

# Textual Annotation Tools for SAGE2

BY  
KRISHNA BHARADWAJ

THESIS

Submitted in partial fulfillment of the requirements  
for the degree of Master of Science in Computer Science  
in the Graduate College of the  
University of Illinois at Chicago, 2015

Defense committee:

Andrew Johnson, Chair and Advisor  
Luc Renambot  
G. Elisabeta Marai

## ACKNOWLEDGEMENTS

I would like to thank:

- My advisor Andrew Johnson, for showing me that hard work and learning are the only things that should matter to a student. I cannot think of a better mentor.
- My committee members Luc Renambot and Liz Marai for their support, guidance, and valuable input throughout the process of building my thesis.
- Pramitha and Ashwin for being the best of friends one can have. I cannot imagine even starting on this journey, let alone finishing it, without their help and support.
- Lance for having made my time at EVL invaluable with all the tiny little talks that we have.
- Jillian for being always ready to listen and offer valuable insights and for being a wonderful friend.
- Tommy, Arthur, and Victor for all the help and feedback they have given me.
- Maxine, Dana, Bob, Angus, Daria, Jonas, John, Kyle, Paul, Chihua, Shi, and the rest of the EVL team for always being supportive.
- My family members for their support and cooperation.

KB

# TABLE OF CONTENTS

<b>TABLE OF CONTENTS</b>	<b>iii</b>
<b>LIST OF FIGURES</b>	<b>iv</b>
<b>LIST OF TABLES</b>	<b>v</b>
<b>LIST OF ABBREVIATIONS</b>	<b>vi</b>
<b>1 INTRODUCTION</b>	<b>1</b>
<b>2 STATE OF THE ART</b>	<b>3</b>
2.1 Annotations on office tools . . . . .	4
2.2 Annotations for the web . . . . .	5
2.3 Annotations for hand held devices . . . . .	6
2.4 Bridging the gap between real and digital world . . . . .	7
<b>3 BACKGROUND: SAGE2</b>	<b>9</b>
3.1 Architectural overview . . . . .	9
3.2 Interactions in SAGE2 . . . . .	10
<b>4 METHODOLOGY</b>	<b>15</b>
4.1 Textual Annotation System . . . . .	15
4.2 Sticky Note . . . . .	16
<b>5 IMPLEMENTATION</b>	<b>17</b>
5.1 Implementation of Textual Annotation System . . . . .	17
5.2 Implementation of Sticky Note . . . . .	23
<b>6 USER STUDY AND EVALUATION</b>	<b>30</b>
6.1 Usefulness measure . . . . .	30
6.2 User study . . . . .	32
6.3 Results . . . . .	34
<b>7 CONCLUSION</b>	<b>45</b>
7.1 Future work . . . . .	46
<b>CITED LITERATURE</b>	<b>47</b>
<b>VITA</b>	<b>49</b>

## LIST OF FIGURES

3.1	SAGE2 web user interface showing a representation of the display . . . . .	11
3.2	Pointer modes: Window manipulation (left) and Deep interaction (right) . . . . .	13
3.3	PDF viewer and its user interface . . . . .	13
5.1	PDF viewer showing annotation window toggle button . . . . .	18
5.2	Annotation window . . . . .	18
5.3	Buttons to create notes . . . . .	19
5.4	Marker and note highlighted with pointer color . . . . .	20
5.5	Changing a general comment to specific comment . . . . .	20
5.6	Left click on the page moves to marker to the position of the click . . . . .	21
5.7	JSON annotation database structure . . . . .	22
5.8	Filename prompt for the sticky note . . . . .	24
5.9	Initial text prompt for the sticky note . . . . .	25
5.10	Sticky note instance created on the display wall . . . . .	25
5.11	Edit note in deep interaction mode . . . . .	26
5.12	User interface of sticky note . . . . .	26
5.13	Sticky note moves with the document . . . . .	27
5.14	Buffer update data . . . . .	28
6.1	Usefulness score of comments on image using Sticky Note . . . . .	35
6.2	Usefulness score of comments on image using Textual Annotation System . . . . .	36
6.3	Usefulness score of comments on multiple page document using Sticky Note . . . . .	38
6.4	Usefulness score of comments on multiple page document using Textual Annotation System . . . . .	38
6.5	Range of comments produced by SN (trials 1-5 on image and trials 6-10 on document) . . . . .	39
6.6	Range of comments produced by TAS (trials 1-5 on document and trials 6-10 on image) . . . . .	39
6.7	Overall usefulness scores of TAS and SN for image and multiple page document . . . . .	40
6.8	Average usefulness scores of TAS and SN for image and multiple page document . . . . .	41
6.9	Ease of use scores for TAS and SN . . . . .	41

## LIST OF TABLES

2.1	Comparison of different annotation tools . . . . .	8
6.1	Context value in usefulness measure . . . . .	31
6.2	Assignment of annotation tools in trials . . . . .	34
6.3	Useful comments out of total comments . . . . .	34
6.4	Annotation tool preference of all 20 participants . . . . .	42
6.5	Difficulties in using Textual Annotation System . . . . .	42
6.6	Difficulties in using Sticky Note . . . . .	42
6.7	Suggestions to improve Textual Annotation System . . . . .	43
6.8	Suggestions to improve Sticky Note . . . . .	44
6.9	General comments on the annotation tools . . . . .	44

## LIST OF ABBREVIATIONS

<b>AOI</b>	Area of Interest
<b>CGLX</b>	Cross Platform Cluster Graphics Library
<b>EVL</b>	Electronic Visualization Laboratory
<b>GNU</b>	GNU's Not Unix!
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>ID</b>	Identifier
<b>JSON</b>	Javascript Object Notation
<b>LLAH</b>	Locally Likely Arrangement Hashing
<b>MS</b>	Microsoft
<b>PC</b>	Personal Computer
<b>PDF</b>	Portable Document Format
<b>RDF</b>	Resource Description Framework
<b>SAGE</b>	Scalable Adapative Graphics Environment
<b>SN</b>	Sticky Note
<b>TAS</b>	Textual Annotation System
<b>UI</b>	User Interface
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>XML</b>	Extensible Markup Language

## SUMMARY

The amount of data that we deal with, on a daily basis, is ever increasing and coupled with the fact that display technology is getting cheaper by the day, this has led to the rise of large/scalable resolution display systems. With more and more people adopting large/scalable resolution display systems in their work places, these display systems are being used for data vizualization as well as collaboration.

The complexity of data being visualized and the nature of collaborative tasks being carried out using large/scalable resolution displays are changing the way we use these displays. It is no longer enough to passively look at data on these displays. There is a growing need to actively engage and interact with the content. As such, large/scalable resolution display systems that handle large amounts of data need to provide adequate functionality to interact with this data. SAGE2 comes as an answer to this. It allows scalable displays to effectively act as platforms for data visualization and at the same time gives these systems the power of multiuser interaction. I propose two textual annotation tools to annotate documents and images in SAGE2. These annotation tools will further the capabilities of SAGE2 in terms of allowing users to collaborate. Through this thesis, I will evaluate these tools in the hopes of gaining insight about their usability, affordances, and limitations.

## 1. INTRODUCTION

Annotating documents is a common practice that has several and varied use cases. In a collaborative scenario annotations are often used to share, thoughts and ideas about documents with others. Once limited to handwritten and printed media, the practice of annotating documents has become commonplace in the digital world as information is increasingly digitized. There are quite a few tools available today to facilitate annotating documents, both stand-alone applications for desktop environments and web based applications for annotating content on the web.

Lately, there has been an increase in the use of Scalable Displays at workplaces not just as passive screens, but as active workspaces that enable collaboration. SAGE2 [1] is a middleware that drives Scalable Displays to provide a collaborative(collocated and remote) environment. SAGE2 allows multiple users to simultaneously engage with the content. Documents and images make up a large part of the content that goes on to these displays, hence the ability to annotate them is a much needed functionality in SAGE2.

“Annotation” is a broad term that is used to describe a variety of textual, graphical, and even audio metadata created and used to augment a piece of data(this data itself could be some text, image, video, or audio). However, in the context of this thesis, it must be noted that, the term “annotation” is used to denote a small piece of text, associated with a document or image in some way. The terms “note” and “comment” are also synonymously used throughout this document to refer to such a piece of text. I have developed two Annotation tools for SAGE2. These tools facilitate the users to add annotations to documents and images. One of the tools, that I have termed “Textual Annotation System”, provides an integrated view of textual annotations with the document or the image being annotated. It provides users with the ability to associate an annotation with a specific portion of a document or an image with the help of a marker. The other tool that I have developed, called “Sticky Note”, is a note taking tool. It lets the users create notes which when placed on a document or an image, stick to that item. The notes stick to the underlying document or image for the duration of that item’s presence on the display, unless the notes are moved from the item. As part of my Master’s Thesis, I have evaluated these two tools in a user study to assess their usability for annotating documents and images. Apart from trying to find out the general effectiveness of the tools, the user study was focused on questions such as:



1. Will the effectiveness and user satisfaction of either tools differ between annotating a multiple page document and annotating an image?
2. Are there any aspects of these tools that will enhance the task of collaboratively annotating documents?
3. Are there any aspects of these tools that are counterproductive to the task of collaboratively annotating documents?

## 2. STATE OF THE ART

The practice of annotating digital media has been around for quite some time now. Very early in the history of digital media, researchers recognized the importance of annotation in collaboration and information sharing. Today, several annotation tools have been developed and researchers are examining their usability and application to collaborative problems. Some of the important parameters that guide the design of annotation systems are:

- Architecture of the tool.
- The nature of media being annotated.
- Devices for which these annotation tools are to be developed.
- Types of annotations that are supported.

Even though there is a good number of annotation tools available today, none of them extend to large display or scalable resolution display environments. In the recent years, several systems have been developed which provide an interactive environment for managing large/scalable resolution displays. While systems like OmegaLib [2] and the Cross Platform Cluster Graphics Library (CGLX) [3] provide the ability to visualize large volumes of data in a single application to be run on large/scalable resolution displays, systems like SAGE [4], Liveboard [5], Impromptu [6], CubIT [7], Mezzanine [8], Montage [9], DisplayCluster [10] are attempts at creating collaborative environments using large/scalable resolution environments. If we look at these existing large/scalable resolution display environments, the reason for the lack of any annotation tools becomes apparent. All of these systems allow the users to interact with contents on the display in a very limited capacity. The interaction in these systems is limited to opening for viewing, moving, resizing, and closing of the few applications and media elements that these systems support. In SAGE2 we call this type of interaction as “window manipulation”. SAGE2 on the other hand has another mode of interaction where the user events directly affect the state of an application. We call this the “deep interaction” mode. We will look at these two modes in a bit more detail when we cover the interactions in SAGE2 in the next chapter. It is this deep interaction mode that has made the

creation of annotation tools possible in SAGE2. This chapter looks at some of the existing annotation tools and Table 2.1 compares these tools against each other.

## 2.1 Annotations on office tools

### Microsoft Office

Microsoft Office provides commenting features to comment on documents, spreadsheets, and slides [11]. Primarily, the idea is to select a piece of text (in case of a document) and using commenting functionality built into its interface, a new comment can be created for this selected text. This creates a “dialogue bubble” into which text can be inserted. MS Office also allows audio comments and ink comments (users write in free hand inside the comment bubble using touch or pointing device) to be inserted in a similar way. The comment then becomes a part of the document. This document can be shared with others as any other file would be shared. Comments are identified as originating from a specific user via the name and initials used by that user as part of personalizing the MS Office tools. Comments to comments or, in other words, replies to comments can also be created by selecting a previously made comment instead of a piece of text while creating a new comment.

### Google Docs

Users can create textual comments in Google Docs [12]. A piece of text is selected and a comment is inserted. However, the key distinction of Google Docs comes in the form of its affordance of sharing a document for editing amongst multiple remote users simultaneously. These different users (when given proper access rights to) can all add comments to the document, apart from editing the document itself. If a user wants to ensure that a specific user sees a comment, he/she can address the comment to that user by including their email address within the comment. This sends an email notification to that user.

### Adobe Reader

Adobe reader provides tools to add textual comments in the form of sticky notes, highlighting options, and drawing markup tools for pdf documents [13]. Any part of the pdf document can be selected and a sticky note or a highlight can be added. The highlighting feature also supports adding a note to it. The drawing markup tools lets users create rectangles, circles, and other simple geometric figures as part of annotations of the document. These markup tools also allow free hand drawing (also known as ink annotations) on the documents. Adobe reader also provides search and filter functionality using which users can search for comments with specific keywords in them.

## 2.2 Annotations for the web

### **Annotea**

Annotea [14] has been developed as a service, rather than a fully developed tool. It uses Open RDF servers with HTTP for client server communication. It does not provide or insist on a client UI and expect users of Annotea to develop their own. Annotea is aimed at collaborative annotation of content on the web. An annotation server administrator may choose whom to give access to the server. When the user browses a web page, the client contacts all the annotation servers in its list and obtains lists of metadata associated with the URI of the web page. Using XPointer data in the metadata, appropriate parts of the document are highlighted. If and when the user clicks on a highlighted portion of the text, the body of the annotation corresponding to that portion is downloaded and along with the metadata, the content is shown in a separate browser window.

### **MADCOW**

MADCOW - Multimedia Annotation of Digital Content Over the Web [15], is a tool for annotating documents, images, videos, and audios on the web. It has a client-server architecture and is inspired by the Annotea architecture. A MADCOW server is an annotation server and a client is a plug-in for a browser. Annotations are produced and stored using XML format. Users can retrieve and filter annotations from different servers. Users can choose to save their own annotations on any of the available servers. By making the client as a plugin MADCOW hopes to integrate the process of annotating with general web browsing in a natural way without forcing the users to use particular browsers, or disrupt the flow of navigation in order to launch an annotation application.

### **iAnnotate**

iAnnotate [16] is a multi user annotation tool that lets users annotate content on the web by using digital ink. It uses Microsoft's Silverlight for the support it provides for digital ink functionality. Annotations created using iAnnotate can be saved, accessed, and shared amongst other user by URLs. iAnnotate assumes the content being annotated will change over time and when the content changes the anchoring of the ink annotations laid on these contents are recomputed to change the positions of the annotations accordingly.

### **ImaNote**

ImaNote - Image Map Annotation Notebook [17], allows multiple users to simultaneously view, zoom, and annotate arbitrary sized images using a web browser. It is an open-source tool distributed freely

through a GNU license. Initially the tool was developed as a social media tool aimed at cultural heritage sector. However, it allows any user to save and share annotations to any image. In ImaNote, the image being annotated is the origin for the communication and knowledge that is generated in the form of annotations. Annotations are presented in electronic post-it notes. Annotations are represented in the form of black and white rectangles and when the user places the cursor over the rectangle, their corresponding post-it notes appear next to the rectangles in a details-on-demand fashion. Users can add textual notes and also include URL to link external files as annotations thus allowing images, audio, and video files to be added as annotations.

