Networked Digital Whiteboard with Handwritten-Symbol Interpreter and Dynamic-Display-Object Creator

Atsuhide Kobashi

Henry M. Gunn High School
Palo Alto, California

September 2009

What is it?

- Tcl/Tk-based whiteboard presentation tool with unique functionalities
  - Audience can participate in the presentation
  - Networked
  - Enables quick construction of complex objects for displaying during presentation

Motivation

- Our situation
  - Educational presentation in classes and conferences
  - Scientific subjects: math, physics, chemistry, etc.
- Problems with the conventional presentation tools (PowerPoint, Impress, etc.)
  - One way information flow
  - Remote (off-site) audience cannot participate
  - Lack of tools to quickly create
    - Complex symbol structures
    - Dynamic display objects
- Solution: Our new application

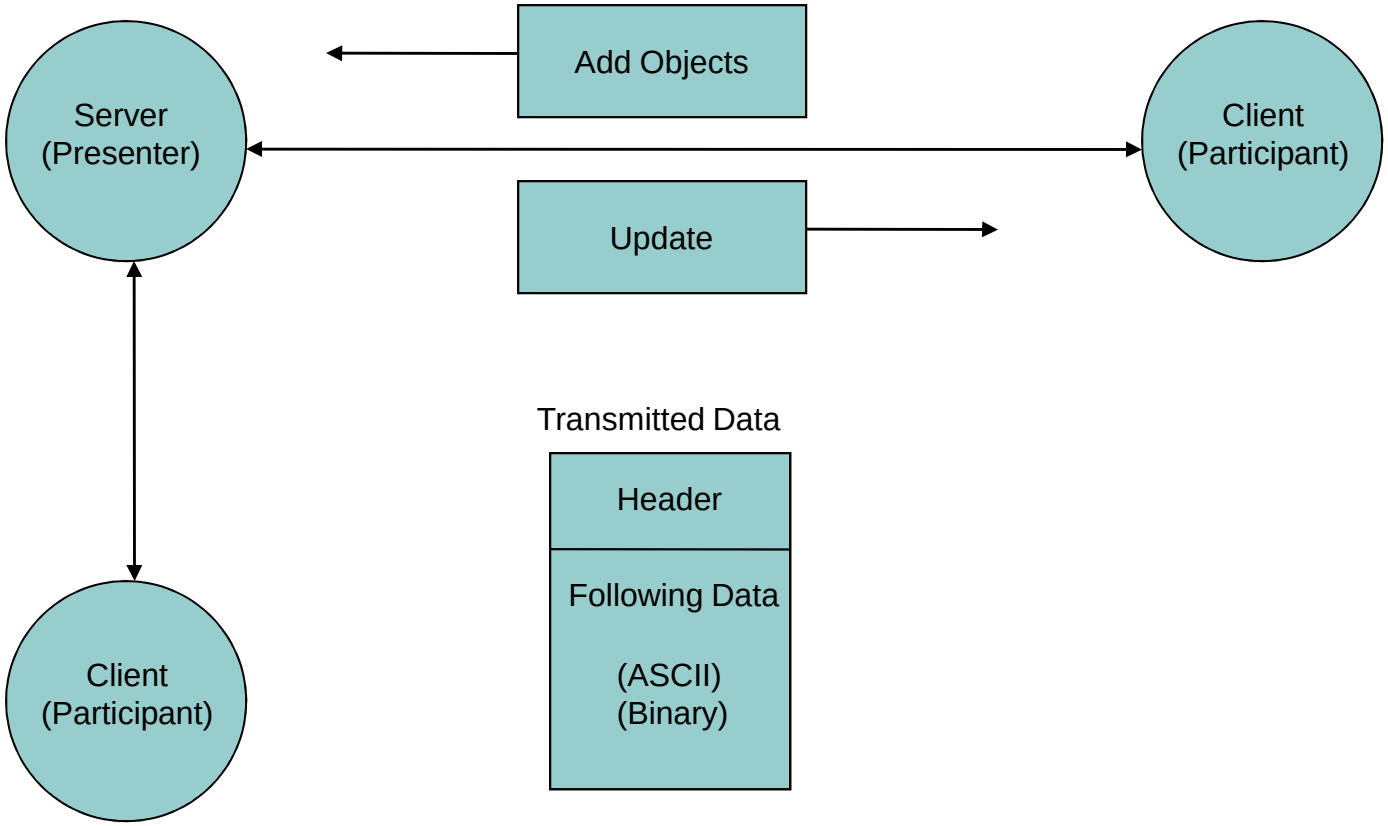
New Function 1

- Two way information flow
  - Enable audience to add objects on the screen
  - Types of objects to be dynamically created:
    - Text
    - Images
    - Lines, arrows
    - Rectangles, ovals
    - Widgets, both static and dynamically manipulable

