



PySide

overview

Marc Pointot
(ONERA/DSNA)



[retour sur innovation](#)

Quite short but practical overview

▶ Qt

- Toolkit overview
- Model/View



▶ PySide

- PyQt4 vs PySide
- Designer & Cython
- Widget bindings
- Class reuse



Qt - Facts

▶ Qt is cute

- Cross platform application framework for GUI
X Window System, Windows...
- C++ kernel + many bindings
Including Python
- Release v5.3 05/2014
- License GPL+LGPL+Commercial
Contact your own lawyer...

▶ Components

- Core: QtCore, QtGui...
- Specialized: QtWebKit, QtSVG, QtSQL...
- Tools : Creator, Designer...

Qt - Example

The image displays the CGNS.NAV software interface, which is a Qt-based application for handling CGNS files. It consists of several windows:

- CGNS.NAV:Control**: A window at the top left showing a file browser with columns for S, T, View, Dir, File, and Node. It lists files like 001 through 006 in the directory `/tmp_user/eos043z/poinot`.
- CGNS.NAV:T001**: A hierarchical tree view showing the structure of a CGNS file. The tree includes:
 - dom1**
 - Zone_t (Shape: (1, 3), D: I4, Value: [[1474, 1368, 0]])
 - ZoneType_t (Shape: (12.), D: C1, Value: Unstructured)
 - GridCoordinates_t (Value: MT)
 - CoordinateX_t (Shape: (1474.), D: R8)
 - CoordinateY_t (Shape: (1474.), D: R8)
 - CoordinateZ_t (Shape: (1474.), D: R8)
 - PENTA_6 (Shape: (2.), D: I4, Value: [[14, 0]])
 - ElementConnectivity_t (Shape: (8208.), D: I4) - This node is selected.
 - ElementRange_t (Shape: (2.), D: I4, Value: [[1, 1368]])
 - QUAD_4 (Shape: (2.), D: I4, Value: [[7, 0]])
 - ElementConnectivity_t (Shape: (128.), D: I4, Value: [[2, 3, 740, 739, 3, 4, 741, 740, 4, 5, 742, 741, 5, 6, ...]])
 - ElementRange_t (Shape: (2.), D: I4, Value: [[1369, 1400]])
 - ZoneBC_t
 - bcu_1_t
 - Famil...
- CGNS.NAV:G004**: A 3D visualization window showing a mesh of a wing-like object. The mesh is colored by element type: red for PENTA_6, blue for QUAD_4, and green for other elements. A legend at the bottom left of the window lists the selected element:
 - dom11/PENTA_6 :{tri }
 - dom11dom11/PENTA_6 :{quad }
 - dom11dom11/QUAD_4 :{quad }
- CGNS.NAV:G004 (Table)**: A table window showing the data for the selected element. The table has columns 1 through 4 and rows 1 through 14.

	1	2	3	4
1	113	207	230	850
2	854	855	980	104
3	117	238	118	854
4	901	930	917	188
5	109	188	214	846
6	989	847	964	214
7	19	20	236	756
8	868	862	869	236
9	220	55	56	957
10	971	802	803	239
11	192	163	186	929
12	953	834	887	216
13	159	272	290	896
14	1039	1032	819	308

Python bindings

▶ PyQt

- The first to come

Some services have hard coded `import PyQt4`

- GPL - Use only in free software

▶ PySide

- Some syntactic & behavior differences
- LGPL - Use allowed in proprietary software

