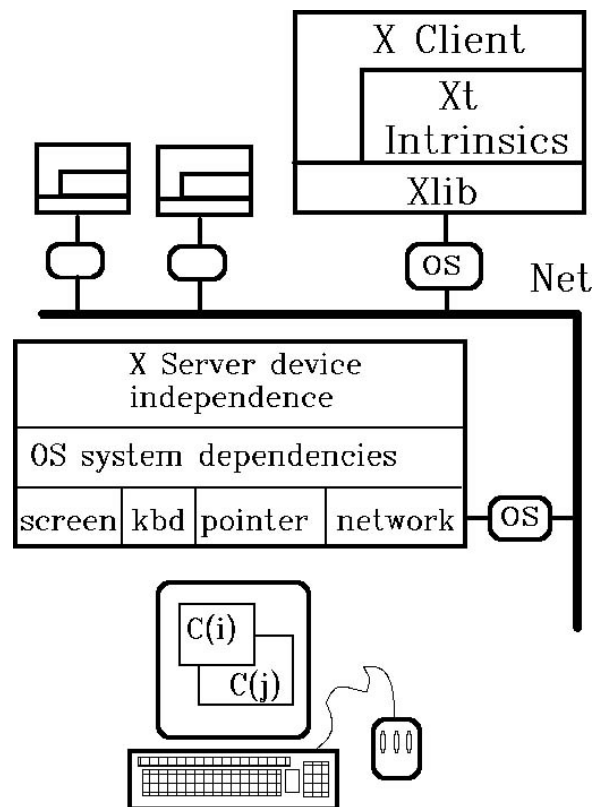
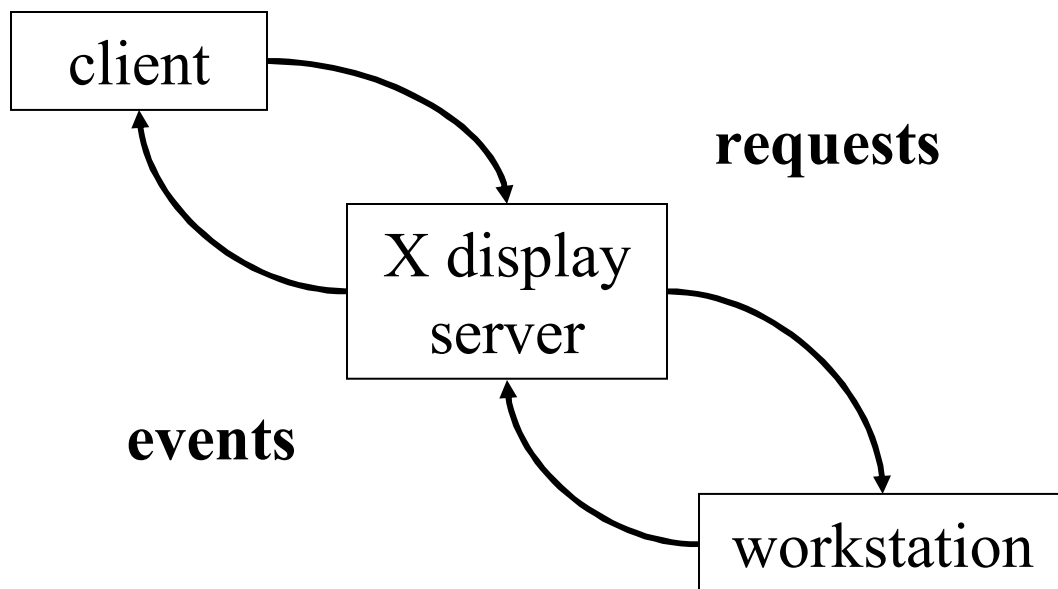
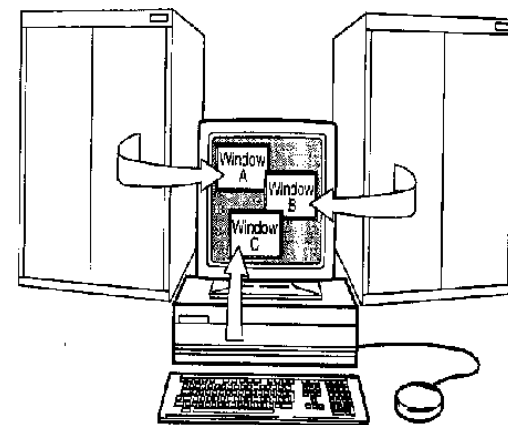


