

# SCWID: A Tool for Supporting Creative Work In Design

*Damon J. Cook, Heather E. Metcalf, Brian P. Bailey*

University of Illinois at Urbana-Champaign

201 N. Goodwin

Urbana, IL 61801

{djcook, hmetcalf, bpbailey}@uiuc.edu

## ABSTRACT

Computer tools offer enormous benefits for the early design process such as remote collaboration, advanced visualization, and the ability to run a design. However, current tools fail to support many elements of creative problem solving, inhibiting the early design process. From the literature on design theory and creativity, and extensive low-fidelity prototyping, we developed SCWID: a tool for Supporting Creative Work In Design. SCWID uses a large display to provide a shared visual context for alternative design ideas and uses multiple local displays for sketching details, navigating a particular idea, and manipulating alternatives. Grounded in creativity theory, the use of our tool facilitates creative thinking in the early stages of design for individual and groups of designers. Our tool can run either on a stand-alone machine or as part of a distributed workspace formed by connecting multiple clients to a server running on a machine that drives any large display.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

**General terms:** Design, Human Factors

**Keywords:** Multiple-Display Environments, CSCW, Multiple-Device Interfaces, Creativity

## INTRODUCTION

Computer tools can provide many benefits for the early design process such as remote collaboration [11], access to design rationale [10], and execution of a design [2]. However, if a tool does not adequately support the process of creative thinking in the early stages of design, it can disrupt that process and inadvertently lower the quality of a designer's final product or raise the cost of creating it [20].

The creative problem solving process consists of a continuous cycle of idea generation, evaluation, and refinement [7, 8]. While this process can be done individually or in groups, the most effective strategy is to alternate between the two [7, 21]. On top of this general process, there have been many advances in the understanding of aids to creativ-

ity including sketching [18, 19], storyboarding [15], and visualization of multiple design ideas [4, 22, 24].

A review of the creativity literature suggests that, to be effective for early design, a tool must support at least these properties: structured sketching, rapid exploration of the design space, working with multiple ideas, collaboration, and reflection and "anywhere" refinement.

Existing tools support *some* of these properties, but none effectively support *all* of them. For example, sketch-based design tools, e.g., [2, 12, 14] support sketching and storyboarding, but do not support collaboration or visualization of multiple ideas. Other tools support collaboration, e.g., [11], but not sketching and storyboarding.

To better support these properties, we developed a new design tool called SCWID (Supporting Creative Work In Design). The tool is comprised of a server executing on a machine driving a large display and client software executing on local displays. The large display provides a shared visual context for visualizing multiple ideas. Using the local displays, designers sketch details on high-resolution screens in an ergonomic manner, as opposed to standing and reaching at a large display. Also, designers can navigate within a particular design and manipulate alternatives from the local display. Our design was informed by extensive low-fidelity prototyping using both individual and groups of users.

Since much of a final design is produced in the early design process [17], our tool can help designers produce more effective design solutions. To enable broad use by the design community, we engineered our tool such that it can run on a standalone machine or as part of a distributed workspace, formed by connecting multiple clients to a server running on any machine driving any large display. This allows the software to be used in dedicated design rooms or to form ad-hoc workspaces using publicly available large displays.

## RELATED WORK

In this section, we present an overview of the creative problem solving process, aids to facilitate it, and discuss computer tools that support different aspects of creativity.

### The Creative Problem Solving Process

The creative process consists of a continuous cycle including an idea generation stage in which many ideas are created, often in rapid succession, and a refinement stage which often includes periods of reflection and incubation before the idea is altered [7, 8]. Iterations of the cycle have

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

UIST'05, October 23–27, 2005, Seattle, Washington, USA.

Copyright ACM 1-59593-023-X/05/0010...\$5.00.

been shown to work especially well when alternating between individual and collaborative work [7, 21].

In [24], a four-phase framework is presented to describe how computer tools can and should better support the creative process: (1) *collect* relevant information from digital libraries and visualizing data processes, (2) *relate* by collaborating with peers, (3) *create* by generating ideas from free associations, exploring solutions, composing artifacts, and reviewing session histories, and (4) *donate* by providing solutions to digital libraries for later creators. Our current implementation supports each of these phases, focusing on the second and third.

### Aids to the Creative Problem Solving Process

There are many techniques that have been shown to enhance creativity. Sketching supports a re-interpretive cycle in the individual thinking process, supports reinterpretation of each other's ideas in group activity, and enhances access to earlier ideas [18, 19]. Storyboarding and other structuring aids in rapid, creative idea generation and helps groups visualize, organize, and connect ideas [15]. Different settings can provide additional inspiration through the introduction of new stimuli, allowing novel mental associations [23]. Visualization of ideas aids in the necessary exploration and transformation of the "conceptual space" by forming good maps and structuring ideas [4, 22, 24].

Specific to creative group work, improving dimensions such as effective communication and conflict management can allow groups to perform significantly better on creative tasks as the number of group members increases [26]. Other beneficial aids come from group awareness and private and public work within a collaborative setting helps in group creative processes [19, 21, 25].

### Tools Supporting Creativity

Current tools support different properties of the creative process to various degrees, but none fully support all of them. For example, tools such as DEMAIS [2], DENIM [14], and SILK [12], provide sketched-based interfaces and storyboarding, but do not support collaboration or specific techniques for working with multiple ideas. Tools like Designers' Outpost [11] support collaboration, but not sketching, storyboarding, or visualization of multiple, distinct ideas.

Interaction techniques have been developed for high-resolution large displays to support working with multiple ideas [9], but they force designers to stand, reach, and physically move about the display. Our system seeks to provide a more ergonomic solution that is targeted for many consecutive hours of design.

