Multimodal Split View Tabletop Interaction Over Existing Applications

Edward Tse[1,2], Saul Greenberg[1], Chia Shen[2], John Barnwell[2], Sam Shipman[2], Darren Leigh[2]

[1]*University of Calgary, Computer Science*
2500 University Dr. N.W.
Calgary, AB, T1W 1S2, Canada
[tsee@cpsc., saul.greenberg@]ucalgary.ca,

[2]*Mitsubishi Electric Research Laboratories,*
201Broadway
Cambridge, MA, 02145,USA
[shen, barnwell, shipman, leigh]@merl.com

Abstract

While digital tables can be used with existing applications, they are typically limited by the one user per computer assumption of current operating systems. In this paper, we explore multimodal split view interaction – a tabletop whose surface is split into two adjacent projected views – that leverages how people can interact with three types of existing applications in this setting. Independent applications let people see and work on separate systems. Shared screens let people see a twinned view of a single user application. True groupware lets people work in parallel over large digital workspaces. Atop these, we add multimodal speech and gesture interaction capability to enhance interpersonal awareness during loosely coupled work.

1. Introduction

In everyday physical collaboration over a shared visual surface, people fluidly transition between working closely together (tightly coupled) and working in parallel (loosely coupled). The situation is somewhat different in the digital domain.

When people are geographically distributed, they routinely work together while viewing *independent applications* (Figure 1 left; A and B are different applications). Because they cannot see each other's screens and bodies, they use other channels (voice, instant messaging) to explicitly state what is visible on the screen. To improve this unwieldy situation, *shared screen systems* duplicated the output of one person's application so that it was also visible on distant screens (Figure 1 middle; A' is a duplicated view of A). Joint action was allowed by a wrapper that gathered and serializing people's input through a turn-taking mechanism, and then passed it onto the application [3]. This is essentially a what-you-see-is-what-I-see (WYSIWIS) view, where each person sees exactly the same visuals and fine-grained changes on their display [14]. Because of the strong linkage between views, shared screen views work well for tightly coupled work. Alternately, *true groupware systems* understood

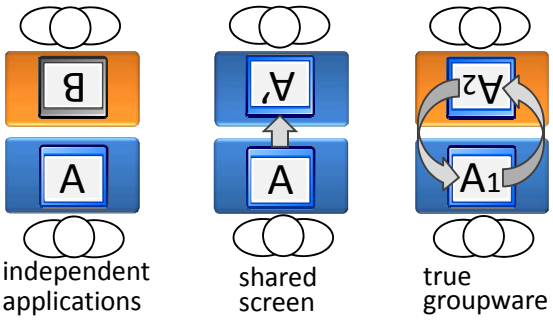

Figure 1. Software configurations for two people working face to face on a split view tabletop

that multiple people were working in the space (Figure 1 right; A_1 and A_2 are instances of the same groupware application that are linked with one another). True groupware systems facilitate loosely-coupled work by allowing simultaneous input, and by relaxing WYSIWIS to allow people to navigate and work independently on different parts of a large digital workspace [14]. Generally, the transition between loosely and tightly coupled work in distributed applications is enabled by the *mechanics of collaboration* [6]. People's awareness of each other's speech, gestures, and gaze actions produce consequential communication around the work surface that facilitates how they engage, interact, coordinate, and transition between loosely-coupled and tightly-coupled work. Common groupware awareness methods supporting these mechanics include telepointers for gesturing [5], and radar overviews to give people a sense of what others are doing if they are working on different parts of scene [6].

Within the context of co-located groupware, things are somewhat analogous. The *independent applications* configuration of Figure 1 (left) happens when people seated next to each other are working on separate computers; their talk can draw attention to each other's display. As they turn to look at one of the screens, they have just transitioned to a simple shared screen system. The limitation is that there is only one input device, so only one person actually interacts with the system, or they manually share that device through turn-taking.

Cite as:
Tse, E., Greenberg, S., Shen, C., Barnwell, J., Shipman, S. and Leigh, D. (2007) Multimodal Split View Tabletop Interaction Over Existing Applications. Report 2007-869-21, Dept. of Computer Science, University of Calgary, Calgary, Alberta, Canada T2N 1N4, June.

