Run II GUI Evaluation

Run II Software Tools meeting

September 12, 1996

preliminary draft doc., info and examples under
http://d0sgi0.fnal.gov/~lueking/gui/gui_group.html

Example Applications

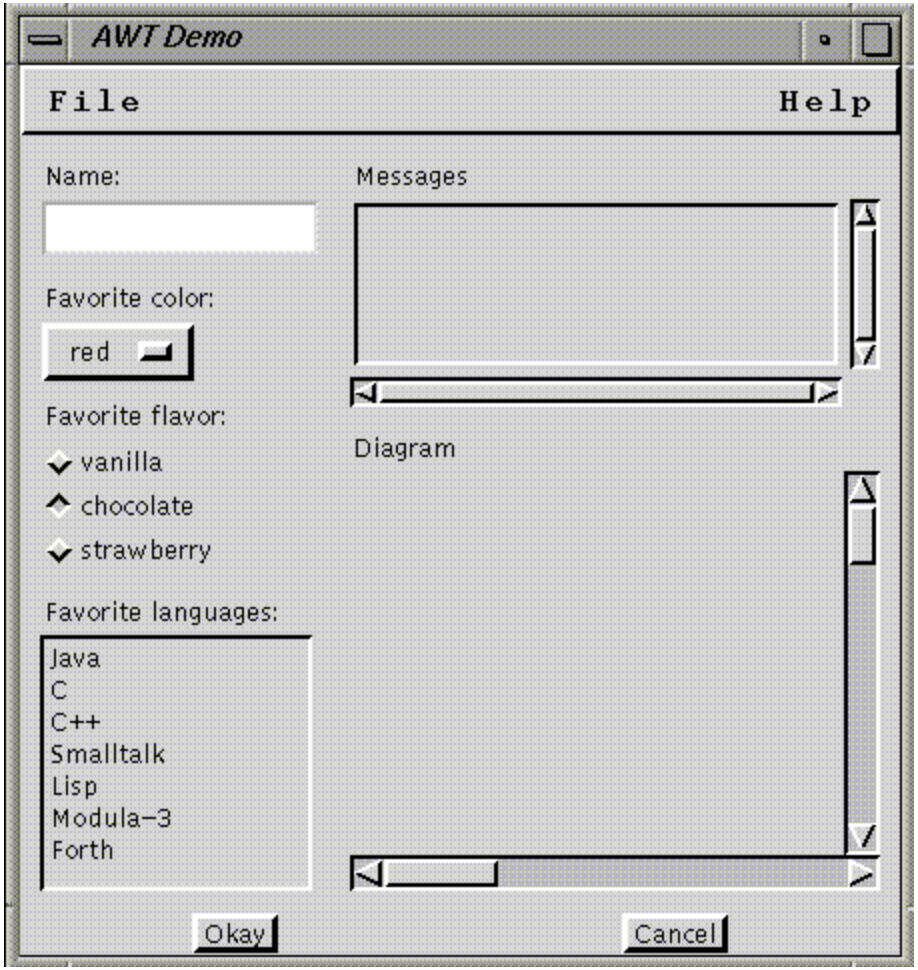
- Global Information Access (possibly Web-based)
  - » Data Base Information
  - » On/Off-line systems
  - » Code/production management
- Local Menu Based Applications
  - » Event reconstruction program.
  - » On-line systems control and monitoring.
- Complex Menu-based Interactive Graphics
  - » Event Display

Considerations

- Requirements
  - » Easy to use: intuitive menu systems
  - » Easy to write and maintain: reusable, extensible, clean format, et cetera.
  - » Platform interoperability: UNIX + NT
  - » Longevity : 10 + years
- Graphics
  - » X/Motif - Window tools, not on PC
  - » wxWindow - Class library
  - » Open GL- Low level graphics language

Considerations

- Scripting Languages with Widget tool kit
  - » TCL/tk
  - » Perl/tk
  - » Python/tk
  - » Java w/ AWT
- High Level Environments
  - » Open Explorer
  - » Open Inventor