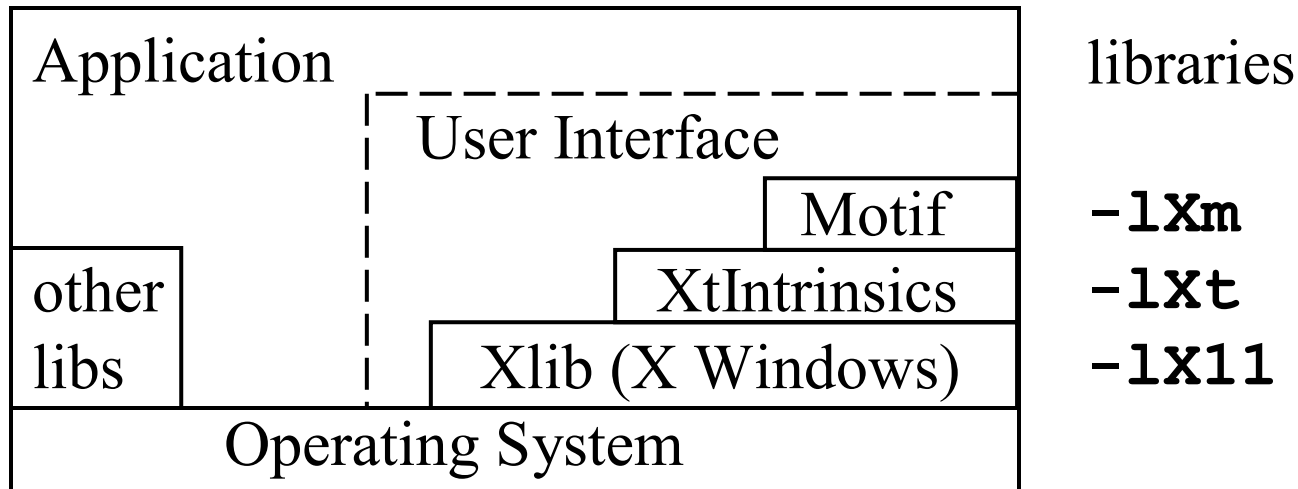
X Windows Motif

Early (1987) network transparent, vendor independent operating environment for workstation based software.

Applications can run on host best suited to support processing.

Information sharing & concurrency





```
cc app.c -o app -lXm [-lXpm] -lXt -lX11
```

High level user interface components from Motif, Xt or other libraries:
widgets (Object Based libraries)

buttons, menus, dialog boxes, viewports, scrollbars, panes, etc.

Low level drawing functionality from Xlib.

draw point, line, arc, fonts

1. Include files for Motif & widgets
2. Initialize Xt Intrinsic
 XtVaAppInitialize(...)
3. Create Widgets { shell, manager, primitive, gadgets }
 XtVaCreate... Widget (...)
 XmCreate ...Dialog (...)
4. Event Handling -- register translation tables and callbacks
 XtAppAddActions(...)
 XtAddCallback(...)
5. Realize Widgets -- display widgets
 XtRealizeWidget(...)
6. Enter Event Loop -- respond to user
 XtAppMainLoop(...)

