A Learning and Collaboration Platform Based on SAGE

Javier Delgado
Coll. of Engineering and Computing
Florida International University
Miami, FL, USA
1(305)348-4106
javier.delgado@fiu.edu

Mark Joselli
Instituto de Computação
Universidade Federal Fluminense
Rio de Janeiro, Brazil
55-21-88828788
mjoselli@ic.uff.br

Silvio Stanzani
Lab. of Architecture and HPC
University of São Paulo
São Paulo, Brazil
55-11-30915617
silvio.stanzani@yahoo.com.br

S. Masoud Sadjadi
Coll. of Engineering and Computing
Florida International University
Miami, FL, USA
1(305)348-3549
sadjadi@cs.fiu.edu

Esteban Clua
Instituto de Computação
Universidade Federal Fluminense
Rio de Janeiro, Brazil
55-21-26592646
esteban@ic.uff.br

Heidi Alvarez
C. for Internet Augmented Research
Florida International University
Miami, FL, USA
1(305)348-2006
heidi@fiu.edu

ABSTRACT
In this paper, we describe the use of a tiled-display wall platform for use as a general purpose collaboration and learning platform. The main scenario of emphasis for this work is online learning by users in different countries. We describe the general efficacy of this platform for our purposes and describe its shortcomings for this purpose empirically. We discuss its advantages and also the shortcomings that we found. We also describe an enhancement made to make it more viable for our target usage scenario by implementing an interface for a modern human interface device.

Categories and Subject Descriptors
K.3.1 **[Computers and Education]:** Computer Uses in Education – *collaborative learning, computer-assisted instruction, distance learning*

General Terms
Performance, Design, Human Factors

Keywords
Cyberinfrastructure, interdisciplinary, collaboration, e-learning

1. INTRODUCTION
Globalization has had a profound effect on the development process of many industries. For example, information technology businesses now globalize the development of software and even managerial positions are no longer limited to members in the home country of a company. The fast-paced growth of the Internet has been fundamental in this movement, as it has made remote collaboration tasks more efficient. However, many obstacles are still faced on a daily basis by workers in globalized environments. In order for future workers to be prepared for this kind of workforce, it is beneficial to get trained in a learning environment that utilizes the tools used by companies. In this work, we describe some of the challenges and requirements for this domain and provide an analysis of the usage of an integrated platform that can be used for collaboration and learning.

This paper describes our experience with SAGE as a learning and collaboration platform. We discuss ease-of-deployment, advantages and disadvantages compared to other solutions, and its general efficacy for this purpose. We describe two specific enhancements that we found necessary in order to make SAGE a more viable platform for collaboration. One was the remote desktop performance and the other was the human-computer interface, which we addressed by implementing an interface for a modern input device. To analyze the remote desktop performance, we performed tests to find where the bottleneck is and we report this. This project itself was carried out by members in different countries and different tools were used throughout our collaboration process, which allows us to provide a frame of reference for the shortcomings of other approaches, which were used extensively throughout this endeavor.

Several products have been released to facilitate collaboration activities that are carried out on a persistent basis by people in globalized workforces. These same tools can be used for remote learning. In this paper, we describe the use of a platform previously not fully exploited for these kinds of tasks. We use the Scalable Adaptive Graphics Environment (SAGE) [7] as our base platform. The SAGE software distribution was not necessarily designed for remote learning, but includes software that can be used for this task.

WCCCE '09, May 1–2, 2009, Burnaby, B.C., Canada.

The rest of the paper is organized as follows. In Section 2 we discuss related work. In Section 3 we present the learning and collaboration requirements. In Section 4 we describe the features of SAGE in terms of deployment as a collaboration platform. In Section 5 we give an overview of the efficacy of SAGE as a learning and collaboration environment. In Section 6 we discuss the remote desktop sharing. In Section 7 we present our input device implementation for SAGE. And finally in Section 8 we present the conclusions.

2. RELATED WORK

Skype [14] is an instant messaging application which provides teleconference facilities involving Skype clients, VoIP compliant devices and regular phones. Skype does not support video conferences involving more than two participants, which makes it impractical for lecture sessions; it is very effective for synchronous communication of project activities.

