RemoteJFC:

A Graphical User Interface Toolkit Approach to Thin-Client Computing

Computer Graphics and
User Interfaces Laboratory

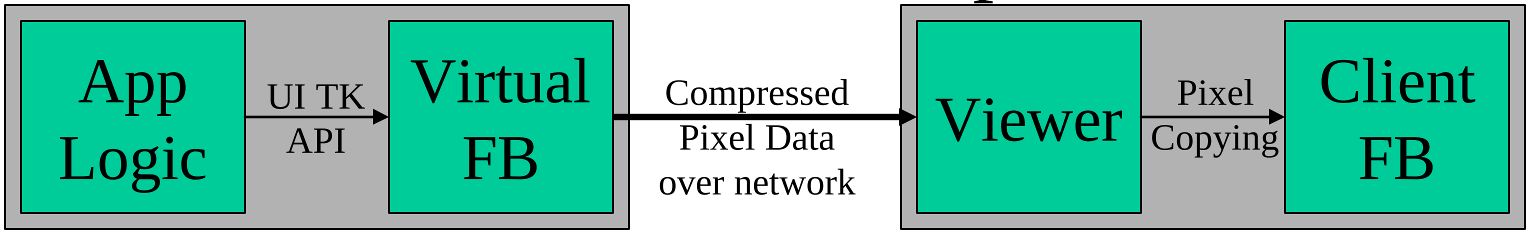
Columbia University

Motivation

- Multi-device UIs are difficult to implement
- Client side code approaches do not work
  - Distributed memory, deployment difficulties, reliability issues…
- Thin-client systems solve some issues
  - Shared state, rapid prototyping…
- Existing thin-client architectures are ad-hoc

Existing Thin-Client Techniques

- Virtual Frame Buffer Transport



App Logic

UI TK API

Virtual FB

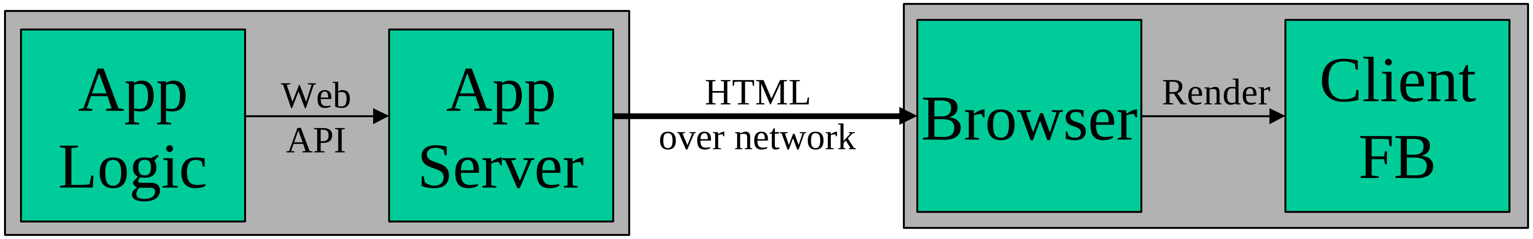
Compressed Pixel Data over network

Viewer

Pixel Copying

Client FB

- Web-based Application



App Logic

Web API

App Server

HTML over network

Browser

Render

Client FB

Goals

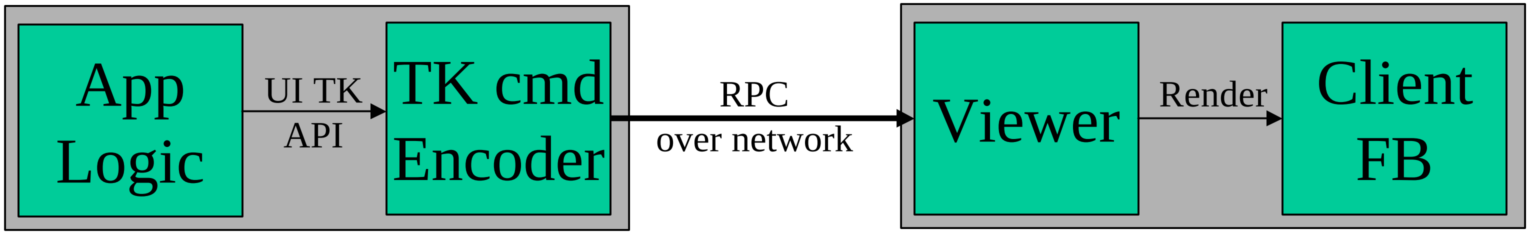
- Performance
  - Low bandwidth
  - Low latency
- Rapid prototyping
  - Shared/single state, no explicit memory synch
  - Standard/familiar API
    - Robust tool support
    - User buy-in

