SWEDISH INSTITUTE OF COMPUTER SCIENCE SICS

Ex11 - An Erlang GUI

Joe Armstrong
joe@sics.se

Why?

No good GUIs for Erlang

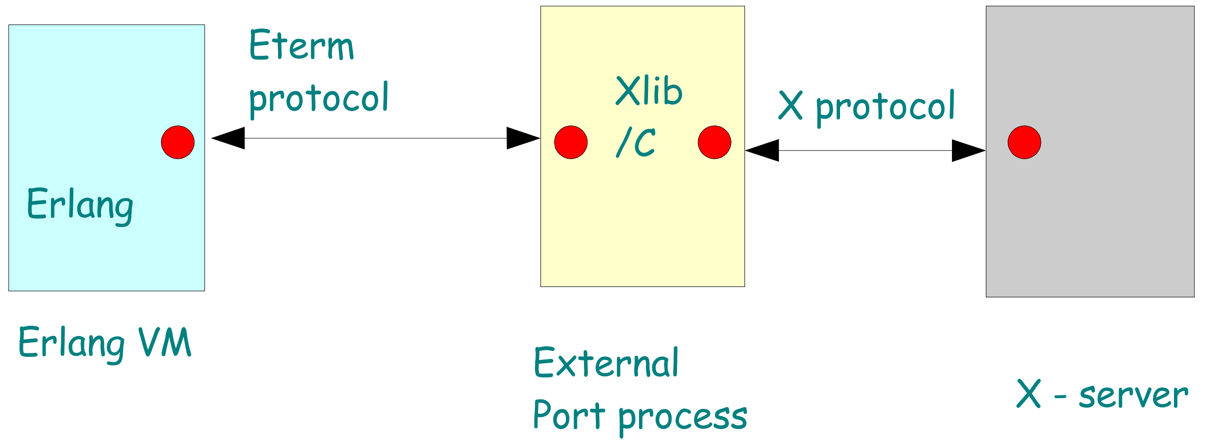
Erlang's message passing maps well onto X Protocol messages

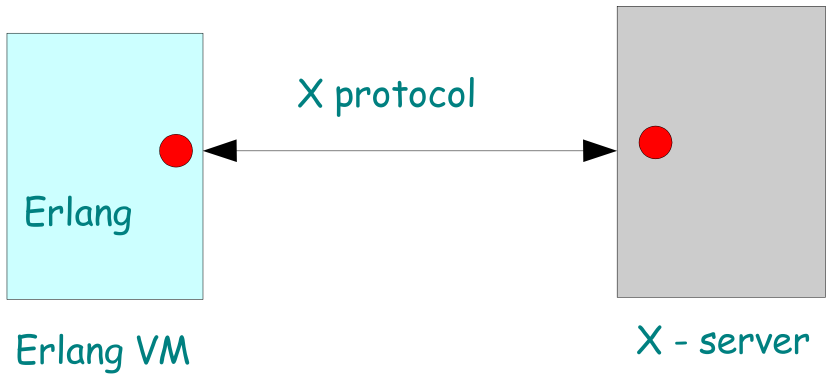
Windows/widgets are concurrent but this fact is not reflected in the API's

Fun

Interfacing Erlang to X widows

● Marshall/unmarshall code





Interfacing Erlang to X widows

+ Efficient
+ All code in Erlang
- Must re-implement a significant sub-section of xlib and make something like athena-widgets or motif

Interfacing Erlang to X widows

Xlib is complex

The X Protocol is simple

You don't need all of the X protocol to do useful stuff

Athena, Xtoolkits, Motif are complex, but unecessarily so

X Protocol

Port 6000 socket based TCP/IP protocol
(or unix domain sockets)

154 protocol message (mostly very simple)