To mitigate this, early single display groupware (SDG) systems provided multiple input devices so people could work in parallel [15]. There are two primary approaches to SDG.

1. *Basic SDG* exploits how most operating systems merge the input from multiple mice into the single input stream seen by the standard single user application. The result is akin to screen-sharing (Figure 1, middle). While each person has a mouse, they still have to negotiate whose turn it is. That is, basic SDG favors very tightly coupled work through strict WYSIWIS turn-taking.
2. *True SDG* uses custom-built groupware applications that recognize and take advantage of multiple mice. Typically, all people have their own cursor and can work in parallel [15] akin to true groupware (Figure 1 right). However, the constraints of the small display usually mean that people are limited to a strict-WYSIWIS view, which in turn favours tightly coupled work. More recently, the development of high-resolution digital walls and tables provide a large enough surface so that people can work somewhat more loosely-coupled when using true SDG: while they all still share the same view, they can work on their corner or side of the surface.

Obviously, true SDG is more flexible than basic SDG in supporting the spectrum of loosely- to tightly-coupled work. The problem is that very few existing real world applications are built as true SDG, thus limiting what people can do. On the other hand, basic SDG can immediately leverage existing applications, but is really amenable only to tightly-coupled work. The question is: can we exploit basic SDG in a way that allows people to work in a loosely-coupled fashion, while still giving them the ability to move towards tightly-coupled interaction by providing strong awareness cues of what the other is doing?

Our answer is *multimodal split view interaction*: a split-screen tabletop configuration that supports both loosely and tightly-coupled joint work over conventional applications projected on a digital table, where the table also promotes awareness through multimodal interaction. The remainder of this paper introduces this concept. We begin with split view interaction, and define three software configurations that constrain how it can be achieved. We continue by adding multimodal speech and gesture interaction capabilities to these split views, which enhance awareness during loosely-coupled work.

2. The Split View Tabletop

The first half of our system is the ***split view tabletop (SVT).*** It is defined as a digital tabletop, where its surface is split into two adjacent projected views, and a person is seated in front of each view. SVT size expectations are that each person can easily see and reach into any part of their view. Seeing and reaching into the other view is slightly more difficult, but can be done by looking up, or by standing and leaning over the table. The basic idea is that people can work in a loosely coupled manner over their own individual views, and can also work in a tightly coupled manner either by shifting their attention to the other view or by linking their views together through software. Several key factors influence SVT: the actual software being projected, seating, size, and input devices used.

2.1 Projected Software

One of our goals is to work with existing 'conventional' applications rather than re-write applications from the ground up. If successful, our SVT can be repurposed for myriads of available applications in co-located work. In this section, we describe three configurations that leverage existing software over SVT. Each is analogous to the distributed configurations described previously, so we reuse Figure 1 to illustrate each software view configuration in SVT. We show how each configuration offers different application sharing abilities, and how each supports different levels of collaborative work.

Individual applications allow people to work on their own separate application, each displayed on one side of the split view surface (Figure 1, left). For example, one person can use a web browser while the other person uses a digital map. The advantage is that people can work over separate applications in parallel yet still have some peripheral awareness of the actions of others simply by glancing up. The disadvantage is that both applications are not aware of each other, so that people have to resort to shifting attention and reaching over to one of the views if they wish to work in a tightly-coupled manner.

Screen sharing takes the screen displaying an unaltered single user application and projects it onto both views (Figure 1, middle) [3]. This can be implemented trivially using variants of the Virtual Network Computing (VNC) protocol [11]. As in the distributed setting, this forces a strict WYSIWIS view and sequential interaction through a turn-taking policy, making it also somewhat equivalent to *basic SDG*. Within SVT, screen sharing means that people can work very closely together over a single application,

even though only a single person can interact at a time. unlike individual applications, rich awareness cues arising from gaze and gestures are available.

True groupware uses applications designed for distance-separated people, where it instead displays an application instance in each view on the digital table (Figure 1, right). This means we can exploit real-time distributed groupware within the co-located setting, which is akin to true SDG. This includes many PC games, as these are actually groupware designed to run over the Internet. This configuration naturally affords relaxed WYSIWIS, where people can work on their own part of the system without affecting others.

