ABSTRACT
Affinity Diagrams are an essential problem solving method that helps teams structure ideas and generate solutions. Collaborative affinity diagrams are created using paper sticky notes for input on a white board where groups and drawings can be made. Although many researchers have tried to leverage technology to support this practice, most live discussion support systems are based on SDG (Single Display Groupware). Some researchers have pursued distributed display solutions, but as yet they have been unable to produce a high usability system that supports this collaborative activity. In this paper, we introduce DADS (Digital Affinity Diagram System), which separates two essential tasks of affinity diagram creation: input and collaboration. Using a dual extended monitor system, DADS is designed for one screen to act as a private input screen while the extended monitor shows a distributed interactive screen for collaboration. Leveraging the network socket infrastructure popularized by multi-player online games, DADS turns each participant into a player with a unique identity in the discussion, and synchronizes all users’ activities across the distributed screens. This paper explores a system architecture that can solve problems in field of DUI, and presents design choices that promote high usability for collaborative affinity diagram creation.

Author Keywords
CSCW; Groupware; Affinity-Diagram; Distributed-Interface; Collaboration

INTRODUCTION
In this paper, we explore how multiple distributed screens can augment the creation of an affinity diagram in brainstorming activities. Affinity diagrams, also known as the KJ method, are a popular and well-researched discussion organizing tool [8]. They are often created by writing individual discussion points on sticky notes and, together with other participants, sorting the sticky notes to uncover the points’ similarities. The affinity diagram gives discussions structure and explicit goals, making them ideal for new groups working together. However, research shows that groups that work together to come up with ideas generate fewer ideas than individuals working alone who collaborate only upon completing the generation phase. [4, 9]

Pen and paper implementations of affinity diagrams present some productivity and psychological problems. Studies of production blocking[4] reveal a source of productivity loss: people take turns expressing their ideas and must wait for an opportunity to express their idea. In most group discussions it is not possible to express ideas immediately when they come to mind. Nijstad’s research [11] showed that this production blocking is harmful and interferes with the cognitive activities involved in generating ideas. Other possible psychological problem include fear of judgment by others [3], and social loafing, where participants rely on the group to generate ideas. It is possible to reduce this productivity loss by using computers for “electronic brainstorming” in which people can type simultaneously. There is evidence that digital tools can eliminate most of the effects of production blocking [5].

PREVIOUS RESEARCH
GUNGEN[10] was a pioneering system for utilizing technology for collaborative work, perhaps the first digital affinity diagram tool. Most systems focus on exactly replicating the physical constraints in a digital format, as in table-top Single Group Displays like Geyer and colleagues’ AffinityTable[6]. They utilize both digital pens and a large format multi-touch screen powered by Microsoft Surface, which allows them to use Surface Byte as a command to do grouping and linking. However, this research did not involve usability testing, and the novelty of the technology raises questions about how practical table-top SGDs are for common use. Similarly, Radle’s TwisterSearch [13] utilizes a tabletop SGD system for collaborative web search but allows distributed input from users’ tablets. Ballendat and colleagues’[1] Proxemic Brainstorm system developed a distributed screen for input and collaboration using a tablet and large touch-screen space. Like the two previous systems we discussed, Proxemic Brainstorm works with stylus input, with each user having a separate tablet for their own ideas. The synchronized system allows them to have distributed screens for input and manipulation. However, the system still needs an extra step of sending and receiving from the two screens, which do not represent true synchronization. Burtner et al.’s Affinity+ [2] is also a
distributed-input system centered on a SGD where all collaborative activity occurs. Gumienny and colleagues’ TeleBoard [7] allows remote teams to share space and integrate distributed devices as input through a digital projector. This system lets users draw on shared virtual space, create idea cards from their personal tablet or phone, and shows video of users in real time for better collaboration. Although these systems all allow distributed input of one kind or another, they are still focused on a monolithic shared display board, which we believe is problematic. Without a full-featured private creative space, users’ ideas will not be as high-quality or as well-supported as they could be.

PROPOSED SOLUTION

In response to the problems we have outlined above, we created a highly usable affinity diagram system that can support small group discussions. The design we used to create a system with these capabilities features full synchronization between distributed interactive screens using network socket infrastructure. This infrastructure transforms users into players and gives each user an identity and the ability to create, edit and manipulate own and others’ ideas to create an affinity diagram. The Digital Affinity Diagram system (DADS) separates private input and distributed interaction into two dual extended screens. Although the screens are separated, the extended functions of the screens do not require users to switch between the input interface and the manipulation interface. We chose a dual-screen design to increase information display clarity and focus for the users. While the main private input screen acts as each user’s personal input interface, the common interactive space allows users to focus on collaboration. Input screen allow users to create, read, update and delete (CRUD) for idea management, and ideas are also simultaneously displayed on the distributed screen. To allow easy manipulation of idea cards, the distributed screen is also equipped with multi-touch sensors which allow users to manipulate and collaborate using touch gesture methods. To support a 3-person discussion, we designed a U-shape seating arrangement that optimizes the dual monitor system and proximity between participants. In our 3-person experiments, a large monitor acts as a shared common screen visible by all participants. For the setup of the dual multi-touch monitor each user’s main laptop is located centrally, while the extended monitor is to their left side, giving space for the mouse on the right side. We chose to tilt the extended monitor to approximately 30 degrees, giving users the best access to the distributed common interactive space as a virtual ‘drawing board,’ and also allow eye-level contact for increased gaze awareness between users.