Enabling Virtual Organizations (EVO) [3] is a Java application which enables video conference meetings. EVO can be used to support lecture sessions and project group meetings. EVO enables a high level of interaction among remote participants by providing several collaboration modalities. A drawback to EVO, as with any application designed to write to a single display output, it the limit that exists in the output display resolution. As the number of remote participants increases, it becomes difficult to accommodate all the shared windows on a single display. Even if a large projector display is used, there may come a point in which the display area becomes saturated. In our GCB class, the lecture session consists of a professor and a group of four to eight students attending the class locally, and at least two remote sites with one to four students in each site attending the class via EVO. This results in at least three different cameras to be displayed on the local site (the local room and the rooms of each of the remote sites). In this sense, the learning interaction would be better if the students could count on the support of a tiled display wall, where all the cameras could be displayed, in high resolution, on a single large (integrated) display. This makes interaction easier as the number of remote participants increases. SAGE can fit this requirement.

3. Remote Collaboration and Learning Requirements and Facilitators

A collaboration and learning environment was created in the context of the Global Cyberbridges (GCB) program [4]. This program aims at enabling a collaboration infrastructure for e-Science using Cyberinfrastructure (CI). Currently, GCB consists of a number of distributed teams of two to three collaborators, with at least one person at each site in a specific team. The collaborators attend lecture sessions about special topics in high-performance and grid computing, and participate in subgroups where they conduct research and perform experiments alongside Cyberinfrastructure (CI) research scientists. The requirements for collaboration and learning were modeled based on the experiences of the GCB project.

The collaboration in GCB involves the following activities: a semester-long lecture on scientific computing, project task planning, project discussion, reporting, and result publication. Considering that all the participants work in different physical locations, and in different parts of the world (where time zones are different), the support for synchronous and asynchronous communication is mandatory. As a result, the functional requirements are the support for teleconferencing, video conferencing, display sharing, and the support for asynchronous activities, such as web forums, email lists, web logs, etc. The non-functional requirements are the Internet connection quality and display resolution quality.

During the learning activity of GCB, synchronous and asynchronous e-learning communication tools are used. The combination of both improves the interaction level and keeps the students motivated, as has been shown in [6]. The synchronous e-learning communication tools, such as video conferencing software, maintain a good interaction level, because it is possible to monitor the receiver's reaction to a message, improving the interaction. The asynchronous e-learning communication tools such as web forums increase the ability to process information, because the person involved in the learning activity can read, comprehend the message, research the topic, and then answer, ask more question(s), and/or make comment(s) [6].

Having taken the class and worked on the project, we were able to evaluate several communication technologies [5]. Teleconferencing tools are used for project planning and discussion and the message forum was used to discuss the results and specific issues on the tasks assigned to each participant. As the forum tool is used to keep track of activities, it can also be used to evaluate the progress of the project. The tools used for the support of synchronous and asynchronous communication are *Moodle* web forums [10], *Skype* for teleconferencing and instant messaging and EVO teleconference and video conference support.

The non-functional requirements comprise Internet connection quality and display resolution quality. The former directly affects the quality of interaction performing the synchronous collaborative tasks. The latter can improve the learning quality, which can take advantage of the quality of media used to support learning. In the context of the GCB experience, the Internet connection quality represented a tremendous drawback in the interaction. Desktop sharing, for example, was rarely used. Instead, the presentations (which were the most typical shared applications) and other materials were sent to the audience prior to the meeting. During the presentations, lecturers had to consciously dictate their position in the document (e.g. slide number in a presentation).

3.1 Specific Examples

3.1.1 Distributed Lectures

Online courses have become very popular, but with current commodity technology, conducting these lectures can still be burdensome for the lecturer. It is often necessary to use multiple tools at once. For example, separate tools may be needed for remote desktop sharing and for transmitting video of the lecturer and/or blackboard. Lecturers should ideally be able to see their entire audience (including those in other sites); this allows them to see if someone has a question or to observe facial expressions. For this, having an integrated teaching platform is beneficial.

It is typical to display graphics when teaching. In many cases, high resolution is not needed; for example, when showing flowcharts. However, there are cases in which higher resolution may help the audience to better analyze the graphics being shown,

e.g. showing aerial images of a city for a lecture on geography. If these types of images are shown by means of a projector, image quality suffers. A solution to this is to use a tiled display wall, which can be composed of standard (high-resolution) LCD monitors.