2.2 Seating Arrangements

Seating arrangements give different affordances to the SVT setting, where they can profoundly influence collaborative practice. Consider the following three common seating configurations.

Face to Face seating provides people with easy visibility of each other's gestures and gaze actions on the work surface, as well as easy viewing of one another during conversation [13][12] (Figure 2 left). This is done simply by glancing up. This arrangement is commonly preferred for co-acting and conversation [7]. The cost is orientation, where studies have shown that displays that are text-heavy are significantly more difficult to read when upside down [20]. As well, a person can directly interact in the other view only by standing up and leaning over the table, or by actually moving to the other side of the table.

Side by Side seating also affords visibility of each other and their work by side glances (Figure 2 middle). One can also reach into the other's view simply by sidling over. Orientation is not an issue, as text is usually upright to both viewers. This arrangement is commonly preferred for cooperative tasks [13], as it easier to read text on a collaborator's display [20]. The disadvantage is that side glances are more effortful than glancing up, and thus happen less frequently. As well, simultaneously viewing another person and the workspace at the same time is somewhat harder as the viewing angles are different. Finally, pairs are in very close proximity, and this could be discomforting if they do not know each other well [7].

Catty-Cornered seating is a compromise (Figure 2 right). As with face to face, a glance up provides easy viewing of the other's work and their body. Its 90^0 viewing angle offers slightly better text readability of their partner's screen [20]. However, eye contact is harder to maintain when people are working over the surface. This seating arrangement is commonly

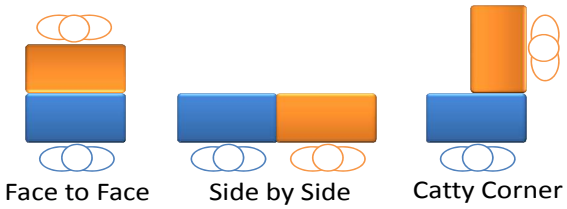

Figure 2. Seating arrangements for two people.

preferred for tasks involving extended conversations as it supports the viewing of other's gestures while not requiring continual eye contact [13].

2.3 Size, Working Area, Reach, and Gaze

The physical size of each person's view can have considerable effect on interaction [7][13].

Small views mean that people can easily reach into both spaces and that they can engage in each other's activity at a glance. The trade-off is limited working area, which is important for parallel work [12].

Arms length views are sized so that a person can just reach all parts of their own view while seated, yet still reach into parts of the other person's view if they stand up or lean over. This is a size where collaborators can still engage in each other's activity at a glance [13].

Large views afford even more space to work in parallel. The cost is that people cannot easily reach all parts of their view while seated, let alone the other person's space. Awareness is also harder to maintain as the other person's display is further away and details become difficult to see [13].

2.4 Input Devices for Pointing / Selection

Input devices for pointing and selecting have a large effect not only on a person's individual work, but on how others can maintain awareness of the gestures of others. Rather than catalog the myriad of tabletop pointing devices and techniques that are under active development, we consider them as broad categories.

Decoupled devices such as a mouse or trackball are physically independent from the display surface. While efficient for individual work, other people may have a hard time noticing small device movements and button presses. A person will likely find it difficult or impossible to interpret what another person's activities mean unless he is looking directly at the telepointer [5] or the artifacts being affected.

Distant freehand devices are directed at the surface but do not directly touch the spot that it is manipulating. Examples are systems that use ray casting or laser pointers on a large display. While the actions of others tend to be more visible, it may still be difficult to interpret one's activities.

Direct touch **devices** correspond directly with the part of the surface being manipulated. Examples include touch tables such as Smart Boards and DiamondTouch [2]. Within a split view setting, the direct engagement of these absolute devices maximizes one person's awareness of the other's activities and their meaning.

3. Multimodal Interaction for Awareness

The second half of our system is its ***multimodal speech and gesture*** interaction capabilities, provided both to facilitate a person's fluid interaction with applications on the table, and to promote awareness between working partners.

