Page 1 [Title Page]

Page 2 [What is it?]

Our new application presented here is a unique Tcl/Tk-based whiteboard system that facilitates discussions among multiple participants in a way no conventional presentation tool can achieve.

Our whiteboard has three unique functionalities. First, it allows the audience to participate in the discussion by enabling them to add objects on the presenter's screen. This makes presentations more interactive than with the conventional tools.

Second, our whiteboard can be used not only for the usual classroom or conference room setups but also for presentations among remote participants. It is made possible by the whiteboard's network communication capability that enables efficient transmission of various types of instructions by taking advantage of the scripting in TCL/TK.

Third, it provides the capability to quickly construct complex objects such as two-dimensionally laid out mathematical symbol structures, or widgets for manipulation on the screen. This functionality helps the participants to make those complex objects on the spot while conducting discussions and affix them to the presenter's screen.

Page 3 [Motivation]

Just like most users of computer presentation tools, we have been using the well known tools provided by Microsoft Office and OpenOffice in our classrooms and conferences. Our discussion subjects are mostly in mathematics, physics and chemistry.

However, we encountered many situations where the audience wished to write or draw on the presenter's screen to express their ideas. In those cases, the only way to satisfy their need was to provide a whiteboard to write or draw on, if a white board was available nearby. Even if this was possible, such a solution was not ideal, since in many cases direct drawing or writing on the presenter's screen was needed to clearly convey the participant's idea.

This is not the only problem we found with the conventional presentation tools. We also wanted remote participants to join in some discussions and we wanted to have a facility to quickly create and display complex structures on the presenter's screen. Thus, we started to develop our new presentation tool to solve these problems listed on the screen. The following slides describe the four new functions that we incorporated in our new tool to solve these problems.

Page 4 [New Function 1]

The first function we incorporated into our new tool was the capability to enable a two-way information flow so that the audience can participate in the discussion by constructing and embedding objects onto the presenter's screen. The current selection of such objects that can be created by the participants are texts, images, drawings, and widgets including those which can be manipulated on the presenter's screen.

Page 5 [New Function 2]

The two-way information flow that I mentioned is enabled by networking. Initially we designed the networking capability only for the participants at the presentation site. In that design the participants' computers were used only as a platform for sending objects to the presenter's screen. Their computer display was not linked to the presenter's screen to update and show the new objects added by the other participants.

There is no problem with this design in classroom and conference setups, since all the participants can directly see the presenter's screen, and there is no need to duplicate it on the participants' displays. However, this does not work out when there are remote participants who cannot see the presenter's screen in the room.

Later, we upgraded the tool so that such remote participants can also join the presentation just like those in the same conference room. To achieve this goal, we redesigned the communication scheme so that all the participants' displays were updated to show the newest state of the presenter's screen. The next slide shows an overview of the upgraded networking scheme.

Page 6 [Networked Two-Way Presentation]

The presenter's computer installs the server and all the participants' computers function as clients. The data-flow from the clients to the server is mostly for sending the objects to be displayed on the presenter's screen. The flow from the server to the clients is mostly for updating the presenter's screen that is duplicated on the clients' computers.

Various types of data are transmitted over our network. Therefore, for each transmission, a header is sent first which specifies the lengths and properties of the following data.

Page 7 [New Function 3]

In our presentations we often use special symbols that are laid out 2-dimensionally in complex structures such as sigma for summation and various integration symbols. It is tedious or sometimes impossible to make complex nested structures using the usual word processors or presentation tools.
Therefore, there must be a tool that provides the participants with such functionality and allows them to swiftly create and display special symbols and structures on the presenter's screen. The quickest and

most efficient way to make complex symbols is by handwriting. Therefore, we started to develop a handwritten-symbol interpreter by using some research results by computer vision researchers and incorporating that capability in our system.

Page 8 [New Function 4]

Unlike most presentation tools our whiteboard can instantly create and display widgets including ones that are manipulable on the screen. For creating those widgets, we developed the Widget Creator tool which will be described in a following slide. Currently, the types of widgets that can be created by the widget creator are text widgets, canvas widgets, scrollbars, and spinboxes.