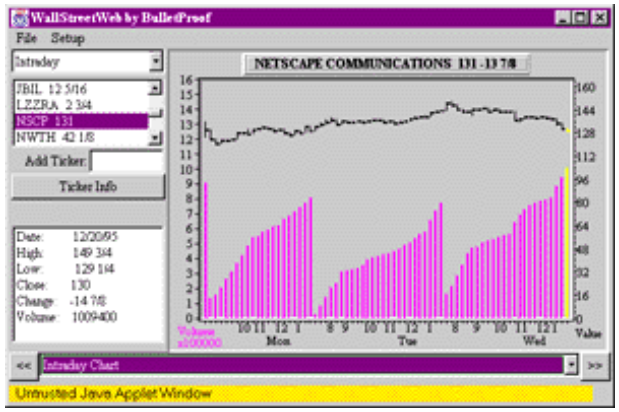


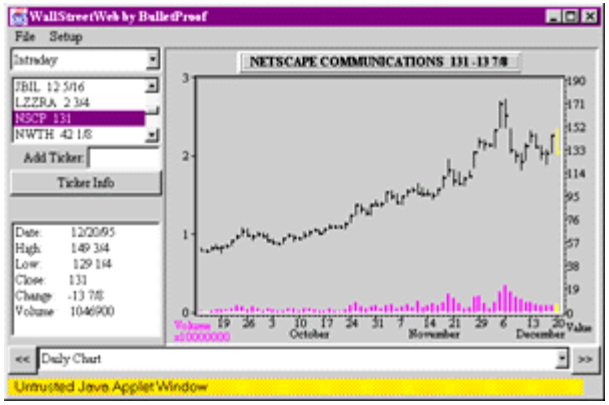
Scripting Languages: A Closer Look

- TCL - Tool Command Language
  - » Pros: Widely used, Supported by Sun, Well documented.
  - » Cons: Peculiar syntax, Not for “serious” programming.
- PERL - scripting and processing language
  - » Pros: Widely used, C-ish syntax, UNIX system, Reg. exp.
  - » Cons: Combersome, Difficult to read script.
- PYTHON -
  - » Pros: Clean syntax, Broad dynamic range, Scales up well
  - » Cons: NOT widely used, Questionable support, Non-C-ish.
- (JAVA)
  - » Pros: Simplified C++, Widely used, Many useful features
  - » Cons: Byte code interpreted, “Closed”, Rapidly evolving
- ✶ See frame page for more info from sss

Global Information Access Tools

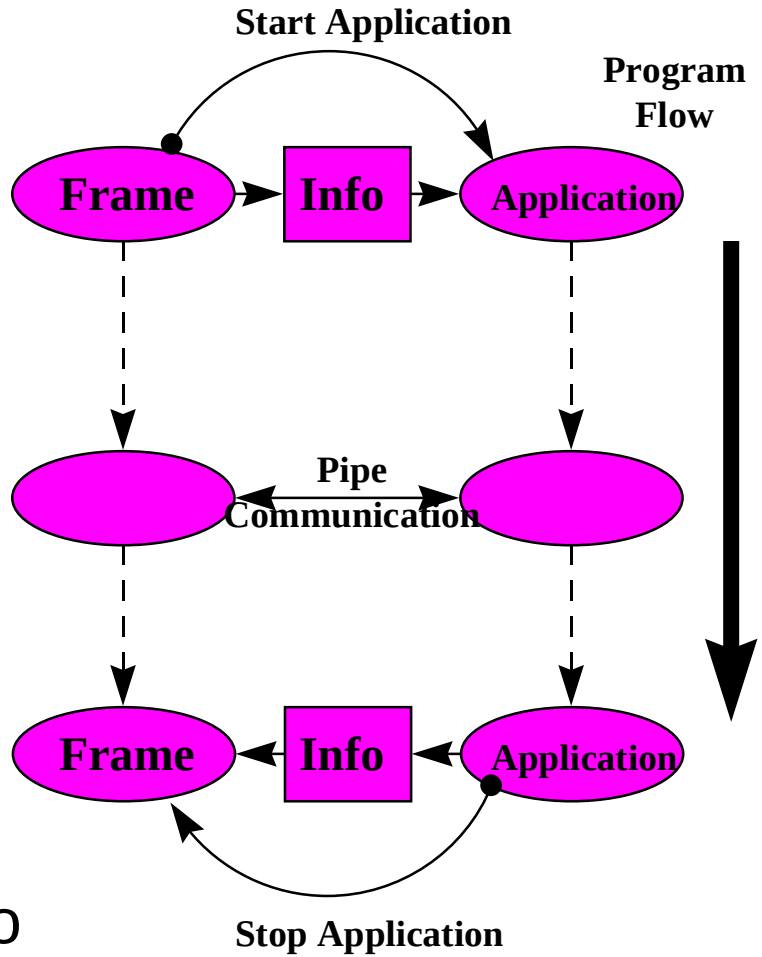
- Tools used
  - » Perl/Python with Common Gateway Interface (CGI).
  - » Java
- Examples
  - » Data Base Information
  - » On/off-line systems
  - » Code configuration info
  - » Production Management
  - » Analysis tools





