

Framework

Modules

Spec

Taco

Python Application Framework (PAF)

New ESRF Beamline control application framework



BLISS

The Present Situation

GUI



Motif or Tcl
Applications on Unix

Data Display



Spec

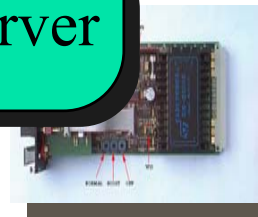


Online Analysis



C,C++ on OS9
VME or PC

Device Server



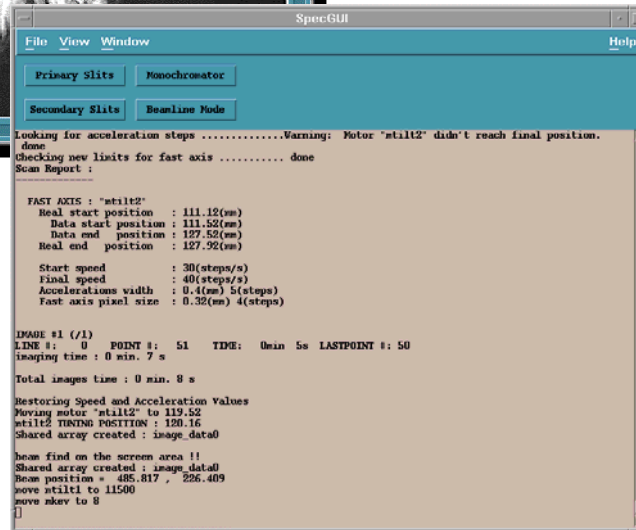
Spec Drivers





BLISS

Example from today

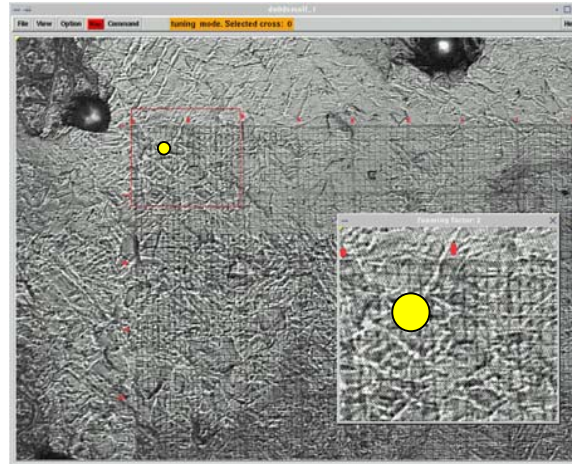




BLISS

Full Integration

- Integrate visualization and control into one application

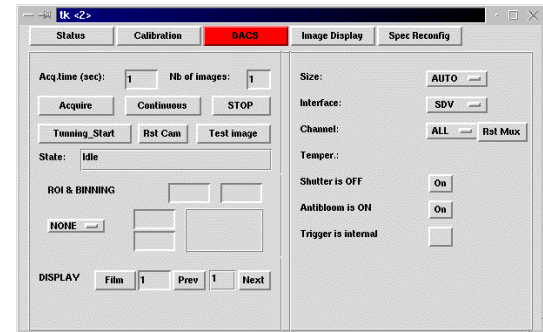




BLISS

Specific and general

- Two types of GUI elements:
 - General panels
(i.e. camera setup)
 - Beamline specific
panels (i.e.
experiment setup)

