Shared Screens and Windows

Kimberly Tee
CPSC 781

Outline

- Introduction
- Implementation Strategies
- Technical Issues
- Non-Technical Issues
- Examples
- Future Work
Readings


Shared-View Systems

- share another’s screen/window
  - initially used to augment face-to-face
  - soon used for distributed, real-time collaboration
Two Approaches

- collaboration transparent
  - uses existing single-user applications

- collaboration aware
  - specifically designed for groups

Abstract Architecture

- system consists of:
  - participants’ window systems
  - application or screen being shared
  - manager

→ shared workspace
  - each participant sees the same view
  - each participant can interact with the application
Centralized Architecture

- system consists of:
  - participants’ window systems
  - one instance of each application or screen
  - one instance of manager

Centralized Environment

- example
  - window system sends input to application_1
  - application_1 produces output → view changes
  - manager distributes output request or changed view to all window systems
    - can be an image file
**VNC Protocol**

- basic message sent:
  
  *put a rectangle of pixel data at a given x, y position*

**VNC Optimizations**

- raw encoding (basic)
  - pixel data sent left-to-right

- copy-rectangle encoding
  - when framebuffer has same pixel data elsewhere
  - sent (x, y) where pixel data is copied

- majority-color encoding
  - describes rectangles of different colors
  - background and sub-rectangles
**VNC Updates**

- can encode each update using a different scheme
- choose most appropriate for:
  - screen content being transmitted
  - available network bandwidth

- demand-driven
  - only sent when explicitly requested
    - adaptive quality

**Replicated Architecture**

- system consists of:
  - participants’ window systems
  - for each participant, an instance of the manager and each application/screen being shared
Replicated Environment

- example
  - window system sends input to application<sub>1</sub>
  - input is distributed to other managers
  - replicas of application<sub>1</sub> receive input
  - replicas produce output
  - view changes
  - output request to window system

Tradeoffs

- replicated has better performance
  - but replicas must be synchronized
Technical Issues

- also want to support:
  - spontaneous interactions
  - group work
  - workspace management
  - floor control
  - data sharing

Spontaneous Interactions

- many interactions are spontaneous
  - should accommodate these interactions
  - minimize startup overhead
  - allow latecomers to join shared sessions
    - replay history of events
    - transfer shared state directly
  - apply process migration techniques
  - enable private windows to be later shared
    - dynamically put manager into communication link
Group Work

- support the mechanics of collaboration
  - provide telepointers
  - support annotation
  - channel for direct communication
    → without affecting shared application

Workspace Management

- traditional window managers not good
  - distinguish shared and private windows
    - identify windows in a particular session
    - determine which session a window is associated with
  - coordinate windows
    - WYSIWIS
    - handle window movements/destinations gracefully
**Floor Control**

- determines who has control of what
  - technological approaches
    - queue: keep list of people who request
    - pre-emptive: pass to requester on demand
    - ring-passing: give current floor holder control
    - open floor: allow anyone to hold floor at any time
  - social protocol
    - if high-quality audio available
    - let participants negotiate access

**Data Sharing**

- determine how data should be shared
  - possible for data to be overwritten
  - individuals may have different permissions
  - create copies of data
Non-Technical Issues

- customizing views
- social conventions
- group dynamics

RealVNC

- (free, open source) variant of VNC
  - no telepointing, annotation
  - scroll around to view
  - open floor control

http://www.realvnc.com/
**Bridgit**

http://www2.smarttech.com/st/en-US/Products/Bridgit/

- developed by SMART Technologies
  - annotation, webcam, spotlight tool
  - screen owner gets priority for control

**Examples**

**MSN Messenger’s Application Sharing**

- uses Windows NetMeeting
  - no telepointing, annotation
  - application owner can accept or deny requests for control

**Examples**
FaceTop

David Stotts, Jason Smith, Karl Gyllstrom

- two video streams over shared desktop
  - light-weight transition between communication and interaction
  - supports synchronous paired collaboration
  - can see facial expressions, gestures

Examples

Remote Access to Physical Devices

- extend VNC
  - remote access to devices
  - use standardized GUI protocol for devices that have no physical display of their own
    - when display becomes available, can then provide graphical information

Future Work
Using Shared Screens for Awareness

- class project – useful for awareness?
  - build shared screen application
  - support awareness → interaction

Future Work

Discussion

- as a viewer:
  - what do you want to see?
  - how much information is too much?
  - do you want to be notified if someone is working in a particular application?
  - do you want a recent history of what someone has been working on?

- as a sharer:
  - how much control over what you share?
  - what would you share?
  - visual indication to indicate what you are sharing?
  - times you don’t want to share?