## 2.3 Annotations for hand held devices

### Notes in Kindle

Kindle provides commenting facility which can be used to add notes to any digital books on kindle [18]. Much like MS Office and Google Docs, a piece of text(a word or a phrase) is selected by the user. This brings up a panel from which the user can either choose to just highlight the selected text or add a note to it. If the user chooses to add a note, then an editable area is presented along with an on screen keyboard, where the user can type in the note and save it. Once a note is saved, a small marker is placed near the text that was selected, to indicate that it has a note associated with it. Kindle also has the notion of public notes, where sections of the text highlighted and commented upon by many users and thus having gained “popularity” can be viewed along personal notes. Users can link their kindle to social media platforms like Facebook and share their notes and highlights with their friends.

### BuddyBooks

BuddyBooks [19] is a real time application for mobile devices that facilitates collocated collaborative reading of documents by allowing users to make annotations on documents using touch. Each user has a copy of the document on their personal slate PCs or mobile devices on which they can create annotations. The server holds a copy of the document and receives information about annotations made from each user and distributes this information to all users so that they can view annotations made by other users. Copy of the document held by the server allows it to accurately provide all the necessary annotation information to any late joiners too.

### MoViA

Using MoViA [20], users can create and share annotations over pre recorded videos while watching them on their mobile devices. It is an Android application and can be run on mobiles devices of any screen

size. Users can create textual, audio, and digital ink annotations. These annotations are temporal in nature, meaning the tool links any annotation created by the user to the point where the video has progressed to at the time of the creation of annotation. Annotations may be shared between users by using existing application on the mobile device such as gmail, dropbox, or any other application. Thus, the users can also control whether the annotations are shared to specific users or publicly.

## 2.4 Bridging the gap between real and digital world

### Share Me

An annotation tool that allows users to make digital annotations on physical documents. It uses document image retrieval (Locally Likely Arrangement Hashing or LLAH technique) and augmented reality to add textual notes and highlights to paper documents with the help of tablets or smart phones [21]. The system allows these annotations to be shared with others. A web interface is also provided which facilitates the users to see digital copies of the physical documents and the annotations made on them via a web browser on desktop machines or on smart phones. Essentially, a user while using this tool holds the smart phone above a document and the document, as seen by the camera of the smart phone, is sent to the server for processing. The server responds with any previously made annotations that are overlaid on top of the image of the document in the smart phone. Different annotations are presented in different color for convenience.

### Memory Specs

Memory Specs [22] is an annotation system that has been developed for Google Glass. It uses document image retrieval (Locally Likely Arrangement Hashing or LLAH technique) to create annotations on ambient text that is present in the form of posters and sign boards. The ambient text in this case needs to be registered beforehand with the server, or on the go by taking a picture. It allows users to attach textual notes, images, and audio notes to the ambient text. Users may later access all annotations (along with the digital representations of the ambient text that was annotated) that they created in addition to publicly available annotations from their friends using a web interface.

Tool	Nature of annotated media	Target device	Type of annotations	Annotation sharing mechanism
Microsoft Office	Documents, spreadsheets, and slides	Desktops, tablet PCs, and hand held devices	Textual, audio, and ink	Conventional file sharing
Google Docs	Documents, spreadsheets, and slides	Desktops and browser enabled devices	Textual	Online sharing with simultaneous access
Adobe Reader	Documents	Desktops, tablet PCs and hand held devices	Highlights, geometric shapes, ink, and textual	Conventional file sharing
Annotea	Web content	Desktops and browser enabled devices	Highlights and textual	Online sharing
MADCOW	Web content	Desktops and browser enabled devices	Highlights and textual	Online sharing
iAnnotate	Web content	Desktops and browser enabled devices	Ink	Shared via URLs
ImaNote	Images	Desktops and browser enabled devices	Textual and URLs	Online sharing
Notes in Kindle	Documents	Kindle	Highlights and textual	Shared through social media
Buddy Books	Documents	Hand held devices	Highlights and textual	Online sharing for collocated groups
MoViA	Videos	Hand held devices	Textual, audio, and ink	Conventional file sharing, cloud, and over social media
Share Me	Images of physical documents	Hand held devices	Highlights and textual	Online sharing
Memory Specs	Images of ambient text	Google Glass	Textual, images, and audio	Online sharing
TAS for SAGE2	Documents and images	Large/Scalable resolution display walls	Textual	Simultaneous access for collocated group
SN for SAGE2	Documents and images	Large/Scalable resolution display walls	Textual	Simultaneous access for collocated group

Table 2.1: Comparison of different annotation tools

### 3. BACKGROUND: SAGE2

Before looking at the functionality, affordances, and limitations of annotation tools built for SAGE2, it becomes imperative to understand what SAGE2 is and take a look at some of the features of SAGE2 that makes it possible to build annotation tools for it.

SAGE2 is a middleware that manages scalable resolution display systems. It is a multiuser, collocated as well as remote, collaboration environment. SAGE2 can be used to have collaborative sessions in which data in a variety of media(pdf documents, images, and video) could be displayed and interacted with. SAGE2 provides functionality to build custom applications for creating specialized data visualizations. These custom applications could be instantiated and interacted within a SAGE2 session. Some of the common scenarios where SAGE2 has been used quite successfully include classroom sessions, project group meetings, and weekly lab meetings. In such scenarios, typically all the participants of the meeting use the display space to display their documents and images to everyone in the room. Everybody is able to interact with the contents on the display through web pages from their own personal devices. SAGE2 also allows participants to share their desktop on the display, giving another way to share information with the rest of the group. SAGE2 has many more features in its repertoire, but in this chapter we will take a look at a subset of these features, such as display of multimedia content and interaction mechanism in SAGE2 as these features are pertinent to the discussion of annotation tools in SAGE2.

#### 3.1 Architectural overview

SAGE2 has a client server architecture, where the server talks to a variety of clients, like display clients, an audio client, and input clients. The display clients are run on systems that drive the scalable resolution display systems. In order to manage scalable resolution displays, the display client assumes that it is a part of a larger canvas being drawn across the entire matrix of displays. Multiple instances of such a client are run covering the entirety of the display wall. Each client instance displays its own “portion”. SAGE2 server is responsible for synchronizing these individual display client instances with one another to provide a seamless windowing environment. Audio client is responsible for managing the audio of, and synchronizing the streaming of video data across the multiple display client instances. Input clients



are of a few different types. For the purposes of this discussion I would like to focus on a particular input client that functions as the web user interface (shown in Figure 3.1) to interact with items being displayed by the display clients. This interface provides users with the ability to put documents, images, and videos up on the display, it allows them to put instances of custom SAGE2 applications on the display as well. It also serves as an interface for interacting with these items on the display wall. The SAGE2 server itself is a bundle of modules that functions as a middleman between the display clients and the input clients. The various functionalities of SAGE2 have been made into modules within the server that coordinate with each other to make SAGE2 work.

### 3.2 Interactions in SAGE2

In a typical SAGE2 session, a user navigates to the url of web user interface using a web browser. Desktop and mobile versions of the interface let users connect using a variety of devices. The interface itself is as shown in Figure 3.1. It has a representation of the display wall and a few controls. Apart from showing the items present on the wall, this representation area also serves two functions:

- It acts as a drag-and-drop area where users can drag a pdf or other multimedia files present on their device and drop them on this area. This will upload the file to SAGE2 server and the server will in turn send a copy of the file to the display clients where the file will be displayed.
- It acts as a window manipulation interface. That is, a user can click and drag any icon (of an item being displayed on the display wall) on this area to move the corresponding item on the display. Other actions like pinching gesture on the trackpad (equivalently scrolling using the mouse) to resize the items, are also possible by interacting in this area.

The web UI also has a number of controls that allows the user to perform a variety of operations such as:

1. Create a pointer on the display wall which can be controlled using the mouse or trackpad of the user's device.
2. Share user's device screen on the display wall.
3. Launch custom applications that are in the repository of the server.
4. Browse through the media files available on the server and select any file to be displayed on the display wall.
5. Tile the contents on the wall, save the session, and clear the wall.



Figure 3.1: SAGE2 web user interface showing a representation of the display

6. Change settings like pointer color and label, change the resolution and frame rate of the screen share.
7. Display help and other information regarding how to run SAGE2.
8. Create a new sticky note instance on the display wall.

### Interaction modes

In a typical single user system, when multiple windows are being displayed, one item will be “in focus” or “active”, meaning that window will be receiving input from the input devices thereby allowing the user to interact with that one window. If the user wants to interact with another application, this is established typically by clicking on this different application window to shift the focus. Also, moving and resizing of application windows is carried out by interacting with specific parts of the window such as the title bar (to move) and edges of the window (to resize).

SAGE2 is a multiuser system. One of the implications of that statement is that the notion of having one item (or window) from amongst several items on the display to be “active” cannot be supported. Unlike single user systems where the mapping between user inputs and application windows is one to many, in SAGE2 it is many to many. Thus an “active” window for one user, will be just another window for a different user. Moreover, since SAGE2 is aimed at display walls of varying sizes and resolution and given the possibility of using touch input to interact with the items on the display, using a particular region of the window like the title bar for moving it is not a feasible solution (title bar might not be within reachable height for the user using touch). This led to the idea of pointers having two different modes, instead of windows being “active” and “not active”. Figure 3.2 shows these two pointer modes. The two different modes are:

1. Window manipulation mode: When the user creates a pointer on the display wall using the control on the web UI, it is in this mode. In this mode, pointer uses the same metaphors as that of a conventional desktop pointer. That is, the user can use the pointer to click and drag on an application window to move it anywhere on the display wall. The user can also resize the window by clicking and dragging on a small area in the right bottom corner of the window or by scrolling when the pointer is above the window.
2. Deep interaction mode: After creating the pointer, user can change the mode of the pointer from window manipulation to deep interaction by using a unique key combination(Shift + Tab) on the user’s keyboard. This is visually indicated by a small ring at the tip of the pointer as shown in Figure 3.2. In this mode, the user will not be able to move or resize the application windows. Instead, the user’s input is sent to the application itself. This lets users to interact with the applications. For example, we have created a custom SAGE2 application for viewing google maps. In the deep interaction mode, scrolling using the mouse wheel on the user’s device will change the zoom level of the map and click-and-drag is used to pan the map.

### **Application interaction**

SAGE2 has a common API in the form of a set of callbacks. One of the callbacks handles events. Whenever a user’s pointer is in deep interaction mode and within the boundary of an application window, any input from the user’s input devices is translated into event by the server and this event is sent to the corresponding application via the event handling callback. The application, in the callback, decides what to do with the event that was generated. If we take the example of google maps application, pressing the key ‘m’ on the keyboard changes the map type of the application. Apart from direct keyboard and mouse interactions, a user can also interact with the application through its user interface. Every



Figure 3.2: Pointer modes: Window manipulation (left) and Deep interaction (right)

SAGE2 application can avail a set of controls (buttons, a slider, and a text input control) that can be used to provide additional ways to interact with the application. These controls are visualized in a radial fashion with the slider and/or text input if any being displayed on the right side of the radial UI. The UI is not spatially bound to the application and is freely movable, like other items on the display, over the entire display area. When a user interacts with these controls, the interactions are in turn translated into events by the server and sent to the application via the event handler. Going back to the same example of google maps, the zoom level of the map can also be adjusted by using the slider in the UI of the application.

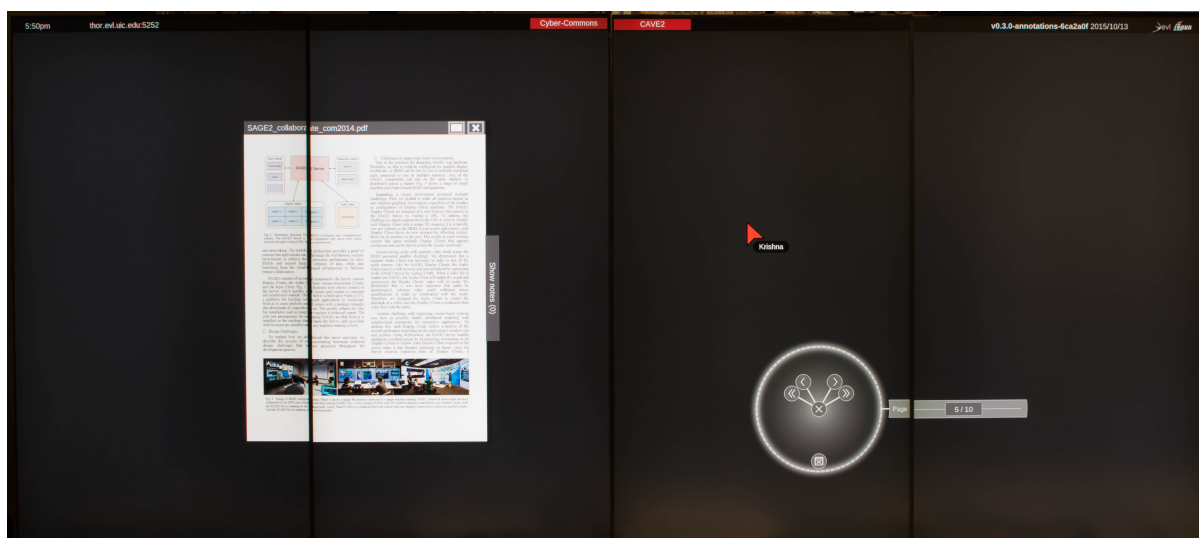


Figure 3.3: PDF viewer and its user interface

SAGE2 has a set of applications used to view multimedia content like documents, images, and videos. Of these, we are particularly interested in the document and image viewer applications for the discussion on annotation tools since these tools will be used to annotate documents and images. SAGE2 supports documents of pdf format and images of jpeg, png, and webp formats. As is the case with all applications in SAGE2, the image viewer has a title bar and the application window. It displays the image over the entire window of the application. The document viewer also has a title bar and an application window where it displays one page of the pdf document over the entire window. That is, only one page of the document is displayed by the viewer at any given point. A user can either use the left and right arrow keys on the keyboard in deep interaction mode or use the “back” and “next” buttons on the application UI to navigate to the previous or next page of the document respectively. Additionally, slider on the UI can be used to navigate to any page of the document. Figure 3.3 shows a pdf viewer (displaying a pdf document) and its application UI.