Existing applications displayed by our SVT system are almost always designed for a mouse, which is a decoupled device. That is, if people are working in a loosely-coupled manner, these small device movements will be hard for others to notice. For example, observational studies [12] of people working over a wall and table display using single user applications revealed that people maintain awareness [6] by "physically moving back to the table to be in close proximity" to other collaborators and using "outlouds to get the attention of others" and attract the attention of people on distant displays by "shouting out and giving directives to him/her as to what to do next."

For this reason we advocate multimodal speech and gesture commands that serve as both commands to the computer and as awareness for other collaborators [19]. Gestures create consequential communication of each other's bodies and activities, while speech serves as verbal communication to others. While single user applications do not understand gestures or speech, we can create gesture and speech wrappers (macros) that activate keyboard and mouse sequences that in turn invoke application functionality [17]. No changes of the underlying application code are required.

Previous work by Tse, et. al. [19] used multimodal speech and gesture commands to enhance how people interacted over a digital table using single user applications. Their studies revealed that people exploited this for communicative purposes such as answering questions, validating understanding and agreement, and affirming statements made in prior conversations [18]. They also used this awareness to coordinate near-simultaneous activity by gracefully interleaving speech and gestures commands across people in the construction of commands.

4. Case Studies

We developed hardware and software to illustrate the multimodal SVT concept. Descriptions of the final systems are provided here, with implementation details deferred to §5.

Figure 3. Split View over Independent Applications

The physical arrangement of our multimodal SVT implementation is illustrated in Figures 3-5. We decided upon the face to face seating arrangement (§2.2) as we were most interested in situations where people moved from loosely-coupled to tightly-coupled work involving co-acting and conversation. We chose an arms-length physical size (§2.3) to balance reach and availability. For input devices (§2.4), we refactored a multimodal speech and gesture recognition system [17]. Gestures were recognized through a DiamondTouch multi-user touch surface [2] as a direct touch device (§2.4), while speech was recognized through headset microphones (§3).

The case studies below show example implementations of all three software configurations (§2.1).

4.1 Independent Applications

Our independent applications configuration is appropriate when multiple people need to work independently over separate applications while still being aware of the actions of collaborators. Figure 3 shows two people planning a trip together by simultaneously browsing the web. Each runs a completely independent instance of the Mozilla Firefox web browser in their view. On the left (and in fuller view in the left inset), the woman is searching for Hotels using Google Maps. On the right the man is finding nearby attractions using an online Lonely Planet Guide.

We created a wrapper around Firefox that allows people to interact with it through gestures and speech. Gesture commands include one finger select, five finger pan/scroll, a two finger gesture to move back/forward or to open/close a tab, and two-handed region selection for highlighting a large region. Speech

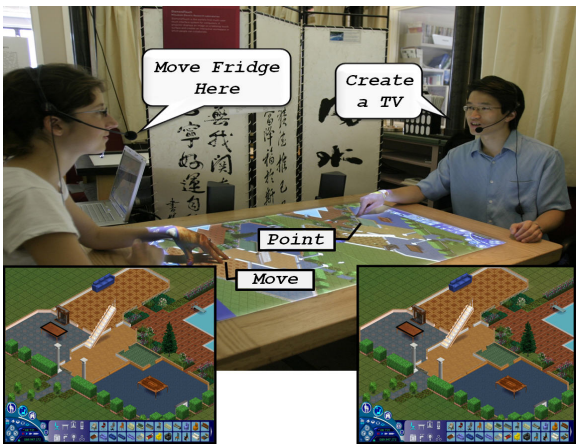

Figure 4. Screen Sharing using VNC and The Sims

commands include "bookmark this page", "go to [bookmark name]", "back", "forward", "home", and "search". The "open keyboard" command opens an onscreen keyboard for typing in URLs or searches. Gestures across the seam are allowed: if the man drags a web link from his browser across to hers, that page is automatically loaded in her browser.

In Figure 3, the woman uses a voice command to bookmark the current page and a one-finger gesture to select the detailed information box of a hotel. She is simultaneously conversing, where she tells the man that the hotel could be a good one. The man responds by glancing over at her selection, and then referring to an attraction in his view that is close by that hotel.