New Function 2

- Communication through network with participants at both
  - Presentation site
  - Remote locations

Networked Two-Way Presentation



New Function 3

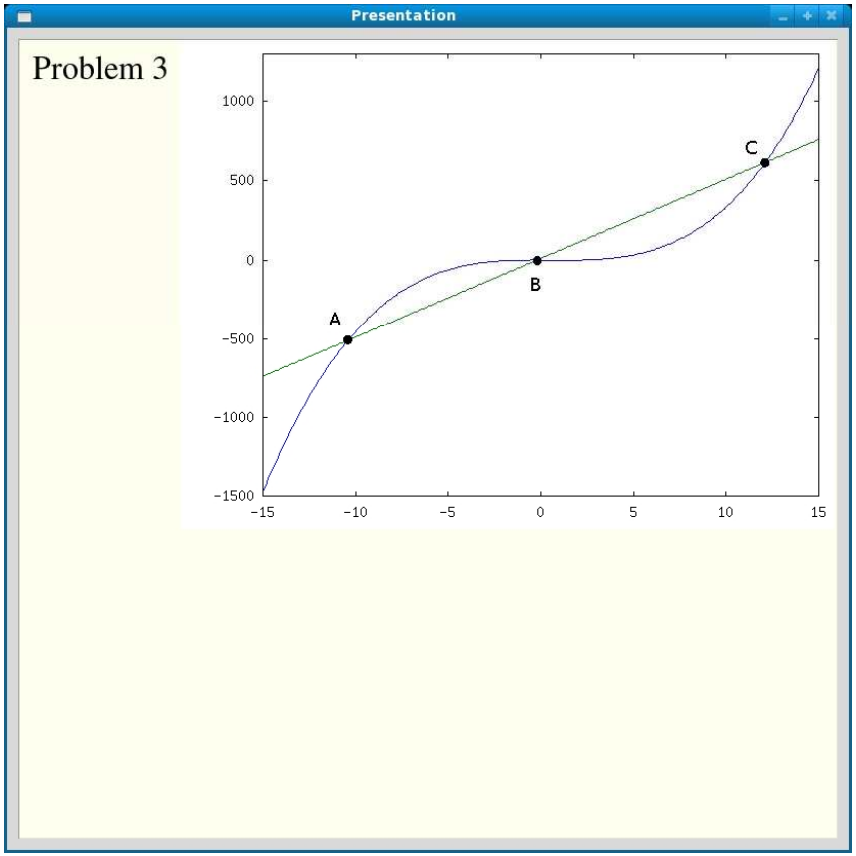
- Capability to efficiently and quickly construct complex symbol structures (for math, logic, scientific symbols) by handwriting

New Function 4

- Capability to quickly create widgets including dynamically manipulable ones

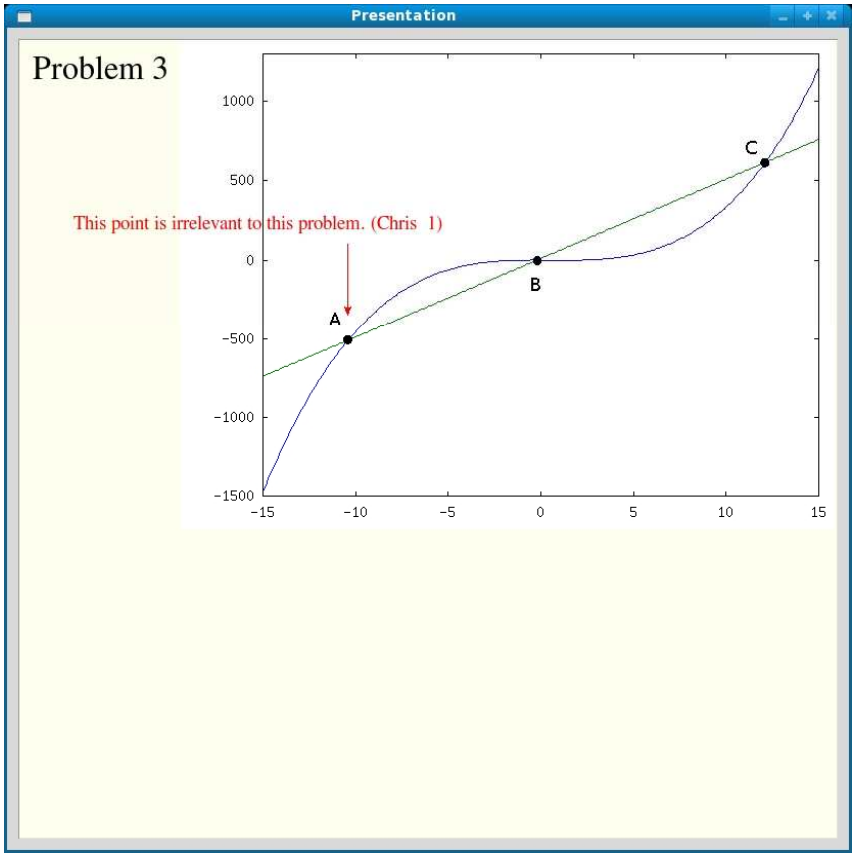
Presentation Screen 1 (Original)