Xlib
Lots of routines (800 ish) -

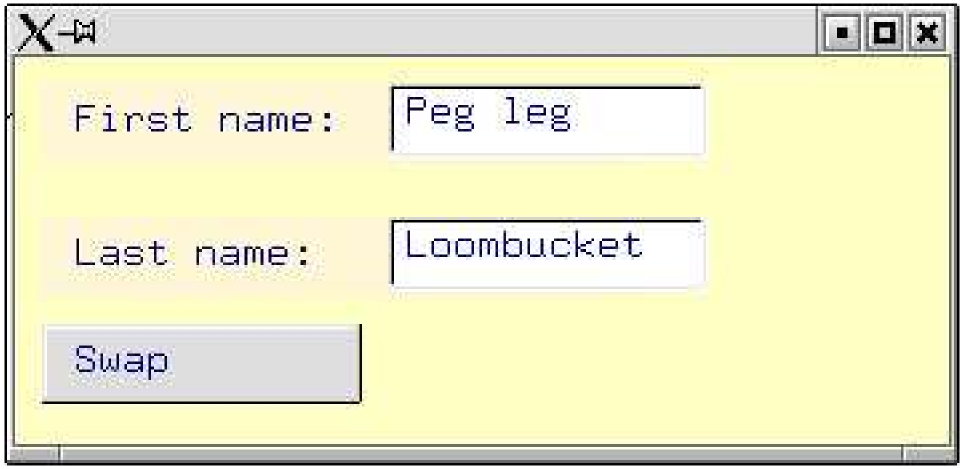
Why the mismatch? - bad concurrency model

ex11_lib.erl

... 50 odd protocol messages ...

```
ePolyText8(Drawable, GC, X, Y, Str) ->
    Len = length(Str),
    Delta = 2,
    BStr = list_to_binary(Str),
    B = <<Len:8,Delta:8, BStr/binary>>,
    req(74, <<Drawable:32, GC:32, X:16, Y:16, B/binary>>).


ePutImage(Draw, GC, Width, Ht, X, Y, Pad, Depth, Data) ->
    req(72, 2, <<Draw:32,GC:32,Width:16,Ht:16,X:16,Y:16,Pad:8,Depth:8,
              0:16,Data/binary>>).
```

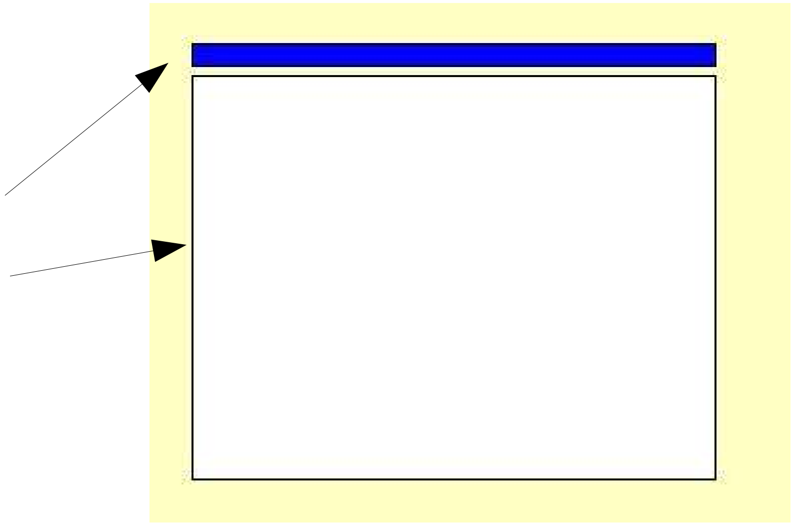
Widgets



```
start() ->
  spawn_link(fun win/0).

  win() ->
  Display = xStart("3.2"),
  Win     = swTopLevel:make(Display,350,145,?bg),
  Label1  = swLabel:make(Win,10,10,220,30,0,?cornsilk,"First name:"),
  Entry1  = swEntry:make(Win,140,10,120, "Peg leg"),
  Label2  = swLabel:make(Win,10,60,220,30,?cornsilk, "Last name:"),
  Entry2  = swEntry:make(Win,140,60,120,"Loombucket"),
  Button  = swButton:make(Win,10,100,120,30,?grey88, "Swap"),
  Button ! {onClick, fun(X) ->
                       Val1 = Entry1 !! read,
                       Val2 = Entry2 !! read,
                       Entry1 ! {set, Val2},
                       Entry2 ! {set, Val1}
                     end},
  loop().
```