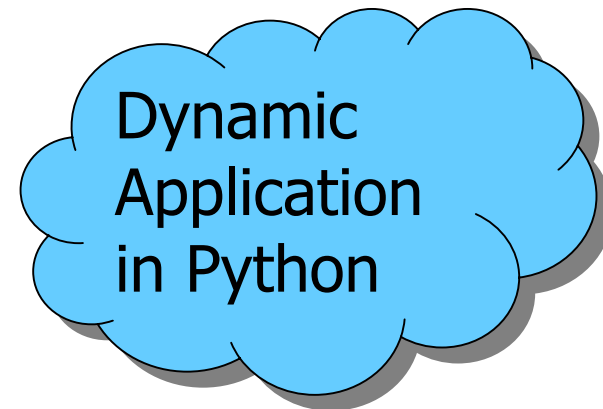
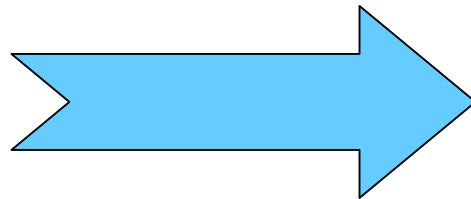




BLISS

Dynamic Application

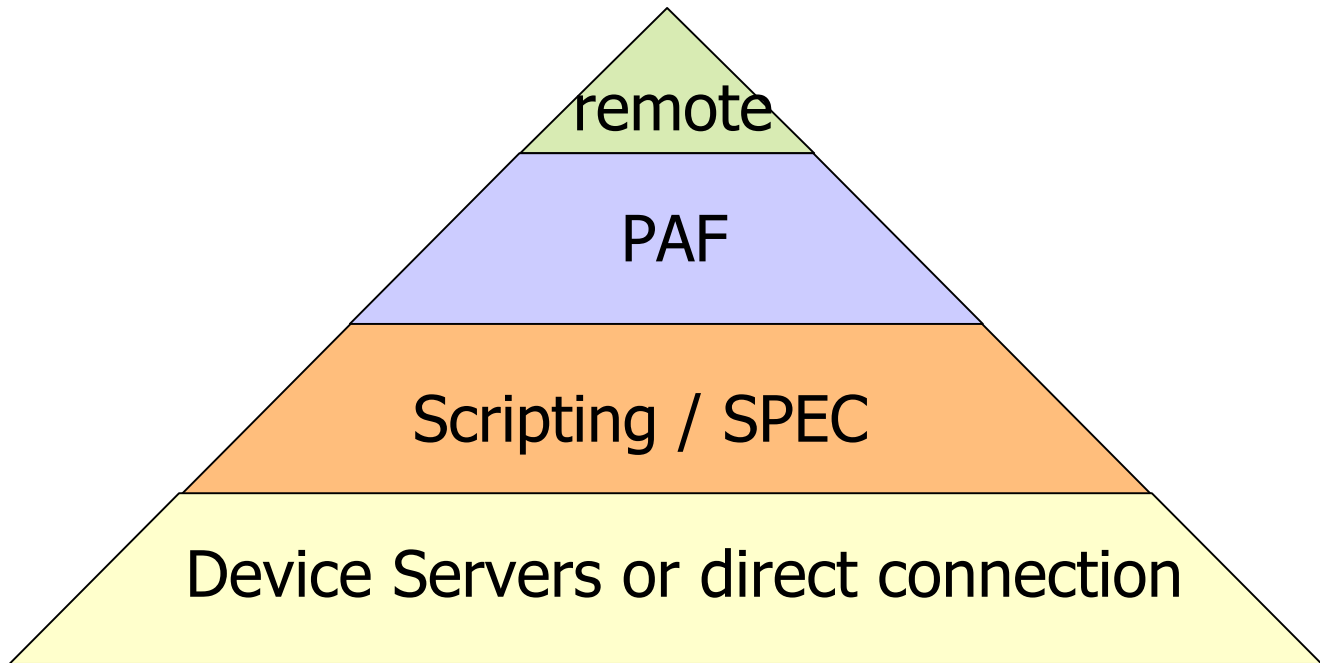
- Role distribution of GUI users:
 - Developer of general panels
 - Integrator on the beamlines
 - Local contact
 - User
- Full integration, general and specific parts





BLISS

The general structure





BLISS

Frame work

- What is it?
 - Python application in Tk
- It consists of
 - **Panels** are (almost) general python class instances
 - **Placers** are panels which can contain other panels
 - **Services** offer general utilities required by most panels