- Tk-based window
- No full-screen mode



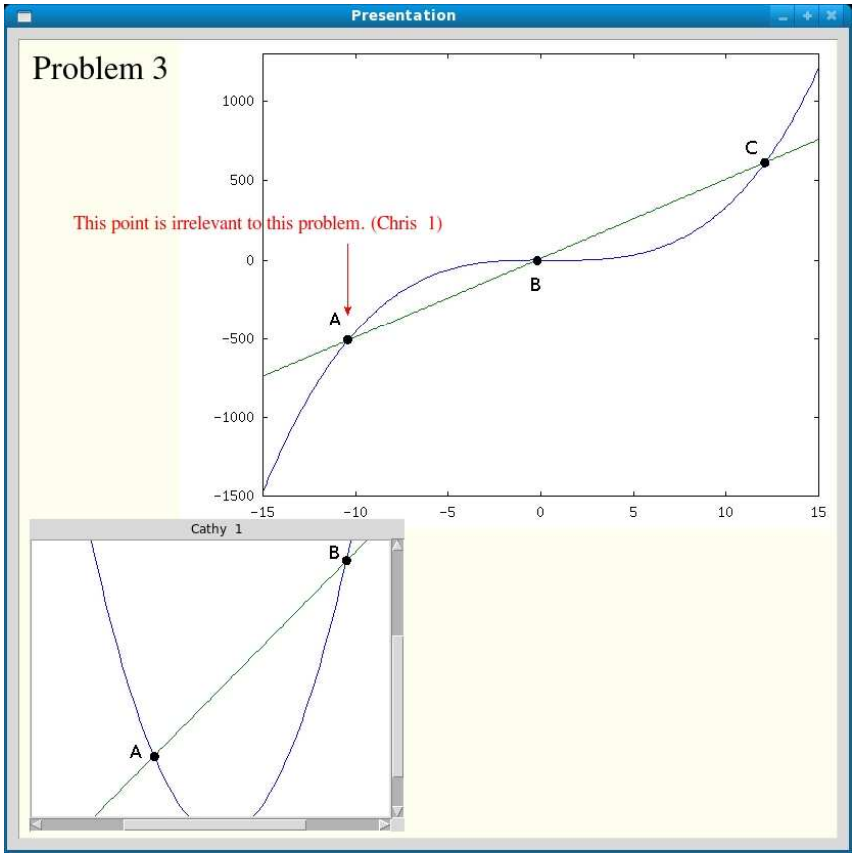
Presentation Screen 2 (Augmented by participant)