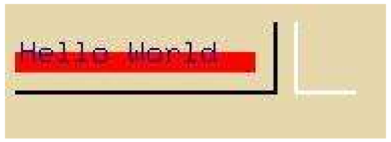
Higher order Widgets

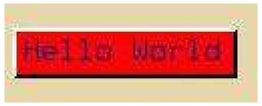
```
DragBar   = swDragBox:make(Win,X,Y,...),
Rectangle = swRectangle:make(Win,XX, ...),
DragBar ! {onMove,
          fun(X, Y) ->
            Rectangle ! raise,
            Rectangle ! {setXY, X, Y+16}
          end}
```



Lower order Widgets

```
win2(Pid) ->
    Win  = xCreateSimpleWindow(Pid,10,10,300,100,
                 ?XC_arrow, xColor(Pid, ?wheat2)),
    Font = xEnsureFont(Pid, "9x15"),
    Pen  = xCreateGC(Pid, [{function, copy},{font, Font},
                           {fill_style, solid},
                           {foreground,
                      xColor(Pid, ?DarkBlue)}]),
    Red = xCreateGC(Pid, [{function, copy}, {font, Font},
                          {fill_style, solid},
                          {foreground, xColor(Pid, ?red)}]),
    xCreateNamedGC(Pid, "black", [{function,copy},
           {line_width,2},{line_style,solid},
           {foreground, xColor(Pid, ?black)}]),
    xCreateNamedGC(Pid, "white", [{function,copy},
                             {line_width,2},{line_style,solid},
                             {foreground, xColor(Pid, ?white)}]),
    Cmds   = [ePolyFillRectangle(Win, Red,
                   [mkRectangle(10,20,110,22)]),
             ePolyLine(Win, xGC(Pid, "black"), origin,
                   [mkPoint(10,43),
                    mkPoint(120,43), mkPoint(120,20)]),
             ePolyLine(Win, xGC(Pid, "white"), origin,
                   [mkPoint(10,43),mkPoint(10,20),
                     mkPoint(120,20)]),
             ePolyText8(Win, Pen, 12, 35, "Hello World")],
    xDo(Pid, eMapWindow(Win)),
    xFlush(Pid),
    loop(Pid, Cmds).
```





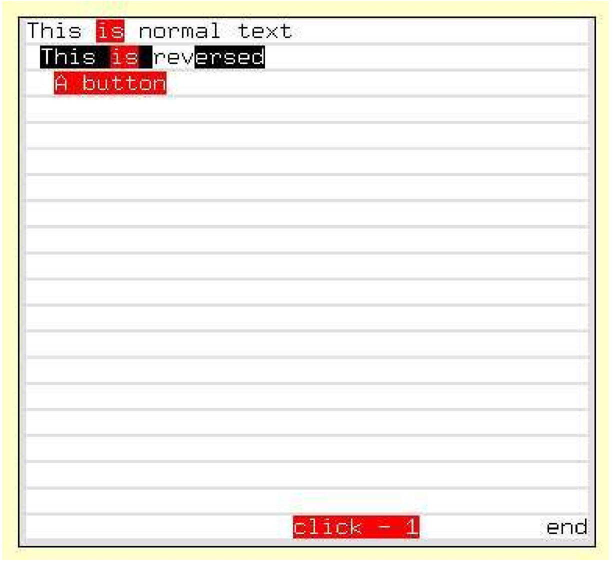
What widgets do you need?

sw.erl
swColorButton.erl
swErlPoint.erl
swScrollbar.erl
swBlinker.erl
swColorText.erl
swFlashButton.erl
swSelector.erl
swButton.erl
swDragBox.erl
swLabel.erl
swText.erl
swCanvas.erl
swEdText.erl
swLifts.erl

swToggle.erl
swEmacs.erl
swProgressBar.erl
swTopLevel.erl
swClock.er
swEntry.erl
swRectangle.erl

What widgets do you need?