BLISS

Example Panel Code

```
class yellowpanel(Label):  
    def __init__(self, parent, eh=None, text="yellow"):  
        eh.register("changetext", self.changetext)  
        Tk.Label.__init__(self, parent, text=text)  
    def changetext(self, newtext):  
        self.config(text=newtext)
```

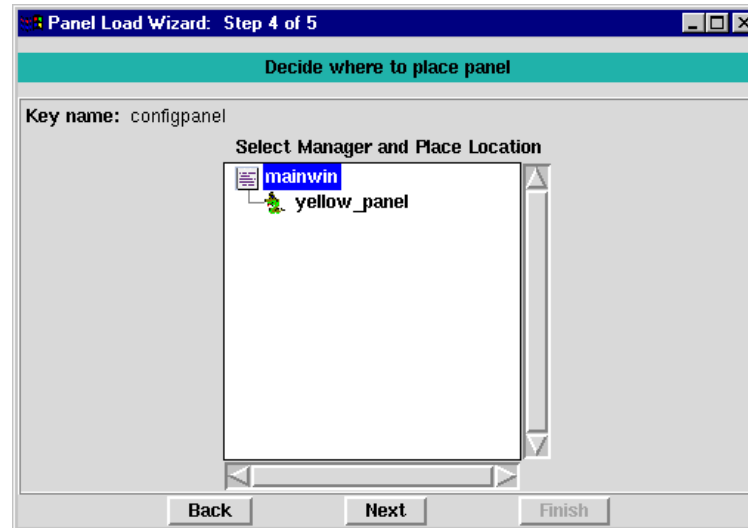
- **What** to load: filename, classname
- How to **reference** it later: instance name
- **Where** to load it: Popup or a "placer panel"
- How to **communicate**: Event Handler
- **Startup** parameters:



BLISS

Load Wizard

Wizard to load and place a panel







BLISS

Services

- Provide standard services for the panels
 - Logging, Printing, Error handling, Messages
 - Parameter store and retrieve
 - Standard Python Console
 - Help system
 - Debugger
- Example:

```
parserv = ParameterService("redpanel")  
parserv.get("background")
```



BLISS

Panel types

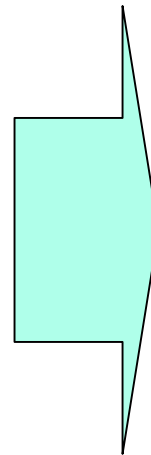
- Hand coded python classes (Either graphical or non graphical)
- Simplified panels (entry forms) to send commands to the scripting layer



BLISS

Entry forms

The screenshot shows the BLISS software interface. On the left is a menu with options: File, Help, Edit, Options, Functions: (with a sub-menu containing 'De-select all'), Create Object: (with sub-items 'panel', 'outputfield', 'inputfield'), and a 'Panel' section with 'Close' and 'Reset' buttons. Below the menu is an 'Edit Mode' section with 'Geometry' and 'Test Panel' options. In the center, a dialog box titled 'Button 1' is open, displaying fields for 'geometry' (containing the expression $(98+0\%) \times (27+0\%) + (10+0\%) + (10+0\%)$), 'version', 'command', 'format', 'name', and 'label' (set to 'button1').