Another issue is scalability. As the remote audience size increases, more display space is needed for the lecturer to be able to see them all. Ideally, the audiences from all remote sites should be viewable from a single display. It also helps if the lecturer is able to control the entire wall from one place. Controlling the wall should require as little effort as possible.

3.1.2 Project Collaboration

Computer science classes typically have components in which a number of students collaborate on a project. In a class with a distributed audience, such as the GCB class, the members of each team will end up in different parts of the world. This is where the need for efficient collaboration tools is most apparent. The team, which may consist of students of different disciplines, not necessarily computer related, benefits from having a means of efficient collaboration using intuitive tools. Many of the same features described for lecturers are critical: high-resolution display, friendly interface to interact with whatever is being displayed, etc.

4. EVALUTATION OF SAGE FEATURES AND DESIGN

A collaboration platform should allow its users to efficiently and easily collaborate. This is seldom the case. Many integrated solutions exist for collaboration, but they are still prone to errors that make them difficult to use and operate. It is often necessary to have an experienced technician present during a "meeting" to set up, monitor, and troubleshoot hardware and software related issues. Some of the issues we faced with the software that we were previously using were broken remote desktop sessions and inexplicable connection problems.

SAGE is a large scale visualization platform developed by the Electronic Visualization Laboratory at the University of Illinois at Chicago. The desktop sharing, live video streaming, and high resolution image displaying software that comes with SAGE allows it to be used as a powerful tool for the support of synchronous collaboration and learning. SAGE provides the following key benefits:

- (*Scalable) High-resolution*. SAGE is one of the tiled-display wall deployment technologies currently available. Multiple screens from different systems can be joined together to form one large display. A novel feature of SAGE that differentiates it from, for example, XDMX [2], is that it allows applications to be written such that each system renders its own part of its display, while still allowing multiple applications to be open at a time. SAGE tiled displays have been shown to scale to over 50 screens [9] while maintaining good visual quality. This allows the viewing of multiple simultaneous shared desktops and/or video conference feeds, up close analysis of rendered images, and other collaborative tasks. In Section 3.1.1, it was explained how, as the size of the remote audience grows the display space needed to show the entire audience also grows. A scalable display modality makes it possible to stream video feeds for several audiences simultaneously, whereas the display area projected by a single projector can only project the audiences of 2 to 5 sites, depending on the size of the audiences. In our target environment, having a large amount of screen space allows users to view several applications at once.
- *Collaboration software included*. The SAGE distribution includes two video players capable of playing back streaming media; both can be used for video conferencing on a SAGE display. It also includes a Virtual Network Computing (VNC) client to allow collaborators to share their displays. This can be used for showing presentation slides, work being done on a remote user's computer in real time, etc.

4.1 SAGE Architecture

SAGE is separated into several components. It has a basic window manager, the "Free Space Manager" or *FSManager*, which keeps track of which applications (i.e. windows) are open and their location on the display. The fundamental difference between the *FSManager* and a traditional window manager is that the *FSManager* is capable of spanning multiple screens by mapping the physical "tiles" to specific sections of the wall. The communication is performed between the *FSManager* and "SAGE Receivers" which run on each node. If a node has multiple graphics outputs, it may run more than one tile, but only one receiver is required per tile.

Drawing functions in SAGE are limited to basic pixel and rectangle manipulation. Porting applications that rely on higher-level drawing functions, such as widget creation, filling enclosed regions of arbitrary shape, etc. to work on a SAGE display is not trivial. It is necessary to modify the application and/or the lower layer drawing libraries it uses.

To provide abstraction between individual SAGE applications and the window/display management, SAGE provides a separate interface, the SAGE Application Interface Library (SAIL). This component obtains pixel information from the application, determines which nodes need to draw which parts of the application's display and sends the appropriate pixel information to the SAGE receivers at those nodes. The SAGE receivers then send the pixel buffers to the tile(s). A message passing paradigm such as MPI can be used to specify which node is to render which part of the screen.

SAGE provides a distinct user interface model wherein the position, orientation, and size of the windows being shown on the screen can be changed from a separate system running one of their *UI clients*. These clients also handle actions generated by input devices. This decoupling of display from display interface allows multiple interfaces to control the same display. Also, multiple pointers may exist, which can be beneficial in collaboration scenarios.