## 4. METHODOLOGY

The designs of Textual Annotation System and Sticky Note are significantly motivated by the user interface aspects and hence the design and implementational details go closely hand in hand.

These two systems were developed to address real needs that were felt when we started using SAGE2 in scenarios such as teaching a class, Image of Research contest (held at UIC to pick the best Images that represent some aspect of research being done at the university) and so on. During class, it is a common practise to push pdf documents and images pertinent to that session's discussion onto the display wall. Over the duration of the class, these items on the wall are discussed about, much like parts of a printed book are discussed about. The need to be able to make comments on the material being discussed became apparent. To have these comments all at one place and be easily accessible with the source material. This was the motivation for building a dedicated annotation tool (TAS) that provides for easy addition and manipulation of notes to documents and images.

SAGE2 enables users to create custom applications by providing APIs and this has led to the development of a variety of applications besides document and image viewers. Over the course of interacting with SAGE2 walls people expressed their desire to have the ability to attach a bit of text to the items on the display. Image of Research contest mentioned above was one of the scenarios where this need was felt. The judges judging the images expressed the need to be able to attach text to the images to indicate rank or preference of images. The requirements in such cases closely aped that of a real world sticky note. Ability to easily attach a piece of text to a variety of content. Due to the lack of any such tools, actual paper sticky notes were used in the Image of Research contest to score the images. These paper notes were placed on the floor in front of respective images on the display. This was the motivation to develop a note taking tool (SN) that lets users to easily create notes and associate them with on display items.

### 4.1 Textual Annotation System

Textual Annotation System (TAS) provides a comment window next to the application window of pdf documents and images. This window lists all the comments that have been made about the respective

pdf or image in latest-at-the-top order. Multiple users can simultaneously interact with this window to create new notes, and edit or delete previously created notes. When we want to make a comment about a document or an image, usually that comment will be in association with an area of interest within the document or image. This area of interest could be a small portion such as a paragraph on one of the pages in a document or a specific location on an image or it could be the entire document or image itself. Based on this very broad classification of comments according the scope of their area of interest, TAS facilitates users to create two different types of comments, namely “specific comment” and “general comment”. A specific comment is a piece of text that has a marker associated with it, that the user can place anywhere on the document or image to indicate the area of interest with which the comment is associated. A general comment is just a piece of text and does not come with a marker. This second type of comment is used in cases where the comment being made is not specific to any particular area of interest within the document or image, but applies to the whole document or image.

## 4.2 Sticky Note

Sticky Note (SN) provides a text editor window that the user can directly type into. Application user interface of SN allows a user to control a few aspects of the note being edited, such as the filename of the note (note is saved as a text file on the server) and the font size for displaying the text. Apart from the functionalities of a text editor, this tool also has a special feature as the name suggests: it is sticky. This means that notes created using SN can be physically attached (stuck) to other windows on the display wall. This stickiness mimics the stickiness of a real life note. In other words, it can be detached from one window and attached to another.

## 5. IMPLEMENTATION

SAGE2 has a modular structure. One way to look at it is, as a collection of functionalities. Since SAGE2 has a client server architecture, functionalities of SAGE2 are split between the server and clients. The server acts as a manager and is largely responsible for data flow as well as decision making, whereas the clients act as interfaces with the user. This chapter looks at the implementation of Textual Annotation System and Sticky Note.

### 5.1 Implementation of Textual Annotation System

Going with the modular structure of SAGE2, I developed a module called Annotation Manager for realizing Textual Annotation System. This module has both server and client counterparts.

#### User interface

The user interface of the Textual Annotation System is the client counterpart of the Annotation Manager module that is discussed in the next section. It is responsible for creating the annotation window next to the application (document viewer or image viewer) window and populating it with all the comments of the respective document or image. The set of features provided by TAS to the users is given below:

1. Show/Hide annotations: When a pdf document or an image is displayed on the display wall, Textual Annotation System creates a vertical button (shown in Figure 5.1) on the right hand side of the application window. This button serves as a toggle button. It can be used to show or hide the annotation window when the window is hidden or visible respectively. The annotation window itself houses all the notes in reverse chronological order. Each note is placed in a separate editable section within the window as shown in Figure 5.2. If a note is specific, then a small Marker with the note's ID on it, is placed at appropriate location on the document or image. However, since the document viewer displays only one page of the document at a time, only the markers placed on the current page are made visible.
2. Add a new note: At the top of the annotation window, two buttons are provided. One for creating a specific note, and a second button for creating a general note as shown in Figure 5.3. When the



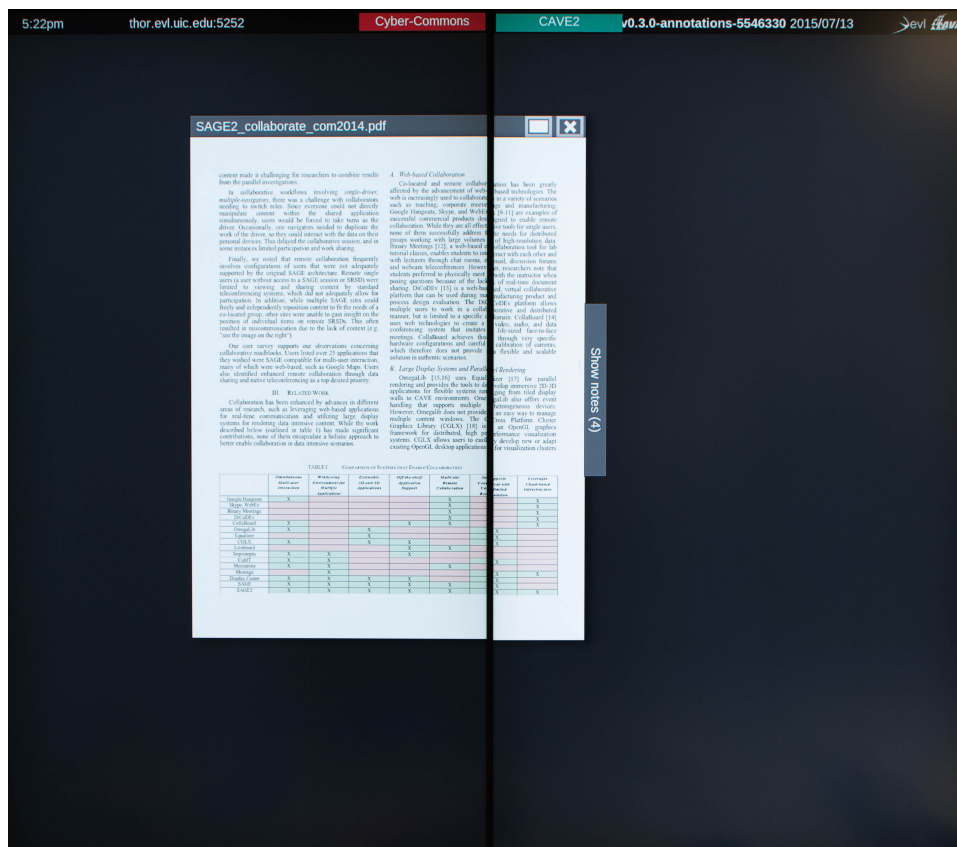


Figure 5.1: PDF viewer showing annotation window toggle button

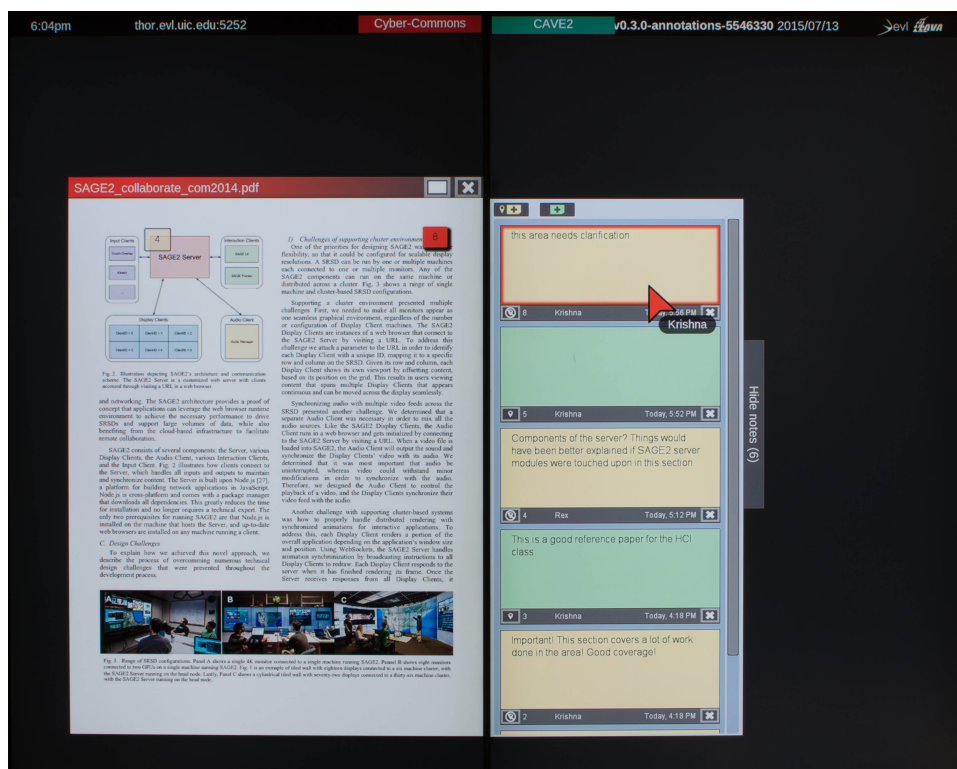


Figure 5.2: Annotation window

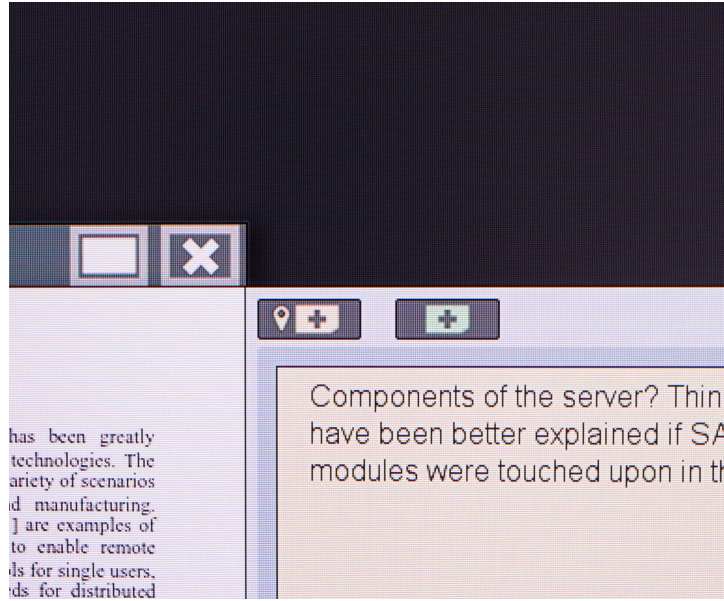


Figure 5.3: Buttons to create notes

user clicks on the general note button, a new editable section is added to the annotation window. When the user clicks on specific note button, in addition to an editable note section being added to the annotation window, a marker corresponding to this note section is also placed on the left top corner of the image or the current page of the document. In order to make the difference between a general note and a specific note stand out, the note sections of general and specific notes are colored differently (light yellow for specific note, and pale green for general note).

3. Edit note: When a user clicks on one of his/her notes, it is highlighted with a border in the color of the user's pointer to signify that it is currently being edited as shown in Figure 5.4. Also, to draw the user's attention towards the marker of the note being edited, the marker is displayed in the same color as that of the user's pointer.
4. Add/remove marker: A user can convert a general note into a specific note and vice versa easily with the click of a button. Each note section is provided with a "toggle marker" button that is placed at the left hand side below the note (shown in Figure 5.4). By clicking this button for a general note, a marker can be added to it thus converting it into a specific note. The newly added marker appears on the top left corner of the image or the current page of the document (shown in Figure 5.5). In case of a specific note, clicking this button will remove the marker associated with the note, from the document or the image.
5. Move marker: A user can change the position of a marker of a note that the user is currently editing by making a click at the new position on the document or image where the marker needs

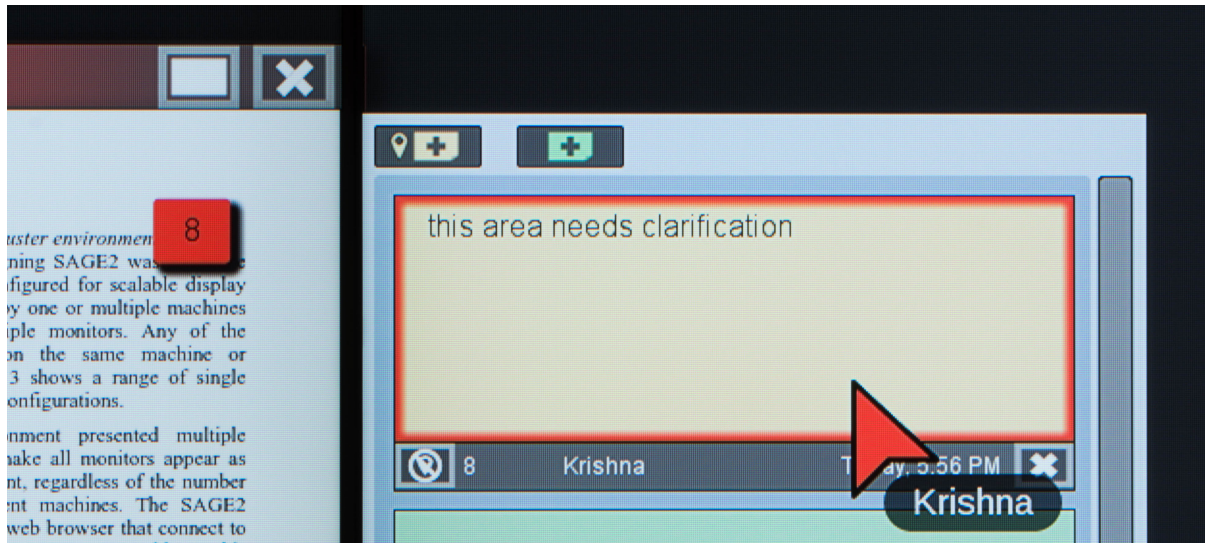


Figure 5.4: Marker and note highlighted with pointer color

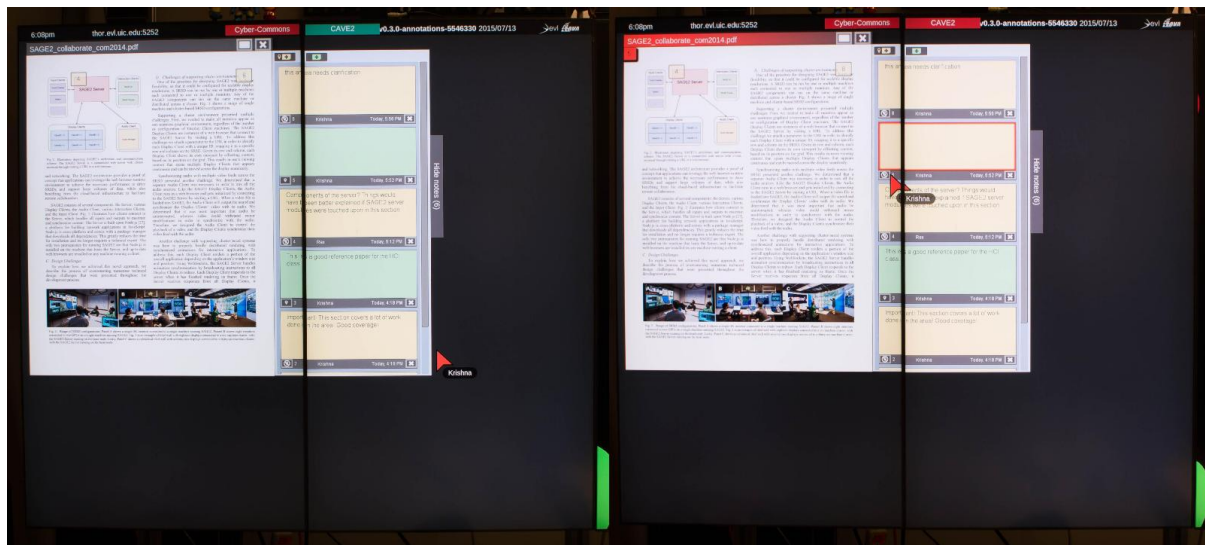


Figure 5.5: Changing a general comment to specific comment

to be repositioned. In a multiple page document, the user may also reposition the marker onto a different page by simply navigating to a different page and making a click on that page.