When tools better support different aspects of the creative process, they can result in more effective designs [1]. Our tool incorporates a set of design goals informed by the creative process to achieve higher quality designs than current tools. These goals include: support for rapid idea generation through sketching and structuring, rapid exploration of many ideas or refinements of a single idea, collaboration

with alternating sessions of group and individual work, and flexibility for reflection and refinement to occur anywhere.

### DESIGN GOALS

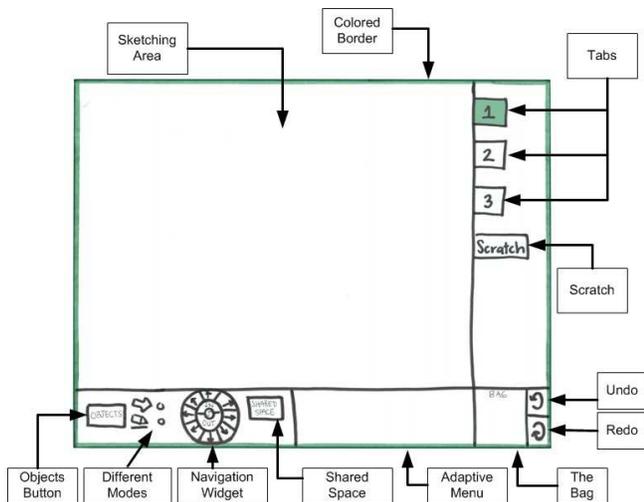
A review of the related work shows that, in order to support the creative problem solving process, tools must effectively support at least the following:

- *Sketching and structuring.* Sketching allows designers to quickly externalize thoughts, allowing for visualization, communication, and external storage of ideas. Sketching also provides a level of ambiguity that facilitates reinterpretation for individuals and across group members [19]. Support for structuring these ideas (such as storyboards) aids in visualization and connecting ideas [4, 22].
- *Rapid exploration of the design space.* A designer is more likely to produce a more creative final design if enabled to explore large numbers of design possibilities or to quickly refine a particular design along different dimensions relevant to the domain [8]. This is typically facilitated by enabling the designer to create mental associations between various alternatives and facilitating the reuse of parts of other design ideas [7, 19].
- *Working with multiple design ideas in parallel.* One key to creating better designs is synthesizing multiple design ideas through comparing and contrasting [7, 19]. Another is creating new designs based on inspiration from past or alternate design ideas [7, 24]. In both cases, this requires viewing different design possibilities in parallel, rather than in sequence. The designer should have the ability to arrange the view of the design ideas in whatever way is most natural or useful for the particular context.
- *Collaboration.* Effective group work processes can result in significantly better work as the number of group members increases [26]. To support effective group work processes, a system must provide designers with both group awareness and the ability to work both publicly and privately [19, 21, 25].
- *Reflection and "anywhere" refinement.* Refining a design idea typically involves periods of reflection (examination of a specific idea and alternatives) [7, 8, 24] and incubation (background thinking about the idea) [7, 21]. For these to be most effective, a tool should provide a way for the designer to work on the idea when and where inspiration strikes them [23]. This necessitates that a tool be flexible in allowing work to be done in many different settings.

Though not exhaustive, we felt that these goals would influence our design the most. Also, the goals are not binary; rather, they lie on a continuum. The better tools support them, the more the tools facilitate creativity. While our current implementation does not fully support all of them, we defined them up front to help guide later design decisions.

### ITERATIVE DESIGN

The basic configuration of the system is a distributed display workspace where a large display shows shared visual



**Figure 1.** The initial low-fidelity prototype featured a sketching area, a control toolbar, and a canvases toolbar.

(global) context of all the design ideas while private displays allow designers to independently sketch details, navigate (pan and zoom) within a specific idea, or switch to an alternative idea. Each design idea is represented on a zoomable drawing canvas, and each canvas is independent.

Building on our earlier work [6], a private display shows details of part of a particular design idea. For each private display, there is a corresponding colored rectangle (frame of reference) shown within a particular design idea on the large display. This shows the viewing relationship between the private display and the context of a particular design idea. If a designer views a different design idea, the frame of reference would then be shown within that idea.

To develop our tool, we followed an iterative design process [13]. This included building and evaluating low-fidelity prototypes, and iterating on the design based on lessons from each round of evaluation. Consistent with this process, our initial prototype started as a minimal, rough design and was extended and refined after each of three iterations.

The prototypes consisted of sheets of paper with transparencies as overlays. During evaluations, computer functions were performed by two researchers with one acting also as the facilitator for each user. Each evaluation focused on a different aspect of the tool. Lessons were used to inform the development of the functional interface.

### Iteration 1

For the initial prototype, collaboration and visualization of multiple ideas was supported by segmenting the large display into equal partitions for each idea. For example, if there were two ideas, the display would be split in half, if there were three ideas, it would be split in thirds, etc.

Figure 1 shows the private display's interface. The interface was broken into three main sections. The sketching area shows details of a particular idea. Tabs along the right rep-

resent alternatives and selecting one moves to the default view within it. There is a specific canvas for private work.

The bottom toolbar contains buttons for creating objects (such as storyboards), controlling the global context, and undo/redo functions. The interface also contains a widget for panning and zooming the design and a "bag" for holding groups of strokes that are being copied to another location. The border of the interface shows the user's unique color to allow identification on the large display.

For the evaluation (figure 2), we asked three individual users to perform tasks including sketching, altering the properties of strokes on the canvas, selecting strokes on the canvas, navigating within a canvas, navigating between canvases (both shared and private), and moving strokes between canvases.