Goals

- Push capability
  - Time critical interaction
  - Consistent information
- Generic “viewer” software
  - No need to update client software when functionality changes

Our Approach

- Most apps are built on a UI toolkit
- Why not have a distributed UI toolkit?



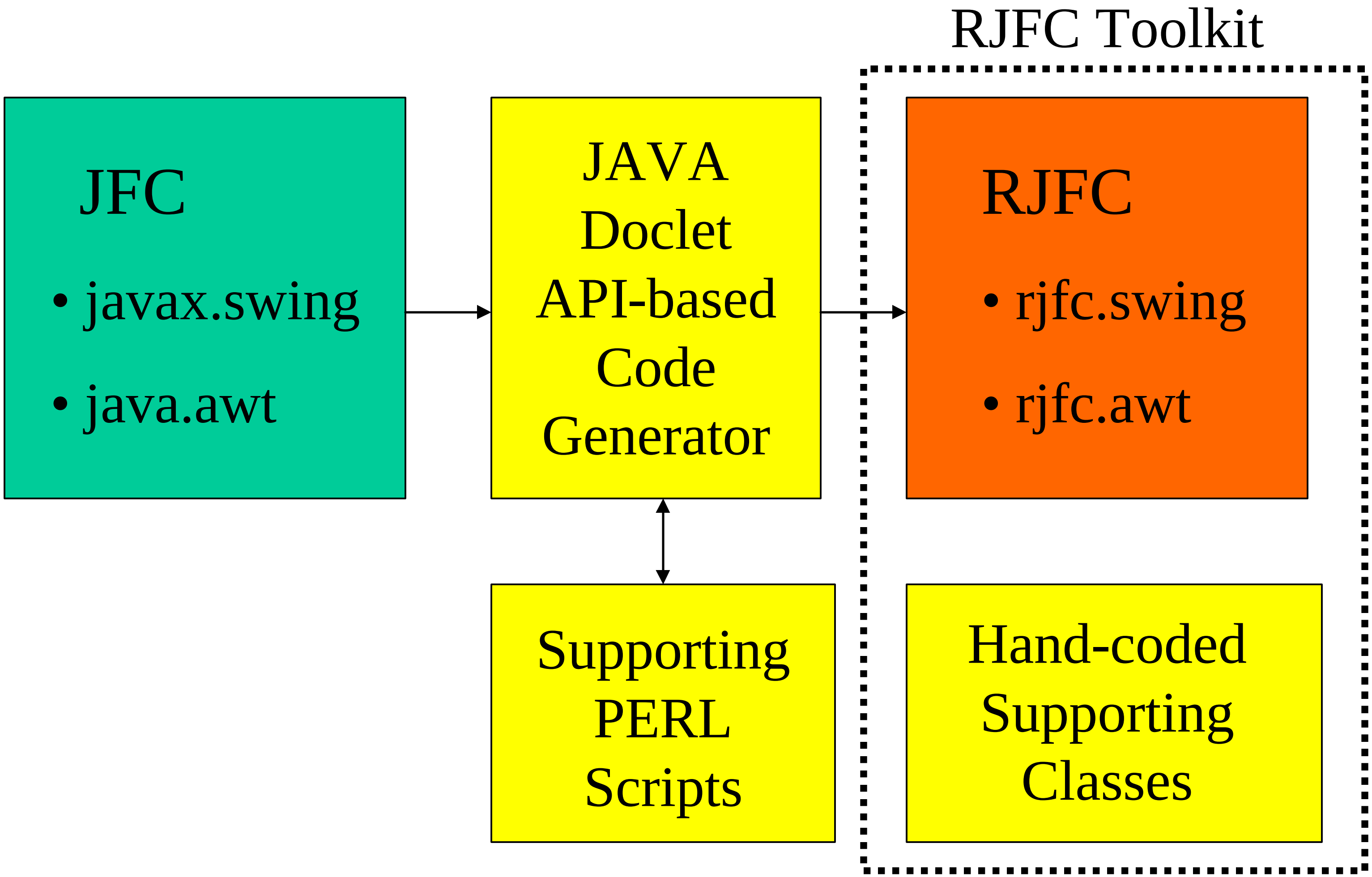
Related Systems

- X-Windows [Scheif86], NeWS [Gosling89]
  - Transport low level drawing commands (e.g. draw line from x to y) across the network
- Repo3D [MacIntyre98], DIV [Hesina99]
  - Distributed 3D graphics systems
  - Shared scenegraph but not thin-client

RemoteJFC Implementation

- Built on the JAVA platform
  - JFC/Swing is the baseline UI toolkit
  - RMI as the network transport mechanism
- System consists of…
  - Server-side library (~1.1M SLOC)
  - Client-side viewer (~2.5K SLOC)

Code Generation



RJFC Toolkit

JFC

- javax.swing
- java.awt

JAVA Doclet API-based Code Generator

RJFC

- rjfc.swing
- rjfc.awt

Supporting PERL Scripts

Hand-coded Supporting Classes

Code Generation

- For each class in the UI toolkit API…
  - Create a equivalent “distributed” class
  - For each method in the original class…
  - Create a method of the same name
    - In the body, call the method in the original API
- Majority of the protocol is implicitly defined by code generator

Support Classes

- RJFCFactory
  - Exposed interface for creating UI components
  - Lives on client, remote reference on server
- Viewer
  - Allow user to choose a server and application
  - Passes remote references of client-side objects to the server
  - Executes “real” UI toolkit code

RJFC API

- Programmer creates class SomeApp