## Annotation Manager

Annotation Manager is the module that encompasses all the functionalities of TAS. At the server side, it manages a database where it stores all the data generated as a result of users adding comments to various documents and images. The following list of functions defined as part of the annotation manager gives a more detailed view of the annotation manager:

1. Initialize database: When the SAGE2 server is started, the annotation manager connects to the



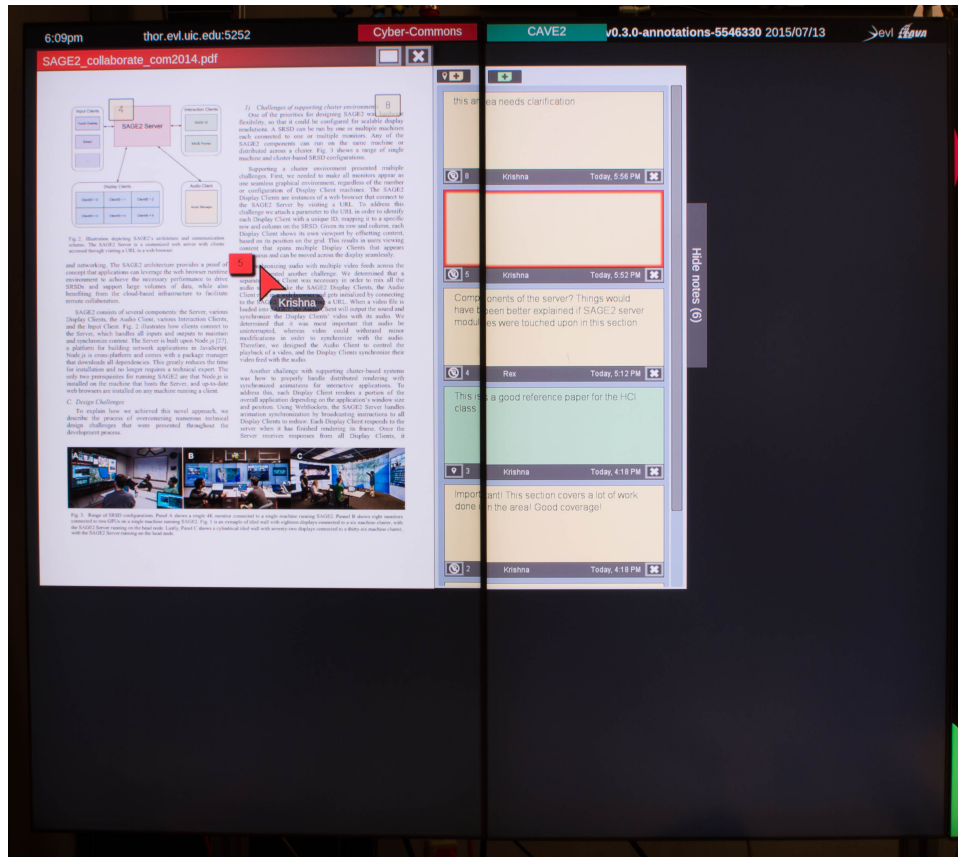


Figure 5.6: Left click on the page moves to marker to the position of the click

database (a JSON file at the server that is used for storage). This functionality is not user operable.

2. **Save annotation:** Whenever a new note is created by the user, annotation manager saves information, such as creation time, user ID of the creator, and other details, regarding this note in the database. It also has the ability to individually save the text of the note and marker (if any) associated with the note. This is to support the two different types of notes (general and specific). When a user creates a general note, text alone needs to be saved, whereas when the user creates a marker for an existing general note thereby making it specific, a marker alone needs to be saved for an already existing note. The database itself is a JSON file with a structure as shown in Figure 5.7
3. **Toggle annotation window visibility:** Annotation manager maintains information about the current visibility status of each annotation window as part of the state of that annotation window. Whenever a user wants to hide/show the annotation window for a document or an image, the user's actions (click on the hide/show button shown in Figure 5.5) are used by the annotation manager to reflect the change in the state accordingly before being passed on to the display for toggling of the actual annotation window's visibility. Aside from toggling the visibility of the annotation

```

{
  "<filename>": {
    "<id>": {
      "id": "<id>",
      "userLabel": "<note owner's name>",
      "createdOn": "<encoded creation time>",
      "text": "<text of the note>",
      "marker": {
        "position": {
          "x": "<left value in %>",
          "y": "<top value in %>"
        },
        "page": "<page number>"
      }
    },
    "<id>": {
      ....
    },
    ....
  },
  "<filename>": {
    ....
  }
}

```

Figure 5.7: JSON annotation database structure

window, the client side also toggles the visibility of all the markers that were placed on the page being currently displayed in case of a document or every marker placed in case of an image.

4. Load annotations: When a new document is pushed to the display, the document name is queried against the database by the annotation manager and it fetches the entire list of annotations that have been previously made for that document. The server then sends this list to the display client along with the document or image to be displayed. The position information of markers (in case of specific comments) is also communicated to the display client along with the rest of the information regarding the annotations at the time of loading.
5. Add a new note: Whenever the user requests for a new note, the annotation manager makes a new entry in the database for the new note, against the document or image being annotated and the server subsequently notifies the display side of the annotation manager to add a new editable note

within the annotation window.

6. Toggle edit mode: Annotation manager maintains a list of all active notes within an annotation window and when a user clicks on one of his/her note, it is added to this list and the display client is notified so that the user will be allowed to edit the note. At the same time the note that was previously being edited by the same user will be removed from the list of active notes. Whenever a specific note is made active in this fashion, if the document being annotated has multiple pages, then the current view of the document is set to that page on which marker associated with the note was last placed.
7. Toggle annotation type: When a user changes the type of a note, this is internally treated as either a marker of a note being deleted (specific note becomes general) or a new marker being added to a note (general note becomes specific). When a marker is to be deleted, the database entry of the note is modified by removing the marker information from it. When a new marker is added to a note, the database entry of the note is modified by adding the marker information to it. As mentioned above, page number of the page on which a marker resides is also stored as part of marker information. Whenever a new marker is added to a note, it is placed by default on the top left corner of the page being currently viewed. The marker information that is stored in the database consists of these default values and changes when the user repositions the marker.
8. Change marker position: When a user moves a marker, the annotation manager saves this new position information in the database.
9. Page change: When a user changes the page (in a multiple page document) being currently viewed the annotation manager hides all the markers that were being displayed (on the page that was previously being viewed) and displays all those markers positioned on the current page.
10. Delete annotation: Deletion of a note is carried out by the annotation manager by first removing the note section from the annotation window and then by removing the corresponding entry from the database. If the note being deleted has a marker, then the marker is also removed.

## 5.2 Implementation of Sticky Note

Sticky Note has three key features in it and these three features were independently developed and brought together to realize Sticky Note. The three features being:

1. A text editing area with user interface at the display client side: This was initially developed as a SAGE2 application utilizing the SAGE2 application API and later merged with the SAGE2

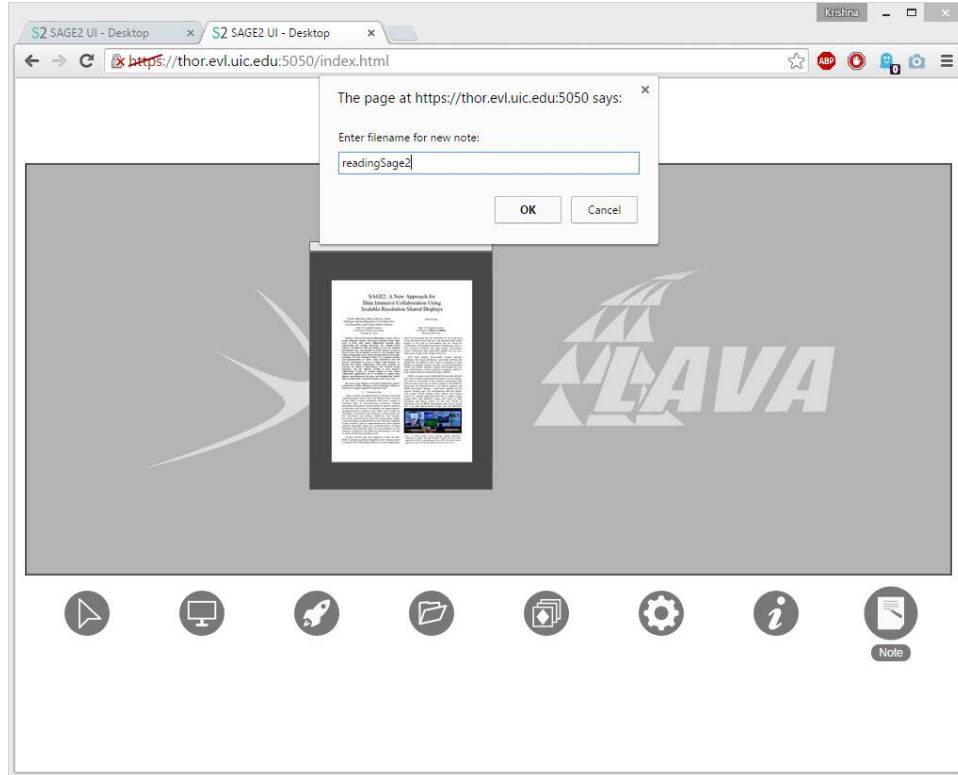


Figure 5.8: Filename prompt for the sticky note

core functionality. This application requests for a file buffer and also for stickiness feature thereby bringing all three aspects together.

2. A text buffer at the server side with file output functionality: This was developed as a module called File Buffer Manager at the server side. Any application can request for a text buffer.
3. The stickiness feature: Another server side module called Sticky Item Manager was developed to provide stickiness to any application that requests for this feature.

## User interface

The Sticky Note provides a set of features for creating and editing notes. These features are discussed below:

1. Create a note: The web user interface of SAGE2 has a button to create a Sticky Note. When a user clicks on this button, the user is prompted to provide a filename for the note. Then, the user is prompted to provide an initial text that goes onto the note. The filename as well as the initial text fields have default values and in that sense user input is optional. When the user clicks 'OK' at these prompts, a note instance is created on the display wall. This is shown in the Figure 5.8, Figure 5.9, and Figure 5.10.