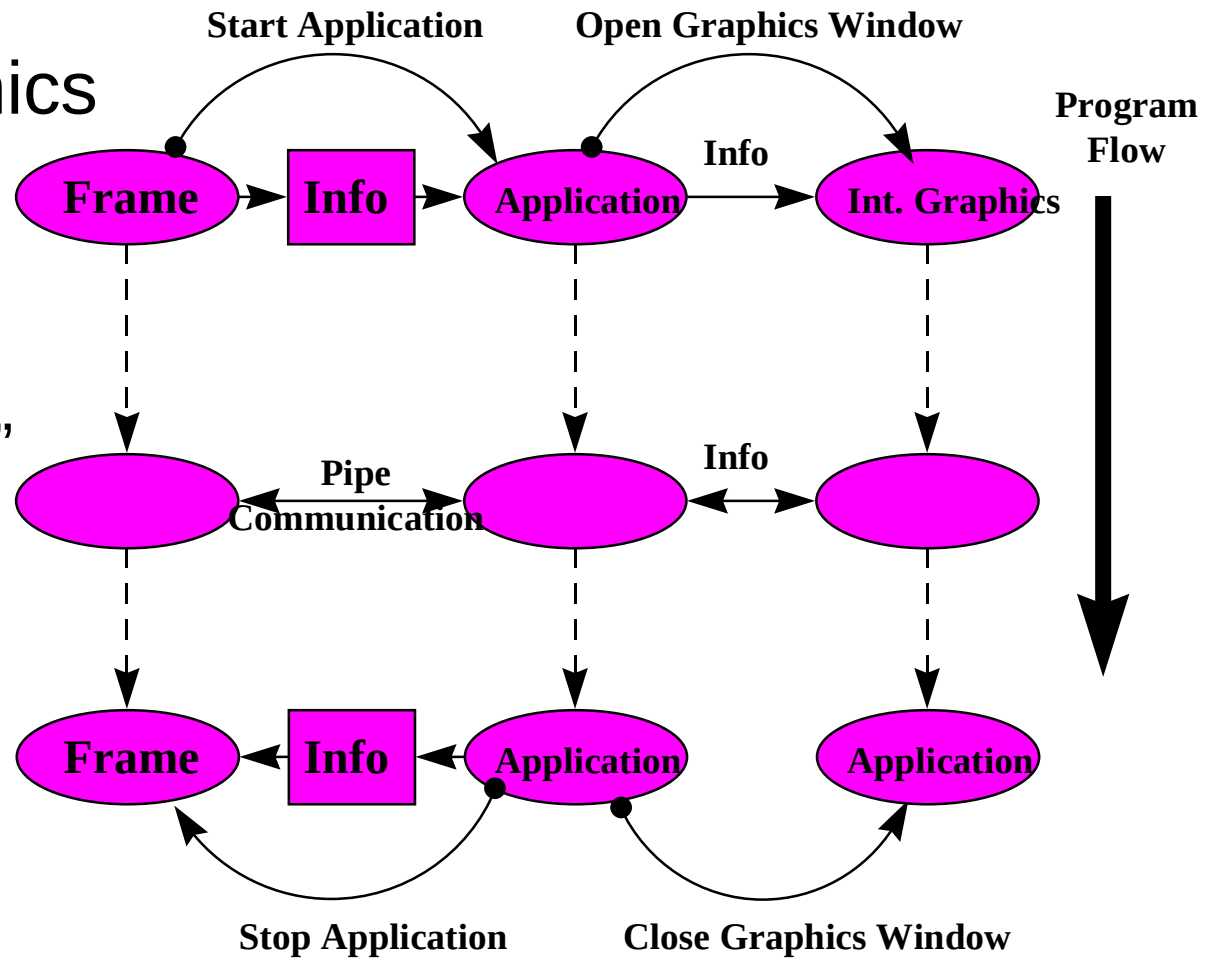
Local Applications: GUI-Frames

- Local “Frame” Example
  - » Frame opens Menu for interactive user
  - » User selects control options
  - » Options are written to script file
  - » Frame starts the application
  - » Script file is read by application
  - » Additional communication through pipes.
  - » When App. ends, info passed to Frame



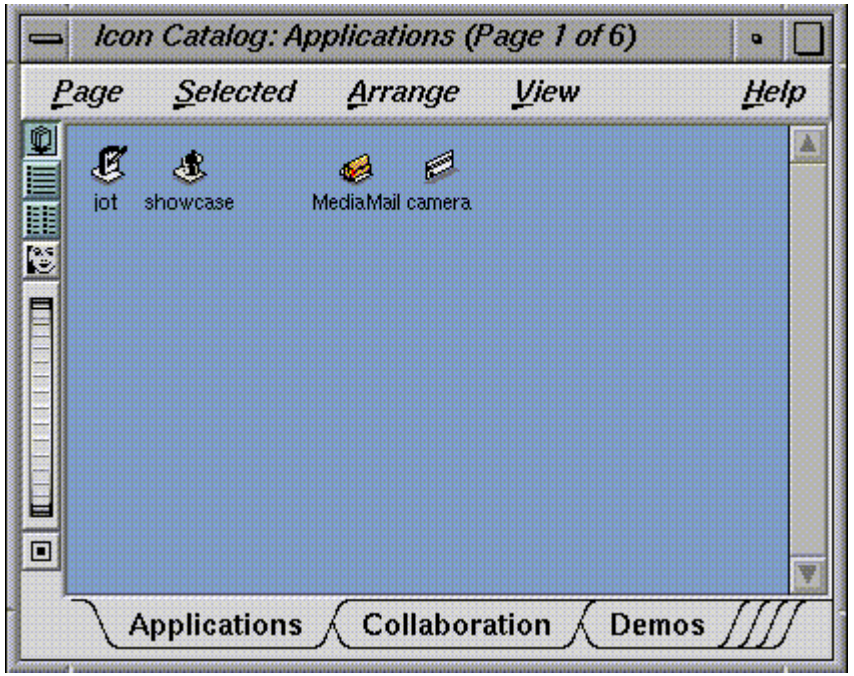
Interactive Graphics

- Interactive Graphics Example
  - » Same as GUI-Frame, but...
  - » Additional “loops”



Widgets

- Available in tk
  - » Labels,buttons, check buttons, radio buttons,iconic buttons, image viewers,list boxes, entries w/o scrollbars, canvases, linear scales, pulldown menus.
- May need Additional
  - » Easily found, e.g. Free Widget Foundation (FWF)
  - » Roll our own



Recommendations

- Scripting Language w/ tk - Perl or Python
- Java - Use where appropriate
- Prototype
  - » Frame example
  - » Possibly interactive graphics
  - » other suggestions…
- Look into wxWin if scripting approach “fails”
- Just scratches the surface; also animation, sound, 3-d graphics...