- Text and arrow added by a participant



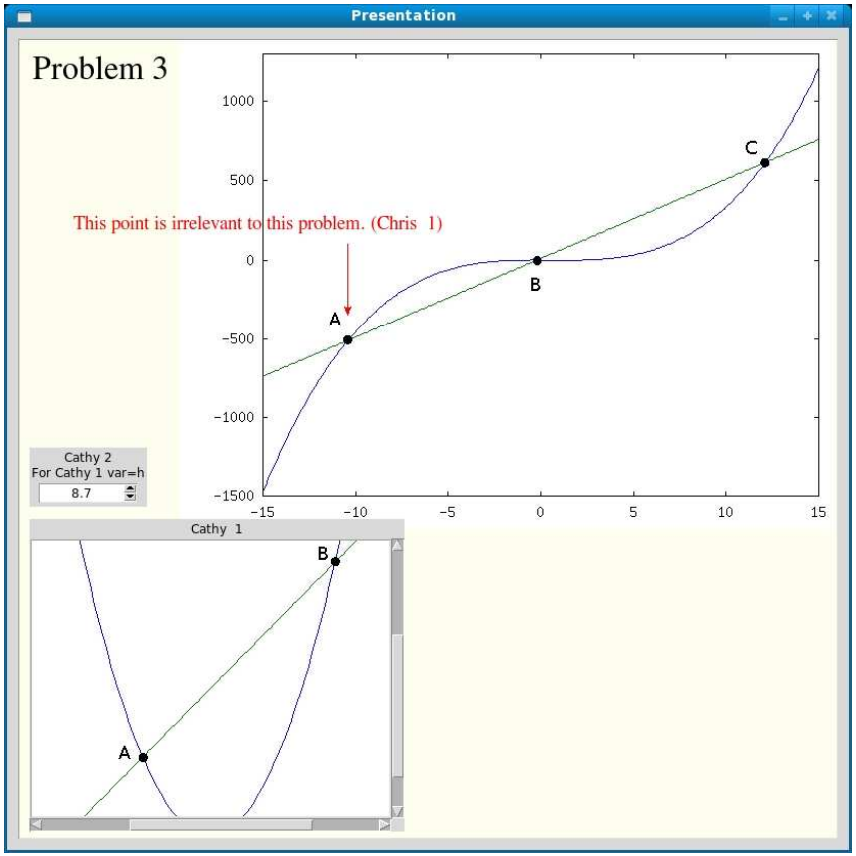
Presentation Screen 3 (Augmented by participants)

- Canvas window with scrollbars added by another participant



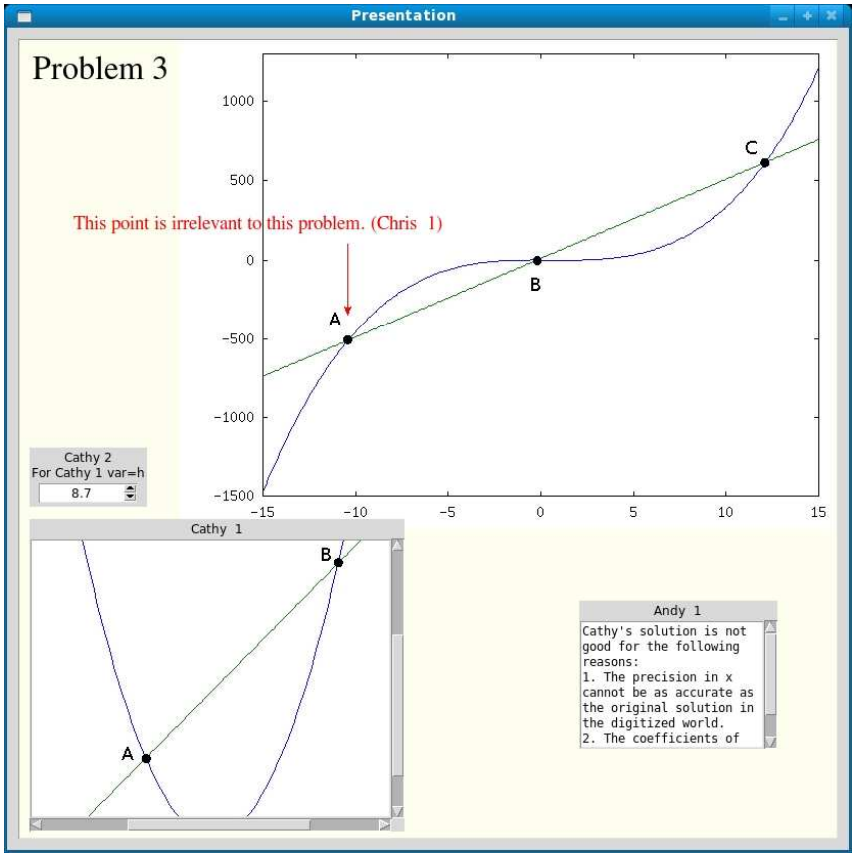
Presentation Screen 4 (Augmented by participants)