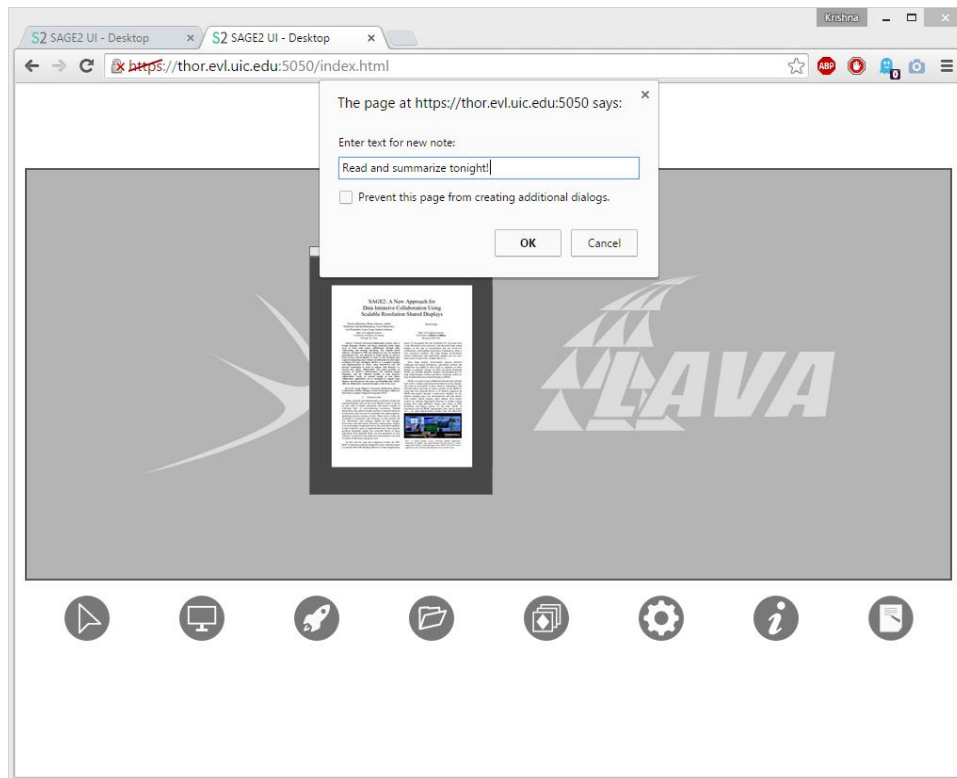


Figure 5.9: Initial text prompt for the sticky note

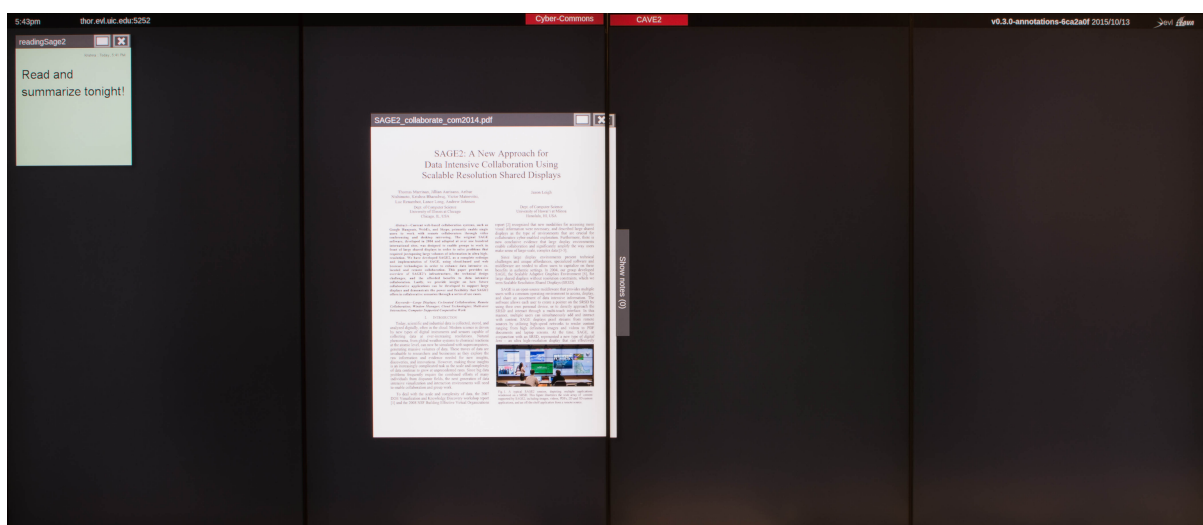


Figure 5.10: Sticky note instance created on the display wall



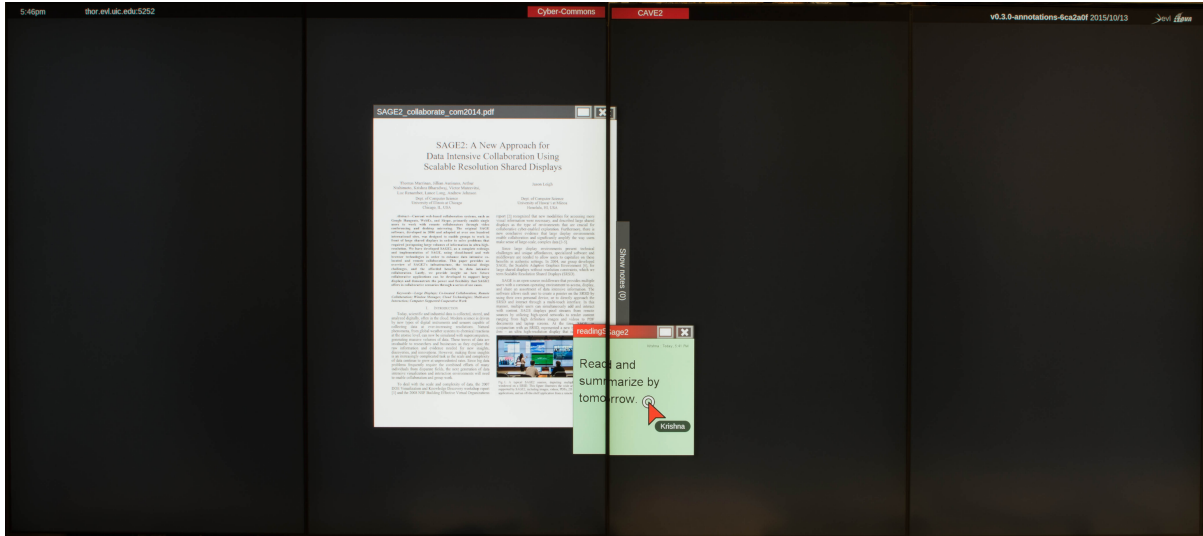


Figure 5.11: Edit note in deep interaction mode

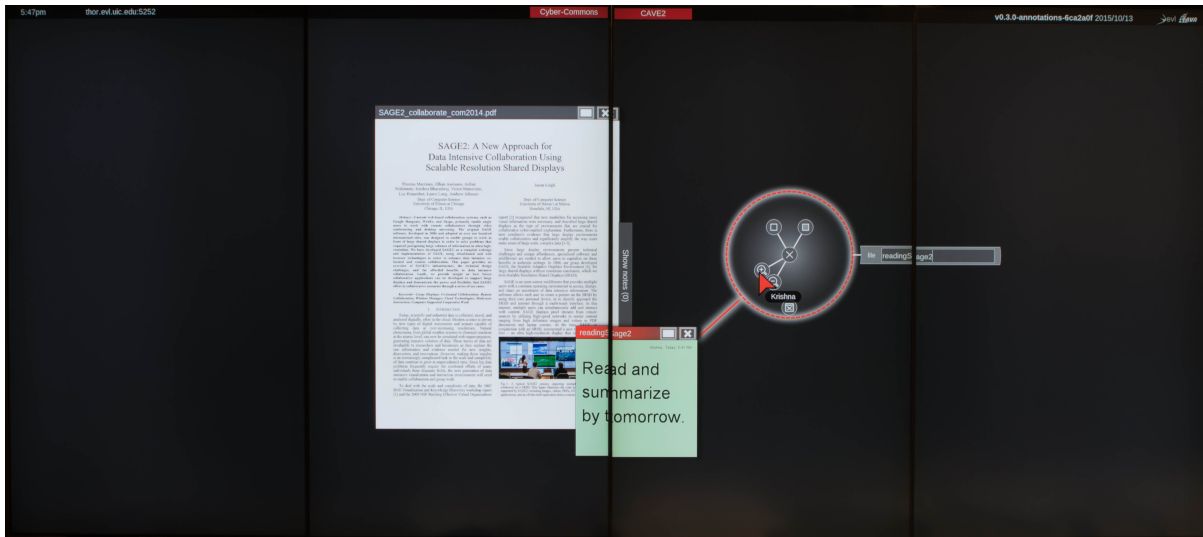


Figure 5.12: User interface of sticky note

2. Edit a note: The user has to go into deep interaction mode (shown in Figure 5.11) and while keeping the pointer within the note window the user can edit the note using key strokes.
3. Change file name of the note: Using the text input field in the UI (Figure 5.12) of the note, the user can change the name of the file that's associated with the note.
4. Change font of the note: The UI of the note also has buttons (Figure 5.12) to change the font of the text in the note.
5. Stick a note: The user can stick a note to any other application window by moving it and placing it such that there is an overlap between their boundaries.

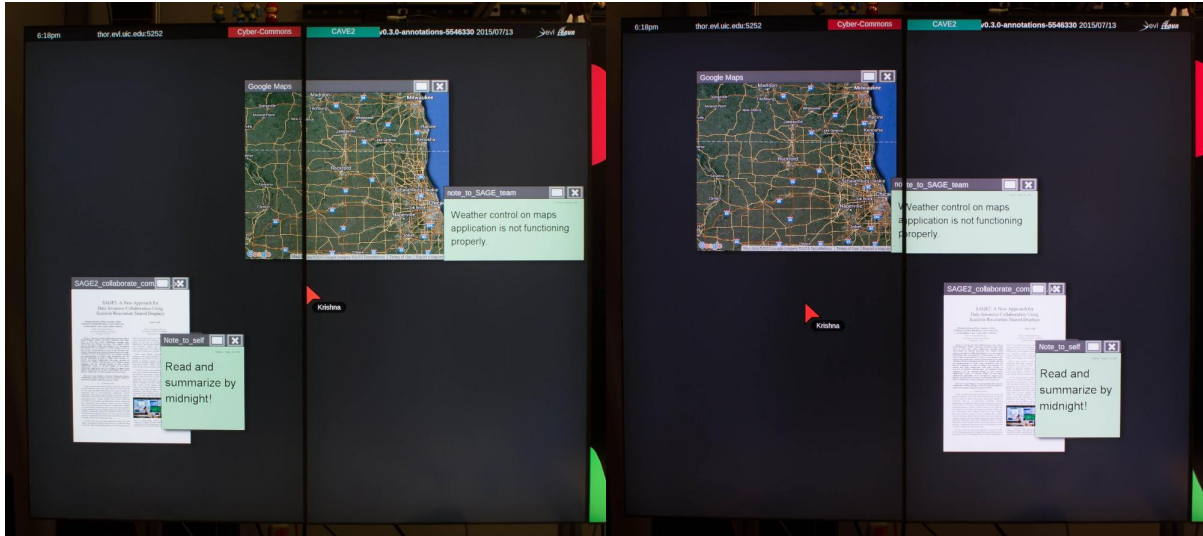


Figure 5.13: Sticky note moves with the document

The editor interface was developed as a SAGE2 application making use of the SAGE2 API and hence its structure is that of a custom SAGE2 application. It has callback routines for events like move, resize, and load.

### File Buffer Manager

1. Request buffer: Any SAGE2 application can request for a file buffer. When a file buffer is requested, the file buffer manager creates and maintains a buffer for every instance of that application. Along with requesting a buffer, the application has the option of binding the buffer to a file for output. If during the course of its execution, an application wants to bind the buffer to a different file, it can do so by sending another request for buffer with a different file name.
2. Close file buffer: When an application that requested a file buffer terminates, it notifies the manager and the manager in turn closes the file buffer associated with that instance of the application. However, before it closes the buffer, it will write any unwritten data in the buffer to the file bound to the buffer.
3. Edit buffer: Once an application requests for a file buffer, all keyboard events to that application are redirected to the buffer. This results in the buffer getting edited and the buffer gets filled with textual data. Every such edit to the buffer is then forwarded to the application in the form of an event called 'bufferUpdate' with a record, that has four values describing the latest change that happened to the buffer, as the data for that event. The record is as shown in Figure 5.14. Every buffer that is created has an internal counter and a limit associated with it (this limit can be configured) and the counter represents the number of changes that have been made since the

```

{
    index: //Position in the buffer where change occurred ,
    offset: //After the change is consumed, this value is added to the index,
    deleteCount: //Indicates how many characters were deleted,
    data: //Data that was introduced at index due to this change
}

```

Figure 5.14: Buffer update data

last file write. Whenever the counter reaches the limit, if a file has been bound to the buffer, then all the changes are written onto the file and the counter is reset. The buffer itself is not affected by this write. If a file has not been bound to the buffer, then the counter is simply reset. When the content of a preexisting file needs to be opened by an application that uses a file buffer, the application may request for another way of editing the buffer. That is, it can request for the entire content of the file to be placed in the buffer at once.

### Sticky Item Manager

Any application may request for the sticky feature through its configuration. The sticky item manager is responsible for realizing the sticky behavior of such applications. In order to achieve this, the manager maintains an internal data structure, very similar to a dictionary, that holds key value pairs. A key in this case is any application window onto which a sticky item is sticking, and the value is a list of all sticky items that are currently sticking to that ‘key’ application. The following list of functions along with this data structure make up the Sticky Item manager:

1. **Attach/Detach Sticky Item:** Every time a sticky item that is present on the display is moved, its boundary is checked to see if it has been moved over another application window. If so, then the sticky item manager adds an entry into its data structure to mark that a sticky item has been attached to another window. At the same time, if a sticky item’s boundary no longer intersects the boundary of the window it was attached to, then its entry is removed from the data structure.
2. **Remove Element:** If an item is removed from the display, the sticky item manager removes it from its data structure as well.

3. Move sticking items: When an application window on the display is moved, the sticky item manager searches its data structure to see if any items are sticking to this window that is being moved. It then fetches the list of items sticking to this window and applies the same translation to those sticking items as the moved window thereby moving them along with the window (Figure 5.13). If the point on the background application, where a sticky item is attached, moves as a result of the application is being resized, the sticky item manager repositions the sticking item suitably.
4. Save/Load sticky item manager state: In order to maintain the relationships, between sticky items and the applications that these sticky items are attached to, beyond a session, the sticky item manager saves its current state along with the rest of the session information whenever a session is saved. Later, when a previously saved session is restored, this information is used to determine what items were attached.

## 6. USER STUDY AND EVALUATION

This chapter describes the user study that was conducted to evaluate the usability of these tools and the results of the study. The study was aimed at gaining insights into how effective these tools are in letting users make comments on document and images.

### 6.1 Usefulness measure

Effectiveness of annotation tool can be measured by the number of useful comments users create for a document using the tool in a session. Comments can be either general or specific. A general comment is a comment that makes a point that is applicable to the document as a whole, or a large portion of the document. Specific comments are those that are applicable to a particular area of Interest(AOI) within the document. In other words, a specific comment makes sense only when it is related to the AOI. Therefore in measuring the effectiveness of an annotation tool, a specific comment will have more value compared to a general comment. Although “usefulness” is a relative term, for the purposes of this evaluation I define usefulness of a comment with the help of the following functions.

$$U(\text{general comment}) = R$$

$$U(\text{specific comment}) = R * C * (1 + \alpha)$$

where: U: Usefulness value

R: Relevance

C: Context

$\alpha$ : Effect of nature of association

The usefulness of a comment depends on its content and context. Content is what is being said. Content of a comment is by its nature, very subjective and hence cannot be compared to the content of another comment to say one is better than the other (and hence more useful). However, we can safely say that the content of a comment should be relevant to the document being commented upon. Context is what is being talked about. In a general comment, the whole document is the context of that comment

C	Association provided	No association
Specific comment	1	0

Table 6.1: Context value in usefulness measure

and this is usually evident in the content of the comment. Hence the usefulness of a general comment is nothing but its relevance (denoted by  $R$ , 0 for not relevant, 1 for relevant) to the document.

For a specific comment, when the content is relevant, it becomes a contributing factor (context of the comment being the other contributing factor) to the usefulness of that comment and when the content is not relevant, it becomes a deciding factor to say that the comment is useless. The context of a specific comment has to be established by associating the comment to AOI. The context of a comment thus takes on the form of a boolean variable  $C$  and its value under various cases is given by in Table 6.1.