  - extends ChildApplication
  - Must have method **`start`**
- Create the UI in body of **`start`**
  - Retrieve objects from a factory instead of instantiating them directly
  - Frame is passed to ChildApplication at instantiation

RJFC HelloWorld

```
public class HelloWorld extends ChildApplication {
  public void start() {
    try{
      RJFCFactory f = server.getFactory(clientInfo)
      RJFrame display = server.getDisplay(clientInfo);
      RJLabel label = f.getRJLabel(“Hello World”);
      display.getContentPane().add(label);
      frame.show();
    } catch (RemoteException e) {
    …
    }
}
```

Comparing RJFC and Swing

```
public void start() throws RemoteException {
  RJFCFactory f = server.getFactory(clientInfo);
  RJFrame d = server.getDisplay(clientInfo);
  RJTextArea TA = f.getRJTextArea(20,20);
  TA.addKeyListener(new TAKeyListener());

  RJScrollPane P = f.getRJScrollPane();
  P.setViewportView(TA);

  RJTextField TF = f.getRJTextField();
  TF.setEditable(false);

  RContainer c = d.getContentPane();
  c.setLayout(new BorderLayout());
  c.add(TF, BorderLayout.SOUTH);
  c.add(CreateMenu(), BorderLayout.NORTH);
  c.add(P, BorderLayout.CENTER);
}
```

```
public MyJFrame() {


  JTextArea TA = new JTextArea(20,20);
  TA.addKeyListener(new TAKeyListener());

  JScrollPane P = new JScrollPane();
  P.setViewportView(TA);

  JTextField SB = new TextField();
  SB.setEditable(false);

  Container c = this.getContentPane();
  c.setLayout(new BorderLayout());
  c.add(SB, BorderLayout.SOUTH);
  c.add(CreateMenu(), BorderLayout.NORTH);
  c.add(P, BorderLayout.CENTER);
}
```

