Tcl / Tk as a Basis for Groupware

Mark Roseman
Department of Computer Science
University of Calgary

roseman@cpsc.ucalgary.ca

Tcl 93 Workshop
Berkeley, California
June 11, 1993
Overview

About Groupware

Personalizable Groupware

GroupKit

Tcl and Tk for Groupware

GroupKit in Tcl

Future Directions

Group Object Model

Overlays

Cross Platform
What is Groupware?

Technology *supporting groups* of people working together

Computer Supported Cooperative Work (CSCW)

Variety of systems

- electronic mail, Usenet News
- shared whiteboards, drawing programs, text editors
- desktop conferencing, media spaces

Same Time / Different Time and Same Place / Different Place

What is collaboration?

Why do we think computers can help us with it?

Computer Science, Sociology, Anthropology, Psychology, Management…

*Technology isn't hard — people are hard!*
Personalizable Groupware

People are different — groups infinitely more so

Entire group must accept groupware for success
  Good interfaces
  Match users' ever-changing needs — within and between groups

Personalizable groupware...
  ... allows different groups to use same system in different ways
  ... allows members of same group to use same system differently

Examples
  Floor control
  Joining a groupware session

Open Protocols
GroupKit — A Groupware Toolkit

Building groupware is a pain
  Technical obstacles
  Human factors difficulties
  Important to build quickly for evaluation

GroupKit
  Requirements are "programmer-centered" and "human-centered"
  Implementation in C++ and InterViews

Communications Infrastructure
Overlays
Open Protocols
Groupware and Tcl / Tk

Easy to build new interfaces quickly
  Quick evaluation, customization by "resilient end users"

Easy to prototype new "gidgets"
  Flexible event bindings

Canvas widget

Tcl Commands are Communications Protocol
  No encoding, message dispatching

Separate interface from application
  Put new "views" on underlying "model"
  Tie together via light-weight callbacks and tracing
GroupKit Revisited

Most of system redone in Tcl and Tk using Tcl-Dp

  brainstorming / voting tools

  shared whiteboard, structured graphics / hypertext editors

  a variety of session management interfaces

Nicer than InterViews version

  2000 lines of code vs. 20000

  much simpler to create simple applications

  modularity is a problem

  overlay support missing
Group Object Model

Need to handle shared group objects
  Lines, rectangles, etc. in a shared drawing program
  Nodes in a hypertext system

Handle primitive behaviors at the toolkit level
  Concurrency control (fine-grained), e.g. locking
  Distributing changes to all instances of objects

Tcl-DP distributed objects provide a good start
  Extend to work at “semantic level”
  Experiment with different concurrency models
Overlay Support

Overlays support generic actions over work surfaces

Act as transparent windows
- e.g. gesturing and annotation
- Should be easy to add to any application

Drawing from application to overlay
- Should be doable with minor changes to canvas

Input from overlay to application
- Can hack with generic event handlers
- Raises issues of composition
- Dependent on changes to Tk event handling
Cross Platform Issues

Cross-platform important for groupware
  group members often on heterogeneous systems
  field testing easier on Macs or PCs

Ideal solution is port Tk to Mac / Windows
  lots of X concepts embedded in Tk
  want native look and feel on other platforms

Practical solution is to keep lower levels the same

<table>
<thead>
<tr>
<th>Tcl + Sockets</th>
<th>Application Data Structures (C + Tcl)</th>
<th>Wrapper for App Model (Tcl)</th>
<th>Wrapper for Interface (Tcl)</th>
<th>Platform specific interface:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Tk</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- MacApp</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>- Windows SDK</td>
</tr>
</tbody>
</table>
Summary

Groupware difficult to construct

- Need good prototyping tools
- Need personalizable groupware systems

Tcl / Tk implementation of our groupware toolkit

Obstacles in Tcl/Tk for groupware

- Need high-level distributed object support
- Lack of fully transparent windows for overlays
- Event handling for overlays can only be hacked currently
- Cross platform development is not supported