4.2 Ease of Deployment

SAGE is available for the three most common desktop platform operating systems: Linux, Windows and OSX. Installation on Linux requires several third-party libraries, such as Simple Directmedia Layer (SDL), portable audio, and GLUT. These are generally not difficult to install for experienced system administrators. Packages are available in the repositories of most

major Linux distributions, and binary versions are available for Windows. SAGE can either use the entire display of a system or just a portion of it. If the whole display is used, another system must be used for user interaction. This paradigm may be difficult for some users, but it is beneficial when the display wall becomes very large, since all the windows in the display are consolidated into one application window in the workstation. It also allows multiple users to manipulate the windows of the display wall.

SAGE also requires the QUANTA toolkit, which is also developed at EVL. The compilation processes for QUANTA and SAGE are similar to other typical software applications. SAGE may require post-installation configuration as well. For example, access to the graphical displays of the tiles through SSH is necessary.

4.3 Cost

A tiled display wall requires a considerable investment, especially if a cluster is not already available. Several LCD monitors are required. Mounting hardware needs to be bought or fabricated to place the monitors. As with EVO, no specific video nor audio hardware is necessary. The graphics hardware used for the tiles should be powerful for best performance.

4.4 Drawbacks

Since a different display paradigm is used for SAGE, standard applications (which are designed to write to the underlying graphics library, such as X11 or Win32) do not work without modification. To fully exploit the scalability of SAGE, applications must be rewritten to work in parallel (i.e. parallel rendering and parallel display).

5. EFFICACY OF SAGE FOR COLLABORATION AND LEARNING

In the previous section, we described some of the general aspects of SAGE that make it attractive for remote collaboration and learning. The next step in our process involved actually using it for remote collaboration tasks. The available software was adequate for collaboration, for the most part. The quality of video conferencing and high resolution image rendering were good. However, we found the performance of two key features to be lacking. One of the features that was lacking was the VNC client. The human interface device (HID) support of the platform also seemed to be lacking for the purpose of collaboration, particularly for giving lectures.

A high-performance remote desktop client is essential for our target environment. One typical use case is for applications that display data that is viewed by all collaborators, e. g. students in different parts of the world watching slides associated with a presentation. The VNC protocol was designed for simplicity [13] and as a result, its performance over wide area networks suffers for some applications [1][8]. Part of our research consisted of testing the SAGE VNC Viewer using a presentation application.

A high-performance remote desktop modality also makes it possible to show non-SAGE applications running on a different virtual desktop of the local host. Since there is no network propagation bottleneck, good performance is potentially achievable, as long as no bottlenecks exist in the translation from the local display to the SAGE display. This is also analyzed.

When working with large displays, traditional input devices can become inefficient and difficult-to-use. The most common input device for manipulating windows in regular workstations, the mouse, is particularly unsuited for tiled displays since there may not always be a surface to put it on and it becomes cumbersome with large displays. To address this, we implement an interface for a modern pointing device.

6. REMOTE DESKTOP PERFORMANCE

The computer screen is very effective for depicting information, but projecting one's thoughts onto a computer is often not. When collaborating remotely, the audience physically present is instantly able to see what the presenter is showing. Virtual collaborators are not able to do so unless a desktop sharing solution is provided. In terms of collaboration, the ability to create something once and redistribute it instantaneously is very efficient. Combining a SAGE desktop with a remote display paradigm furthers the efficiency. But it is important for the software to perform adequately. A speaker should assume that whatever depiction (e. g. a slide from a presentation) is being explained is visibly available at the collaborating institution(s). Since sharing presentation slides is the most common desktop sharing task, we use this as a benchmark for assessing the quality of the remote desktop experience in terms of collaboration. This application requires short bursts of high-bandwidth data transfer (e.g. to show quick animations, such as slide transition effects).

6.1 Existing Remote Desktop Solution

In addition to the already-mentioned shortcomings of the protocol, the SAGE VNC Viewer client is not optimized for performance. A single node renders the image and SAIL propagates the changes to the affected display nodes. As the window gets larger (i.e. uses more screens), the performance degrades. The performance issue is partially mitigated by employing the “tight” implementation by Constantin Kaplinsky [9] to reduce the amount of data transferred. However, protocol-related optimizations do not address the problem of scaling to many screens.