Behind the Scenes

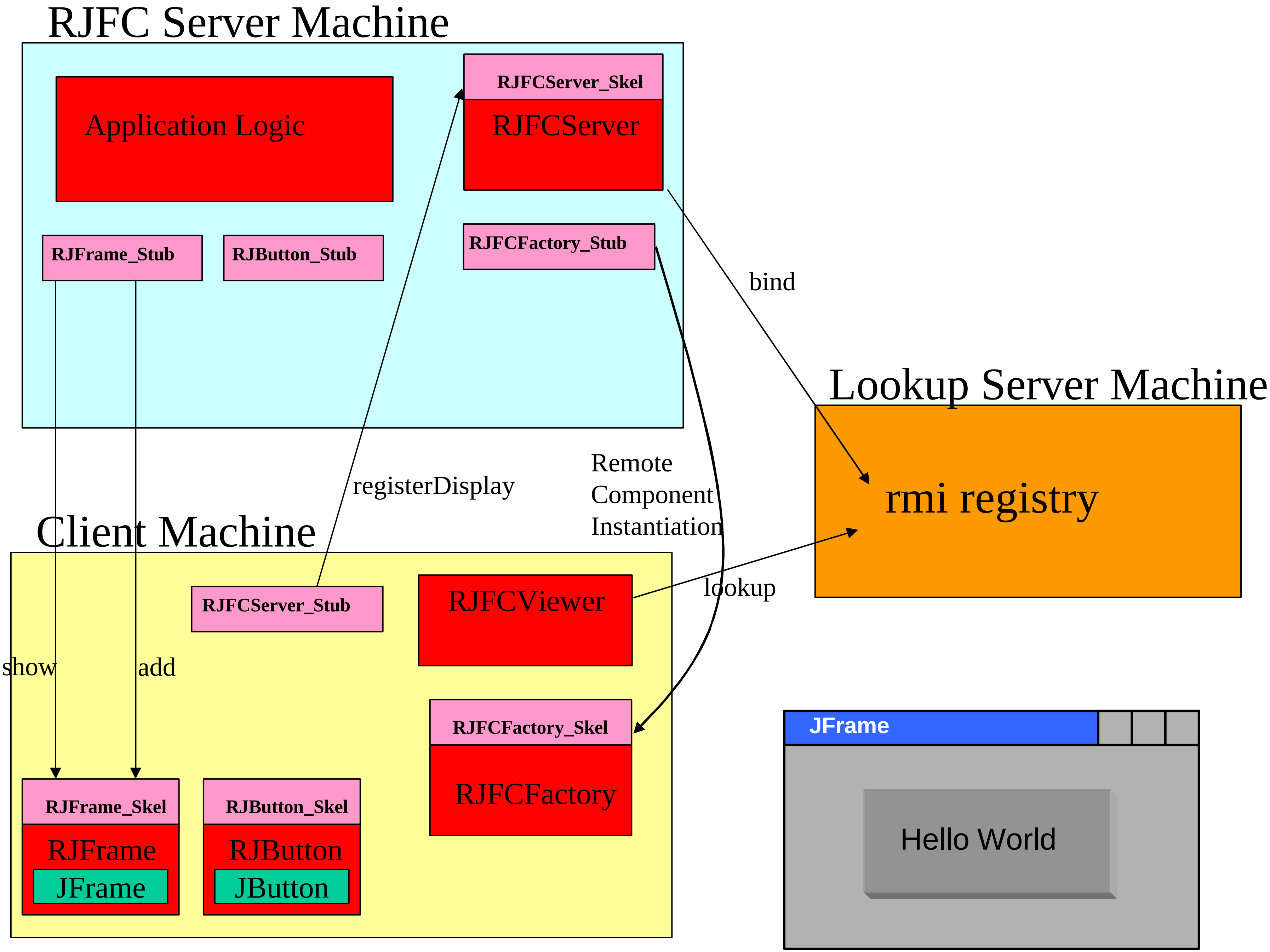
- User launches Viewer, connects to server
- Viewer executes the registerDisplay method
  - Remote references to RJFCFactory and RJFrame are passed to the RJFCServer
  - All operations on RJFrame and RJFCFactory are actually executed on the Viewer
  - Actual JFC components are instantiated by the RJFCFactory in the Viewer’s memory space