Page 9 [Presentation Screen 1 (Original)]

This is an example of the presentation slide of our whiteboard. Unlike other presentation tools, it does not use the full screen. Instead we use a single toplevel window as the screen, because our system needs parts of the root window to accommodate other GUI tools for communicating with other
parties and creating objects to be displayed on the screen. All the slides are created using TK.

Page 10 [Presentation Screen 2 (Augmented by participant)]

This slide shows an example of objects added by a participant. The comment printed in red and the red arrow were added by participant Chris. This is an example of direct writing on the presenter's screen.
The parenthesis at the end of the comment shows the name of the participant and the index of this object among all the objects created by Chris.

Page 11 [Presentation Screen 3 (Augmented by participants)]

In this slide another participant Cathy added a canvas window with two scrollbars: horizontal and vertical. This is an example of adding manipulable objects, since the user can move the scrollbars to view other parts of the canvas. The name of the creator and the index of the object, "Cathy" and "1", are displayed at the top of the window. This canvas window set was created during the presentation using the Widget Creator which I will explain later.

Page 12 [Presentation Screen 4 (Augmented by participants)]

This slide shows the addition of a spinbox which is linked to a variable that controls the output to the canvas window created by the same participant. The label at the top of the spinbox window shows the name of the creator and index of the object among all the objects created by the same creator just like
the other added objects. In the case of the spinbox, the label also shows the name of the variable which is linked to the spinbox's displayed value. As the user changes the value of the spinbox, the output to the canvas - in this case the quadratic curve - changes as well . This is another example of adding a dynamically manipulable object.

Page 13 [Presentation Screen 5 (Augmented by participants)]

In this slide another participant Andy added a text widget with a scrollbar to show his lengthy opinion on the claim made by Cathy. The label at the top of the text widget has the same format as the labels of other added objects. This text widget set was also created using the widget creator during the presentation session as a response to the claim made by Cathy.

Page 14 [Master Controller]

This is the Master Controller designed to be used by the presenter. This tool provides the presenter with the ability to communicate with the participants through the communication windows, and control the discussion by changing the displayed page on the screen.

The presenter can advance or backup the page one by one at each press of the big arrows, or jump to a remote page using the spinner arrows in the spinbox. The communication windows allow the presenter to quickly converse with select participants in a troubleshooting or moderating role. While this communication facility is not necessary in small conference or classroom setups, it plays a vital role when holding discussions with remote participants. The presenter's messages can be addressed to a select participant, but the conversation can be viewed by all parties. Each participant also has a similar tool to this Master Controller. But the client's controller does not have the page changing capability.

Page 15 [Draw/Write Tool]

This is the Draw/Write Tool available to all participants as well as the presenter. It provides the functionality for creating both graphic and text objects, including the capability to create complex symbols from handwriting. From left to right, the top row of buttons provides the standard capabilities for drawing rectangles, ovals and lines, free-hand drawing, inserting texts, and erasing. The "Hand Write" button opens a new "Write" window in which the user can draw a symbol to be converted into a clean font. The "Write" window will be shown in the next slide.

Page 16 [Hand-Written Symbol Interpreter]

This is the GUI for handwriting symbols. To convert the handwriting, the user simply uses the mouse to draw one of the acceptable symbols in the tool, and presses the convert button. The symbol will be converted to a clean shape and will be placed on the spot specified by a mouse click on the target plane. The size and the color of the symbol is set by the Draw/Write tool shown in the previous slide.

Page 17 [Widget Creator]

This is the Widget Creator tool I mentioned in the explanations of some previous slides. It enables the user to quickly create three types of widgets during the presentation: texts, canvases, and spinboxes.

Each type of widget has a particular set of relevant entry fields. I will show a few examples of creating widgets in the following slides.

The user can also determine the contents of the text or canvas widget by selecting a content they prepared beforehand, such as graphs, charts, images, texts or a combination of them. The "Contents" selector which is set at the middle of the Widget Creator provides this capability. The user can prepare and install any content into the Widget Creator beforehand, and select it to fill a text widget or a canvas widget during the discussion. The user can modify the contents if necessary using the Draw/Write tool. This facility enables the quick creation of complex contents even during the presentation. The user can also create the contents from scratch on the spot, using the Draw/Write Tool.