The context of a specific comment can be indicated by visually associating the content with the AOI. TAS achieves this form of visual association with the help of a marker and SN achieves it by allowing the user to attach a note to any point of the window that displays a document or an image. When SN is being used to make comments on a multi page document, such a visual association is not possible as the note sticks to the window, irrespective of the page being displayed. In this case, an indirect association can be established by citing (giving some form of reference to) the AOI in the comment. However this form of association is less efficient compared to visual association because here the association will always be unidirectional, that is from the comment to the document but not the other way round. Thus the nature of association determines the difficulty in making sense of a comment and factors into computing the usefulness of a comment. Further, the citations themselves differ from each other in how well they point to the AOI. Citations can very clearly indicate the AOI (giving the page number, positional indicators and so on), and on the other hand some citations are implicit references to keywords (task of association completely falls on the reader). These factors are captured in the variable  $\alpha$ . The following points describe the different cases that are possible and gives the value of  $\alpha$  in these cases:

- For Textual Annotation System:
  - When there is a single AOI, if the marker has been placed properly at AOI,  $\alpha = 1$ , if the marker is not placed at the AOI then  $\alpha = 0$  (note that  $C$  will also be 0 in the case)
  - When there are multiple AOI, that is when the same comment could be applied to multiple AOI in the document, then  $\alpha = 1$  as long as the marker has been placed at one of those AOI.
  - When different parts of the document together make up the AOI for a comment, that is when the comment being made is collectively about more than one part of the document  $\alpha = 0$  (and  $C=0$ ), since only one marker is available per comment and placing it at any constituent part of the AOI will not make the comment useful.

- When a page of a document or an image gets a large number of comments such that the association of a comment to its AOI via the marker no longer remains a trivial visual association task, but the large number of markers on the page/image forces the user to scan through the markers to locate a particular marker, this reduces the usefulness of such a comment. In this case  $\alpha = 0.70$  (0.5 for association in the direction from AOI to note, and 0.2 for association in the direction from note to AOI. This reflects the fact that scanning for the markers is only slightly better than scanning for the actual AOI)
- For Sticky Note:
  - When there is a single AOI and the note is visually linked properly by placing at the AOI,  $\alpha = 1$ .
  - For a citation of a specific comment  $\alpha = 0.5$ . This means that the link works in only one direction. It is not possible to associate an AOI to its comment in this case. Here it must be noted that in case the AOI is made up of different parts, providing a citation to each part is possible. This will not affect the usefulness of the comment adversely unlike in the case of TAS.
  - When the association is unidirectional (citation,  $\alpha=0.5$ ) and the citation is implicit, the size of the document further adds to the difficulty in associating a comment to its AOI. For the multi page document used in this study, since each page has graphical data, I have chosen to use the number of pages as the determining factor of  $\alpha$ . Thus  $\alpha = 0.5 / (\text{Number of pages})$ . If the document had textual data, then in addition to pages, number of paragraphs and other such measures would have further reduced  $\alpha$ .
  - When there is no association provided for a specific comment,  $\alpha$  takes a value of 0 indicating that that comment is useless, this case will always have  $C=0$ , thereby making  $U=0$ .

If a comment has a  $U$  value greater than 0, then it is counted towards measuring the effectiveness of annotation tools. Further, the value of  $U$  acts as an indicator of how useful a comment is for someone who goes through the document, should he/she decide to read the comment.

## 6.2 User study

The study was conducted at the Electronic Visualization Laboratory at UIC making use of the Cyber-Commons tiled display wall. This tiled display is a matrix of six panel wide and 3 panel high, thin bezel LCD displays coming to a physical size of 20 feet by 8 feet and a resolution of 18 mega pixels. The subjects sat facing the tiled display at a table approximately 6 feet away from the display. Although, no

restrictions were placed on the subjects, all of them remained seated throughout the sessions. The study was aimed at gaining design insights to create efficient multi-user annotation systems for SAGE2. The user study compared two different annotation tools for SAGE2 on the basis of number of annotations generated, placement of positional indicators, and user satisfaction in using the tools. A group - of two subjects at a time - was observed as they collaboratively annotated a document and an image. The subjects were free to converse with one another throughout the task. Two subjects were involved in each session to simulate a collocated collaborative scenario as SAGE2 is primarily used in such scenarios and this would give a better understanding of the affordances and limitations of the annotation tools in a group scenario. To facilitate the experiment, a facilitator sat in the room during the session answering questions regarding the use of the tools. Each session was video and audio recorded to capture the users' interaction with the tools. At the end of each session, a snapshot (or context) of the wall was saved and restored at the beginning of the next session so that the subjects of that session got the annotations added to the document and image by the previous subjects as their starting point. Each session lasted about 60 minutes where with the first 15 minutes going into training. Following the training, the subjects performed two tasks (explained below) and about 20-25 minutes was allotted for each task. At the end of allotted time for each task, the subjects were asked to stop the tasks. Ten such sessions were conducted involving a total of 20 subjects (two per session). Of the 20 subjects, 16 were male and 4 were female and came from an age range of 20 to 50 years. All of them used computers in their daily work and had prior experience with large display systems since they were all associated with EVL, UIC (students and non teaching staff members).

The subjects were informed ahead of time to bring their own laptops to the study. All of them used their own laptops to connect to and interact with the display. For SAGE2 pointer interactions, all of them used the track pads on their laptops. The subjects were given a fifteen minutes training on how to use the tools. Explanation was given regarding how the controls work, how the two different annotation tools work, and how they can use the tools to add annotations to documents and images.

For the tasks of this user study a multi page (six pages) pdf document and an image was used. The contents of the document and the image served as a basis for setting up a scenario that would require the subjects to annotate the document and the image. The document contained floor plans of a new building that the College of Engineering, UIC is planning to build. As a scenario, the subjects were told that the College of Engineering is interested in getting their feedback about the plans. The image was an initial draft of a technical poster that one of the grad students at EVL had created. As a scenario, subjects were told that the grad student was interested in getting their feedback about the content, layout, and design of the poster to improve the quality of the poster. The subjects were given a brief oral description of the document and the image to which they were supposed to add annotations.



Trials	Task 1	Task 2
1 - 5	TAS on document	SN on image
6 - 10	SN on document	TAS on image

Table 6.2: Assignment of annotation tools in trials

	Textual Annotation System		Sticky Note	
	Comments made	Useful comments	Comments made	Useful comments
Multiple page document	58	57	84	80
Image	58	56	39	36

Table 6.3: Useful comments out of total comments

The training was followed by the first part of the study in which the subjects went through the document and added annotations to the document using one of the annotation tools. For the second part of the study, the subjects added annotations to the image using the other annotation tool. Table 6.2 shows the assignment of tools amongst the two tasks, across the trials of the study. At the completion of the task, the subjects took a brief survey about their experience (using pen and paper). The purpose of this survey was to gain additional insights about the subject's experience and see if they have any suggestions on how to improve the design of the annotation tools.

## Data gathering

All the annotations made during the ten trials using both TAS and SN were retrieved from the server along with meta data such as the time of creation, owner of the note, and so on. Video and audio recordings of the sessions were used to verify this data. For computing the usefulness of comments, as described above, each comment was analyzed and values of  $R$ ,  $C$ , and  $\alpha$  for each comment were tabulated.

## 6.3 Results

Textual Annotation System and Sticky Note were both used to annotate a multiple page document and an image. In this section we will look at how these two tools fared in case of a multiple page document and how they fared in case of an image.

Usefulness scores were computed for each comment and if a comment gained a usefulness score greater than zero it was counted as being useful in Table 6.3.

### Annotating an image

In the first five sessions, Sticky Note was used to annotate an image. Notes created by the ten subjects in these five sessions were placed(stuck) directly on the image close to the AOI of the note in case of a

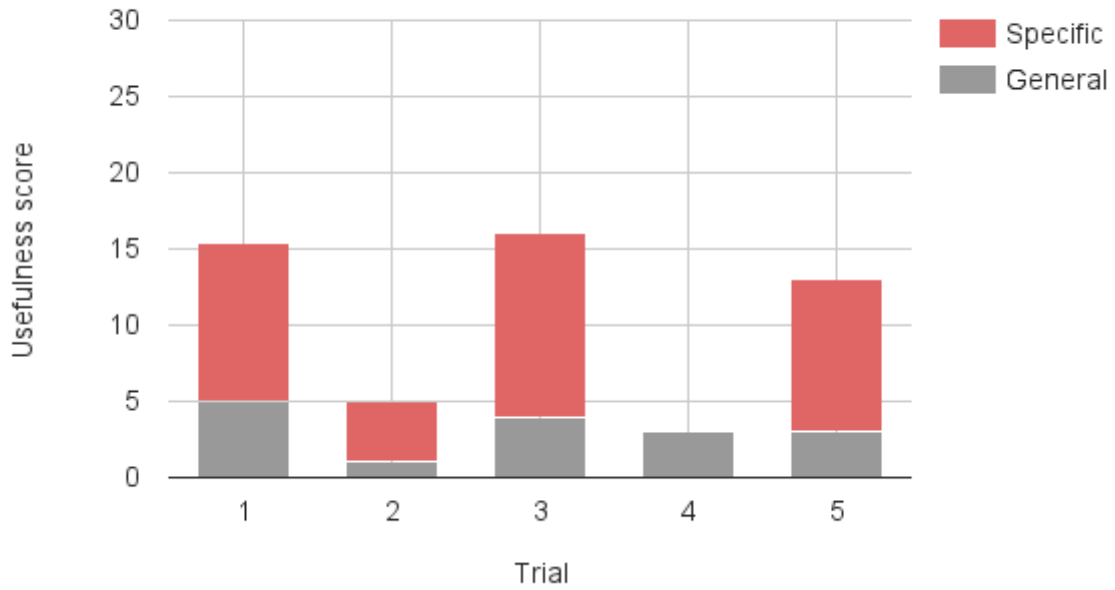


Figure 6.1: Usefulness score of comments on image using Sticky Note

specific note. Due to the direct visual association of the comments with their AOI, all specific comments had  $\alpha$  values of 1. In the very first session subjects started placing the general notes off the image to a side on the display. This convention was followed throughout the five sessions. For this task, the users worked independently on separate areas of the image occasionally consulting each other for the other person's opinion on a comment they were making. A few sample annotations are given below:

- General comments
  - “this looks like enough information for 3 posters. Way too visually busy.”
- Specific comments
  - “small font size on the poster might be uncomfortable to the reader”
  - “I feel like the results portion should be larger”

In the following five sessions that made up the second half of the study, Textual Annotation System was used to annotate the image. Markers were used to connect the comments with their AOI. As a result these comments had  $\alpha$  values of 1. However, at the beginning of the 4th session in this set, that is at the beginning of session 9, it was observed that the subjects had to scan the scattering of markers to locate the marker associated with a specific comment instead of instantaneously locating it. Thus, I deduced that the number of markers on the image must have crossed that minimum number after which the

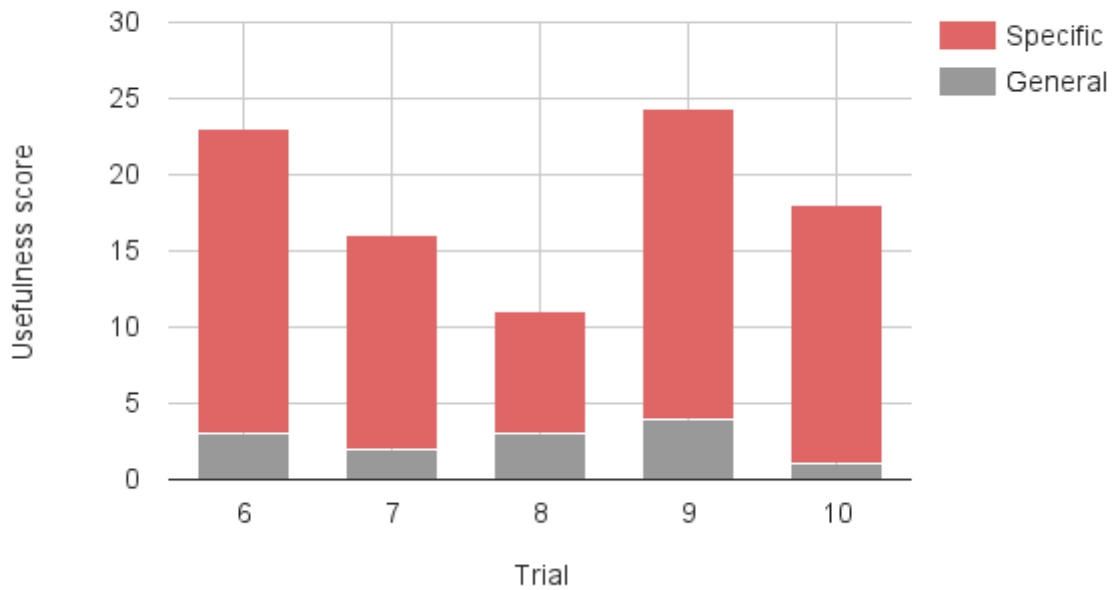


Figure 6.2: Usefulness score of comments on image using Textual Annotation System

lookup doesn't remain a trivial association task, but scanning the markers becomes necessary. Hence, the specific comments made in the last two sessions in this part of the study have  $\alpha$  value of 0.7. For this task, the users worked independently on separate areas of the image occasionally consulting each other for the other person's opinion on a comment they were making. A few sample annotations are given below:

- General comments
  - “The titles of each section take up a lot of space. Make them smaller”
- Specific comments
  - “Text is too close to border... maybe make it left aligned?”
  - “These panels are good explanations of the design goals and methodology”

### Annotating a multiple page document

In the second half of the study, Sticky Note was used to make comments on a multiple page document consisting of six pages. Out of the ten subjects who used Sticky Note to make comments, four subjects provided page numbers as well as keywords as citation to the comments whereas the other six subjects only provided keywords as citation. This resulted in reduced  $\alpha$  values for the comments of these six

subjects. Despite large number of comments being generated in this part of the study, the usefulness scores have come down due the reduced  $\alpha$  values as it is evident in the Figure 6.3. For this task, the comments were a mixture of independent work and collaboration. Four subjects opened their own sticky notes and made comments whereas the rest of them often wrote in each other's sticky notes. All ten subjects wrote their comments in one or two sticky notes each, that is they made multiple comments under the same sticky note. A few sample annotations are given below:

- General comments
  - “Can we go to another building without leaving this one?”
- Specific comments
  - “pg4: Some classrooms missing occupancy” (Explicit citation)
  - “Offices seem to be small as well on 5th floor” (Implicit citation)

The first half of the study had subjects making comments on the multiple page document consisting of six pages, using Textual Annotation System. Markers were used to associate comments with their AOI resulting in  $\alpha$  value of 1 for all the specific comments and since the total comments made were spread across multiple pages, none of the pages were over crowded with markers. For this task, the users worked independently on separate areas of the image occasionally consulting each other for the other person's opinion on a comment they were making. A few sample annotations are given below:

- General comments
  - “where are the doorway access symbols for each room?”
- Specific comments
  - “this auditorium extends into sublevel. a lot like lecture hall E”
  - “why so many auditoriums? already one in basement (that we think should move to first floor).”