As shown by this example, this configuration is good for loosely-coupled work during a joint task, where people can occasionally bring attention to some of their activities, e.g., by having the other person view it. While joint viewing is easy, joint interaction is difficult as one person would have to reach over the table to access the other view.

4.2 Screen Sharing

The screen sharing configuration is appropriate when people need to work tightly coupled over the same view. Figure 4 shows two people interacting with a shared view of The Sims by Maxis, a single user simulation game that allows a player to create a virtual home and to control the actions of its inhabitants. As seen in Figure 4, the views of both people are identical.

We created a wrapper around The Sims specifically for a furniture layout task. Speech commands include "create [object]", "<1st/2nd>" floor" while gesture commands include one finger placement, five finger movement and one fist object stamping. There are also multimodal commands that require both speech and gesture, such as "create [object] [one finger point]",

Figure 5. True Groupware using Warcraft III.

which creates an object at the location being pointed to. This interaction is powerful, for statements like "create a table [one finger point]" not only commands the system, but provides awareness to other collaborators about their actions. This is necessary for people to interleave their actions through turn-taking.

Figure 4 illustrates how this works in practice. The man is indicating through speech that he wants to create a TV while using a gesture to point at the spot, while the woman is 'simultaneously' moving a fridge.

Of course, this shared view could have been implemented by having a single instance of the Sims appear across the entire surface. Yet the split view could be advantageous if multiple people want to work together without concern of one's body occluding the other person's work area [13]. As well, the split view means that each person can reach the entire space.

A concern with screen sharing is that true simultaneous interaction cannot be guaranteed. However, awareness minimizes conflict: when people know what the other is doing, they mediate their actions accordingly. Still, global actions can cause interference, e.g., if one person pans the map as another person is creating an object on a particular spot.

4.3 True Groupware

True groupware is appropriate when people need to work both in a loosely- and tightly-coupled manner, and when their combined activities reflect changes in the common workspace. Figure 5 shows two people interacting with Warcraft III, a groupware game originally designed for distributed Internet players. As seen in Figure 5, the views of both people into the game's map surface can differ. Yet both views share the same common game environment, so actions by either person occurs in both places.

Akin to other configurations, we map speech and gestures to keyboard / mouse actions. Gestures include one finger unit selection, one hand panning and two hand side bimanual selection [19]. Speech commands, some combined with gestures, include "[selection] label as unit #", "<move/attack> here [point]", "build [building]", and "stop". Players can also move troops across the seam by saying "move here [point]" and touching in their partner's view.

Figure 5 illustrates how this works as each person is pursuing duties on different parts of the scene. The woman is directing worker construction in one area through a speech/gesture action ('build a farm here [point]), while the man is directing troop actions in a region in a different area ('label as unit one [select region]). An overview map on the bottom left of each view shows the entire map surface, and immediately reflects each other's actions.

Leveraging distributed groupware to the split view setting is obviously powerful, for it allows simultaneous action in a relaxed-WYSIWIS setting. Of course, people can also align their views to achieve WYSIWIS, although this has to be readjusted during panning (as scrolling is not synchronized). Awareness support within the distributed groupware system, such as radar views and feedthrough of other's actions in the scene [6], is enhanced by seeing the speech and gestural acts of others. People can leverage gestures over the seam, e.g., the man selects units in his view, and moves them to the woman's view by saying "move here", and pointing to the woman's view.

4.4 Moving between Configurations

SVT can be configured so that people can switch to a more conventional tabletop mode that presents a single view onto the surface. Ideally, they should be able to switch between variations of the configurations above: this part of our software remains to be implemented, but is straightforward. For example, the travel planners in Figure 3 could move into tightly-coupled work over Google Maps by switching from the independent application mode to a shared screen mode. If they want to work even more closely together, they can switch it again so only a single view of Google Maps is shown across the entire table. Similarly, the Warcraft players in Figure 5 can move from a true groupware view into a shared screen or the single view configuration if they wish to move into tightly-coupled strict-WYSIWIS interaction. Alternately, one player can bring up a different view as an independent application, e.g., to look up a cheat sheet for Warcraft III on the Internet.

