Extending ParaView

Utkarsh Ayachit, Dave DeMarle

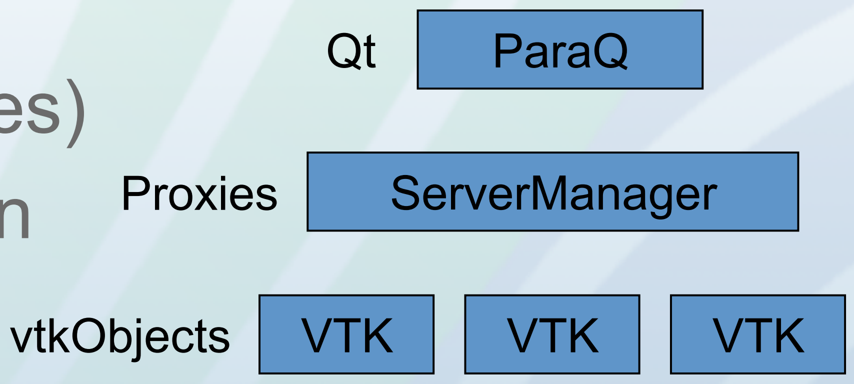
Introduction

- ParaView is an Open Source application and architecture for visualization and analysis of massive data sets.
- Open Source Architecture - it is supposed to be reusable
- ~1 Million lines of code
- Recent work makes it trivial to Revise it

The Learning Curve

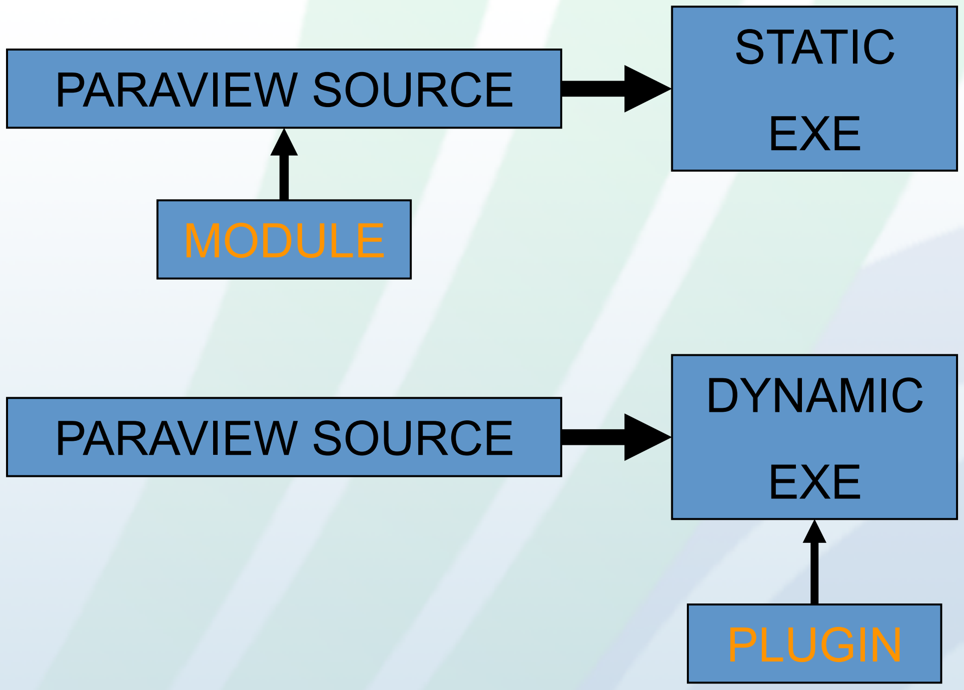
- Takes too long to learn enough to change it in meaningful ways

- Topics to master
  - VTK
  - ServerManager (Proxies)
  - paraQ client application



Three approaches to revising ParaView

- Edit code directly (open source after all)
- Plugins and Modules
  - CMake Macros that codify the way to bring code into, or make code that is loadable by ParaView
- Custom Applications
  - Completely new executables that reuse the servermanager layer (ex tcl/tk app, c++ apps, python apps, PVEE webvis app)



Problems with the three Approaches

- Edit code directly

  too invasive/not modular enough, too hard to keep current, too much code to keep track of
- Plugins and Macros

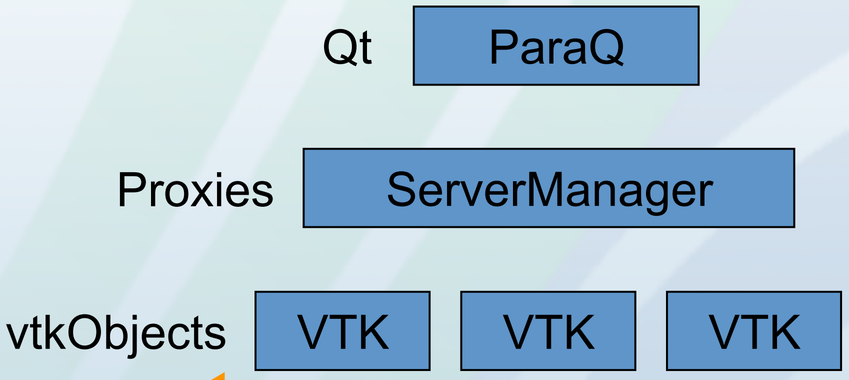
  Good at Adding, difficult to Subtract
- Custom Applications

  at ServerManager layer?