Compared to other remote desktop paradigms, VNC performs very well when showing the display of systems connected through high-speed (e.g. gigabit) links. VNC has been shown to provide faster performance than more sophisticated remote desktop protocols, such as NX, in some cases [11]. VNC is considered a “low-level” desktop sharing approach, since it is pixel based. The other type of desktop sharing approach (“high-level”) is based on processing drawing instructions. In [11], the authors suggest that “high-level” protocols can be slower than the “low-level” protocols since the instructions used (and/or the protocols themselves) were designed for local displays; the fact that low-level protocols do not have to synchronize the state of the desktop is one possible reason why this is the case. However, to achieve adequate performance over a wide area connection, it is necessary to overcome the shortcomings of the VNC protocol.

Another advantage of VNC is that the protocol is based on image transfer. This makes it easier to port it for use with SAGE, since the design of SAGE is also based on simple bit maps. NX, for example, which is an open source alternative remote desktop paradigm, uses a more complex design. It uses a proxy, *nxproxy*, at the server side, to compress messages and an agent, *nxagent*, at the client side that caches and displays the images. It displays the

images in a window similar to the *XNest* [17] nested X server. Since *XNest* is essentially an X server that is a client of a parent X server, porting the NX client would involve porting most functions of the X Desktop API. Doing so would be time consuming and error prone.

The SAGE *VNCViewer* is based on a standard VNC implementation. It transfers its pixels to the SAGE display using a SAIL-defined function that transfers a pixel buffer from the application to the SAGE display. This is performed at a specified interval, which is given by the invoker of the *VNCViewer* at run time. By default, it is one frame per second. The VNC protocol is well-suited for SAGE since it is based on similar, simple drawing primitives such as rectangle drawing. Also, as with SAGE, it knows to update only affected regions of the screen. The only VNC feature that is not available in SAGE is the efficient handling of window "move" events [12].

6.2 Performance Evaluation

A series of tests were performed to gain empirical performance results when with the use of *VNCViewer* in different collaboration scenarios. As a benchmark, a presentation rendered using *OpenOffice.org Impress* was played back. Regular presentations consisting of basic text and graphics typically do not require a large amount of bandwidth. However, if there is some kind of animation or slide transition being used to help the audience grasp a topic, some information may be dropped by the remote desktop paradigm. This is especially true of "pull" models such as VNC, wherein updates to the frame buffer are requested periodically, depending on available bandwidth, rather than the server "pushing" a set of frames as is done with other protocols. Conversely, the push model can lead to de-synchronization between what a presenter is saying and what is actually being displayed. The reference presentation we created consists of several effects, including fading (of graphics and text), moving objects, and changing of background graphics. We specifically noted that VNC did not provide animations as smooth as NX did when used over a commodity broadband link. To measure the loss in display quality quantitatively, slow motion benchmarking as described in [11] was employed. In this technique, a remote desktop trial is run at a regular pace and then in slow motion. The slow motion case ensures that more information is transferred. In the end, the total bandwidth transmitted is compared between the two runs.

The following systems were involved in the experiments:

- *Mind*: A 16-node, dual Xeon compute cluster connected to a 15-tile SAGE display wall. Un-accelerated ATI Radeon Graphics cards are used in each node.
- *Mileena*: A Pentium 4 workstation using Gentoo Linux and *tightvnc* 1.3.9
- *Athena*: An Intel Core Duo workstation using Gentoo Linux and *tightvnc* 1.3.9
- *Simiano*: An Intel Core Duo workstation using Fedora Core Linux, version 9 and *tightvnc* 1.3.9.

In all cases, Mind was the system acting as the VNC client. *Mind* and *Mileena* are located in the same laboratory at Florida International University, but they are not connected directly to the same switch. *Athena* is located 22 miles away from the University and uses a Direct Subscriber Line (DSL) connection rated at 3 Mbps and 384 Kbps download and upload rates, respectively. *Simiano* is located in the University of São Paulo. For each test, regular-speed and slow-motion benchmarks were executed with Mind running a regular VNC client and with the SAGE VNCViewer at 1 fps and 20 fps. A video was created of the presentation being executed with automatic slide transitions every second, using *recordmydesktop* capturing at 24 frames per second. The slow motion video was generated by playing the video through *mplayer* at a rate of 1 frame per second and transfering the output to a separate file using the *yuv4mpeg* output. Bandwidth utilization was measured using *Wireshark*.