5. Implementation

Multimodal SVT interaction over existing single user applications is a new concept. Yet we were able to build our environment by repurposing various hardware / software systems at our disposal (Figure 6).

5.1 Hardware

Hardware comprises 2 touch-sensitive tables, 2 projectors, 2 computers, and speech input.

The Split View Digital Table. The digital table seen in Figures 3, 4 and 5 is called the Double DiamondTouch (Figure 6, row 1). We designed as a face to face (§2.2) and arms-length (§2.3) surface. The table's total size is 148 x 116 cm, comprising an active area of 98 x 65 cm and a 25 cm wide solid oak bezel. This active area is made of two smaller DiamondTouch surfaces [2] laser cut to produce almost no seam (or break) between the two surfaces. The oak bezel and its beveled edges provide a comfortable resting area for people's arms that would not unexpectedly perform an action on the display surface. The beveled corners were designed to prevent arm strain over extended use.

Projecting Multiple Views. We used two projectors (Figure 6, row 5) and two computers (Figure 6, left and right half) to top-project two views onto the table, one in each half. Our two 1024x768 LCD projectors give a total resolution of 1536x1024.

Switching. As described in §4.4, one can switch between a single large shared display from one computer to two separate views from two computers. To do this, we used a KVM (Keyboard, Video, Monitor) switch (Figure 6, row 5). Projector A is connected to the primary display for Computer 1, while Projector B is connected to the KVM switch. The KVM switch is connected to both Computer 1's secondary display and Computer 2's primary display.

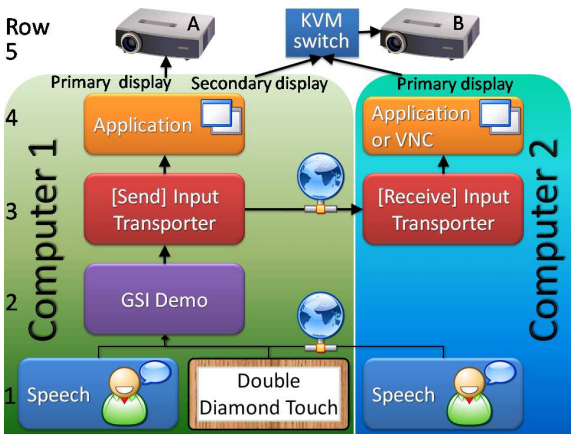

Figure 6. The SVT Infrastructure

Thus toggling the switch either displays Computer 1's contiguous view across the entire table, or Computer 1 and 2's view in the corresponding split views. The orientation of the projected image is adjusted so that it is the right way up for the seated viewer. This manual switching process is automated, where we use Phidgets [4] to programmatically control the KVM switch. A person uses simple voice commands "split view mode" or "shared view mode" to transition between the two.

Detecting touch input from multiple people. The DiamondTouch provides the necessary multi-user simultaneous touch input [2], as well as a reliable way to uniquely identify which person belongs to each touch. This is necessary to disambiguate who is doing what in our SVT environment. Both DiamondTouch surfaces are connected to a single DiamondTouch hardware controller board that has modified firmware to handle the larger board size. To the end programmer, the Double DiamondTouch appears as a single large DiamondTouch surface. All its input is sent to a single computer (Computer 1 in Figure 6), where a driver on that computer receives that input so that it can be used in application development.

Detecting Touch Input From Multiple People. We used the technique described in [17][18] to gather simultaneous speech input. Two Labtec LVA 7330 noise cancelling microphones are each connected to off the shelf speech recognition software. Recognized speech is then sent to a single computer for further processing (Computer 1 in Figure 6).

5.2 Software

Software is built atop GSI Demo [17], a system originally developed for a single display multimodal tabletop surface (Figure 6, row 2. With GSI Demo, people program by demonstration to map speech and gesture actions onto keyboard and mouse events for multiple people. After training, GSI Demo listens for people's speech and gesture commands, and invokes the appropriate mouse/keyboard counterpart.