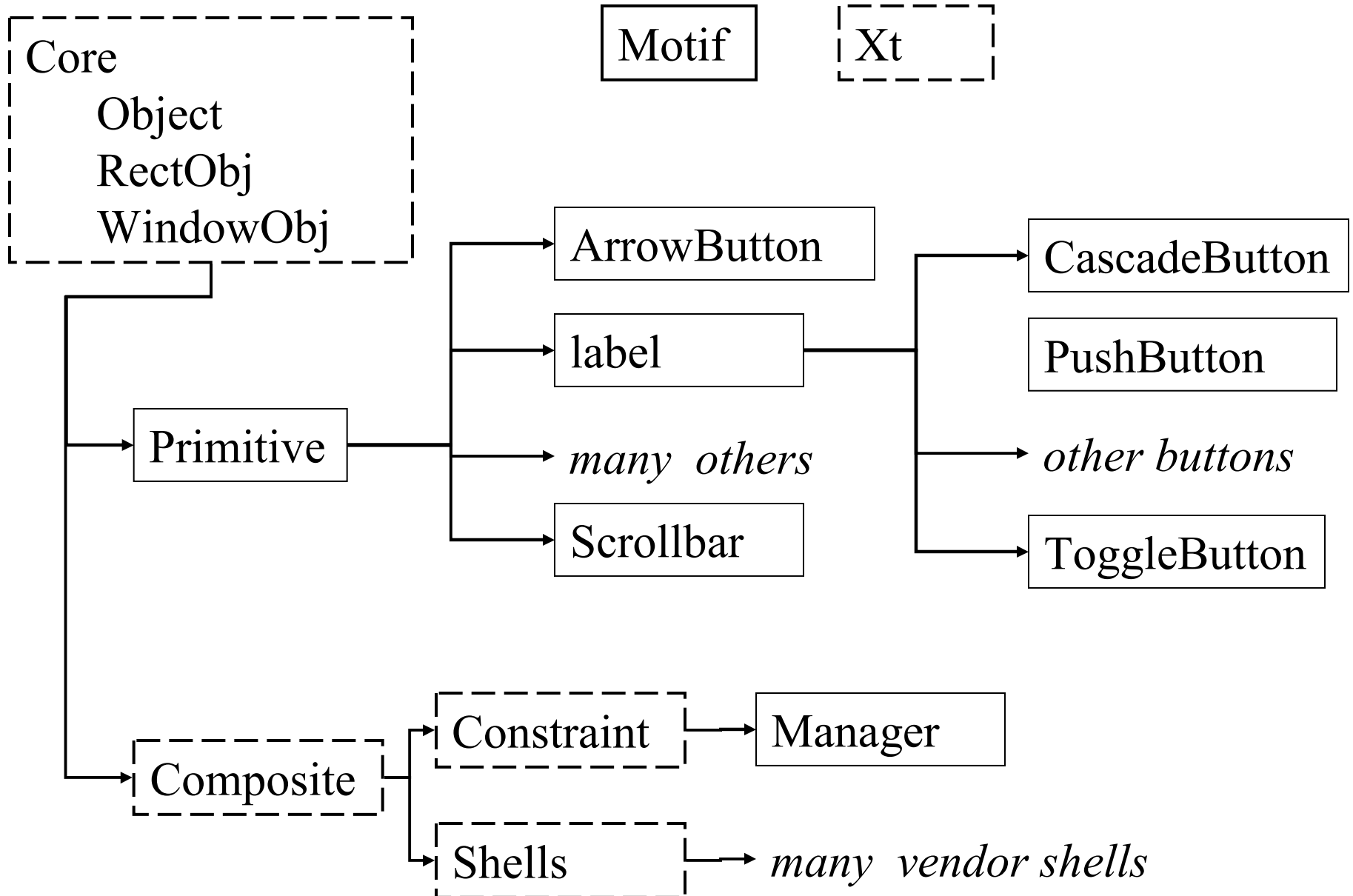
XtVaCreateManagedWidget(...)

Enables the creation and setting of resources for managed, and XtVaCreateWidget(...) unmanaged, widgets.

```
widget = XtVaCreateManagedWidget(  
    widget name, // used for resource dbms  
    widget class,  
    parent, // a manager widget  
    // remaining arguments are list of resource settings.  
    // These use the form: XmNresourceName, resourceValue  
    // XmNbackground, wheat  
    NULL // end of resourceName, resourceValue pairs);
```

Resources can be set dynamically after widget creation in the application, or in various developer specified, system specified, and user specified resource database files...

