# GUI Examples using wxPython

11<sup>th</sup> Geant4 Workshop LIP, Lisboa

GUI parallel session 10 October 2006

Hajime Yoshida Naruto University of Education, JST

#### 2000 statement on the interactivity refrained

• There is an obvious requirement for Geant4 to fit into existing Interactive Frameworks. Examples of Interactive Frameworks are: Momo, Explorer, AVS, the BaBar Framework, Gaudi, OPACS, JAS, WIRED, ROOT. They each come with their own way of interacting with the applications inside.

• Geant4 already offers interactivity through intercoms (G4UImanager, several concrete G4UIsession classes - G4Uterminal, etc.); Momo uses this approach.

 <u>An alternative approach commonly used is to wrap application classes directly</u> <u>with wrappers/adaptors.</u> Tools for automatic creation of wrappers exist or are in development, e.g., SWIG (for Tcl, Python, etc.), JACO (Java Access to C++ objects).
Conclusion: it appears that Coant4 is now sufficiently open that both techniques

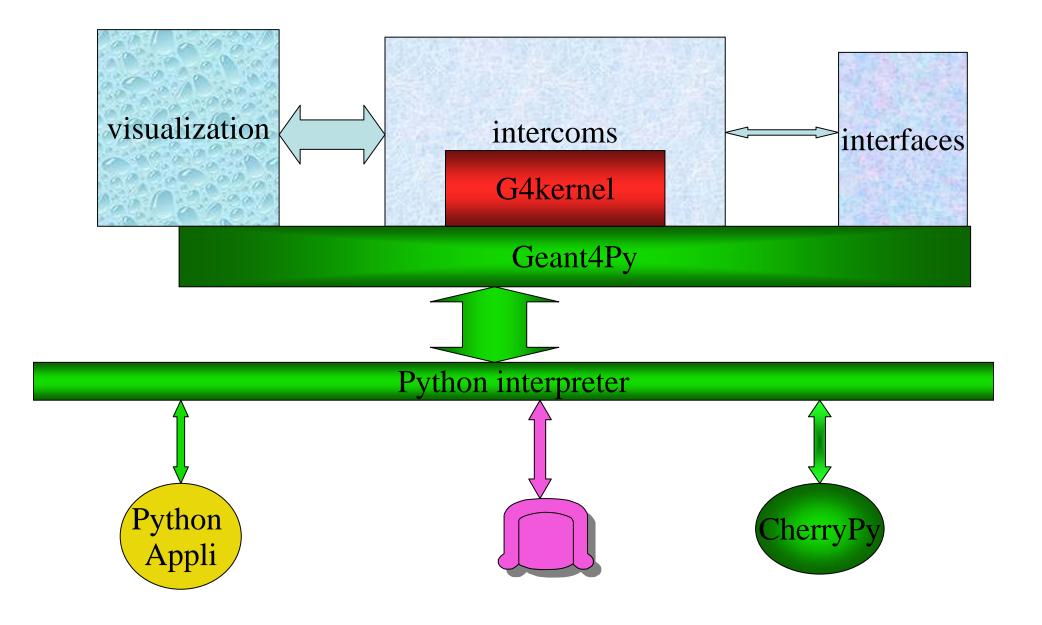
• Conclusion: it appears that Geant4 is now sufficiently open that both techniques can be used but this remains to be tested in a real application.

This would be sufficient for most frameworks. Reservations which arise from the current dominance of the intercoms way of interacting:

<u>There might be, or might come to be, functionality</u> that is only accessible through intercoms, <u>which would restrict direct wrapping techniques and other direct object</u> <u>oriented interfacing techniques</u>.

There is a dual use of intercoms ((a) for inter-category communication and (b) for a command interpreter) and they might need distinguishing in future.

- Now in 2006, we have Geant4Py. <u>WEb page in KEK.</u>
- •We found no restrictions in wrapping G4 objects.



