

Window Systems

Goals: Virtual Devices

- Virtual display abstraction
- Multiplex physical input devices
- Simulated or higher level "devices"
- Limited resource management

2 Views of the Window System

- User Interface
- Application Interface
 - Imaging model
 - Input model

Higher Level Imaging Models e.g. Postscript-based

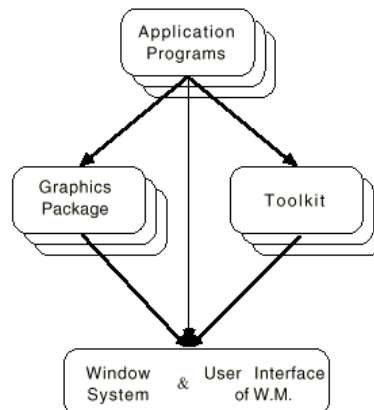
- e.g. NeWS & NeXT [AKA OpenStep]
- Real valued coordinate system
- Support for full transformations
 - e.g., scale & rotate
- Richer primitives
 - e.g., Curves
- Stencil and paint model

Higher Level Imaging Models: one more word

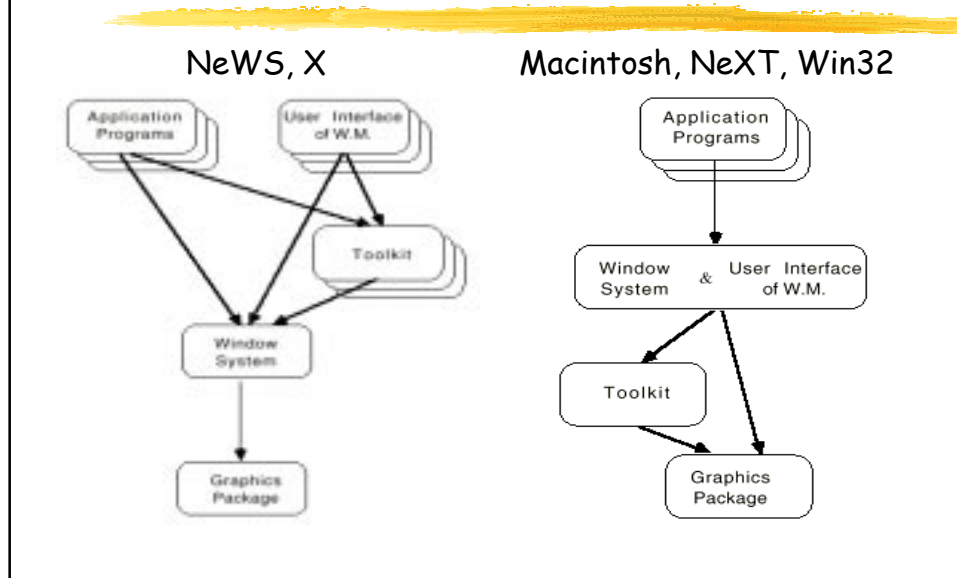
- See the "Primal Screens" article for why NeWS died
- Despite this, I prefer them and feel they will come back

Early Output Architecture

- E.g. Sapphire, SunWindows



Current architectures



Input models

- Handling input devices tedious
 - Want an abstraction for input
 - As disks, etc. are to file systems
- The "uniform event" input model
 - An event is a record of an input action
 - Events placed in a queue
 - processed asynchronously
 - "producer/consumer" between system/user

An event record

- What caused the event
 - e.g., left mouse button went down
- Where was the mouse
- When did the event occur
- Value associated with device action
 - e.g. ascii value of key, position of knob
- Additional Context
 - e.g. modifiers

Example

■ The Java 1.2 MouseEvent

<http://java.sun.com/products/jdk/1.2/docs/api/java.awt.event.MouseEvent.html>

Special events

- Not just for simple input device actions
- e.g.

Using events: Return of basic paradigm

```
Main_Event_Loop()
  Set_input_mask();
  repeat
    Wait_for_event(E)
    case E of
      ...
      dispatch event E
      ...
    end;
    redraw_screen();
  until done;
```

Life with Events

- Continuous handling
- Higher level abstractions
 - Asynchrony: FSMs

Issues

- One channel (no "back channel")
 - Flow control characters inserted in queue
- Fixed set of events
 - Handling new *kinds* of devices hard
- Events are asynchronous
 - ⇒ User asynch with program

Synchronization Issues

- How typically dealt with?
- Implication?

Example:

Synchronization Issues

- Separate queues don't help
- Each thread needs one unified queue
 - Can be one per thread