Widget Class Hierarchy

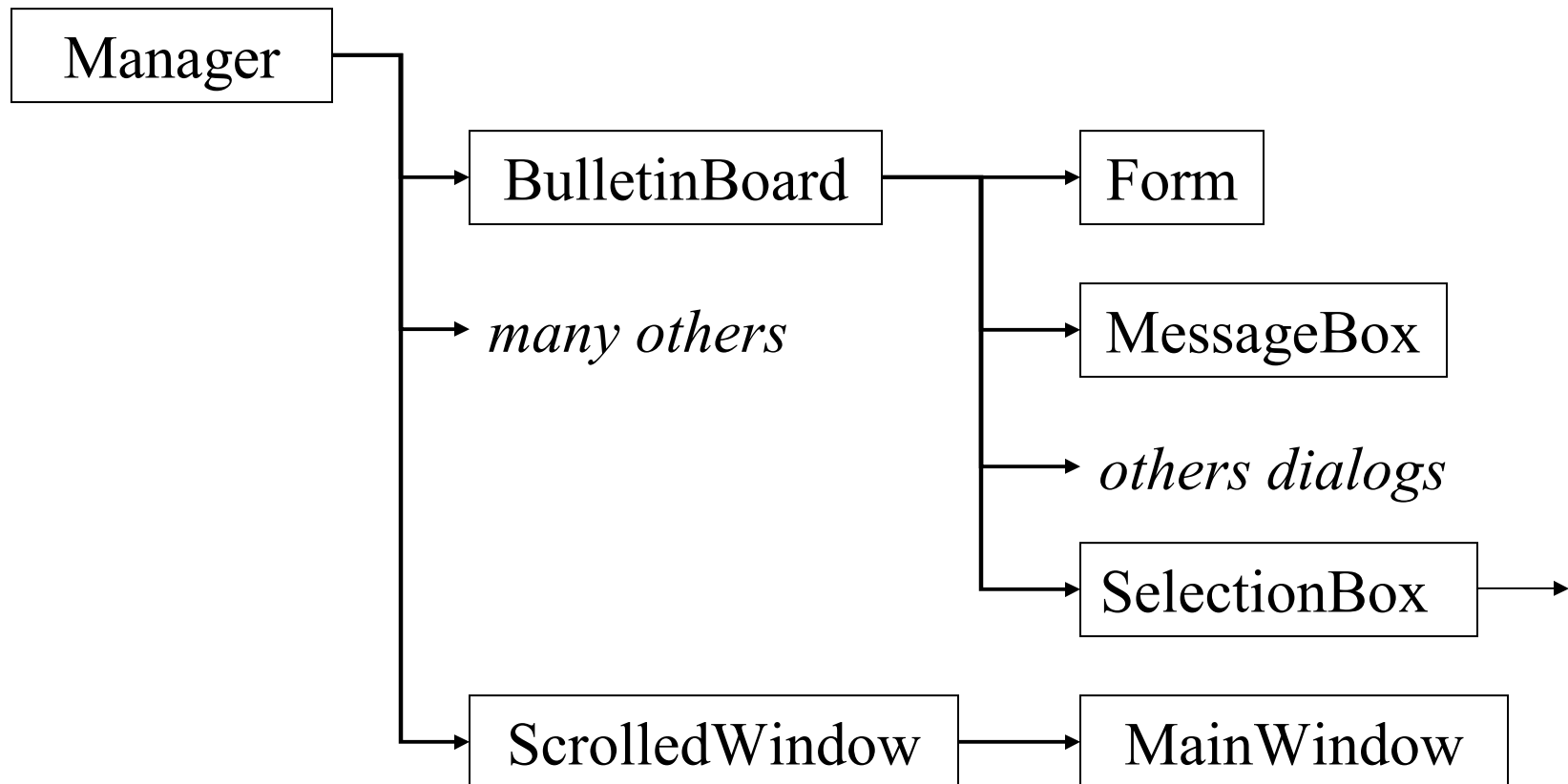


Primitive widgets are visible UI controls with default resources:

