result, it is not surprising that participants in the tangible condition rated their frustration level significantly lower than those in the touch condition. This line of argument is supported by the difference in rating between the tangible and mouse conditions which, though not significant, was trending in that direction. It should be noted that the frustration ratings of participants in the mouse and touch conditions could have been influenced by an interface bug in which the system believed a block was present where there wasn't one. This system bug was present in the mouse and touch conditions only. These "ghost blocks," appeared relatively infrequently (appearing as a result of over-zealous clicking and tapping) and were easily identified and removed, but a small number of users reported their presence as frustrating.

The significant difference in the temporal demand ratings between the tangible and mouse conditions, with participants in the tangible condition averaging significantly higher ratings, was surprising. The TLX questionnaire frames the temporal rating with the following question, "How hurried or rushed was the pace of the task?" It was not predicted that there would be any differences in ratings, but most surprising was that the result appears to go against the performance results. By all objective measures, participants in the mouse condition moved through the experimental task significantly more quickly than their counterparts in the tangible condition. One possible explanation for this result could be an interaction between multiple factors. To review the experimental design, participants were given one hour to complete as many of the 32 experimental trials as they could. This time limit, however, was not made known to them, and in fact users were explicitly told to work at whatever pace felt comfortable for them. At the end of the hour, if the participant had not completed all 32 trials, the system would notify the user that their time was up, and then the experimenter would have them fill out the TLX questionnaire. Though there was not a significant difference found by condition in the overall number of trials completed, more participants in the mouse condition than in the tangible condition completed all 32 trials within the hour. Thus, with the resulting system message notifying them that their time was up still fresh in their mind, which perhaps led to a belief that they had not performed the task fast enough, it may have primed them to rate the task as more hurried and rushed, due to their perceived slow performance. It should be noted, however, that the participants had no idea how many trials were present in the experimental task, how long the time limit was, and were not given any indication of how well or poor they did. Therefore, any perceived notion of a "slow" or "poor" performance would have been of their creation.

FUTURE WORK

In the current study design, it was virtually impossible to understand and categorize the strategies used by participants during the experimental task. Like any problem solving task, there are any number of strategies available in order to discover each trial's arrangement. Further research should incorporate a think-aloud protocol into the design of the experiment. Incorporation of the think-aloud protocol would lead to a better understanding of how interface closeness affects participant strategy.

Furthermore, in the study's current incarnation, there was no way to test the learning of the participants across the experimental task. Previous research [18] has said that different interfaces promote different types of learning. For example, interfaces with high implementation cost promote higher order insight learning, while interfaces with low implementation cost promote mere trial and error. Though it was somewhat possible to categorize the three interface conditions into these categories (e.g. the mouse condition appears to have promoted trial and error), this categorization, however, does not necessarily mean that participants in one interface learned less than another. In order to test whether the amount learned about how best to solve the experimental task differs by interface condition, a sort of final test should be incorporated into future work.

CONCLUSION

This paper has presented an empirical investigation conducted to determine the effects of interface closeness on abstract problem solving. The study compared interfaces at three levels of closeness (mouse, touch and tangible) on a novel problem solving task which centered on deductive reasoning and hypothesis generation. We found that touch and tangible interfaces offered significant benefit over traditional mouse interfaces. Individually, the touch interface yielded the best combination of speed and efficiency. The tangible interface was slightly more efficient than the touch interface, though also the slowest. However, we suggest that the speed deficits of the tangible interface were partially a result of the novelty of the interaction. Finally, the mouse interface was as fast as the touch interface, though significantly less efficient than the other two. These results indicate that closer interfaces can foster reflection and planning, underlining their importance for interfaces designed specifically for problem solving tasks.

ACKNOWLEDGMENTS

Luke West, Samuel D. Jaffee, and Brianna J. Tomlinson

REFERENCES

1. Blackwell, A.F. 1996. Metacognitive Theories of Visual Programming: What do we think we are doing? In *Proceedings IEEE Symposium on Visual Languages*, 240-246.
2. Fails, J.A., Druin, A., Guha, M.L., Chipman, G., Simms, S. and Churaman, W. 2005. Child's play: a comparison of desktop and physical interactive environments. In *Proceedings of IDC 2005*, 48-55.
3. Fitzmaurice, G. W., Ishii, H., and Buxton, W. 1995. Bricks: Laying the Foundations for Graspable User Interfaces. In *Proceedings of the 1995 SIGCHI conference on Human Factors in Computing Systems*, ACM, 442-449.
4. Goldstone, R.L. and Son, J.Y. 2005. The transfer of scientific principles using concrete and idealized simulations. *The Journal of the Learning Sciences 14*, 69-110.
5. Hayes, N., Broadbent, D.E. 1988. Two modes of learning for interactive tasks. *Cognition 28*, 249-276.
6. Hippler, R., Klopfer, D., Leventhal, L.M., Poor, G.M., Klein, B., Jaffee, S. 2011. More than Speed?: An Empirical Study of Touchscreens and Body Awareness on an Object Manipulation Task. In *Proceedings of HCI International 2011*. Springer-Verlag.
7. Horn, M. TopCode: Tangible Object Placement Codes, http://users.eecs.northwestern.edu/~mhorn/topcodes/
8. Klahr, D., Triona, L.M. and Williams, C. 2007. Hands on what? The relative effectiveness of physical vs. virtual materials in an engineering design project by middle school children. *Journal of Research in Science Teaching 44*, 183-207.
9. Marshall, P. 2007. Do tangible interfaces enhance learning? *In Proceedings of the First International Conference on Tangible and Embedded Interaction*, ACM, 163-170.
10. Marshall, P., Cheng, P.C.-H., and Luckin, R. 2010. Tangibles in the balance: a discovery learning task with physical or graphical materials. In *Proceedings of the Fourth International Conference on Tangible, Embedded, and Embodied Interaction*, ACM, 153-160.
11. Metz, K. 1995. Reassessment of developmental constraints on children's science instruction. *Educational Researcher 65*, 93-127.
12. Moyer, P. S., Bolyard, J. J., and Spikell, M. A. 2002.What are virtual manipulatives? *Teaching Children Mathematics 8*, 372–377.
13. NASA TLX: Task Load Index. http://human-factors.arc.nasa.gov/groups/TLX/
14. Norman, D. 2002. *The Design of Everyday Things*. Basic Books, New York, NY.
15. Noyes, J. M. and Garland, K.J. 2003. Solving the Tower of Hanoi: does mode of presentation matter? *Computers in Human Behavior 19*.
16. O'Hara, K.P. and Payne, S.J. 1998. The effects of operator implementation cost on planfulness of problem solving and learning. *Cognitive Psychology 35*, 34-70.
17. Rogers, Y., Scaife, M., Gabrielli, S., Smith, H. and Harris, E. 2002. A Conceptual Framework for Mixed Reality Environments: Designing Novel Learning Activities for Young Children. *Presence: Teleoperators & Virtual Environments 11*. 677-686.
18. Svendsen, G.B. 1991. The influence of interface style on problem solving. *International Journal of Man-Machine Studies 35*. 379-397.
19. van Joolingen, W. 1999. Cognitive Tools for Discovery Learning. *International Journal of Artifcial Intelligence and Education 10*, 385-397.