Where Everybody Lives

- Server-side:
  - – All application logic
  - – Remote references to…
    - RJFCFactory
    - RJComponents (including RJFrame)
- Viewer (client) -side
  - – RJFCFactory and all RJComponents
  - – Instances of all user created JFC componets

References to the “Real” UI

- RJComponents cannot contain references to the actual UI toolkit components!
  - RJComponents are constantly passed around
  - Create RJButton, pass RJButton to add() method in content pane of RJFrame
- Viewer-side Hashtable to keep track of RJComponent -> JComponent links



RJFC Server Machine

Application Logic

RJFCServer_Skel

RJFCServer

RJFrame_Stub

RJButton_Stub

RJFCFactory_Stub

bind

Lookup Server Machine

rmi registry

registerDisplay

Remote
Component
Instantiation

Client Machine

RJFCServer_Stub

RJFCViewer

lookup

show

add

RJFCFactory_Skel

RJFCFactory

RJFrame_Skel

RJFrame

JFrame

RJButton_Skel

RJButton

JButton

JFrame

Hello World

Performance - Bandwidth

Event	To Srv	To Client	To Srv	To Client	To Srv	To Client
Connect	724613	12553	12261	27343	69181	56623
Log In	0	0	82152	4687	0	0
Open Application	39607	9364	24613	4509	0	0
Idle (1 min, static mouse)	0	0	12660	6200	0	0
Idle (1 min, anim. mouse)	0	0	24810	9217	0	0
Idle (1 min, no mouse)	0	0	6390	2200	0	0
Idle (1 min, full screen)	0	0	1709	2960	0	0
Typing (1 min, 382 chars)	377159	135392	79617	74304	0	0
Cut Paragraph	125969	38786	79618	74304	0	0
Paste Paragraph	91811	29430	1437	1899	658	461
Copy Paragraph	153062	41224	2508	2979	295	460
Find in Paragraph	154306	40750	5157	2312	1390	1965
Save Document	187768	49384	10621	5684	1238	2004
New Document	60940	19498	1360	2123	689	875
Open Document	114686	25120	6590	3144	1423	1396
Resizing from full screen	741322	60056	180576	16185	0	0
Drag 1/4 size window	697433	64530	134016	212275	0	0
Drag mouse across screen	308324	99300	1471	3726	0	0
Tear down	9618	3006	1779	2097	1667	2210
	VNC		RDP		RJFC	

Performance - Latency

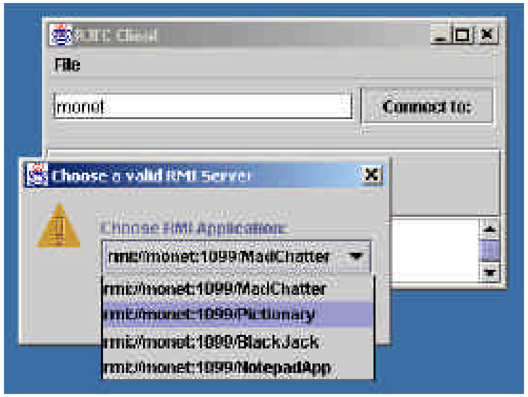
- Excellent performance once connected
- Startup latency is relatively high
  - Viewer calls registerDisplay() on RJFCServer
  - RJFCServer repeatedly calls methods on RJFCFactory (which resides on Viewer)
- Solutions:
  - "bank" / "bundle" calls to RJFCFactory
    - More difficult than it's worth due to JAVA language
  - Pipeline calls to RJFCFactory with threads
    - This is currently implemented in the latest RJFC