- Spinbox added by the participant who created the canvas window
- Spinbox linked to a variable that affects the output (quadratic curve) on the added canvas



Presentation Screen 5 (Augmented by participants)

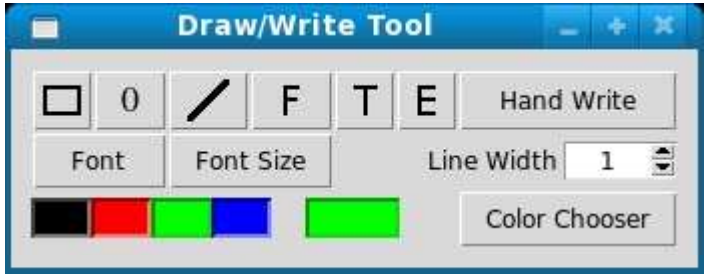
- Text widget window with a scrollbar added by another participant



Master Controller

- Used by the presenter
- Client side controller has no page-change capability

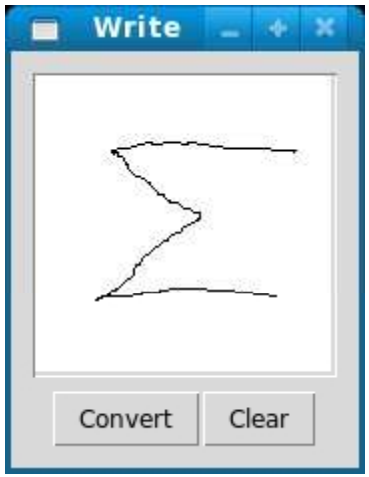
Draw/Write Tool



- Used for all drawing and writing including the handwriting of symbols

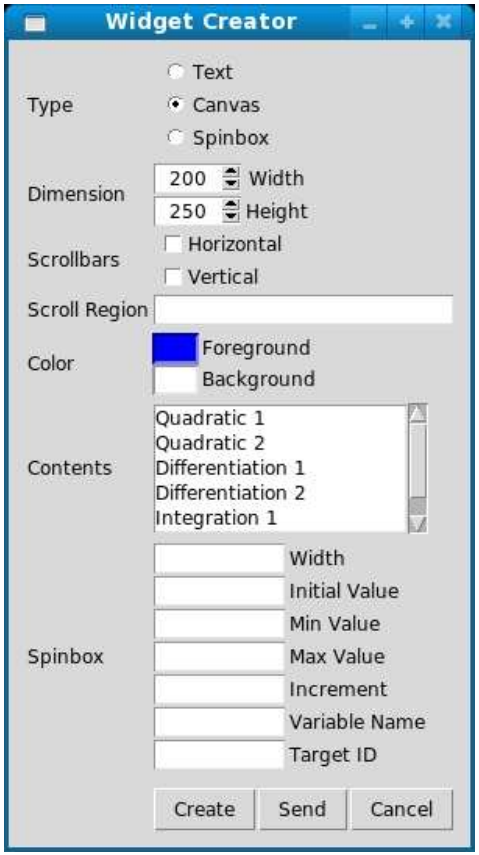
Hand-Written Symbol Interpreter

- Invoked by the Draw/Write tool
- Size and color set by the Draw/Write tool
- Location set by mouse click on the target plane