Processing input. We consolidate the speech and gesture input of multiple people to a single computer so that we can process it more easily. This makes some tasks, such as turn taking management, much easier. Figure 6 shows the SVT software infrastructure behind this. Input from multiple microphones and the Double DiamondTouch (Row 1) are eventually received by the single computer using GSI Demo (Row 2). GSI Demo then plays back the appropriate keyboard/mouse actions as if the user had entered them [17] (Row 4).

As an aside, the original GSI Demo was developed for adding multimodal input atop a single display. In multimodal SVT, GSI Demo now mediates input and output from multiple computers. To do this, it uses the distributed shared dictionary data structure provided by the GroupLab Collabrary [1] to send and received speech and gesture input to the appropriate computer (Figure 3, Row 3). Events sent include: speech volume /recognition/hypothesis; gesture down/move/up events.

Mapping input to screen coordinates. The raw input coordinates of a gesture cannot be used directly, as input is provided in table coordinates. This differs from the screen coordinates for either computer. For example, the top left of Computer 1's display does not correspond to the top left of the table. To solve this, we created input mapping rules that convert gesture coordinates into screen coordinates. These rules consider the orientation, resolution and size of each display. In particular, the input transporter (Figure 6, Row 3) uses a configuration file to set the seating arrangement, display and software configuration. Using this information, all transported input is converted to screen coordinates for each computer using a simple linear transformation.

Mapping speech / gesture to keyboard / mouse events. To implement screen sharing, the input transporter creates a VNC-like system [11], where it sends screen snapshots captured by the GroupLab Collabrary [1] every 100 milliseconds to all client computers (Figure 6, Row 4). All input is serialized by the input transporter, and this is passed on as keyboard / mouse events to the computer running the single application. GSI Demo also includes several turn-taking protocols that mitigates interference when people try two work simultaneously [3][17][19].

For both true groupware and independent applications configurations, the appropriate speech / gesture map is loaded onto each machine. Input for each computer is then managed as if it originates from each of the DiamondTouch surfaces that correspond to a particular computer, i.e., input from a particular view is passed onto the appropriate computer running the groupware instance or independent application.

In all cases, the single user applications shown in Figure 6, Row 4 receive simulated keyboard and mouse events and are completely oblivious to the use of speech, gestures and transported input.

Handling input across the seam. The Input Transporter API lets a programmer map custom actions onto drag or touch actions across boundaries. For example, if two desktop systems are running, the programmer can create a mapping that detects if a file is dragged across the boundary, and then invoke a 'copy file' onto the desktop of the other computer.

6. Related Work

There is a long history of research in distributed groupware [5][6], shared screen systems [3][9], single display groupware [15], large digital tables and displays [2][12][16][20] and multimodal interaction [19]. However, several works stand out in regards to multimodal split view tabletops.

Within screen sharing context, turn-taking protocols in distributed shared screen applications [3] and more recently in the large display SDG setting [16][19] regulate and limit how one person can interfere with another's activity. Simultaneous interaction with existing applications can be simulated if commands are combined into unitary chunks and then interleaved with others. This has been done in various SDG systems using PDAs [10] and multiple mice [15].

There are other systems related to split screen tables [12]. Within the gaming world, there are a plethora of multi user console and arcade games that split a single screen, where each person works in their own view. Unlike our generalized solution, these games are typically implemented as special-purpose true groupware. Within office productivity world, many applications let its user split a document into two independently scrollable views. Almost all are limited to a single point of input, although Li et al. [9] produced an application that could leverage a split view for multi user simultaneous input. Sing, Gupta and Toyama (of Microsoft Research India, research.microsoft.com/research/tem/) split a single computer display vertically: each person independently works on their half using their own keyboard and mouse (equivalent to independent applications).

Finally, two separate personal devices can be connected into a single display. With Connectables [16], two people move their small tablet displays into close proximity: sensors notice this and combine them into a shared workspace [16]. Alternately, force sensors detect what sides of two tablets bump against each other, and use that information to combine and adjust the orientation of the view automatically [8].