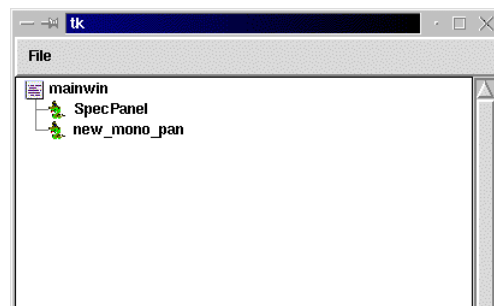


The screenshot shows the 'Monochromator Position' window. It has a title bar 'tk <2>'. The window is divided into several sections: 'Monochromator Position' with input fields for 'Energy (keV)', 'Wavelength (D)', and 'Bragg (deg)', and a 'Move Mono to' section with an 'Energy_KeV' dropdown and 'Move' and 'STOP' buttons; 'Monochromator mode' with a 'Standard' dropdown; 'Monochromator crystal' with a 'Si_1_1_1' dropdown; and 'Monochromator Calibration' with a 'Setup tuning parameters' button, a 'Tunning coef. (% fwhm)' field set to '0', a 'Find 2nd Crystal Offset' button, and a section for 'start (Kev)', 'end (Kev)', and 'Nb. Point' with corresponding input fields. At the bottom, there is a 'Status' field and 'Build position Table' and 'STOP' buttons.



PAF

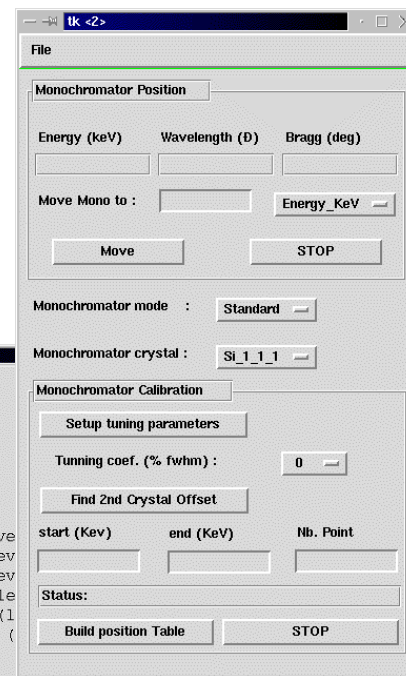
BLISS



```
tk <4>
Browse to http://www.certif.com for complete documentation.
=
Reading file "SPECED/site.mac".
Reading file "SPECED/./jmacros/expgsetup.mac" (level 2).
Starting darren SPEC application
**** Setting up plot options ....
Reading file "/bliss/users/spruce/src/pxicap/spec/prodc.mac" (level 1)
Reading file "/bliss/users/spruce/src/pxicap/spec/energy.mac" (level 1)
Reading file "/bliss/users/spruce/src/pxicap/spec/lambda.mac" (level 1)
Reading file "/bliss/users/spruce/src/pxicap/spec/osc_gui.mac" (level 1)
Reading file "/bliss/users/spruce/src/pxicap/spec/detector.mac" (level 1)
Reading file "/bliss/users/spruce/src/pxicap/spec/rot_image.mac" (level 1)
**** Changing working directory to ./data
Can't change directory: No such file or directory.

Doing SETUP.
Reading file "SPECED/setup".
Reading file "SPECED/darren/setup".
Simulate was off, is now ON.

2.DARREN> mono_movemot "0" ""
```





BLISS

Spec and Python

- Complete Integration of Python into SPEC
- Macro function can be written in Python or in built-in language
- Python can access SPEC functions, variables and data arrays
- SPEC as a command line for interactive Python



BLISS

Example code

Spec macro function:

```
def test16() `{  
    return motor_par(0,"step_size") `}'
```

Python code:

```
def test16():  
    return spec.motor_par(0,"step_size")
```

When used from SPEC i.e.: **SPEC> print A[mono]/test16()**

- Spec finds that test16 is a python function
- Python calls a spec function with standard Python types
- The return values are accepted on both "sides"



BLISS

What's missing

- Currently only one Python thread can access SPEC core at the same time (Global SPEC lock)
- Use in Production code
- Rewrite macros in Python