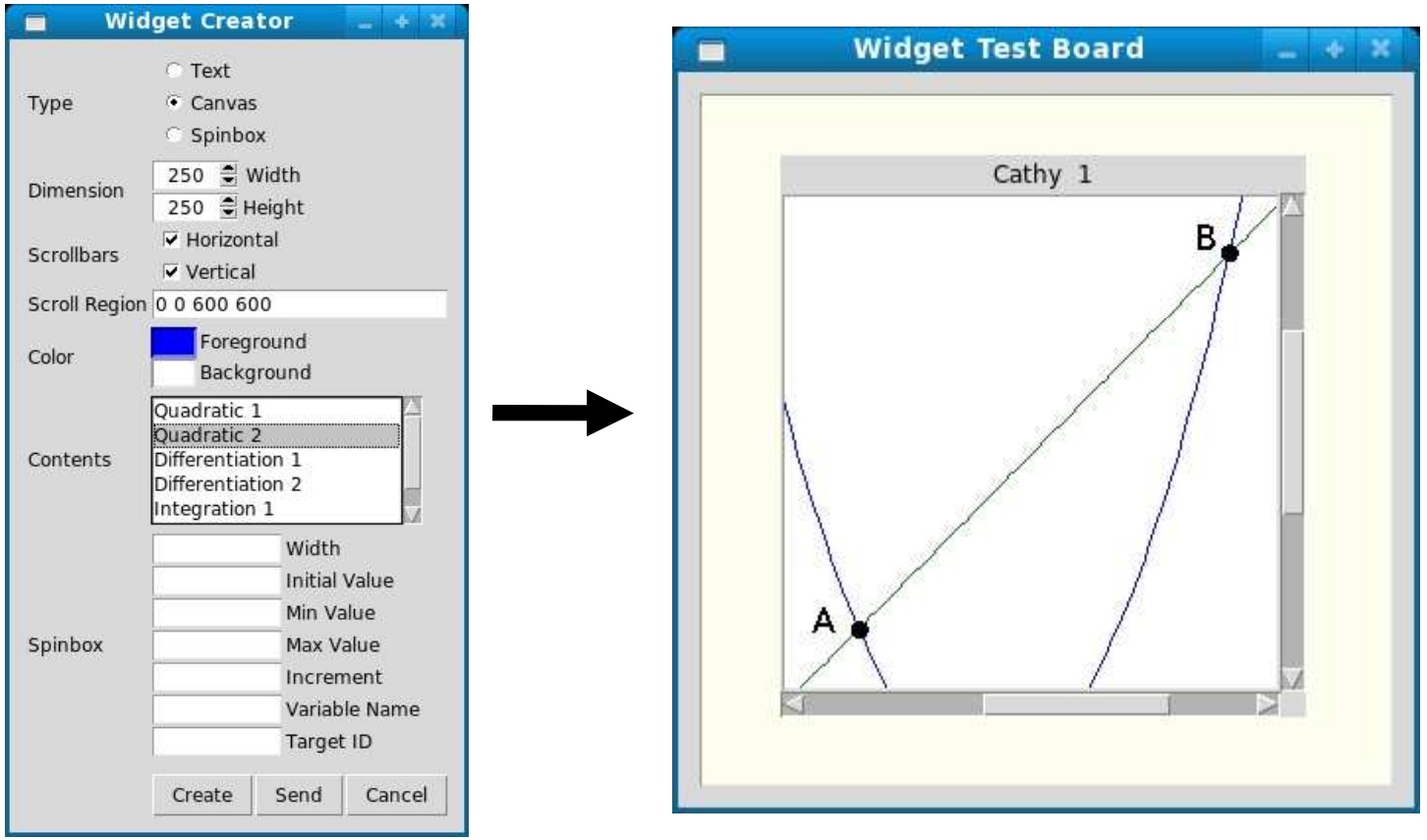


Widget Creator

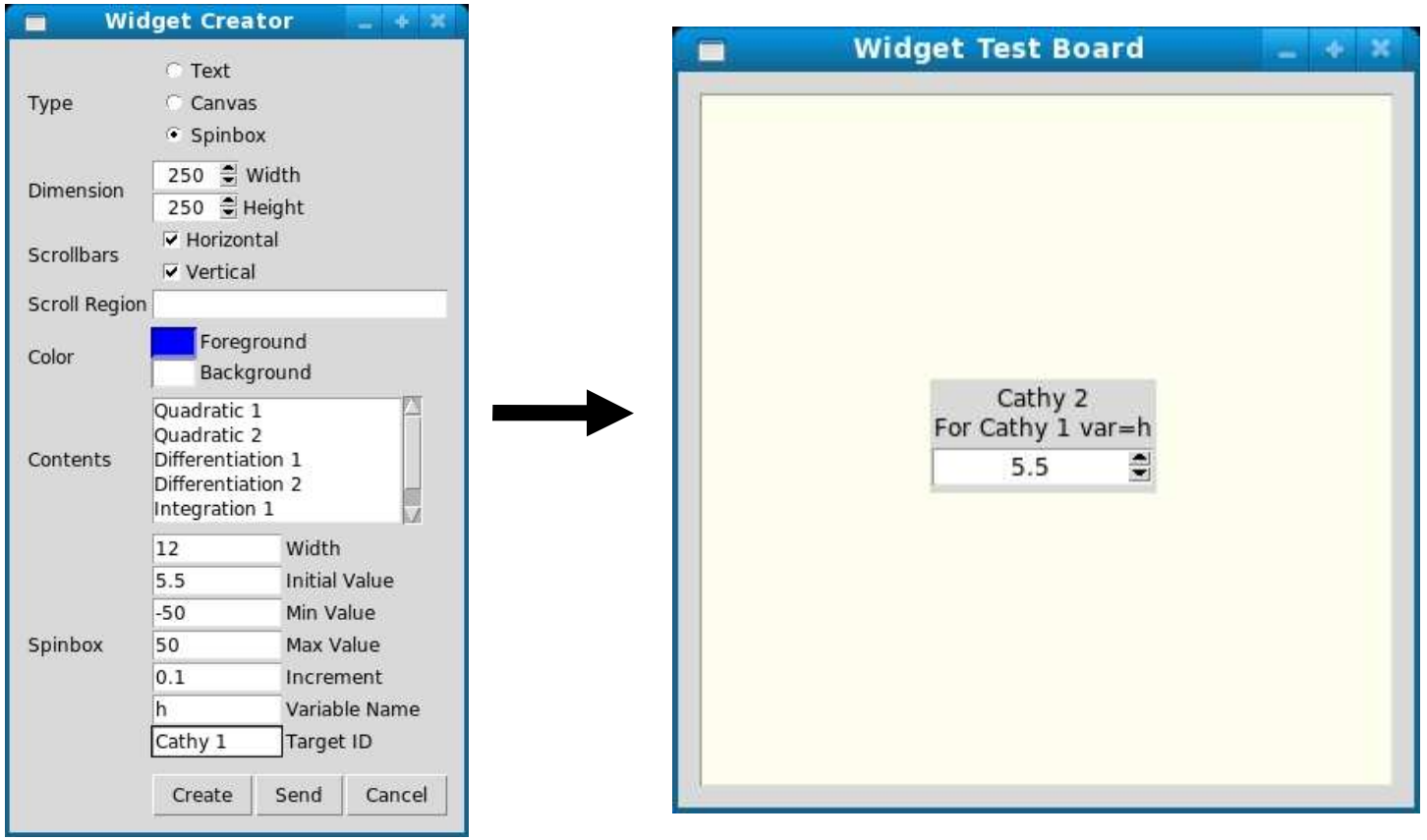
- Three types of widgets
- Contents selector helps quick composition
- Spinbox can be linked to a variable that changes output



Canvas created by Widget Creator



Spinbox created by Widget Creator



Details on the Three Functions

- Networking
- Handwritten symbol interpreter
- GUI and Widget Creator

Behind the Networking

- Use of TCL's network-related commands only
  - Easy to develop
- Use of custom header to indicate
  - Types of data (multiple combinations)
  - Length of each type
  - How to handle each segment
  - Owner ID and sequence index
  - Optional extra information field
- Economy on transmitted data
  - Use of wish by both server and clients reduces the amount of transmitted code (suitable for networking)
  - For the widgets created by the Widget Creator, only the specifications entered in the Widget Creator are transmitted

Behind the Hand-Written-Symbol Interpreter

- Many algorithms available
- No perfect or even near human level
- Requires lots of CPU power
- Use of C module needed for speed
- Still under development and modules for a small number of symbols developed
- Font for the converted results
  - Outputs from Tex used
  - For the symbols not available from Tex, custom raster symbol creation required

Behind the GUI and Widget Creator

- All GUI Tk-based.
- Merits of Tk
  - Easy and quick to create GUI
  - Small amount of code fits well for transmission
  - Easy to inspect the transmitted code in development stage, since it is written in human readable script.
- Safe independent operation for each participant guaranteed by multiple interpreters

Current Problems

- Overall slide creation process
  - Takes more time to make presentation pages than the conventional presentation tools
- Graphic input tools
  - More variation needed
- Widget creator
  - Wider variety of widgets need to be handled
- Multiple interpreter design
  - Use of threaded system desirable for smooth operation
- Hand-written symbol interpreter
  - Coverage of more symbols and better recognition algorithm needed
- Other functionalities
  - Page advance/backup capability on the client side without affecting the display of the server

Conclusion

- Current state still version 0.1
  - Rough edges exist. Refinement needed.
- Currently TCL/TK knowledge required to compose presentation slides. (Less so for clients to add objects.)
  - More convenience modules needed to compete directly with PowerPoint and Impress.
- High potential to become a useful tool for users in various fields, not just education.
  - No other tool can enable networked two-way interactive presentation with dynamically manipulable objects.