PySide overview hereafter mixes Qt/PySide features

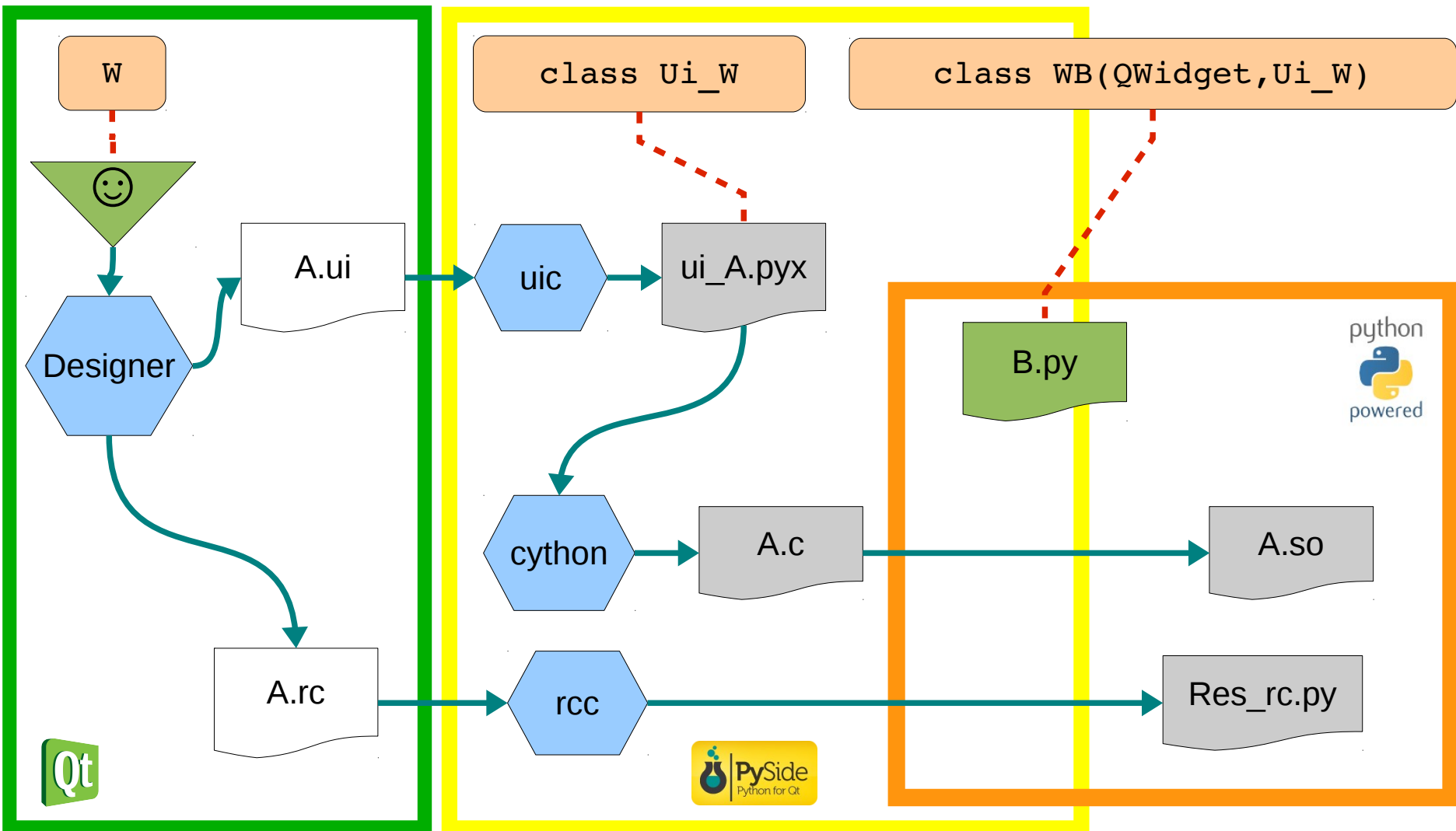
PySide - Facts

- ▶ Full Python binding
 - Qt classes as Python classes
 - Python types as parameter types
- ▶ Release 1.2.2 04/2014
- ▶ Reference documentation

`http://pyside.github.io/docs/pyside/`

- ▶ Production process
 - Uses many steps
 - Better with setup & source management

PySide - Production process



PySide - Example

```
from PySide.QtCore import *
from PySide.QtGui import *
from Gview import gui

import numpy as NPY
import vtk
from os.path import splitext

class GMain(QWidget,gui.Ui_controlWindow):
    # -----
    def __init__(self):
        QWidget.__init__(self,None)
        self.setupUi(self)
        self.b_load.clicked.connect(self.b_loadOneFile)
        self.b_X.clicked.connect(self.b_xaxis)
    # -----
    def b_xaxis(self,pos=None):
        if (self.c_mirror.isChecked()): self.setAxis(pos,-1)
        else: self.setAxis(pos,1)
    # -----
    def closeEvent(self, event):
        print '*** Gview: close'
        event.accept()
```




PySide

overview

Marc Pointot
(ONERA/DSNA)



retour sur innovation

Outline

Quite short but practical overview

▶ Qt

- Toolkit overview
- Model/View



▶ PySide

- PyQt4 vs PySide
- Designer & Cython
- Widget bindings
- Class reuse



Qt - Facts

▶ Qt is cute

- Cross platform application framework for GUI
X Window System, Windows...
- C++ kernel + many bindings
Including Python
- Release v5.3 05/2014
- License GPL+LGPL+Commercial
Contact your own lawyer...