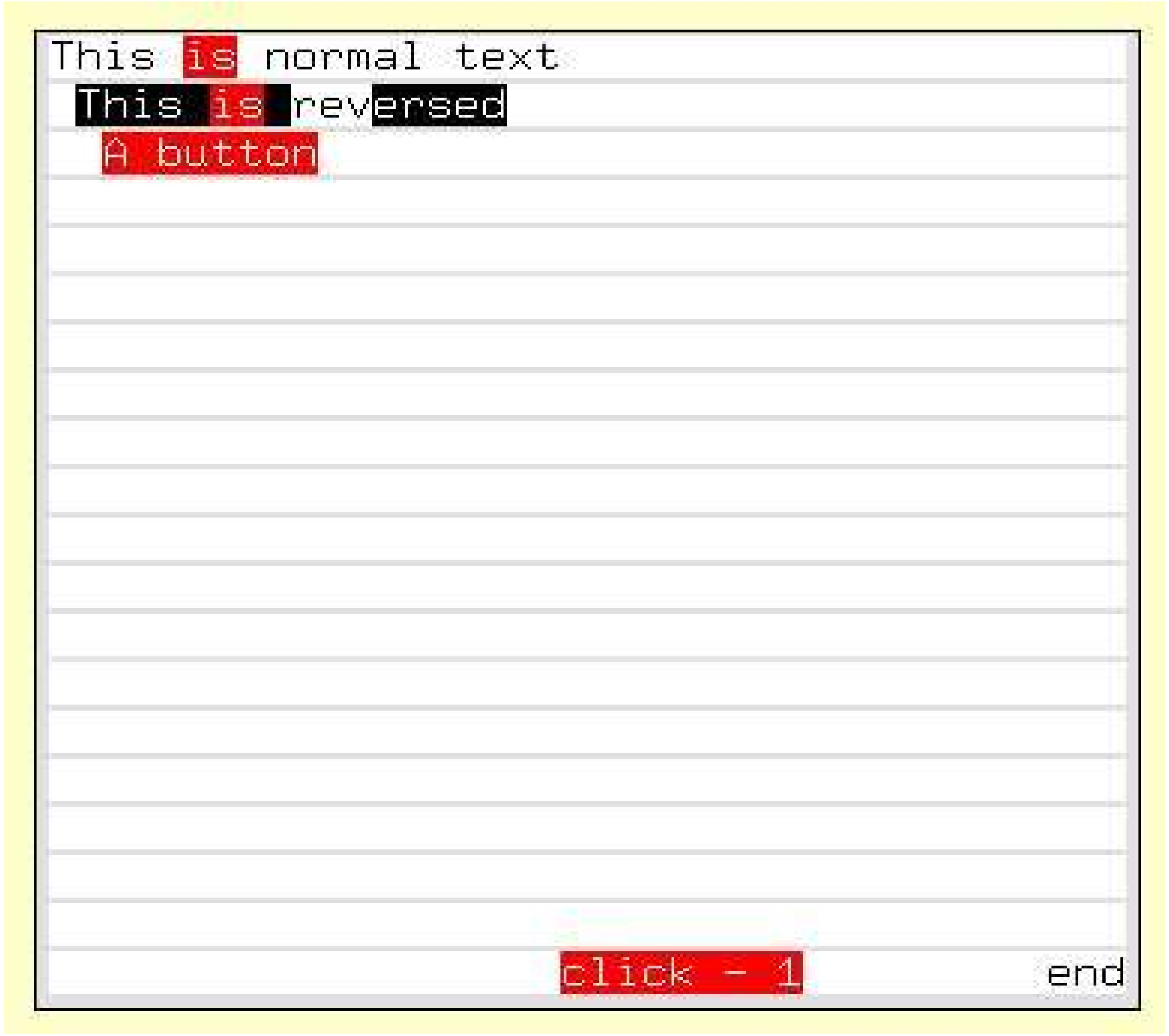
```
win() ->
    Display = xStart("3.2"),
    XX = 40, YY=20,
    {Width, Ht} = sw:sizeInCols2pixels(XX, YY),
    Win     = swTopLevel:make(Display, Width+20, Ht+20, ?bg),
    Rect    = swColorText:make(Win, 10,10, XX,YY,1,?grey88),
    S = self(),
    Rect ! {onClick, fun(X) -> S ! {click, X} end},
    Rect ! {onKey,   fun(X) -> S ! {key,   X} end},
    Rect ! {newPen, normal, ?black, ?white},
    Rect ! {newPen, rev, ?white, ?black},
    Rect ! {newPen, button, ?white, ?red},
    Rect ! {display, 1,1,normal,"This is normal text"},
    Rect ! {display, 2,2,rev,"This is reversed"},
    Rect ! {display, 3,3,button,"A button"},
    Rect ! {display, 6,1,button,"is"},
    Rect ! {display, 7,2,button,"is"},
    Rect ! {display, 10,2,normal,"rev"},
    Rect ! {display, 20,20,button,"click - 1"},
    Rect ! {display, XX-2,YY,normal,"endXYZ"},
    loop(Rect).

loop(Rect) ->
    receive
       {click,{X,Y}} ->
          Rect ! {blink, X,Y},
    loop(Rect).
```


```
This is normal text
 This is reversed
  A button
            click - 1      end
```