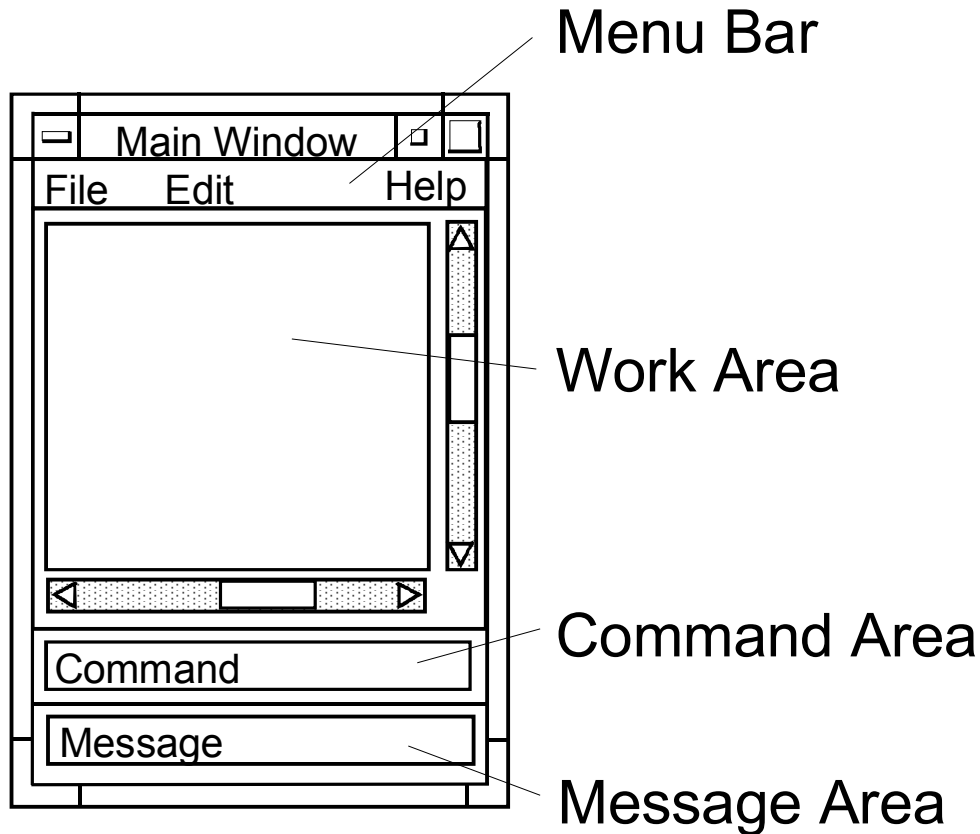
visible properties: fore / background, font, strings

callbacks for event handling

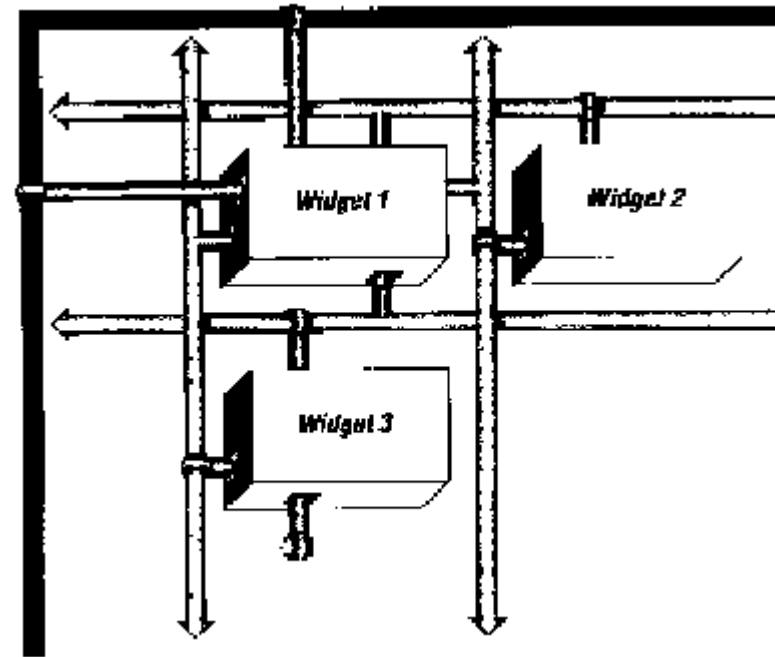
Manager widgets control geometry and layout of children (managed) widgets.