▶ Components

- Core: QtCore, QtGui...
- Specialized: QtWebKit, QtSVG, QtSQL...
- Tools : Creator, Designer...

Qt - Example

The screenshot displays the CGNS.NAV:Control and CGNS.NAV:T001 software interfaces. The CGNS.NAV:T001 window shows a hierarchical tree of data for 'dom1', including ZoneType, GridCoordinates, CoordinateX, CoordinateY, CoordinateZ, PENTA_6, ElementConnectivity, ElementRange, QUAD_4, and ZoneBC. A 3D mesh visualization is shown in the bottom right, with a central white object and surrounding colored mesh elements. A table of data is visible in the bottom left of the CGNS.NAV:T001 window.

	1	2	3	4
1	113	207	230	850
2	854	855	980	104
3	117	238	118	854
4	901	930	917	188
5	109	188	214	846
6	989	847	964	214
7	19	20	236	756
8	868	862	869	236
9	220	55	56	957
10	971	802	803	239
11	192	163	186	929
12	953	834	887	216
13	159	272	290	896
14	1039	1032	819	308

Python bindings

▶ PyQt

- The first to come
 - Some services have hard coded `import PyQt4`
- GPL - Use only in free software

▶ PySide

- Some syntactic & behavior differences
- LGPL - Use allowed in proprietary software

PySide overview hereafter mixes Qt/PySide features

PySide - Facts

- ▶ Full Python binding
 - Qt classes as Python classes
 - Python types as parameter types

- ▶ Release 1.2.2 04/2014

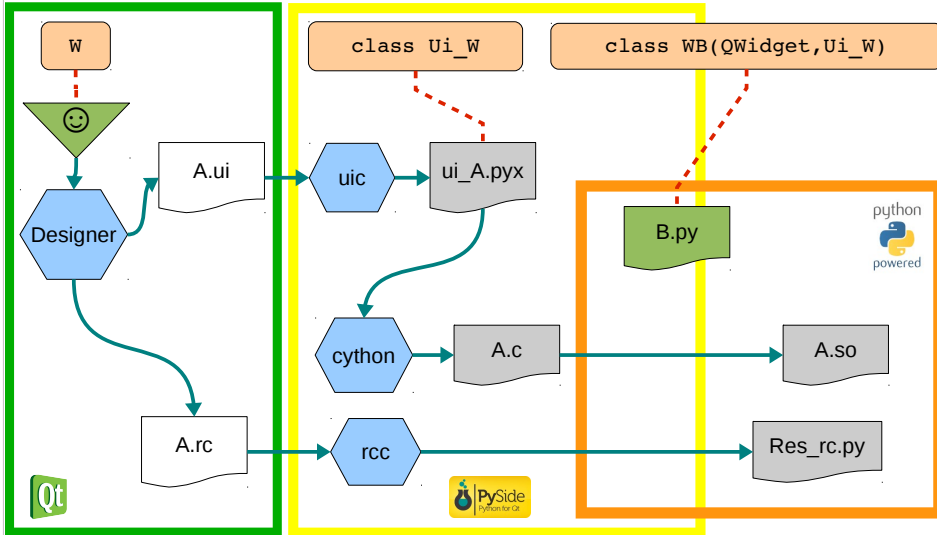
- ▶ Reference documentation

<http://pyside.github.io/docs/pyside/>

- ▶ Production process

- Uses many steps
- Better with setup & source management

PySide - Production process



PySide - Example

```
from PySide.QtCore import *
from PySide.QtGui import *
from Gview import gui

import numpy as NPY
import vtk
from os.path import splitext

class GMain(QWidget,gui.Ui_controlWindow):
    # -----
    def __init__(self):
        QWidget.__init__(self,None)
        self.setupUi(self)
        self.b_load.clicked.connect(self.b_loadOneFile)
        self.b_X.clicked.connect(self.b_xaxis)
    # -----
    def b_xaxis(self,pos=None):
        if (self.c_mirror.isChecked()): self.setAxis(pos,-1)
        else: self.setAxis(pos,1)
    # -----
    def closeEvent(self, event):
        print '*** Gview: close'
        event.accept()
```