## Overview

The following two charts show the range of comments for each trial. Data for Sticky Note at individual session level shows that the first five sessions where Sticky Note was used to comment on image has least number of comments compared to the other three parts of the of the study. Also, it reflects the fact that in case of multiple page document which was used in the study the number of comments made using

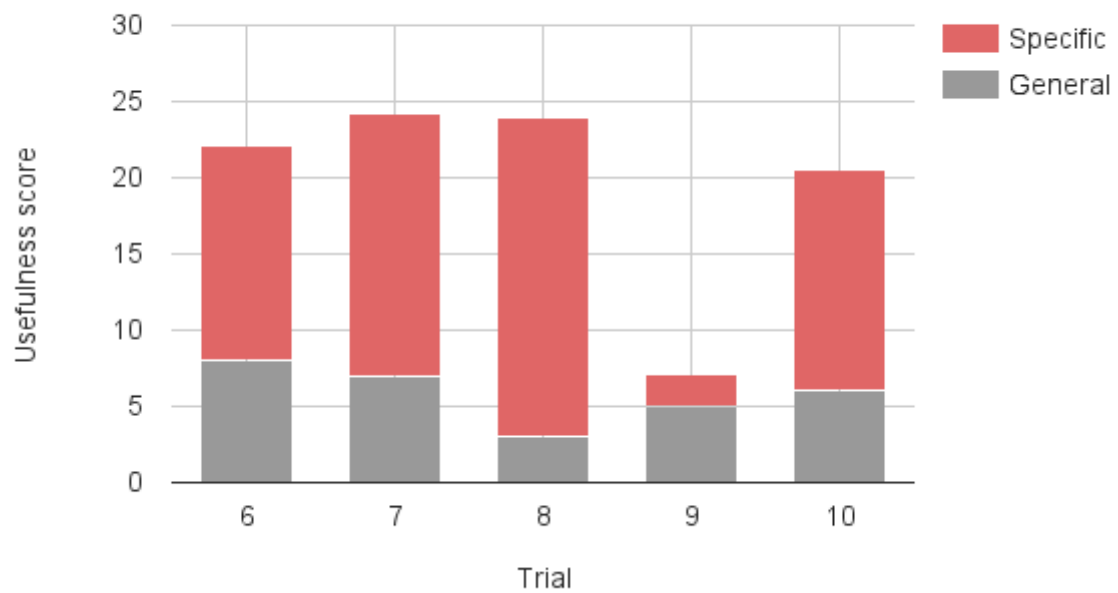


Figure 6.3: Usefulness score of comments on multiple page document using Sticky Note

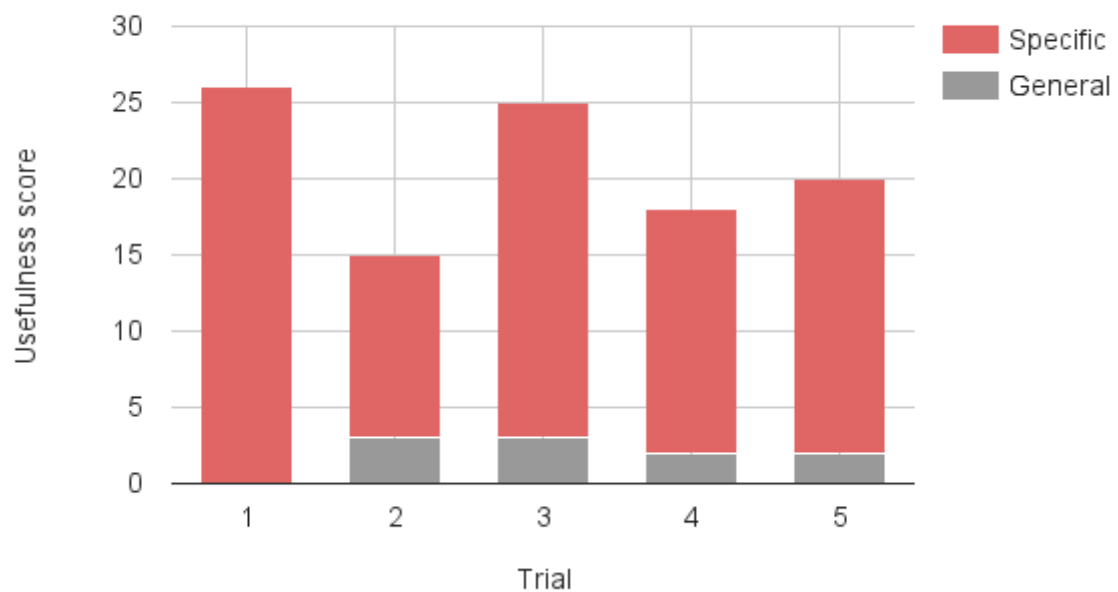


Figure 6.4: Usefulness score of comments on multiple page document using Textual Annotation System

Sticky Note is higher compared to all the other three scenarios. Further, more general comments were made using Sticky Note compared to Textual Annotation System.

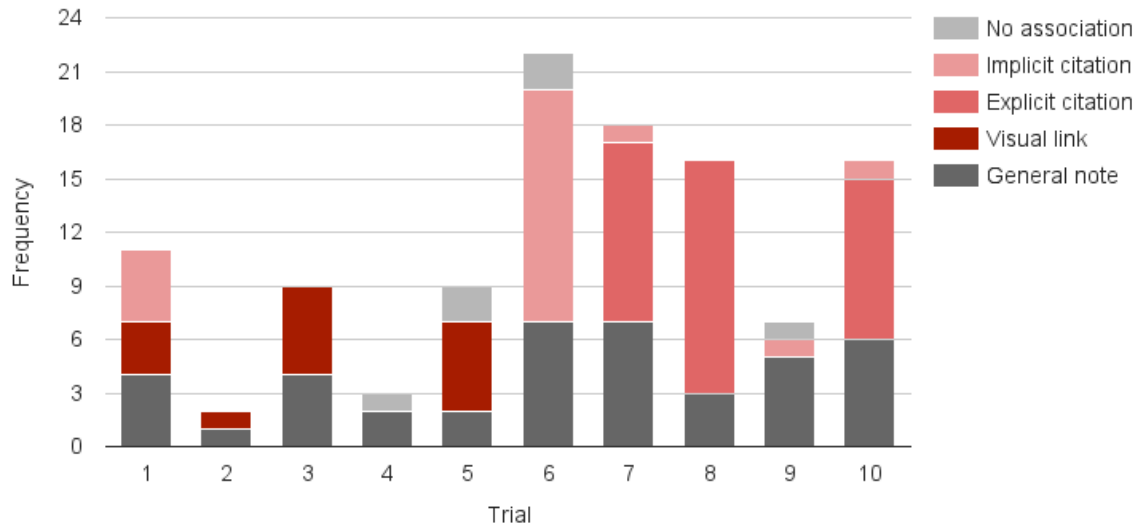


Figure 6.5: Range of comments produced by SN (trials 1-5 on image and trials 6-10 on document)

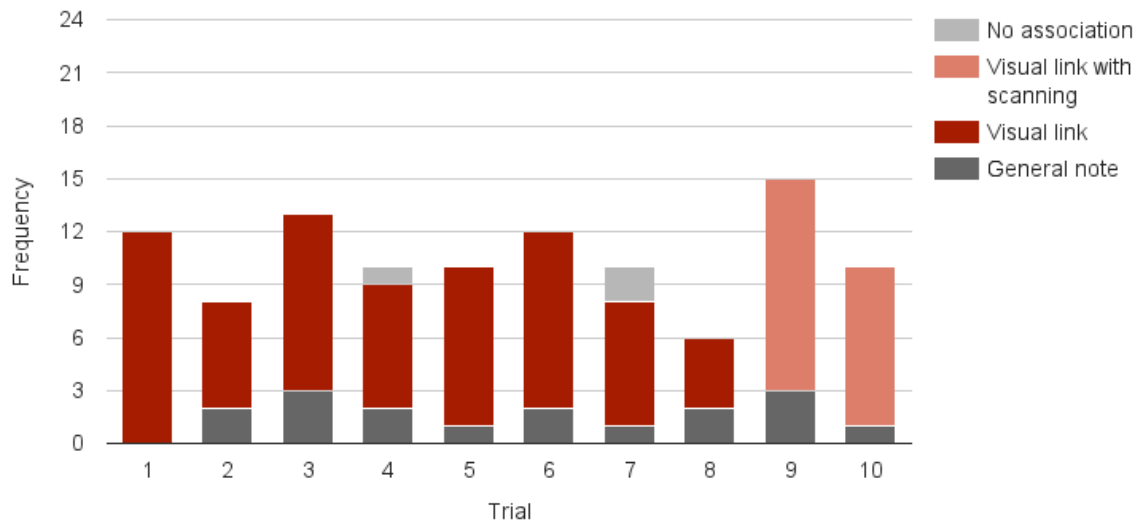


Figure 6.6: Range of comments produced by TAS (trials 1-5 on document and trials 6-10 on image)

When we look at the average usefulness score (shown in Figure 6.8) of these two tools against image and multiple page document we see that the performance of Sticky Note with a multiple page document is very low. This is because of the fact that direct visual association is not possible in this scenario. However, this is also the case where most comments were made. A large number of comments, with a good number of them being specific comments, despite the low average usefulness score suggests that association of comments to their AOI is harder. On the other hand, the sheer number of comments also suggests that this form of creating annotations is perhaps easier compared to others. In case of an image, even though the usefulness of specific comments made by Sticky Note is not handicapped by lower alpha

values, the small total number of comments indicate the existence of some kind of deterrent in creating annotations in this manner. Observation of subjects during the sessions showed that visual occlusion caused as a result of directly placing the notes close to their AOI resulted in later subjects moving the previously placed notes to a side so that they could look at that part of the image. Additionally they had to move these notes back to their original positions. This significantly added to the task of creating annotations.

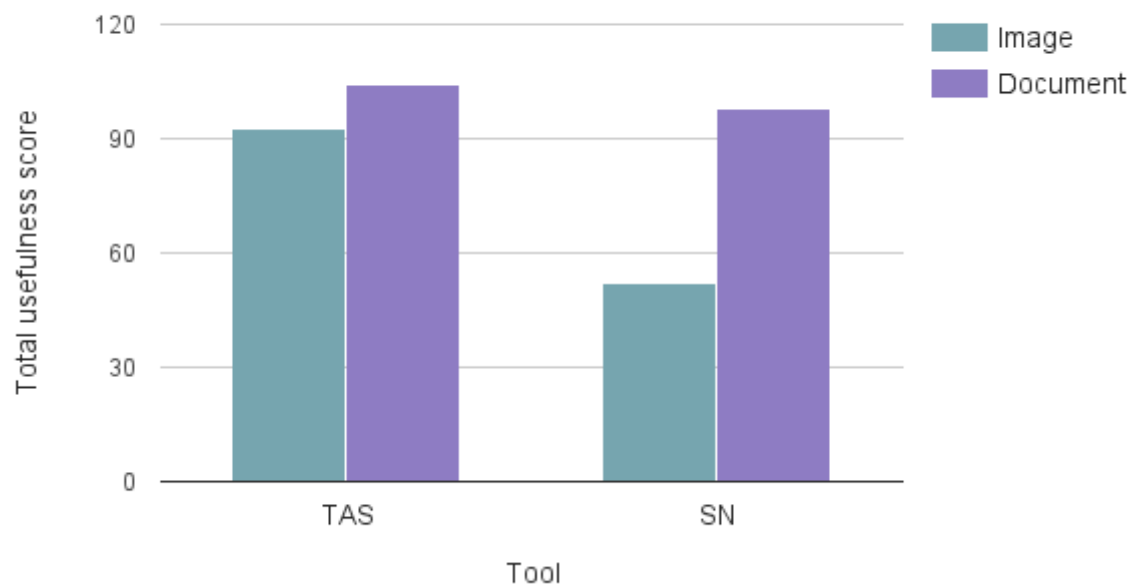


Figure 6.7: Overall usefulness scores of TAS and SN for image and multiple page document

## Survey results

A survey was conducted at the end of the user study. The purpose of this survey was to gain additional insights about the subject's experience and see if they had any suggestions on how to improve the design of the annotation tools

The first part of the survey asked the subjects to rate the annotation tools based on their ease of use on a scale of 1 to 10 with 1 being 'difficult' and 10 being 'easy'. The results are shown in Figure 6.9

The second part of the survey asked the subjects for their preferred annotation tool between the two choices they had. Table 6.4 tabulates the results. One user skipped this part of the survey.

The next part of the survey asked the subjects to describe the difficulties that they felt while using the tools. Excluding the difficulties that the subjects reported which were a direct result of limitations of the SAGE2 interaction system, the rest have been tabulated in Table 6.5 and Table 6.6.