MainWindow convenience widget for generic application with optional menuBar, workArea, CommandArea, and MessageArea widgets



Form widget is a powerful manager with attachment constraints that can attach to the form or other widgets on top, left, right, bottom



Widget have predefined callback resources.

e.g. PushButton widget class defines:

XmNactivateCallback

XmNarmCallback

XmNdisarmCallback

Callbacks are added (ie "*addListener*") with:

```
XtAddCallback(Widget w, String resourceName,  
               XtCallbackProc functionName,  
               XtPointer clientData);
```

The resourceName specifies the widget's defined callback resource.

clientData is any scalar data you wish to be passed to the function when the callback is invoked

pointer to class (struct) or argv type approach to pass many values.

Other events handler mechanisms: translation tables and actions

Callback functions

The function associated with the callback must have a specific argument list

```
void functionName( Widget w, XtPointer clientData,  
                  XtPointer callData);
```

XtPointer is a general pointer type casting is often needed -- especially w/ C++

clientData references the data in the XtAddCallback(...) statement.

callData references a CallbackStruct event structure returned by the event bound to the widget's callback resource.

X uses many CallbackStructs for different event classes.

The basic one is:

```
typedef struct {  
    int reason;  
    XEvent      * event;  
} XmAnyCallbackStruct;
```

Menus

menus in menuBar with pulldown, cascade, popup, menuItems

Dialogs

convenience message, information, warning, error, fileI/O

custom: transient frame with managers and primitives

Convenience widgets

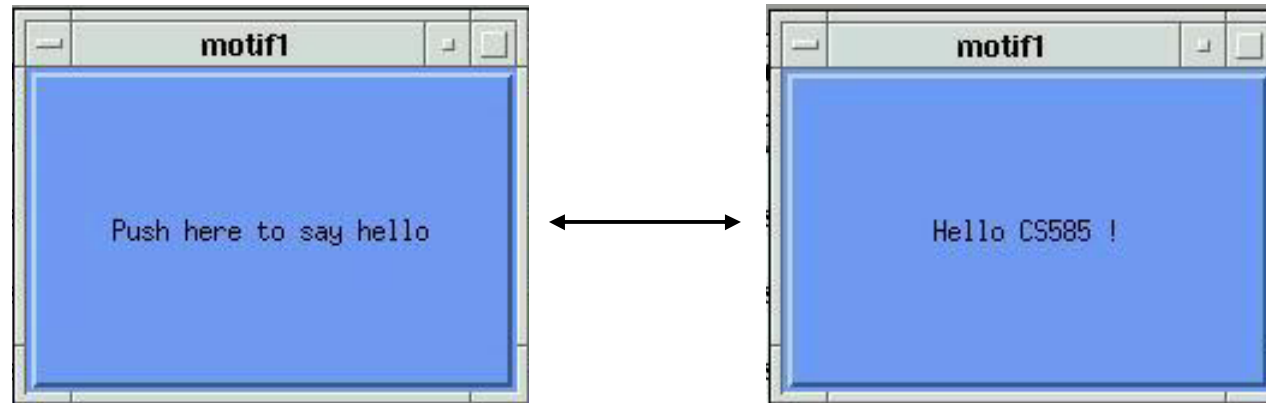
Menus, dialogs, MainWindow, others

many widgets have convenient forms with default arguments

Widget layout design requires layered manager widgets

Rich graphics drawing functions. GC (graphics context)

Resources include: bitmaps, images, cursors, fonts, colors



Example `motif1.cc` illustrates

- events: push button callback and keyboard event handling.
- resources: `XtSetValue(...)` and `XtVa*` techniques