Geant4Py Wrapper

#### Use of Python Wrappers case 1 = C++ + Python

- Create an application in C++ and wrap its classes as necessary
  - Examples are found in g4py/site-modules/
    - ExN03 geometry and physics list
    - Voxelized water phantom etc.
  - Performance isn't deteriorated,
- Much more interactive than the "old" scheme
  - Use of interactive Python shell : ipython
  - See attributes of the exposed objects
  - Powerful Pythons data structure etc. are exploitable
- Python based GUI tool kits can be employed for better user friendliness
- Connection with analysis tools is straightforward

#### Use of Python Wrappers case 2 = purely Pythonic scripting

- Ezgeom
  - No C++ coding is necessary to create your own detector, beam line etc.. Python script can do all.
- Typical e.m. Physics list is provided.
  - Importing them is sufficient to use them in your Python script
- Performance isn't so bad
- Integration with analysis tools and use of fancy GUI tools are just same as the case 1

## WxPython: GUI toolkit

- Tkinter
  - Is the default GUI of Python, since it was included from the beginning
  - Is simple to use but lacking advanced widgets like tree etc.
  - Multi platform but original Tk's look and feel
- WxPython
  - Has been the second popular GUI toolkit after Tkinter
  - Wxwidgets (C++ library) is wrapped with Python
  - Many advanced features and widgets
  - A little lengthy scripting (than Tkinter) to profit its power
  - Multi platform, keeping each platform's look and feel
  - A good book has come

#### Unicode is supported

<b>wxPython: (A Demonstration)</b> <u>File Demo H</u> elp		
Overview → New since last release → wxGenericDirCtrl → wxImageFromStream → RowColSizer → Unicode → Windows → Common Dialogs → Controls → Window Layout → Miscellaneous ↔ wxPython Library → Cool Contribs	Unicode Demo Demo	
	Chinese: Python 是最好的編程語言!	Python is the best programming language!
	Lithuanian: Pythonas yra žaviausia šneka	Python is the best
	Lithuanian: Aš mėgstu šokoladą	l like chocolate
	Korean: 파이썬은 최고의 프로그래밍 언어이다!	Python is the best programming language!
	<sup>Вulgarian:</sup> Питон е най-добрия програмен език!	Python is the best programming language!
	<sup>Russian:</sup> Питон - лучший язык программирования!	Python is the best programming language!
	window handle: 2101298 Running demo Unicode.py	

#### boa\_constructor; IDE

🔯 Boa Constructor - wxPython GUI	Builder	IJŇ		
🖾 🕅 📓 😨 🥔 🤣 🖉				
New Frame bars Containers/Layout Basic Controls Buttons List Controls Utilities Dialogs				
ļ		/i.		
🖳 Inspector		IJŇ		
🖻 🗙 🖫 🗊 🕼 🔗 🗶	File Edit Views Windows Help Columns1			
Constr Props Evts Objs	🔄 🖆 🛅 🔯 💌 🛷 💥 🏈 <sup>Columns2</sup>			
	Shell Explorer 🗊 Boa 🛄 *(wxF			
AutoLayout false BackgroundCc (wxColourPtr)	Source Explore Data			
✓ ClientSize (274, 90)		-1		
Columns (Columns)				
Constraints (Constraints)	imageList1 : wxImageList			
DropTarget				
Enabled true	File			
EvtHandlerEna true	· · · · · · · · · · · · · · · · · · ·			
Font (wxFontPtr)	Columns0 Columns1 Columns2			
ForegroundCol (wxColourPtr)				
Label "				
NormalImageL None				
Shown true				
Sizer				
SmallImageList None				
TextColour (wxColourPtr)				
Title				

### Extending user-friendliness

- Plot tools : <u>matplotlib</u>, plot library a la Matlab
- Web server : <u>CherryPy</u>, purely Pythonic Web server
  - Powerful template language supported
  - Session and cookie management etc.
- GUI toolkit : Tkinter, <u>They say about wxPython</u> - May replace the old GUI tools of Geant4
- <u>Geant4 for Education</u> project, combining the above user friendly environment
- SIG: Python for Education

## **Examples and Demonstration**

- •Educational examples with wxPython GUI
  - -Lesson1 : purely Pythonic script
    - measurement of mass attenuation coefficients in various materials with variable dimensions
    - And other observations
  - -Lesson2 : wrapped C++ clases
    - The classes of Michel Maire's exampleN03 are wrapped
      - sandwich calorimeter geometry is modifiable with GUI
      - electromagnetic processes can be switched on/off with GUI
- •Visualization tools can be switched easily from one to another: OpenGL, VRML or Wired
- •Preliminary implementation of GAG
- •Demonstration
  - -Suse Linux + Pvthon + wxPvthon