Additional Features

- Optional shared application footprint between multiple clients
  - Well-formed API for client connect/disconnect
  - Provides shared memory facilities
  - Enables community oriented shared applications (e.g. “chat” and “auction”)

Screen Shots

RJFC Viewer



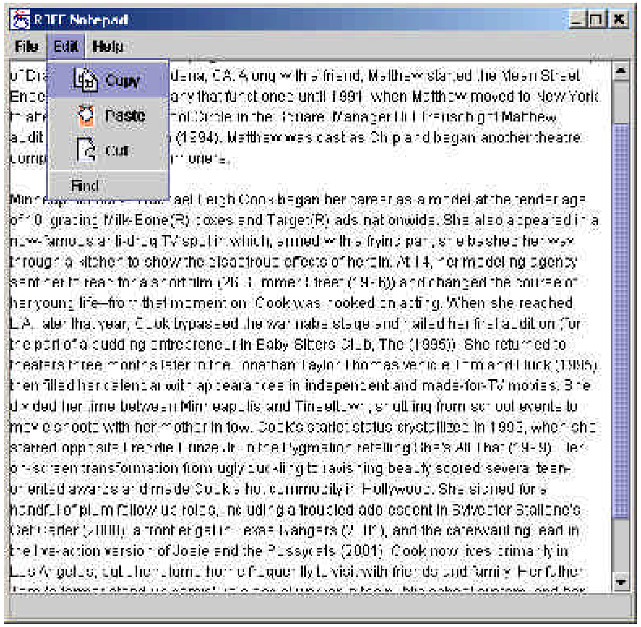
monet

Choose RMI Application:

rmi://monet:1099/MadChatter

rmi://monet:1099/MadChatter
rmi://monet:1099/Pictionary
rmi://monet:1099/BlackJack
rmi://monet:1099/NotepadApp

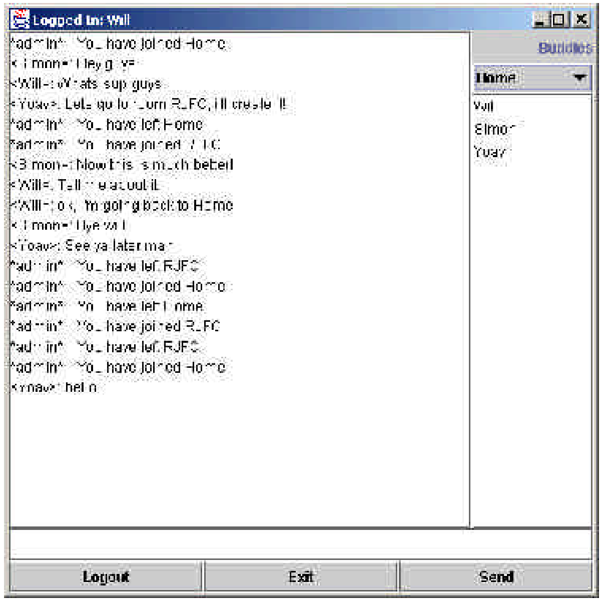
RJFC Notepad



u[illegible]dena, CA. Along with a friend, Matthew started the Mean Street
Ense[illegible]ory that functioned until 1991, when Matthew moved to New York
Theatre[illegible]nt Circle in the Square. Manager [illegible] Matthew
audit[illegible] (1994). Matthew was cast as Chip and began another theatre
comp[illegible] ...