As users interacted with our prototype, we asked them to think aloud. After each task was completed we also asked a series of questions to understand the user's expectations and reactions to our system. Based on analysis of the verbal protocol, observations, and discussions, we learned that:

- *Controls for the global context need to be clearly distinguished from local controls.* Due to similarities between local and global controls, users often experimented with aspects of the interface and ended up inadvertently affecting the global context. In a collaborative setting, these unintentional manipulations of the shared context could disrupt the work of other users. To address this issue, we placed global controls in a separate pop-up menu accessible from the toolbar.
- *Controls should be represented in the same way that users relate to the elements being controlled.* Though the tabs representing each canvas were analogous to common features in many current tools, users were confused because they identified each design idea by its content rather than its name. We addressed this issue by replacing the tabs with a map (worlds in miniature) view of the shared visual context. The map was placed on the toolbar and would expand when clicked. A designer could then select the desired canvas. The overview expands in order to allow the user to better manipulate the canvases.
- *Controls for navigation between areas in different canvases should be different than controls for navigation within a canvas.* When asked to pan their view within a canvas, users found the control widget useful. However, when asked to navigate large distances or between canvases, users did not consider zoom techniques (zooming out, panning, and zooming back in) to be appropriate. Rather, they wanted a discrete interaction to move to another canvas. To address this, we decided that a user could select a canvas in the map view to navigate to it and could draw a stroke around an area in the canvas to navigate to that canvas and set the view to that area.
- *Users want to be able to work privately without losing the surrounding strokes.* From the related work we know that users require different modes to collaborate effec-



**Figure 2.** For our first formative evaluation, our low fidelity prototype was made entirely from paper and transparencies. In the upper left, a user evaluates the interface by running through common tasks. The interface shows a drawing canvas, a toolbar, and an area for tabs that allow users to switch between multiple canvases. In the upper right, a researcher holds additional sheets of paper against the display wall to show the user the results of an interaction.

For our second and third formative evaluations, shown in the lower left, electronic pictures of our physical documents were used for the large display and one researcher manipulated them from a computer off to the side. Multiple users were brought in to evaluate the system for collaboration issues. A researcher worked with each of them to allow for simultaneous interaction with the interface.

tively. We reasoned that a private canvas would solve these issues. However, users frequently expressed that they would want to do their work within the context of other shared work. We addressed this issue by introducing a “hidden” mode where users draw strokes on shared canvases that are only visible on their private display.

### Iteration 2

For the second evaluation we asked three pairs of users to perform tasks including loading canvases with design ideas into the shared context, juxtaposing them for discussion, and saving and closing specific canvases. We used our initial prototype with modifications from the first iteration.

Again, users were asked to verbalize their thoughts as they completed tasks. After each task, the users were asked to write responses to questions on paper and were then asked to discuss their answers with each other and the researchers. Based on observations of the tasks, written responses, and verbal discussions, we learned that:

- *Tools should allow for personal territory within the global context.* Initially we laid out loaded canvases in rows based on the order that they were loaded. Users were very surprised that the canvases that they loaded were not physically close to them on the large display. They reasoned that if they loaded the canvas, it should

appear near where they were sitting. We addressed this issue by allowing a user to select the initial location.

- *Users need locking mechanisms in shared tools.* In this case, we refer to a lock as a way for a user to temporarily gain sole control over a canvas. We initially thought that private canvases would provide enough flexibility for users to do work that group members could not modify, but this did not allow for showing the design ideas to others in a read-only manner. Our users in this evaluation felt a strong sense of ownership for their design ideas and did not want others to be able to manipulate them without express permission. We addressed this issue by introducing a way to “lock” a canvas or an area within that canvas so that only the specified user could sketch on, edit the contents, move, resize, or close it.
- *Canvases should emulate the ways users are accustomed to interacting with application windows.* An issue which arose was whether canvases in the shared context should be allowed to overlap each other or whether they always be completely visible (necessitating resizing as canvases are moved about or loaded into the space). Users typically preferred all design ideas to be visible at all times, but they preferred a system that allowed overlapping canvases to one that would cause global re-arrangement whenever an idea was added or removed. We addressed

this by altering our prototype to allow freeform movement and resizing of canvases within the global context.

### Iteration 3

For the third evaluation, we asked three pairs of users to perform tasks related to the hiding and locking mechanisms introduced after previous iterations. Among other tasks, they were asked to sketch in both shared and hidden modes and to sketch in both normal and locked modes. In both cases they were asked to work within one canvas on a single design idea. The evaluation method for this iteration was the same as in the second iteration. We learned that:

- *Users should not be allowed to do hidden work within canvases over which they do not have sole ownership.* Our users were extremely against allowing a collaborator to affect a shared area in a way that is not visible to other group members. We addressed this issue, in the context of our lessons from the first iteration by allowing any number of private canvases and providing quick menu shortcuts to copy groups of strokes for pasting into a private canvas. A user can cut or copy the necessary strokes, modify them as desired, and replace the old version with their new version when appropriate.
- *Only area content needs to be locked.* Originally, we included many different options for what a lock would do. Users found these very confusing and preferred a solution where they could simply turn a lock on or off. They argued that providing different locks for both the entire canvas and for an area in it was unnecessary because the canvas can be locked against closing if any part is locked and because if the group is collocated, social convention will solve the rest. We altered our prototype to include only an on/off lock for areas of the canvas.
- *Locked areas should be clearly marked.* One of the main reasons that users agreed that locking was important was in order to “reserve” an area that the user knew in advance they would use for their current work. However, they were adamant that the system should clearly distinguish both that the area is locked and who it is locked by. We addressed this issue by highlighting locked areas with the same color as the user’s border rectangle.

After the third evaluation, we felt that the design had solidified enough to move forward with an implementation.

### SYSTEM DESCRIPTION

Based on our design goals, the lessons we learned from low-fidelity evaluations, and lessons taken from the evaluations of other tools (e.g. [1]) we created SCWID: A tool designed for Supporting Creative Work In Design.