What widgets do you need?

SwColorText

Emacs
Drop down menus
Buttons
File Selector
Entries
Forms
Progess bars

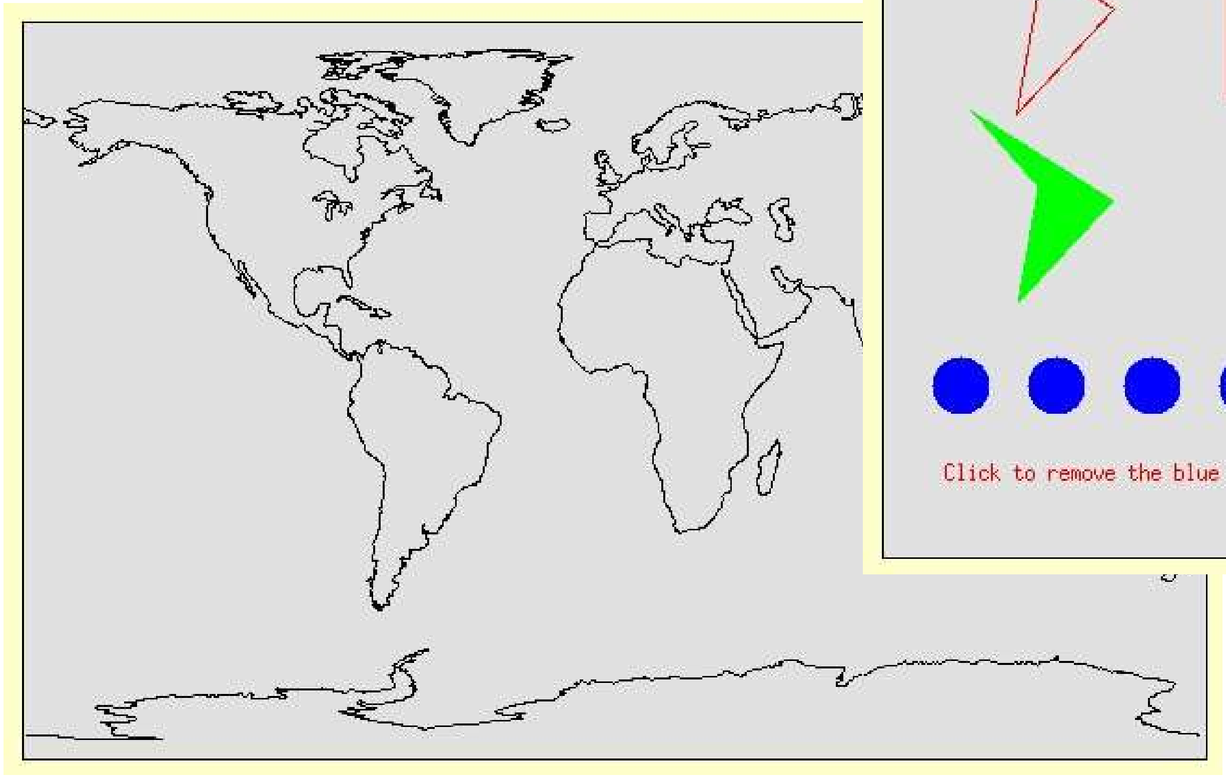
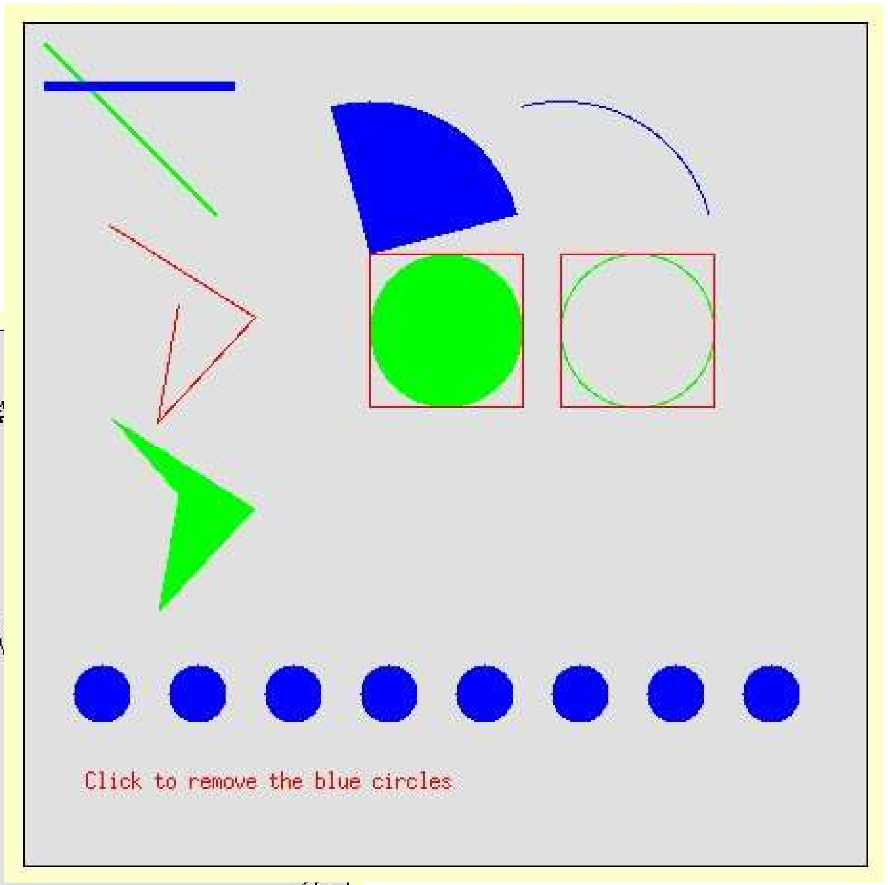


```
This is normal text
 This is reversed
  A button
```