Minneapolis native Rachael Leigh Cook began her career as a model at the tender age
of 10, gracing Milk-Bone(R) boxes and Target(R) ads nationwide. She also appeared in a
now-famous anti-drug TV spot in which, armed with a frying pan, she bashed her way
through a kitchen to show the disastrous effects of heroin. At 14, her modeling agency
sent her to read for a short film (26 Summer Street (1996)) and changed the course of
her young life—from that moment on, Cook was hooked on acting. When she reached
L.A. later that year, Cook bypassed the warm-up stage and nailed her first audition for
the part of a budding entrepreneur in Baby Sitters Club, The (1995). She returned to
theaters three months later in the Jonathan Taylor Thomas vehicle Tom and Huck (1995),
then filled her calendar with appearances in independent and made-for-TV movies. She
divided her time between Minneapolis and Tinseltown, shuttling from school events to
movie shoots with her mother in tow. Cook's starlet status crystallized in 1999, when she
starred opposite Freddie Prinze Jr. in the Pygmalion retelling She's All That (1999). Her
on-screen transformation from ugly duckling to ravishing beauty scored several teen-
oriented awards and made Cook a hot commodity in Hollywood. She signed for a
handful of plum follow-up roles, including a troubled adolescent in Sylvester Stallone's
Get Carter (2000), a frontier girl in Texas Rangers (2001), and the caterwauling lead in
the live-action version of Josie and the Pussycats (2001). Cook now lives primarily in
Los Angeles, but she returns home frequently to visit with friends and family. Her father
[illegible]