SCWID consists of a server running on a machine with a large display, and client software executing on private displays. The large display shows the global context by providing separate canvases placed freely within it.

The client divides the screen into three main sections (see figure 3). The sketching area provides a controllable view (can be panned and zoomed) into a specific canvas. The

sketch area is surrounded by a border with the user’s unique color. We also provide pie menus [5] for manipulating selections.

The right side of the client contains a local controls toolbar for general user interactions including a file menu, a navigation widget, buttons for common editing tasks, and a storage area for groups of strokes or objects (similar in function to a Windows clipboard). The local controls toolbar also contains mode buttons to allow a user to select sketching or selection mode. The toolbar provides access to different canvases by listing private canvases that a user can move between and a map view of the space. When selected, the map view covers the client and mirrors the view of the large display to allow for manipulating canvases. A separate toolbar allows a user to control the global context.

Along the bottom of the client is a toolbar that hosts the visual sketching language, based on [2]. Among other features this toolbar provides controls for defining behavioral strokes and executing a design.

### Scenario of Use

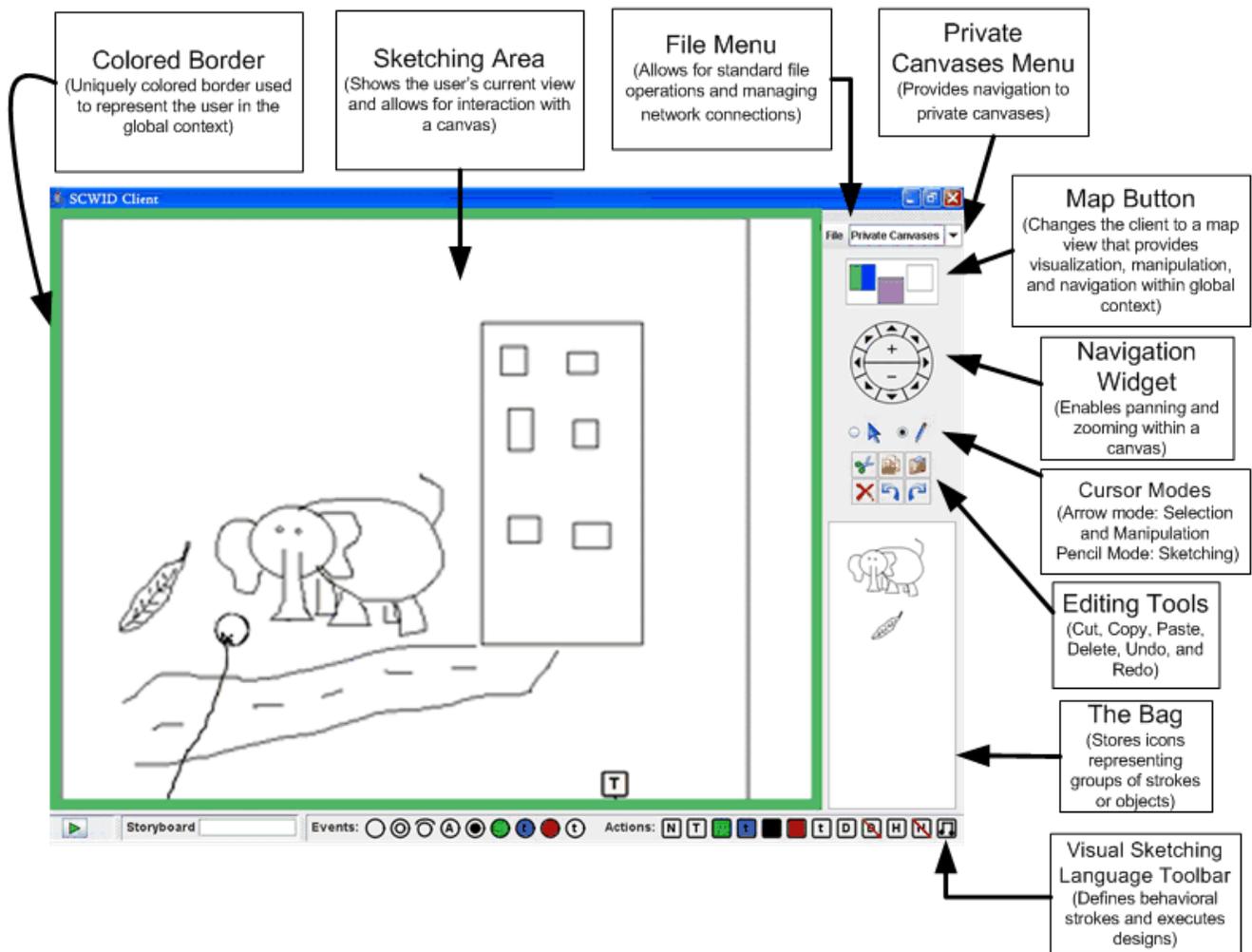
In this section we provide a representative use case scenario and use it to show how our tool supports each of our design goals and benefits that come from this support.

With a tight budget and short deadline looming, a group of designers is trying to come up with a creative educational game that teaches children math and physics concepts. They are running our software and are using a large tiled display for the shared context and networked tablet PCs for the private displays. This scenario follows them from the individual creation of a design idea, to a group meeting focused on creating and evaluating each other’s and more ideas, and ends with them individually refining the better ideas.

### Sketching and Structuring

Before meeting with the rest of the group, Janel creates a new canvas in SCWID to sketch out an idea. She creates two storyboards to structure the canvas and sketches in one of them. She makes use of the visual sketching language offered by SCWID to add behavioral strokes indicating sound and movement to her sketch and to connect the storyboards. At any time, she can run the design to quickly interact with a working prototype of the idea.