Figure 1 shows the average bandwidth utilized throughout the sessions. *Simiano* is not included since the performance was too slow. The average of five separate sessions were used, since the amount of data transfer varies due to there not being any specified update interval in VNC. It can be seen that when the VNC server is far away, a lot of data is lost, which results in poor remote desktop playback quality. Visual assessment agrees with this: the interactive features of the presentation were lost when *athena* was the VNC server. The slow motion benchmarks always transfer approximately the same amount of data regardless of the frame rate of the SAGE client. With non-local runs, a lot of information is discarded when there is a low frame rate on the SAGE client.

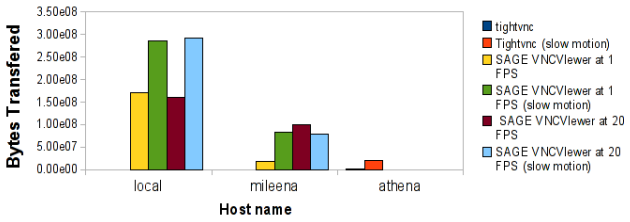

Figure 1. Amount of VNC-specific data transferred while playing a pre-recorded screen capture of our reference presentation. Data are shown for different hosts at regular speed and slow motion.

When comparing runs with different frame rates and with using the SAGE *VNCViewer* at regular speed, the amount of bytes transferred is not changed much in the local execution, whereas for remote desktop executions, having a faster frame rate results in more bandwidth being transferred. This was unexpected, since intuitively the lack of a network bottleneck on the local side would offset the advantage from having a higher frame rate. Conversely, when the VNC server was at a separate client system, several times more bytes were transferred when a higher frame rate was requested. This indicates an inability of the client to process all of the updates[1]. Another experiment was carried out at the University of São Paulo, with similar results. For this experiment, two systems connected through a LAN were used to measure the amount of bytes transferred during a VNC session. The following systems were involved in the experiments:

- *Simiano*: An Intel Core Duo workstation using Fedora 9 Linux Running SAGE Display on a single screen.
- *Willykit*: A Pentium 4 workstation using WindowsXP and RealVNC 4.1.2.

1 The VNC server pools update requests and combines them into one when it detects that the client is unable to handle each one

Figure 2 shows the amount of data transferred when using 1 and then 5 frames per second (FPS). Besides the larger initial peak for the 5 FPS run, there is no sign of it using more bandwidth overall. A test conducted afterward using a Secure Copy (SCP) transfer (Figure 3) showed that the link is capable of much higher bandwidth than was used by VNC, indicating a bottleneck at the client. The fact that CPU utilization increased supports this point. This indicates that VNC could benefit from a way of maximizing the bandwidth utilization.

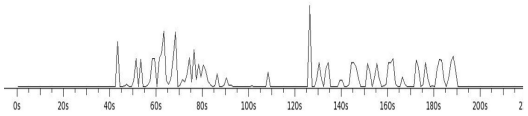

Figure 2. Two VNC sessions transferring identical content. The first was at 1 FPS (from 40 to 100 seconds) and the second was at 5 FPS

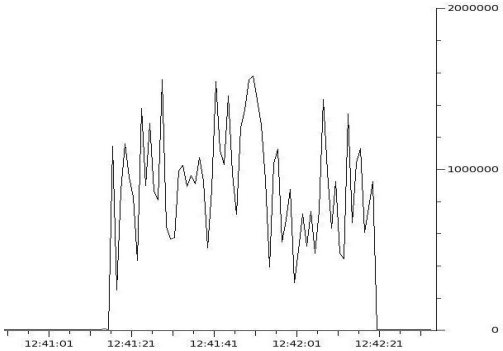

Figure 3. Using the same hardware setup as was used for the plot shown in Figure 2.

7. A MODERN INPUT DEVICE FOR DISPLAY WALLS

SAGE provides a new kind of platform and it benefits from new kinds of user interaction in order to exploit its capabilities. It supports traditional user interaction through the use of commands, a graphical application, or joysticks. None of these interactions are very useful for lectures when they need a direct interaction with the display and are not close to the computer. In order to have this kind of interaction, SAGE requires support for a new input device.