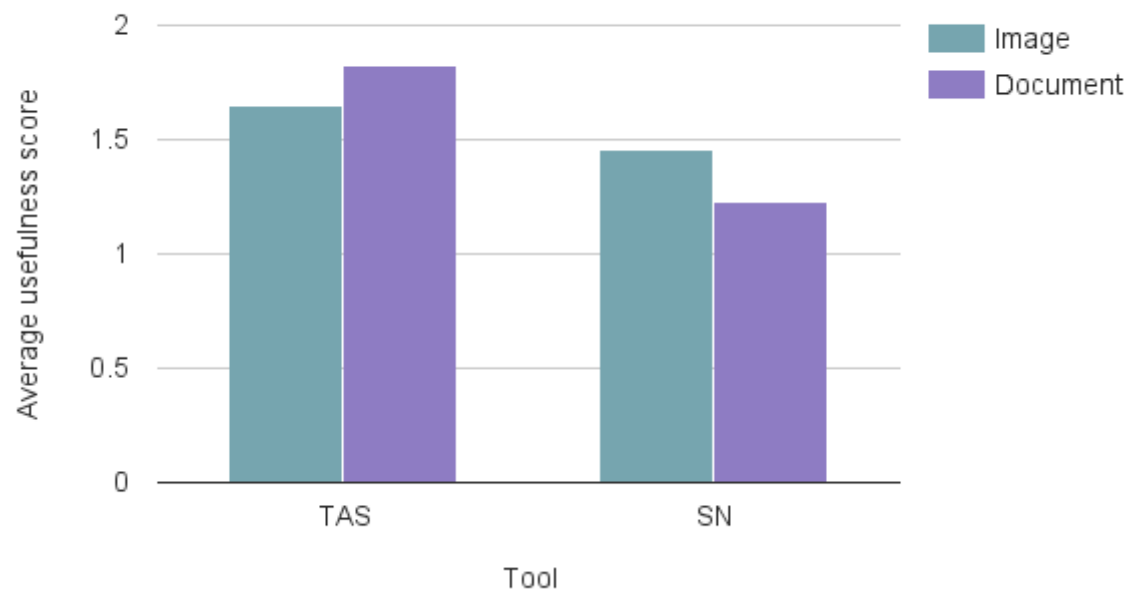


Figure 6.8: Average usefulness scores of TAS and SN for image and multiple page document

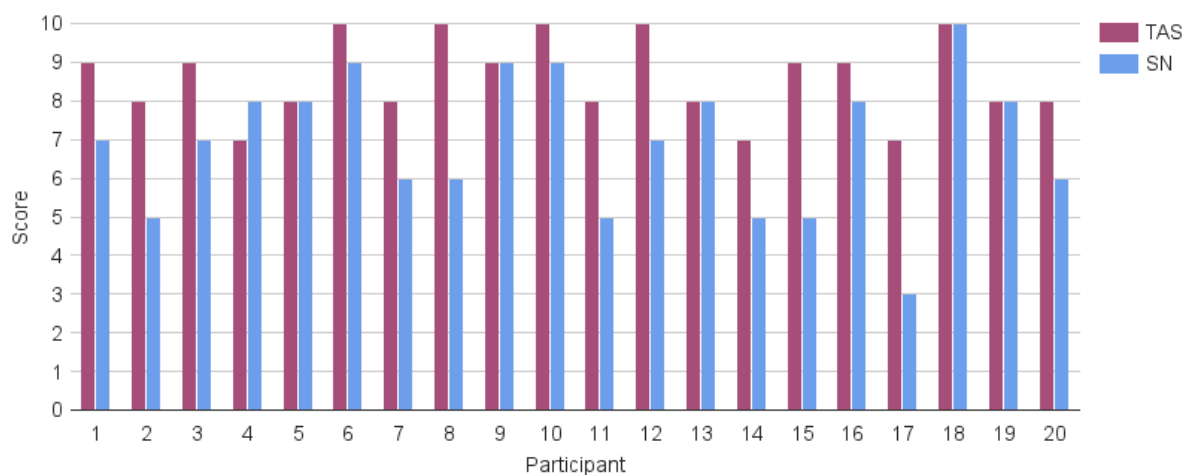


Figure 6.9: Ease of use scores for TAS and SN

The next part of the survey asked the subjects for suggestions on how to improve the tools. Table 6.7 and Table 6.8 show suggestions that the subjects made towards the two tools (Suggestions made for alleviating issues arising from SAGE2 interaction system have been excluded) .

The last part of the survey asked the subjects to provide general comments regarding the tools. About half of the subjects did not have anything to say. A few comments were repetitions of some of the list of suggestions listed. The rest of the comments are as given in the Table 6.9:



Tool	Number of test subjects that showed preference
Textual Annotation System	18
Sticky Note	1
No tool	1

Table 6.4: Annotation tool preference of all 20 participants

Difficulty	Number of users who reported it
Unintentional repositioning of active markers due to left clicks made out of habit.	5
Locating a marker in a large document.	1
Marker obscuring content of the document/image.	1
Too many notes(markers) induces clutter.	1
Difficult to peruse or edit when someone else is editing a note.	2
Could not resize text block.	2

Table 6.5: Difficulties in using Textual Annotation System

Difficulty	Number of users who reported it
Note occludes parts of the image/document when visually linked (stuck).	2
After a few notes are placed on a document/image, they cover up most of the document/image.	1
Creating a new note through web UI.	4
Frequent need to keep changing pointer modes.	4
Mouse should be over the window to edit	2

Table 6.6: Difficulties in using Sticky Note

Improvements	Number of users who suggested it
Ability to reorder the notes in the annotation window.	1
Provide a different mechanism to reposition marker so that it is not moved unintentionally.	1
Clicking on marker should highlight the comment.	3
A third type of note, global(general) for a page.	1
Multiple markers for the same note.	2
Option to display notes only related to current page.	1
Ability to manually set marker caption.	1
Reply to notes(notes to notes).	1
Color code notes based on users.	1
Separate the notes being edited from the other notes and provide scrolling for the latter.	1
Segregate global and local notes.	1
Separate text input field in the web UI to edit notes.	1
Ability to merge notes into one another.	1
Hiding notes and markers selectively.	1

Table 6.7: Suggestions to improve Textual Annotation System

Improvements	Number of users who suggested it
Ability to minimize notes into markers.	1
Ability to toggle Stickiness feature.	2
Ability to turn on transparency of notes at will (when not viewing or editing).	2
Eliminate pointer mode switching.	1
Adding note should be a function of the document.	1
Create note using mouse at pointer position instead of web UI.	2
Ability to stick notes to individual pages.	1

Table 6.8: Suggestions to improve Sticky Note

General Comments	Number of users who made it
There is a small learning curve with both tools.	1
Both tools are useful and easy to use.	2
Visual links should be provided between note and marker(for Textual Annotation System)	1
Annotations worked better as small marker covered the actual document, with text elsewhere.	1
Prefer Textual Annotation System	3

Table 6.9: General comments on the annotation tools

## 7. CONCLUSION

The process of annotating media is a widely practised activity that plays an important role in the digital world. Today, many annotation tools are available that facilitate annotation of content on the web, annotation of not just text, but images, audio, and video, annotation of content on mobile and handheld devices. Due to their nascent state, large/scalable resolution display systems do not have any support for annotating content on these displays.

SAGE2, is a colocated collaborative environment that manage scalable resolution display walls. SAGE2 caters to a wide range of scenarios from classroom meetings, project group meetings to collaboration with domain experts. It is the first large/scalable resolution system to support multi user simultaneous interactivity that goes beyond moving and resizing windows on the display. As part of my thesis, I developed two annotation tools: Textual Annotation System and Sticky Note, for annotating documents and images in SAGE2. Textual Annotation system provides a dedicated annotation window and uses markers placed on the document to associate the comments with the content. Sticky Note can be directly attached to any point on an image, and in case of a multiple page document, users use citations within the note to associate the comment to the content. These two tools were evaluated in a user study to gain insights about their usefulness and other aspects of the tools that influenced how the users used these tools. Both tools were found to be useful by users. The usefulness of the Textual Annotation System was found to be higher than that of Sticky Note. However, the study showed that ease of adding comments (with citation) afforded by a Sticky Note prompted the users to add more number of comments using it. The larger goal and hope in building these annotation tools has been to enhance the overall usability of SAGE2 as a system. Even though these two annotation tools were built for SAGE2, their evaluation has revealed a few lessons that can be applied to other annotation tools as well. They have been listed below:

- Artifacts of the annotation tools should avoid occluding the content being annotated.
- Ease of creating individual comments influences the number of comments generated.
- It is essential that the navigation between AOI and their respective comments is bidirectional (from suggestions by users)

## 7.1 Future work

Some of the improvements suggested by the users during the study hold merit for consideration as enhancements to these tools that can be taken up as future work.

- Highlighting, ink annotations, and replies to annotations in Textual Annotation System
- Equipping the Sticky Note with the ability to shrink individual note to the size of a small marker.
- Multiple markers for one note in Textual Annotation System
- Locating note using marker (by clicking on the marker) in Textual Annotation System.

## CITED LITERATURE

- [1] T. Marrinan, J. Aurisano, A. Nishimoto, K. Bharadwaj, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh. Sage2: A new approach for data intensive collaboration using scalable resolution shared displays. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, 2014 International Conference on, pages 177–186, Oct 2014.
- [2] A. Febretti, A. Nishimoto, V. Mateevitsi, L. Renambot, A. Johnson, and J. Leigh. Omegalib: A multi-view application framework for hybrid reality display environments. In *Virtual Reality (VR)*, 2014 iEEE, pages 9–14, March 2014.
- [3] K.-U. Doerr and F. Kuester. Cglx: A scalable, high-performance visualization framework for networked display environments. *Visualization and Computer Graphics, IEEE Transactions on*, 17(3):320–332, March 2011.
- [4] L. Renambot, A. Rao, R. Singh, B. Jeong, N. Krishnaprasad, V. Vishwanath, V. Chandrasekhar, N. Schwarz, A. Spale, and C. Zhang. Sage: the scalable adaptive graphics environment. In *Proceedings of WACE*, vol. 9, page 2004–09, Sept 2004.
- [5] Scott Elrod, Richard Bruce, Rich Gold, David Goldberg, Frank Halasz, William Janssen, David Lee, Kim McCall, Elin Pedersen, Ken Pier, John Tang, and Brent Welch. Liveboard: A large interactive display supporting group meetings, presentations, and remote collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’92, pages 599–607, New York, NY, USA, 1992. ACM.
- [6] Jacob T. Biehl, William T. Baker, Brian P. Bailey, Desney S. Tan, Kori M. Inkpen, and Mary Czerwinski. Impromptu: A new interaction framework for supporting collaboration in multiple display environments and its field evaluation for co-located software development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’08, pages 939–948, New York, NY, USA, 2008. ACM.
- [7] Markus Rittenbruch. Cubit: Large-scale multi-user presentation and collaboration. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces*, ITS ’13, pages 441–444, New York, NY, USA, 2013. ACM.
- [8] Oblong Industries Inc. Introducing mezzanine: The future of conference room collaboration. white paper. <http://www.oblong.com/mezzanine/overview/>, 2013.
- [9] David Lee, Sean A. Munson, Ben Congleton, Mark W. Newman, Mark S. Ackerman, Erik C. Hofer, and Thomas A. Finholt. Montage: A platform for physically navigating multiple pages of web content. In *CHI ’09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’09, pages 4477–4482, New York, NY, USA, 2009. ACM.
- [10] G.P. Johnson, G.D. Abram, B. Westing, P. Navr’til, and K. Gaither. Displaycluster: An interactive visualization environment for tiled displays. In *Cluster Computing (CLUSTER)*, 2012 IEEE International Conference on, pages 239–247, Sept 2012.
- [11] Microsoft Corporation. Office support. <https://support.office.com/en-us/article/Insert-or-delete-a-comment-8d3f868a-867e-4df2-8c68-bf96671641e2>.
- [12] Google. Commenting in google docs. <https://support.google.com/docs/answer/65129?hl=en>, 2013.

- [13] Adobe Systems. Share, comment, and review. [http://help.adobe.com/en\\_US/reader/using/WS82af097c4236e2eb-3301d0fa130e80f7907-8000.html](http://help.adobe.com/en_US/reader/using/WS82af097c4236e2eb-3301d0fa130e80f7907-8000.html).
- [14] José Kahan and Marja-Ritta Koivunen. Annotea: An open rdf infrastructure for shared web annotations. In *Proceedings of the 10th International Conference on World Wide Web, WWW '01*, pages 623–632, New York, NY, USA, 2001. ACM.
- [15] Paolo Bottoni, Stefano Levialdi, Anna Labella, Emanuele Panizzi, Rosa Trinchese, and Laura Gigli. Madcow: A visual interface for annotating web pages. In *Proceedings of the Working Conference on Advanced Visual Interfaces, AVI '06*, pages 314–317, New York, NY, USA, 2006. ACM.
- [16] Beryl Plimmer, Samuel Hsiao-Heng Chang, Meghavi Doshi, Laura Laycock, and Nilanthi Seneviratne. iannotate: Exploring multi-user ink annotation in web browsers. In *Proceedings of the Eleventh Australasian Conference on User Interface - Volume 106, AUIC '10*, pages 52–60, Darlinghurst, Australia, Australia, 2010. Australian Computer Society, Inc.
- [17] Lily Díaz, Markku Reunanen, Blanca Acuña, and Atte Timonen. Imanote: A web-based multi-user image map viewing and annotation tool. *J. Comput. Cult. Herit.*, 3(4):13:1–13:11, April 2011.
- [18] Amazon. Notes in kindle. [http://www.amazon.com/gp/help/customer/display.html/ref=hp\\_bc\\_nav?ie=UTF8&nodeId=200953940](http://www.amazon.com/gp/help/customer/display.html/ref=hp_bc_nav?ie=UTF8&nodeId=200953940).
- [19] Jennifer Pearson, George Buchanan, and Harold Thimbleby. Investigating collaborative annotation on slate pcs. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services, MobileHCI '12*, pages 413–416, New York, NY, USA, 2012. ACM.
- [20] Bruna C.R. Cunha, Olibário J. Machado Neto, and Maria da Graça Pimentel. Movia: A mobile video annotation tool. In *Proceedings of the 2013 ACM Symposium on Document Engineering, DocEng '13*, pages 219–222, New York, NY, USA, 2013. ACM.
- [21] K. Tanaka, M. Iwata, K. Kunze, M. Iwamura, and K. Kise. Share me - a digital annotation sharing service for paper documents with multiple clients support. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 779–782, Nov 2013.
- [22] Katsuma Tanaka, Kai Kunze, Motoi Iwata, and Koichi Kise. Memory specs: An annotation system on google glass using document image retrieval. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, UbiComp '14 Adjunct*, pages 267–270, New York, NY, USA, 2014. ACM.

## VITA

NAME: Krishna Bharadwaj K N

### EDUCATION:

- M.S., Computer Science, University of Illinois at Chicago, Chicago, Illinois, 2015
- B.E., Computer Science, Visvesvaraya Technological Univeristy, Karnataka, India 2007

### PROFESSIONAL EXPERIENCE:

- Research Assistant – Electronic Visualization Laboratory (EVL), University of Illinois at Chicago, 2013 – present
- Research Assistant – Energy Resources Center (ERC), University of Illinois at Chicago, 2012 – 2013

### TEACHING EXPERIENCE:

- Lecturer, National Institute of Engineering, Mysore, India, 2008 - 2012

### PUBLICATIONS:

- T. Marrinan, J. Aurisano, A. Nishimoto, **K. Bharadwaj**, V. Mateevitsi, L. Renambot, L. Long, A. Johnson, and J. Leigh. Sage2: A new approach for data intensive collaboration using scalable resolution shared displays. In Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on, pages 177-186, Oct 2014.