Using our tool, Janel is able to quickly sketch out the content of her partially-formed idea before she forgets it. Dividing the space with storyboards allows her to contain her thoughts within different areas so that she can easily distinguish them later. Structuring the space also allows her to connect the different thoughts logically as scenes. We further allow her to work quickly by providing basic editing functionality so that when she wants reuse parts of her ideas in several storyboards within the canvas, she need only draw a circle around the strokes, click the copy button, navigate to the other areas and paste the strokes. Finally, using a visual sketching language allows her to not only note what interactions should happen, but to actually ex-



**Figure 3.** The functional interface is broken into three major sections. The sketching area allows a designer to interact with the area of interest in a canvas. The bottom toolbar provides controls for behavioral strokes and execution of the design idea. The side toolbar provides local controls such as navigation, cursor modes, and the bag, which allows selections of strokes or objects to be stored in an iconified form for later use. An interesting feature to note is the map button, which not only opens a map view of the global context, but also displays an up-to-date thumbnail picture of the global context.

perience them by executing the design without having to create a high-fidelity prototype.

### Rapid Exploration of the Design Space

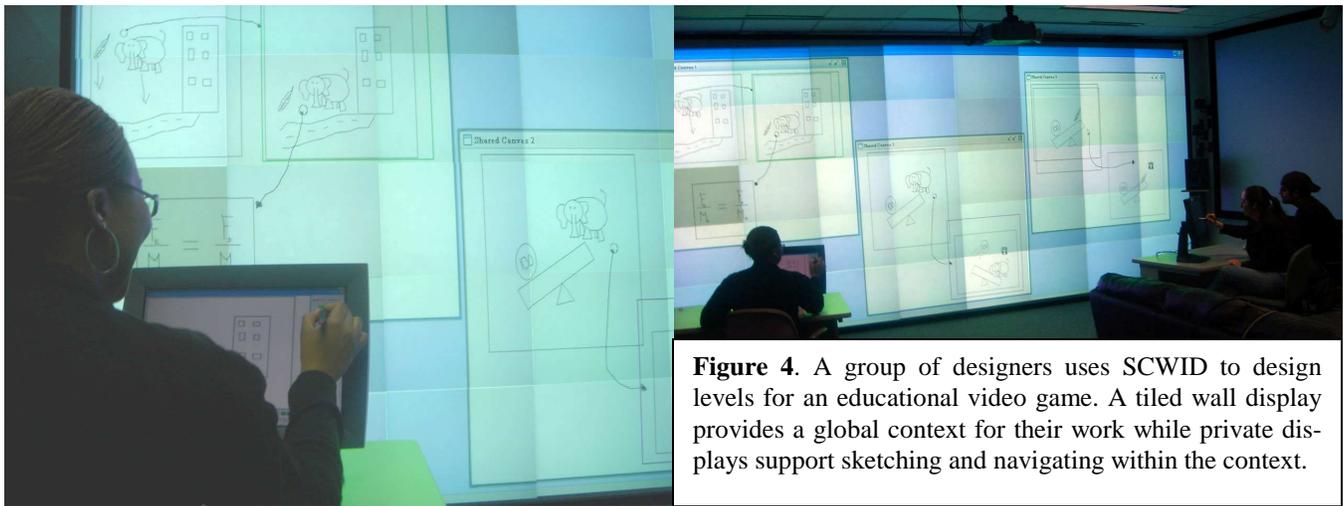
Later, the designers meet to generate ideas using the brain-sketching technique [16]. To inspire new ideas, they start by loading one design each from the work they had done in advance (see figure 4). Then they each create a blank canvas by selecting “new canvas” from the file menu and specifying an initial location near the loaded design. They use the blank canvas to sketch an idea inspired by the loaded design. After creating each new idea, they switch to a different set of ideas (initial loaded idea and those inspired by it) and create another new idea inspired by those in the series.

While brain-sketching, Janel creates several storyboards for each of her ideas. Several times she decides to combine elements from previous design ideas into a new one. She

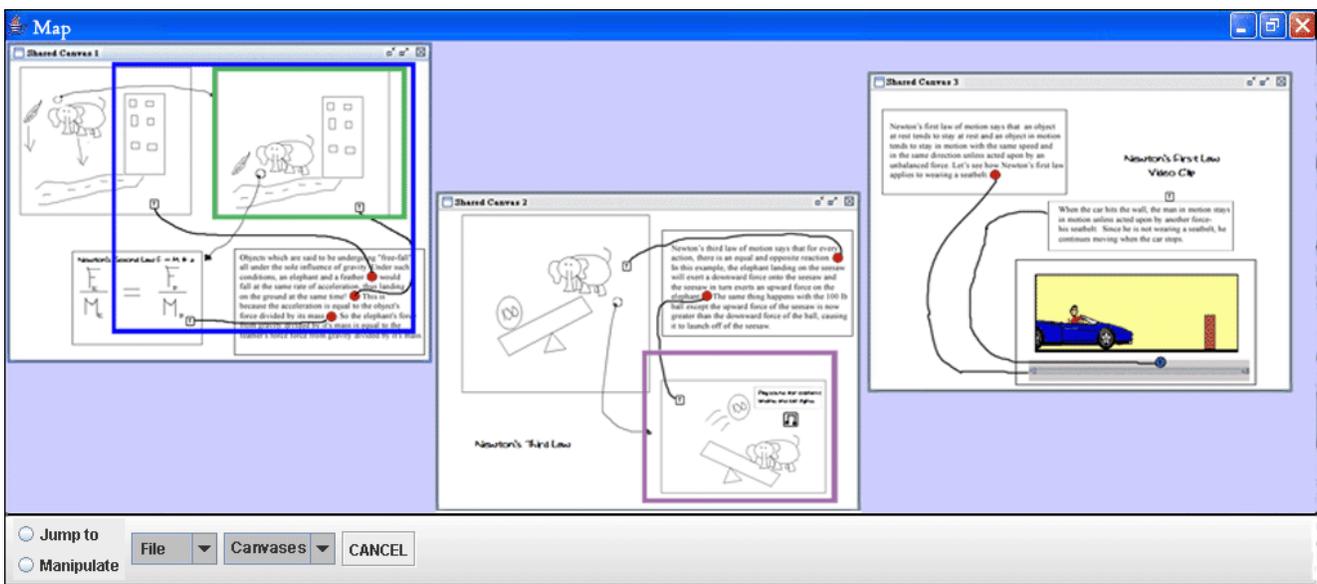
navigates to each canvas by selecting the map view and jumping to her previous view in each canvas. In some cases, she knows the specific area in a canvas she wants to view. By drawing a gesture around it, the tool switches to that canvas and sets the view based on the gesture.