This section presents a new kind of input device that is more ergonomic than the traditional input devices supported by SAGE for working with the tiled display. A modern input device that fits this application is the Nintendo Wii™ Remote ("wiimote"). It provides a more natural interface to the display wall that does not require a surface on which to be placed. It also provides a similar interaction experience on an arbitrary display to touch-screen devices, which by experience have been very successful.

7.1 The Wiimote

The wiimote communicates with the Wii™ console or the PC though a Bluetooth wireless link. This link follows the Bluetooth Human Interface Device (HID) standard, which is the same interface used by traditional Bluetooth input devices, such as mice and keyboards. The wiimote sends state information with a frequency of 100 Hz and is composed of eleven input buttons, plus a shutdown button. It also has a special input though an infrared camera on top of it. This input has a resolution of 1024x768 and a field of view of about 45 degrees, and is able to "see" up to four IR points' positions. In order to use the IR input an IR source must be provided. In the case of Wii™ it uses the sensor bar, composed of two infrared light emitting diodes (LEDs). Based on the position information, the system creates a pointer device.

The wiimote has another special input, important for this work, that consists of a 3-axis linear accelerometer that detects movements along the three axes (in the three-dimensional Cartesian coordinate system) in the range of 3 g's (gravities) with a precision of 10%.

7.2 Implementation

The *wiiuse* [16] open source library was used for performing the low-level connection between the UI Client system and the wiimote. Both the IR sensor and the accelerometer sensor can be used with the SAGE display wall. The IR sensor would be the most intuitive pointer for the wall, but because a SAGE wall can potentially span several meters, the small field of view of the wiimote's sensor (45°) may be inadequate as a pointer for SAGE. As a result, we use the wiimote accelerometer and the wiimote buttons. The following inputs were defined and implemented: the accelerometer data of the X and Y axis are used in order to control the cursor on the screen. The "A" button is used to initiate the movement of application windows though the display. The "+", "-", are used to scale the size of the windows and the "home" button to rotate the applications. Also the cursor buttons can be used to navigate through the SAGE displays, providing a fast way to move across the wall, which can be difficult for normal joysticks when the wall is very large.

The functionality was implemented using two threads. The first one is responsible for getting the messages from the *FSClient* (a client of the *FSManager*) and process the message to manipulate the SAGE objects (e.g. the wall resolution, the SAGE applications positions, etc.). The second thread is responsible for getting the wiimote state, through *wiiuse*, from the bluetooth device, and sending messages to the *FSManager,* according to the wiimote input, as can be seen in Figure 4.

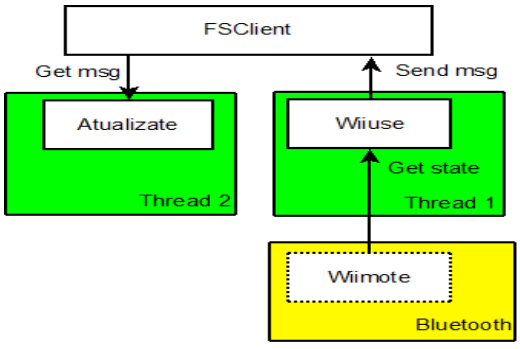

Figure 4. Two components, running in separate threads, interact with the *FSClient* and with the wiimote itself to form the interface to the display wall.

This application can also process input of messages through the keyboard, making it functionally similar to *FSConsole* (which is an application for the management of SAGE windows that provides a text-based interface to input commands to send messages to the *FSManager*). This is useful for the execution/shutting down of applications.

7.3 Evaluation of the Wiimote Interface

The use of the Wiimote in the SAGE environment provides an improvement in the degree of freedom when working with the SAGE display wall. Previously, the most common way to interact with the display wall and applications was through the use of the *SAGEUI* and the *FSConsole*. The SAGEUI is a graphical application that has the same functionalities as the *FSConsole*, but provides a more intuitive and user-friendly GUI. However, the *SAGEUI* requires that the user interact with a computer showing the current status of the wall (i.e. placement of windows). Referring again to the example in Section 3.1.1, the lecturer in this case would need to be in front of a computer showing the *SAGEUI*, which would give very little freedom in the case of needing a direct interaction with the wall if he/she is far from a computer.