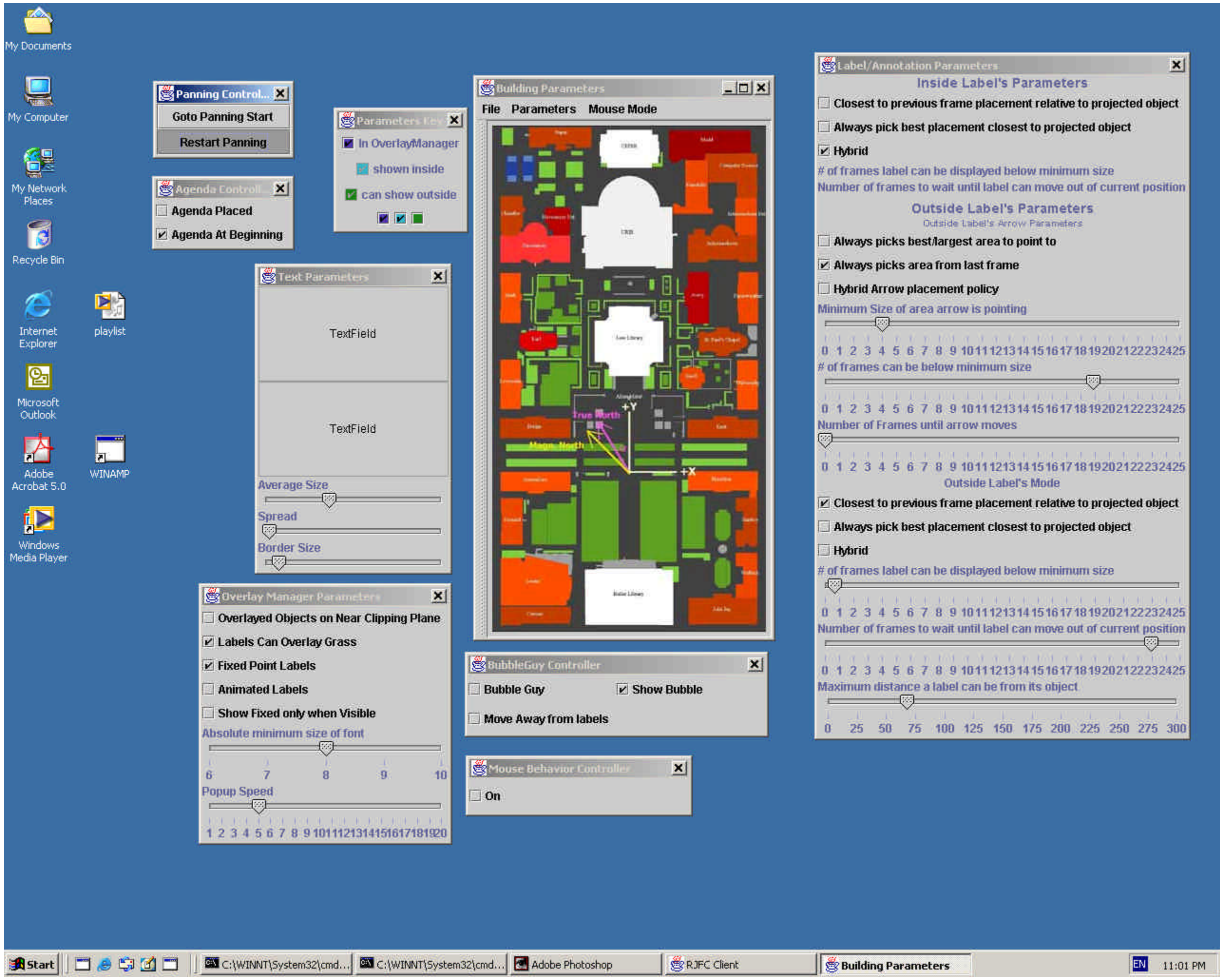
RJFC Chat



admin: You have joined Home
<Simon>: Hey guys
<Will>: What's sup guys
<Yoav>: Lets go to room RJFC, I'll create it!
admin: You have left Home
admin: You have joined RJFC
<Simon>: Now this is much better!
<Will>: Tell me about it.
<Will>: ok, I'm going back to Home
<Simon>: Bye will
<Yoav>: See ya later man
admin: You have left RJFC
admin: You have joined Home
admin: You have left Home
admin: You have joined RJFC
admin: You have left RJFC
admin: You have joined Home
<Yoav>: hello

Will
Simon
Yoav

RJFC In Action



Panning Control...

Goto Panning Start

Restart Panning

Agenda Controll...

☐ Agenda Placed

☑ Agenda At Beginning

Parameters Key

☑ In OverlayManager

shown inside

☑ can show outside

Text Parameters

TextField

TextField

Average Size

Spread

Border Size

Overlay Manager Parameters

☐ Overlayed Objects on Near Clipping Plane

☑ Labels Can Overlay Grass

☑ Fixed Point Labels

☐ Animated Labels

☐ Show Fixed only when Visible

Absolute minimum size of font

6 7 8 9 10

Popup Speed

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Building Parameters

File Parameters Mouse Mode

BubbleGuy Controller

☐ Bubble Guy ☑ Show Bubble

☐ Move Away from labels

Mouse Behavior Controller

☐ On

Label/Annotation Parameters

Inside Label's Parameters

☐ Closest to previous frame placement relative to projected object

☐ Always pick best placement closest to projected object

☑ Hybrid

# of frames label can be displayed below minimum size

Number of frames to wait until label can move out of current position

Outside Label's Parameters

Outside Label's Arrow Parameters

☐ Always picks best/largest area to point to

☑ Always picks area from last frame

☐ Hybrid Arrow placement policy

Minimum Size of area arrow is pointing

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

# of frames can be below minimum size

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Number of Frames until arrow moves

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Outside Label's Mode

☑ Closest to previous frame placement relative to projected object

☐ Always pick best placement closest to projected object

☐ Hybrid

# of frames label can be displayed below minimum size

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Number of frames to wait until label can move out of current position

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Maximum distance a label can be from its object

0 25 50 75 100 125 150 175 200 225 250 275 300

RJFC implementation of 3D AR view management control system [Bell, UIST01]

Conclusions

- Distributed UI research is under-utilized
- Most web-based applications would benefit from this using approach!

Possible Future Directions

- RJFC compiler
  - – Automatically generate RJFC from JFC code
- Viewer side caching of objects
  - – Allow the viewer to dynamically “download” objects that have executable code in them
  - – Draw on JIT and other compiler techniques
  - – Violates thin-client principles to some extent
- Framework for analyzing the IBC of UIs

Acknowledgements

- William Chiong and Yoav Hirsch
  - They did all of the “real” work
- Blaine Bell
  - First user, primary driver behind improvements
- Supported by:
  - NSF Grant IIS-98-17434 under DLI2
  - Donations from Microsoft and Intel.

Website

- For downloads and more information:
  - http://rjfc.cs.columbia.edu