What widgets do you need?

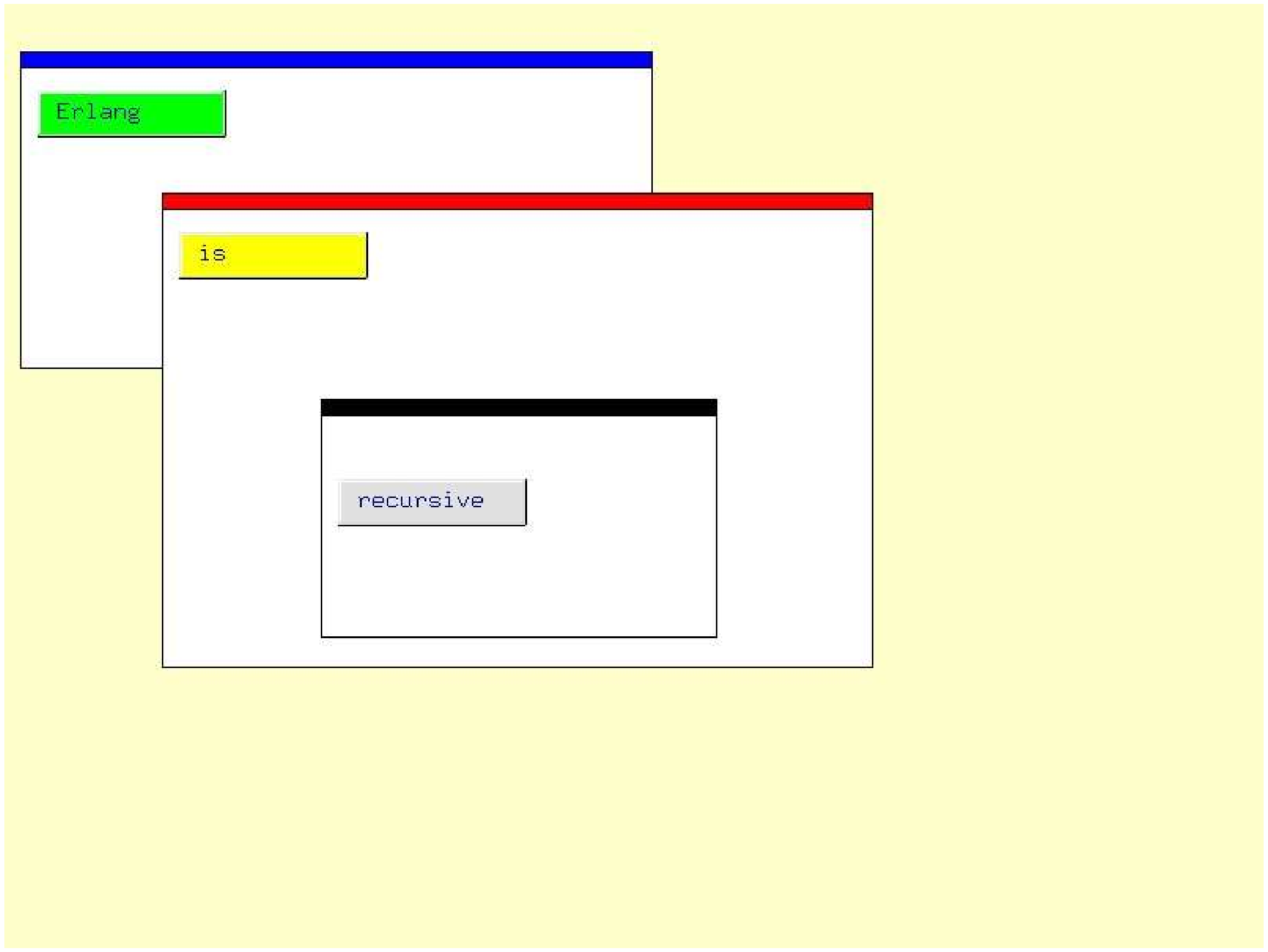
SwCanvas

map
plots
clock

What widgets do you need?

Desktop or MDI



What widgets do you need?

Desktop or MDI
Text widget
Canvas widget

with a simple intuative API

```
Text ! {onClick, fun(X, Y, Char) -> ...}
Text ! {onKeyPress, fun(Key) -> ...}

MDI  ! {moveWindow, X, Y}
MDI  ! {onWindowMoved, fun(Win, X, Y) -> ...}

Pos = MDI !! {whereisWindow, Win}
```

Now what?

Programming model (solved)

X11 model (almost solved), needs testing on non truecolor terminals, old terminals without 24 bit color, connecting is difficult (badly configured machines etc.)

Running on windows – suck – sigh – needs X11 server (cygwin, etc too difficult for normal user to install)

Conclusion:

Make widget model run on portable graphics libraries
Just need to implement the big three on GDK/FLTK/whatever
(or win32 native API)

Rant on a bit?

GUI toolkits suck big time. GTK ... etc. Are a total confused mess of low_level and highlevel stuff.

GUI programming should be easy, but it is appaulingly messy.

GUI libraries offer the wrong abstractions. Text widgets etc. Are a prime example they are appaulingly complex

Tools make matters worse (this is why we don't see dynamic GUIs)

Instead of correcting the problem they hide it.

GUIs can be simple?

Borland BGI

Oberon

8½ the Plan 9 windowing system

Acme

Wily

Press here to send mail or
you can quit the program

to:mike
subject:

```
write(wm, ”@window
Press ~here~ to send
mail or
you can ~quit~ the
program

to:mike
subject:”)
```

Finally

Goal – Drop dead beautiful, easy to use intuative GUI toolkit (my three widgets) – anti-aliased fonts, alpha blending

Help wanted – I'm very bad at C++/GUI programming

Acknowledgements:

Tobbe – first version of ex11 "Proof of concept"
Tony – added many features
Vlad – added binary syntax for protocol parsing
Joe – rewrote everything (except authentication)
Sean – odd hacks
Frej – Explained how X works