The other way to interact to the wall is with the use of joysticks. These joysticks can only move and zoom applications. Also, normally these devices have wired interfaces to the computer, requiring the lecturer to be close to a computer. In addition, these devices are not made for user interaction. They normally need the use of both hands, and normally are not very ergonomic when used standing up. The use of a wiimote gives a new degree of freedom for the user interaction of SAGE. It is ergonomic, it only needs one hand for interaction, it is wireless, it allows the user to interact directly to the SAGE wall, and is capable of performing all the possible application windows transformations supported by SAGE.

8. CONCLUSIONS AND FUTURE WORK

Based on our evaluation, we see SAGE as a viable platform for remote learning and collaboration. With some optimizations to the remote desktop implementation, it could be used more effectively for this purpose. The *wiimote* interface gives users an improved interface to the display. Compared to other integrated collaboration solutions, it has the distinct advantage of supporting high-resolution graphics display.

Another feature we would like to add is a native web browser. Since many applications are written with a web browser as their graphical user interface, having a native SAGE web browser would allow users to use many more applications. This would be beneficial for collaborators and/or lecture sessions requiring special software that cannot be trivially ported to run on SAGE.

9. ACKNOWLEDGMENTS

The authors appreciate the support from Global CyberBridges, which is under the National Science Foundation office of cyber-infrastructure, CI-TEAM OCI-0636031. The NSF PIRE program, grant number 0730065, is also appreciated as it will help extend this research as well. Javier Delgado appreciates research grants HRD-0833093, CNS-0426125, CNS-0520811, CNS-0540592 and IIS-0308155.

10. REFERENCES

[1] Baratto, R.A., Kim, L.N., and Nieh, J. 2005. Thinc: a virtual display architecture for thin-client computing. In SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles, volume 39, pages 277–290, New York, NY, USA, December 2005. ACM Press.

[2] Distributed Multihead X Server (XDMX). http://xdmx.sourceforge.net

[3] EVO. http://evo.caltech.edu/evoGate

[4] Global Cyberbridges Project. http://www.cyberbridges.net/

[5] Hang, X., Villegas, D., Sadjadi, M., and Alvarez, H. 2007. Formative assessment of the effectiveness of collaboration in gcb. In Proceedings of the International Conference on Information Society (i-Society 2007), Merrillville, Indiana, USA, October.

[6] Hrastinski, S. Asynchronous and Synchronous E-Learning. EDUCAUSE Quarterly, vol. 31, no. 4 (October–December 2008). http://connect.educause.edu/Library/EDUCAUSE+Quarterly/AsynchronousandSynchronou/47683?time=1227443959.

[7] Jeong, B., Renambot, L., Jagodic, R., Singh, R., Aguilera, J. Johnson, A., and Leigh, J. 2006. High-performance dynamic graphics streaming for scalable adaptive graphics environment. In proceedings of the ACM/IEEE conference on Supercomputing, page 108, New York, NY, US.

[8] Lai, A. M. and Nieh, J. 2006. On the performance of wide-area thin-client computing. ACM Trans. Comput. Syst., 24 (2):175–209.

[9] Leigh, J. et. al. The global lambda visualization facility: an international ultra-high-definition wide-area visualization collaboratory. 2006. Future Gener. Comput. Syst., 22(8): 964–97.

[10] Moodle. http://moodle.org

[11] Nieh, J., Yang, S.J., Novik, N. 2003. Measuring thin-client performance using slow-motion benchmarking, ACM Transactions on Computer Systems (TOCS), v.21 n.1, p.87-115, February 2003. DOI=10.1145/592637.592640

[12] Remote Framebuffer Protocol. http://www.realvnc.com/docs/rfbproto.pdf

[13] Richardson, T., Stafford-Fraser, Q., Wood, K.R., Wood, A.H. 1998. Virtual Network Computing, IEEE Internet Computing, vol. 2, no. 1, pp. 33-38, Jan/Feb., 1998.

[14] Skype. http://www.skype.com

[15] Tightvnc . http://www.tightvnc.com.

[16] WiiUse. http://www.wiiuse.net

[17] XNest. http://www.x.org