BLISS

Existing Python Modules

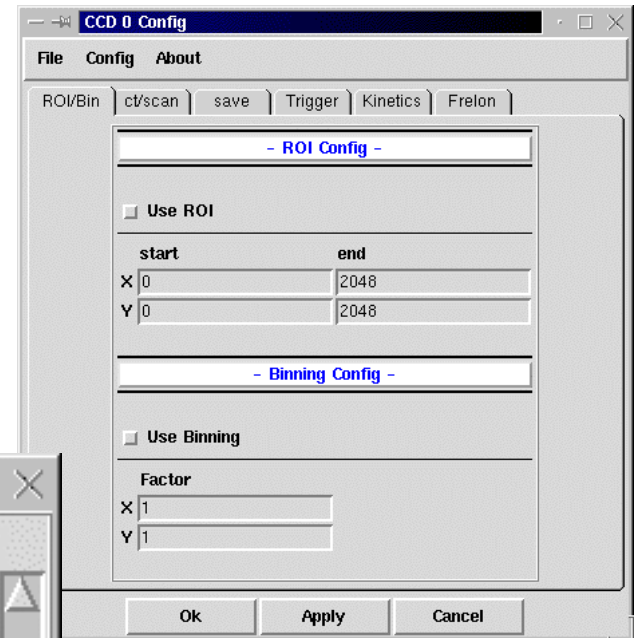
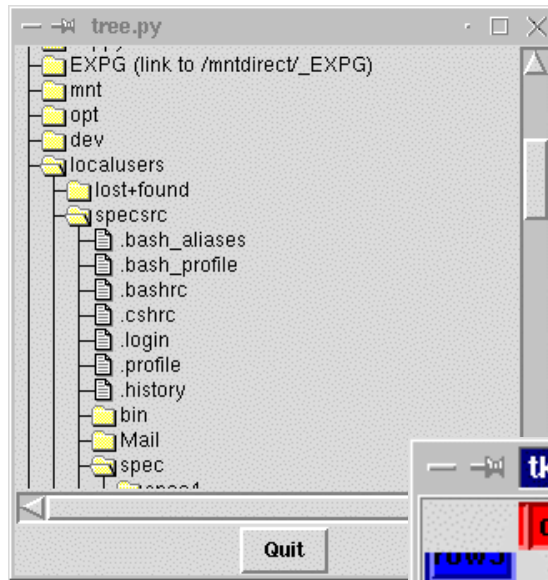
- Data Widgets: 1D, 2D, Data Model, Fitting
- TACO: Server and Client
- File reading: EDF, Spec Scan and MCA
- General Widgets: Tree, Table, Notebook, Parameter input, Wizard
- User input: Python Console, VT100 Widget
- Communication: Event Handler