Once she is viewing the canvas, she uses her non-dominant hand with a six degrees of freedom (6DOF) device to pan and zoom to the exact view that she wants. She selects the appropriate strokes and drags them to the bag. There, they form into an icon representing the strokes she selected. She does this to obtain selections from three of the previous designs and returns to her original canvas. Once she is viewing the original canvas, she drags the icons from the clipboard and they expand to become the strokes again.

Janel and her group members are able to rapidly explore the possible designs because they can quickly sketch out ideas as they occur. They are able to not only draw from previous



**Figure 4.** A group of designers uses SCWID to design levels for an educational video game. A tiled wall display provides a global context for their work while private displays support sketching and navigating within the context.



**Figure 5.** The client's map view allows a designer to view and manipulate the global context from a private display. The "jump to" mode allows quick navigation to a canvas. Right-clicking initiates an attentional "beacon" on the large display.

designs intellectually, but they can actually navigate to them to reuse pieces of previous designs. The ability to navigate quickly allows for these actions as well as moving between areas within a single canvas.

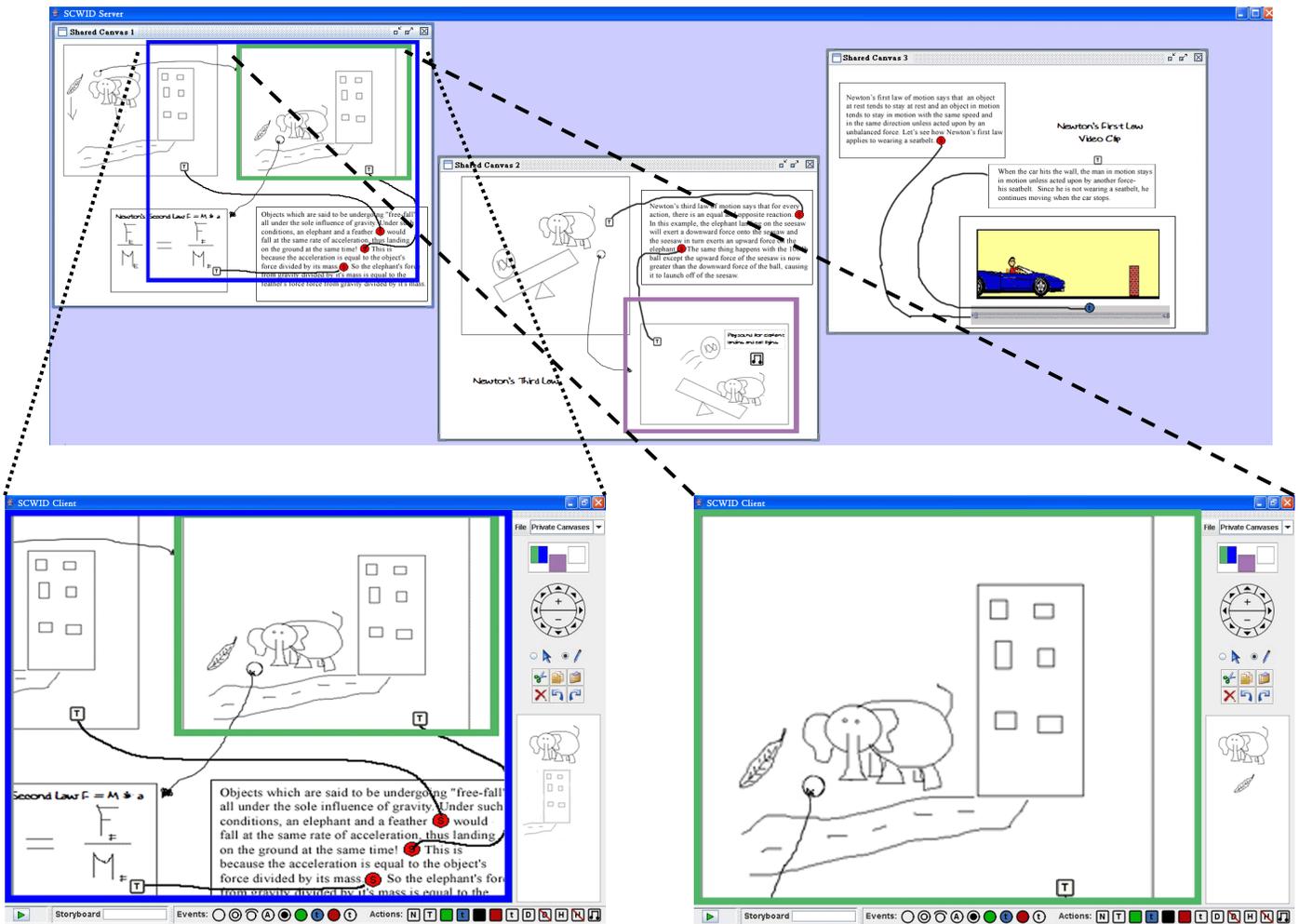
### Working With Multiple Design Ideas in Parallel

After creating a number of ideas, the group members need to compare and evaluate them. Mike is selected to be in charge of manipulating the large display and he selects three design ideas to discuss. He opens up the map view (figure 5) on his private display and moves the three canvases to the different areas of the display. He resizes them to be easily viewed by the group. They are all able to navigate through the canvases to review and discuss them (see figure 6).