**Groupware Architecture**

The system’s groupware was developed using Simple Object Access Protocol (SOAP) architecture, which relies on the exchange of information through XML sent through a local HTTP protocol, and mainly developed under Microsoft’s .Net 4.5 framework. In addition, we also took advantage of the touch framework library that is currently maturing in Windows 7 and 8. This library allowed us to include a large screen multi-touch monitor in our solution. Users’ private input spaces were accessed on laptop systems running Windows 7, and the distributed common interactive space extended the screen with a 22” multitouch monitor. The multi-touch monitor uses a four-point infrared sensor to detect accurate multi-touch from users, and has an 8ms response time.

**Distributed real-time synchronization system**

To synchronize events and operations between servers and distributed clients, DADS relies on a client-server type of network socket infrastructure. The network socket service we choose is called Photon Engine, developed by Exit Games[12]. The network infrastructure used provides reliable low-latency data transport services on top of a user datagram protocol (UDP). Network socket can also be used as a Remote Procedure Call (RPC) system and server-side run-time for application logic. The socket also acts as a delivery medium for code execution and callback between multiple clients and servers. For example, if the client calls some code on the server side, the server-side code executes and may return some arguments back to the client. The benefit of the network socket is that it serializes the code logic by packeting it in small bytes of integers and creating a continuous call for event updates (between 10-50 calls per second) to simulate synchronous real-time behavior. Without Photon, we would have to deal with network packets directly, either using TCP/IP or UDP sockets. This would lead to an inefficient code execution stream, and increase latency in the system. Photon handles all instant notification needs, including: 1) All action updates created by users, 2) All operations on
the common interactive space editor (interactive user cursors, moving and editing any shapes. In sum, the Photon Engine server application will process all of DADS’ main functions and automatically update any changes made to the SQL server database in real-time.

**Private Input Interface**

The private input interface is designed to maximize visibility and navigation capabilities for users viewing both their own points and others’ points. Following network socket design standards, each discussion topic is treated as a separate room. Users are asked to give their name and select a color to represent their points on the common board. After identifying themselves, they choose a discussion room to participate in, and upon entering, each user can be identified by their chosen color. The private input interface is a workspace intended for users to create points by inputting points and information. Figure 2(A) shows the 3 column dashboard. In the top right hand, users can select different topics and switch from one topic to another. The middle column lists all the opinion titles written by each participant. The bottom right corner is where users select and view different participants’ points. The lower center column lists the titles of other participants’ points. Finally, the full screen right-hand column is where users construct their own points. Three distinct areas allow users to give each point a title, description, and attach source documents that support each point.

**Distributed Interactive Space**

In the distributed interactive interface, ideas written in the private input space are displayed as sticky-note-like cards with the author’s identity color. The card also displays the point’s title and the author’s name. These cards can be rearranged by touching and dragging them to a new location on the multi-touch screen.

We believe that successful collaboration requires the distributed common interactive space to be synchronized smoothly across different users. Users’ card manipulation commands are constantly sent to the server, and updates are broadcast to other clients. Low latency ensures the smoothness of the distributed interactive space’s control and manipulation responses. The following software design features were created to solve the challenges in DUI including concurrency conflicts and the pointing issue.

- **Vectored Free Draw** - To prevent concurrency conflicts in distributed screen, we designed the annotation drawing system to be a two-step vector system. When users first click or touch the draw button, it puts the screen into drawing mode (see Figure A) with a control interface (Figure B) on the bottom right allowing users to control the size of the pen and delete strokes from the drawing. Users can start annotating anywhere on the screen, but the annotation is not saved until users click the Finish Drawing button. It is then converted to a vector file allowing manipulation in the distributed screen. The two-step vector drawing process helps prevent concurrency issues, since individuals only control the annotation as a finished vector. At the same time, the system also makes organizing annotations in the distributed screen easier, increasing usability.

- **Grouping and Linking** - In a collaborative affinity diagram, grouping and linking cards organizes the structure of the group’s ideas. Grouping is done by using certain gestures on the multi-touch monitor. If a user presses the "Group" button and touches cards in the common interactive space with a circular motion between several cards,
a border around these cards is created and these grouped cards can be moved as a unit. Groups of cards can be re-organized by moving any card within the group. Inserting new cards into an existing group happens if any 2 corners of an ungrouped card touches the group. If this occurs, it initiates a regroup function that allows users to insert a new card into the group. Users can link cards or groups to each other by pressing the ‘Link’ button and clicking two cards in sequence.

Figure 5. Digital pointer with username and color ID helps users indicate an area of interest

- Digital Pointer - The system’s pointer function is designed to allow users to point to an object on the distributed screen. This is achieved by allowing the users to right click the mouse (on the laptop) or click the Pointer button (in the distributed interface) to show other users what they are referring to. This extra pointer capability helps users direct attention to a point of interest and steer the conversation to a particular topic. Based on our early tests, we discovered that this solution was quickly embraced as part of the discussion.

CONCLUSION AND FUTURE WORKS

This paper aims to show that by incorporating a distributed user interface design, the usability of affinity diagram groupware could be improved. In addition, some of the challenges that Distributed User Interface (DUI) faces in group collaboration can be addressed. Our solutions include solving the concurrency issue with the implementation of low latency socket architecture. The system’s performance shows that manipulation changes are created almost instantly without any delays that would decrease the system’s usability rating. Conflicts were solved through the lock and release algorithm, and finally the rest of the functions we discussed were designed to increase and enhance the creation of collaborative affinity diagrams using distributed screen methods. Further research will include a usability study that allows users to test the usability of DADS directly against another method.

ACKNOWLEDGMENTS

We would like to thank the Japanese Government (MEXT) for funding and grants for our research, and anonymous reviewers for their helpful comments on this workshop submission.

REFERENCES