BLISS

General Widgets

■ Tree, Table, Parameter Input

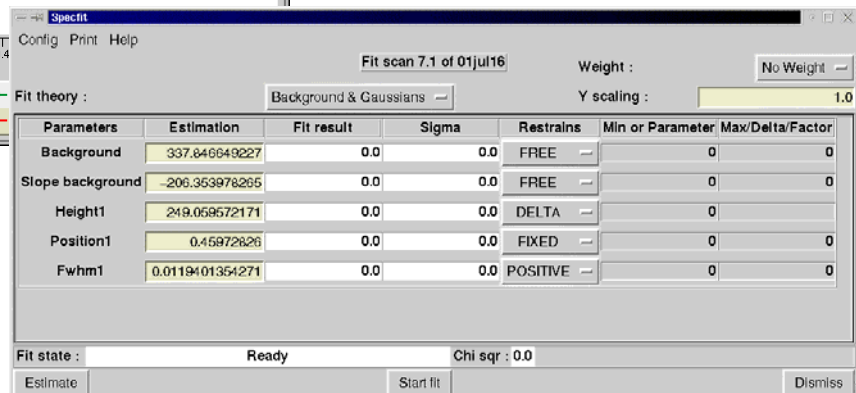
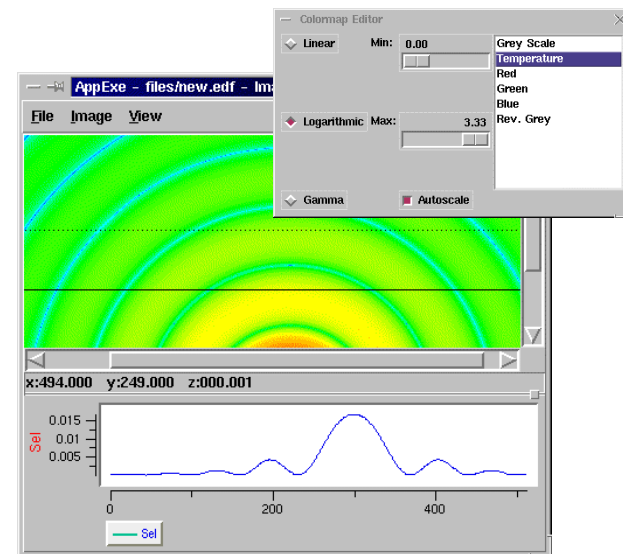
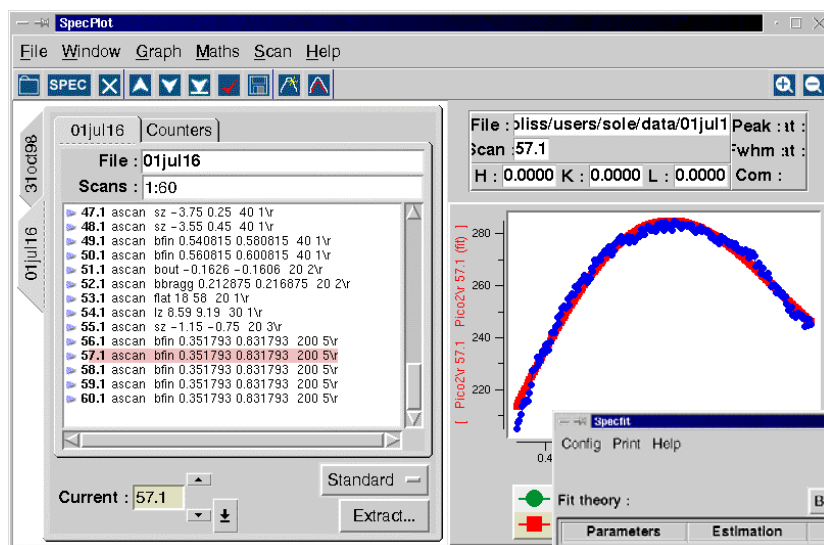




BLISS

Data Widgets

1D, 2D, Fitting





BLISS

Keyboard Input

■ Vt100, Python Console

```
tk
expglin>ls
.
.#,test
..
CmdConsole.py
expglin>

ls
Console.py      TERMIO.py      termios.sl
Console.pyc     doc            tkterm
OLD            pythonvt100.py vt100.py
RCS            specvt100.py  vt100.pyc
```

```
tk <3>
Python 2.1.1c1 (#1, Jul 18 2001, 12:40:07)
[GCC egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)]
Copyright (c) 2001 Python Software Foundation.
All Rights Reserved.

Copyright (c) 2000 BeOpen.com.
All Rights Reserved.

Copyright (c) 1995-2001 Corporation for National Research Initiatives.
All Rights Reserved.

Copyright (c) 1991-1995 Stichting Mathematisch Centrum, Amsterdam.
All Rights Reserved.
>>> |
```



BLISS

Questions

- Which language? -> Python
- Which GUI toolkit? -> Tkinter
- When will it be ready?



BLISS

More Info

- Framework: Darren Spruce
- Data Model: Alexandre Gobbo
- 2D Module: Gilles Berruyer
- 1D Module, Event Handler: Vicente Rey
- Tree: Laurent Claustre
- Fitting Module, Newplot: Armando Sole
- Option Module: Emanuel Papillon
- VT100, Spec Interface: Jörg Klor
- Python Spec: CSS Gerry Swislow
- TACO Interface: Marie Christine Dominguez, Jens Meyer
- Table Module: Nicolas Pascal
- And many more