When the group is finished discussing the first set of design ideas, Beth makes some refinements to one idea. As she does this, Mike selects different design ideas to discuss. He resizes the first three (including the one Beth is working on)

and moves them aside before bringing in and enlarging the next selections. Beth's private display is not affected by Mike's interactions with the system.

By allowing canvases to be quickly moved and resized within the global context from each private device's map view, we provide an easy way to juxtapose ideas for consideration and discussion. Separating the files so that each canvas is saved in a different file allows users to do work on one system and load the work into another or to save only the relevant parts of a design session. Providing the personal screens with a view into a canvas rather than into the global context allows a canvas to be resized, moved, or even overlapped by another without affecting a designer working within that canvas. Of course, the designer can get a view of the global context by pressing the map view button or simply glancing at the large display.



**Figure 6.** Screenshots of the SCWID server and two clients: At the top, the large display shows three different design ideas in separate canvases. Two designers interact with specific areas within the leftmost canvas through their private displays (at bottom). The designer on the left is represented on the large display by a blue rectangle while the designer on the right is represented both on the large display and on the blue designer’s private display as a green rectangle. A third, not pictured private display allows another designer to interact with the middle canvases.

### Collaboration

After discussing the design ideas, the group selects one idea to refine. They divide the work and start refining the idea. Both decides that she would rather work on her part privately and wait to share her work with the group until she has finished it. She copies the relevant strokes and pastes them into a private canvas. She refines the section there and pastes her work back into the shared canvas when finished.

Janel starts sketching out her part of the design. She soon realizes that her sketch will require a large amount of space to complete. She glances at the large display and sees that Mike’s frame of reference is close to hers, indicating that he is working on a related part. She indicates to Mike how far she will need to draw by flashing a “beacon” on the large display and asks him if she can use the space. After he agrees, she zooms out and draws a stroke around the area she will need. She selects the lock option and the area becomes highlighted with the same color as her rectangle. Now Mike can clearly distinguish the area that Janel will

need and is also restricted from accidentally sketching into that area. When finished, Janel selects to turn off the lock allowing the view and functionality return to normal.

The group’s collaborative efforts are supported by SCWID in that users are able to work both privately and publicly as needed and to coordinate with their group members more easily by being able to see who is working nearby via the shared display and map view button. By looking at the map view button, which shows current positions of canvases and the colors of the users working within those canvases, users have awareness of the location of other group members.

The beacon function is available in the map view as a quick way to draw attention to a specific area. Using this and the locking mechanism allows users to indicate and reserve an area. The lock could also be used to indicate that some design elements are permanent and stop accidental manipulation of those elements.

### Reflection and “Anywhere” Refinement

After finishing their work, each group member can take a different design with them to refine on their own. Each of them saves the canvases that they think will be useful to them later and part ways. Though none of them have individual access to the tiled display, they are able to refine the design ideas using only the client program on their tablet PCs. In this case the program acts as a traditional zoomable interface without a separated context.

The designers are supported in the reflection and incubation stages of their creative process by both the flexibility of the system and the distributed file structure. The flexibility allows them to continue refining designs in a wide variety of settings while the file structure allows them to work without a server and without requiring the entire design context.

### IMPLEMENTATION

Our system is designed to benefit designers with any computer hardware available and that are working either individually or in a group. To support this flexibility, the system allows for nearly any hardware configuration, operating system use, or network connections.

To support different hardware configurations, the tool is designed with a client-server architecture to allow the software to be run on multiple machines through TCP connections, on the same machine with multiple monitors (using local TCP connections), or as a standalone client. Virtually any large display can be used for the shared context or, if no large display is available, the tool can still provide many benefits to a designer.

SCWID supports designers regardless of input device. Though designed for maximum effect with a stylus for sketching and a 6DOF device for navigation, the tool works just as well with only a mouse. Keyboard shortcuts are provided for some functions, but are neither required nor a significant factor in the design of the system.

To allow for operating system portability, SCWID is written with Java 1.5 using the Piccolo zoomable user interface toolkit [3]. This allows designers to use any computer that supports a Java runtime environment, which, with further development will include handheld and other small devices.

This flexibility also allows for the benefits of the system to be available with any network, whether wired or wireless. This allows groups of designers to form ad-hoc networks in any location with a large display.

Add this flexibility to the general creativity principles inherent in the design and we are excited about the benefits we can provide to a very wide range of designers.

### DISCUSSION

In this section, we discuss the current implementation of our tool meets our design goals:

- *Sketching and structuring* is supported by using stylus input and storyboarding to create and group design content. A visual sketching language extends the concept of sketching into the temporal and interactive domain.

- *Rapid exploration* is supported by providing visualization of many design ideas on the large display and providing quick ways to draw pieces from different designs to form a new one.
- *Work with multiple designs in parallel* is supported by allowing simple interface controls for moving and resizing canvases within the global context.
- *Collaboration* is supported through the use of both shared and private canvases as well as methods to provide awareness of the activities of group members.
- *Reflection and “anywhere” refinement* are supported by the flexibility of the system which allows canvases to be easily separated from the global context and which allows the tool to be used in virtually any computing environment.