7. Conclusion

We explored the design space of multimodal split view tabletop interaction over existing single user applications. We also offered a generalized approach to leveraging three types of existing software in SVT: independent views, shared screens and true groupware. We also added multimodal input as a way to promote awareness between collaborators. Three proof-of-concept systems illustrate how this works in practice.

We do not expect multimodal SVT to replace conventional single view digital tables. Rather, it offers alternate configuration that could be used as particular situations warrant it. Each method has its own benefits and tradeoffs that direct that choice.

8. References

[1] Boyle, M. and Greenberg, S. (2005) Rapidly Prototyping Multimedia Groupware. *Proc. Distributed Multimedia Systems '05*, Knowledge Systems Institute.

[2] Dietz, P.H.; Leigh, D.L., (2001) DiamondTouch: A Multi-User Touch Technology, *Proc ACM UIST'01*.

[3] Greenberg, S. (1990). Sharing views and interactions with single-user applications. *Proc ACM/IEEE Conference on Office Information Systems*, 227-237.

[4] Greenberg, S., Fitchett, C. (2001) Phidgets: Easy Development of Physical Interfaces through Physical Widgets. *Proc. UIST '01*, ACM Press, 209-218.

[5] Greenberg, S., Gutwin, C., and Roseman, M. (1996). Semantic Telepointers for Groupware. *Proc OzCHI '96*, IEEE Computer Society Press. 54-61.

[6] Gutwin, C., and Greenberg, S. (2004) The importance of awareness for team cognition in distributed collaboration. In E. Salas, S. Fiore (Eds) *Team Cognition: Understanding the Factors that Drive Process and Performance*, APA Press, 177-201.

[7] Hall, E. (1966) *The Hidden Dimension.* Anchor Books

[8] Hinckley, K. (2003) Synchronous Gestures for Multiple Users and Computers, *Proc. ACM UIST '03*.

[9] Li, D. and Lu, J. (2006) A lightweight approach to transparent sharing of familiar single-user editors. *Proc. CSCW '06.* ACM Press, 139-148.

[10] Myers, B., Stiel, H., and Gargiulo, R. (1998): Collaborations using multiple PDAs connected to a PC. *Proc ACM CSCW'98*, ACM Press. 285-294.

[11] Richardson, T., Stafford-Frasser, Q., Wood, K. and Hopper, A. (1998) Virtual network Computing. *IEEE Internet Computing*, 2(1), 33-38.

[12] Rogers, Y. and Lindley, S. (2004) Collaborating around vertical and horizontal large interactive displays: which way is best? Interacting with Computers (16), 1133-1152. Elsevier.

[13] Somer, R., (1969) *Personal Space: The Behavioural Basis of Design, Spectrum*, ISBN 0-13-657577-3.

[14] Stefik, M., Bobrow, D., Foster, G., Lanning, S., Tatar, D. (1987) WYSIWIS revisited: early experiences with multiuser interfaces. *ACM TOIS*, 5(2), 147–167.

[15] Stewart, J., Bederson, B., Druin, A. (1999) Single display groupware: A model for co-present collaboration. *ACM CHI'99*, 286-293.

[16] Tandler, P., Prante, T., Müller-Tomfelde, C., Streitz, N., Steinmetz, R. (2001) Connectables: dynamic coupling of displays for the flexible creation of shared workspaces. *Proc. UIST '01*, ACM Press, 11-20.

[17] Tse, E., Greenberg, S. and Shen, C. (2006) GSI DEMO: Multiuser Gesture / Speech Interaction over Digital Tables by Wrapping Single User Applications. *Proc. ICMI'06*, ACM Press. 76-83.

[18] Tse, E., Shen, C., Greenberg, S. and Forlines, C. (2007) How Pairs Interact Over a Multi*modal Digital Table. Proc. ACM CHI '07, ACM Pre*ss.

[19] Tse, E., Shen, C., Greenberg, S. and Forlines, C. (2006) Enabling Interaction with Single User Applications through Speech and Gestures on a Multi-User Tabletop. *Proc. AVI'06*, ACM Press. 336-343.

[20] Wigdor, D., Balakrishnan, R. (2005). Empirical investigation into the effect of orientation on text readability in tabletop displays. *Proc. ECSCW '05*.