Page 18 [Canvas created by Widget Creator]

This slide shows an example of creating a canvas widget set using the Widget Creator. The Widget Creator generates the canvas widget set in the widget test-board window, once all the specifications are entered into each relevant field. The example chose a pre-made content "Quadratic 2" to be placed on the canvas. The user can prepare and install as many contents as they like in the tool. Simple modifications to the contents can be done on the widget test board if necessary.

Page 19 [Spinbox created by Widget Creator]

This is another example of a usage of the Widget Creator. It shows the creation of a spinbox linked to the canvas widget set created in the previous slide. The specifications relevant to this creation process are the seven fields at the bottom of the tool. The field for "Target ID" indicates it is linked to the previous canvas widget set. The "Variable Name" field specifies the name of the variable to which the value of the spinbox is assigned.

Page 20 [Details on the Three Functions]

So far I have explained what our whiteboard does and how it operates. In the following three slides, I will explain details on the three listed functions.

Page 21 [Behind the Networking]

We used the Tcl commands only for the networking-related functions. Thanks to Tcl, our development process was made much quicker and simpler.

As I mentioned on a previous slide, our whiteboard uses a header for the transmission of multiple types of data. The header contains the following information: types of data, length of each segment, owner ID, sequence index, and the optional extra field that can be used in special cases and unexpected future necessities.

We designed the system to reduce the amount of transmitted data as much as possible. The use of "wish" on both the server and client sides helps minimize the amount of transmitted code, since even small amounts of Tcl code can accomplish a lot of work on the receiving side. For the widgets created by the Widget Creator, the amount of information sent through the network is even less than that for a comparable Tk code, since the system only sends the data entered in the relevant fields of the Widget Creator, instead of the entire Tcl script needed to create the widget.

Page 22 [Behind the Hand-Written-Symbol Interpreter]

Many algorithms for handwritten symbol recognition are available from publications of computer vision. However, we found that no algorithm can make the system as good, or even nearly as good as a human's recognition capability. Therefore, with the currently available algorithms, the user needs to learn the limitations of the system and fit their writing style to it.

To implement the algorithm, we created modules written in C instead of Tcl script, since the algorithm is very complex and requires high computation power.

For the output from the hand-written-symbol interpreter, we are using the results generated from Tex, since the symbols we have implemented so far are all available from that system. If symbols and fonts are needed that are not covered by Tex, we will need to make our own raster images.

Page 23 [Behind the GUI and Widget Creator]

We created all of the GUI and dynamically generated widgets using Tk. Our use of Tk helped make a speedy development process and reduced the amount of code needed to be sent through the network. We used multiple interpreters for the safe independent operation of objects created by multiple participants.

Page 24 [Current Problems]

Finally, I will touch upon the current problems of our whiteboard. Our whiteboard is still in the development stage and has many weaknesses and shortcomings.

One of the problems is the slide creation process. It requires more time and skills than with the conventional presentation tools. Another weakness is in the graphic input capability. The current graphic input facility only has a minimum set of tools and must be substantially augmented to make it a popular presentation tool.

The Widget Creator's capability has to be augmented as well, since currently it can only generate three types of widgets.

For a smoother operation with the multiple interpreters, we feel the use of a thread extension may be necessary to prevent CPU hogging by some processes.

As for the handwritten-symbol interpreter, we need to cover more symbols, since currently it can interpret only three special symbols (sigma for summation, pi for products, and the standard integration symbol).

Another problem frequently pointed out by past users is the convenience on the client side. Our current system does not allow clients to view any pages other than the one displayed on the server. We are planning to install a capability that allows clients to view other pages on their display without affecting the server's screen.

Page 25 [Conclusion]

Our whiteboard is still version 0.1 and has some rough edges. It needs substantial refinement for public distribution. One of the biggest problems with the current version is that it requires TCL/TK knowledge to make the slides of the whiteboard. In order to make it a popular tool for everybody, this problem has to be solved first.

However, despite all the problems and shortcomings I mentioned, we believe our whiteboard has a high potential to become a popular tool for various fields, not just education, since there is no comparable system that can enable a networked two-way interactive presentation with dynamically manipulable objects.