The use of our tool is most appropriate for designs that are of large-scale, designs that will be highly collaborative, or designs that demand highly creative solutions. For other situations, existing design methods and tools may be more appropriate. While a usability study has not yet been conducted, Mack and Nielson [20] and Virzi, et al. [27] have showed that low-fidelity prototyping finds many of the same usability issues that would be found in an empirical test of a high-fidelity prototype.

### FUTURE WORK

Our future work is to conduct usability studies to further enhance and refine the interface and evaluate how the use of our tool affects designers and the design process relative to existing tools. Also, since current sketching language was taken almost directly from a previously developed tool, we are investigating extensions to the language that allow it to express more complex interactions with minimal overhead.

Creative thinking is a core part of almost any ill-structured problem solving activity. Since the goals driving our tool were not domain specific, we want to try and abstract from our existing tool a more general purpose software framework that could be applied in domains other than interactive systems design.

### ACKNOWLEDGEMENTS

We thank Fen Lu for her help with coding our project. We also thank all of our users for their insightful feedback.

### REFERENCES

1. Bailey, B.P. and J.A. Konstan. Are Informal Tools Better? Comparing Demais, Pencil and Paper, and Authorware for Early Multimedia Design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2003, 313-320.
2. Bailey, B.P., J.A. Konstan and J.V. Carlis. Demais: Designing Multimedia Applications with Interactive Storyboards. In *Proceedings of ACM Multimedia*, 2001, 241-250.
3. Bederson, B.B., J. Grosjean and J. Meyer. Toolkit Design for Interactive Structured Graphics. *IEEE Transactions on Software Engineering*, 2003.

4. Boden, M.A. Agents and Creativity. In *Cognitive Technology: In Search of a Humane Interface*, Elsevier, 1996, 119-127.
5. Callahan, J., D. Hopkins, M. Weiser and B. Shneiderman. An Empirical Comparison of Pie Vs. Linear Menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Washington, D.C., United States, 1988, 95-100.
6. Flider, M. and B.P. Bailey. An Evaluation of Techniques for Controlling Focus+Context Screens. In *Proceedings of Graphics Interface*, London, Ontario, 2004.
7. Gabora, L. Cognitive Mechanisms Underlying the Creative Process. In *Proceedings of the ACM Conference on Creativity & Cognition*, 2002.
8. Grossman, S. Brainstorming Updated. *Training and Development Journal*, 84-87, 1984.
9. Guimbreti re, F., M. Stone and T. Winograd. Fluid Interaction with High-Resolution Wall-Size Displays. In *Proceedings of the ACM Conference on User Interface Software and Technology*, 2001, 21-30.
10. Herbsleb, J.D. and E. Kuwana. Preserving Knowledge in Design Projects: What Designers Need to Know. In *Proceedings of the ACM conference on Human Factors in Computing Systems*, 2003, 7-14.
11. Klemmer, S.R., M.W. Newman, R. Farrell, M. Bilezikjian and J.A. Landay. The Designers' Outpost: A Tangible Interface for Collaborative Web Site Design. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2001, 1-10.
12. Landay, J.A. and B.A. Myers. Interactive Sketching for the Early Stages of User Interface Design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1995, 43-50.
13. Lewis, C. and J. Rieman. *Task-Centered User Interface Design*, 1993.
14. Lin, J., M.W. Newman, J.I. Hong and J.A. Landay. Denim: Finding a Tighter Fit between Tools and Practice for Web Site Design. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 2000, 510-517.
15. Lottier, L.F. Storyboarding Your Way to Successful Training. *Public Personnel Management*, 15 (4), 421-427, 1986.
16. Lugt, R.v.d. Brainsketching and How It Differs from Brainstorming. *Creativity and Innovation Management*, 11 (1), 43-54, 2002.
17. Lugt, R.v.d. Developing a Graphic Tool for Creative Problem Solving in Design Groups. *Design Studies*, 21 (5), 505-522, 2000.
18. Lugt, R.v.d. Functions of Sketching in Design Idea Generation Meetings. In *Proceedings of the ACM Conference on Creativity & Cognition*, Loughborough, Leic, UK, 2002, 72-79.
19. Lugt, R.v.d. How Sketching Can Affect the Idea Generation Process in Design Group Meetings. *Design Studies*, 2005.
20. Mack, R.L. and J. Nielsen. Usability Inspection Methods: Executive Summary. In *Readings in Human-Computer Interaction: Toward the Year 2000*, 1995, 170-181.
21. Paulus, P. and H.-C. Yang. Idea Generation in Groups: A Basis for Creativity in Organizations. *Organizational Behavior and Human Decision Processes*, 82 (1), 76-87, 2000.
22. Prante, T., C. Magerkurth and N. Streitz. Developing Cscw Tools for Idea Finding - Empirical Results and Implications for Design. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, New Orleans, Louisiana, USA, 2002, 106-115.
23. Shapero, A. Managing Creative Professionals. In *Managing Professionals in Innovative Organizations: A Collection of Readings*, Ballinger, 1988, 215-222.
24. Shneiderman, B. Creating Creativity: User Interfaces for Supporting Innovation. *ACM Transactions on Computer-Human Interaction*, 7 (1), 114-138, 2000.
25. Souder, W. and R. Ziegler. A Review of Creativity and Problem Solving Techniques. In *Managing Professionals in Innovative Organizations: A Collection of Readings*, Ballinger, 1988, 267-279.
26. Taggar, S. Group Composition, Creative Synergy, and Group Performance. *Journal of Creative Behavior*, 35 (4), 261-286, 2001.
27. Virzi, R.A., J.L. Sokolov and D. Karis. Usability Problem Identification Using Both Low- and High-Fidelity Prototypes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, British Columbia, Canada, 1996, 236-243.