    too time consuming

  at Client layer?

    client not modular enough **yet**



Kitware

Custom Applications

- To make a targeted vis application
- Application design is non trivial effort
- Ideally reuse effort that went into existing Client
- Top down design:
  – Copy/Paste client's source code then cut down
  – Inelegant and not as easy as it sounds
- Bottom up design:
  – Start with a minimal core, then add
  – Existing app doesn't have a minimal core!
  – Paper is about changes that make bottom up possible

Kitware

Motivation

- ParaView is intended to be a general purpose visualization and analysis tool
- Existing plugins and macros make it possible to add even more (domain specific) features
- But how do you remove the stuff that a domain expert doesn't care about?
  - Reduced selection of file formats
  - Reduced selection of filters
  - Reduced set of view types
- How to make big changes to key GUI elements?

Problem : Monolithic application

- Executable compilation and startup is arcane
- Qt components of the app are completely interdependent
- Behaviors are hard coded into the application logic

COMPONENT DEPENDENCIES

pqPipelineBrowser

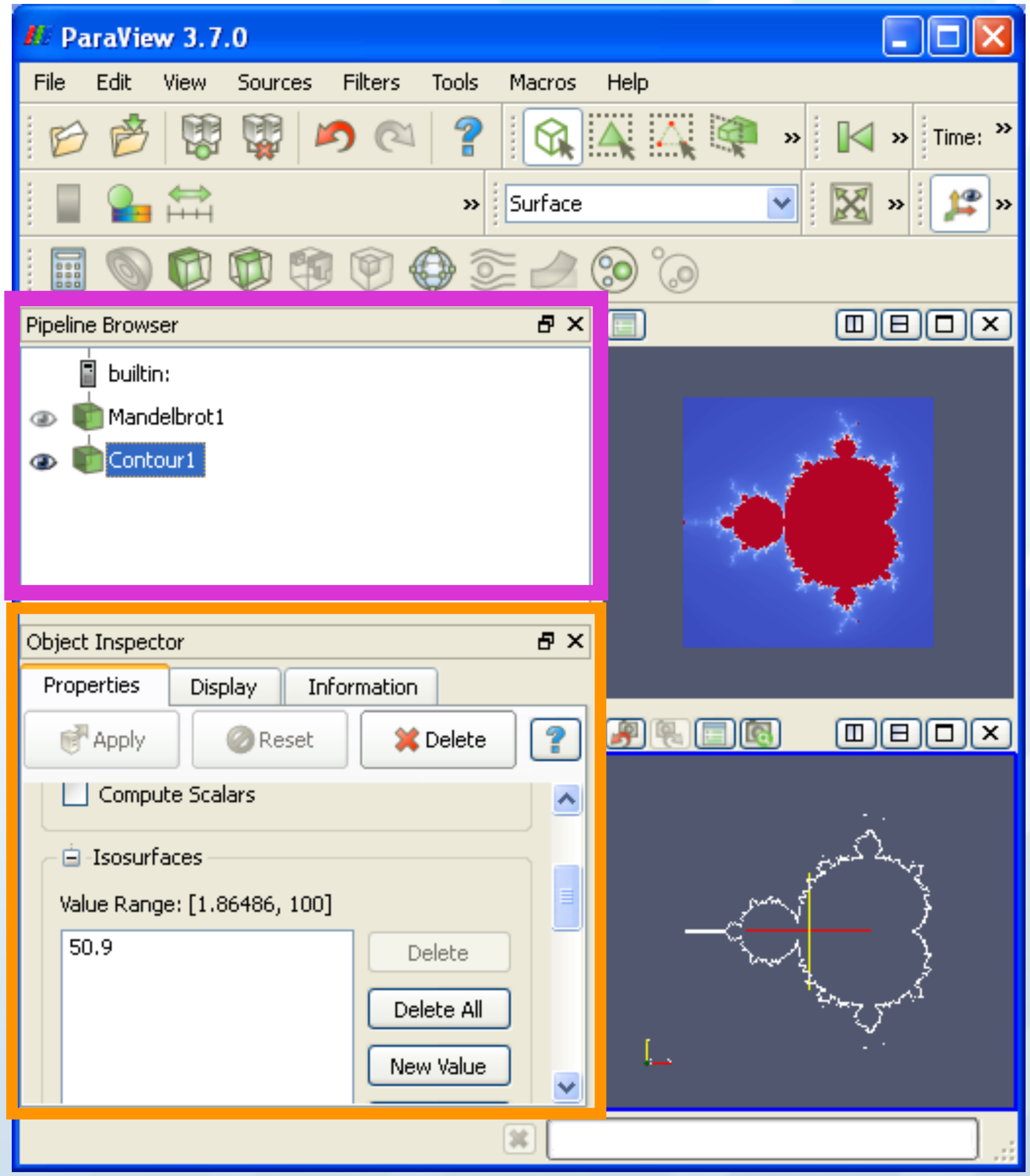
signal: newActiveFilter

pqProxyTabWidget

slot: onNewActiveFilter

Direct dependencies, can not have one without the other.

Many many more…



HARD CODED ASSUMPTIONS

Allow MultiView

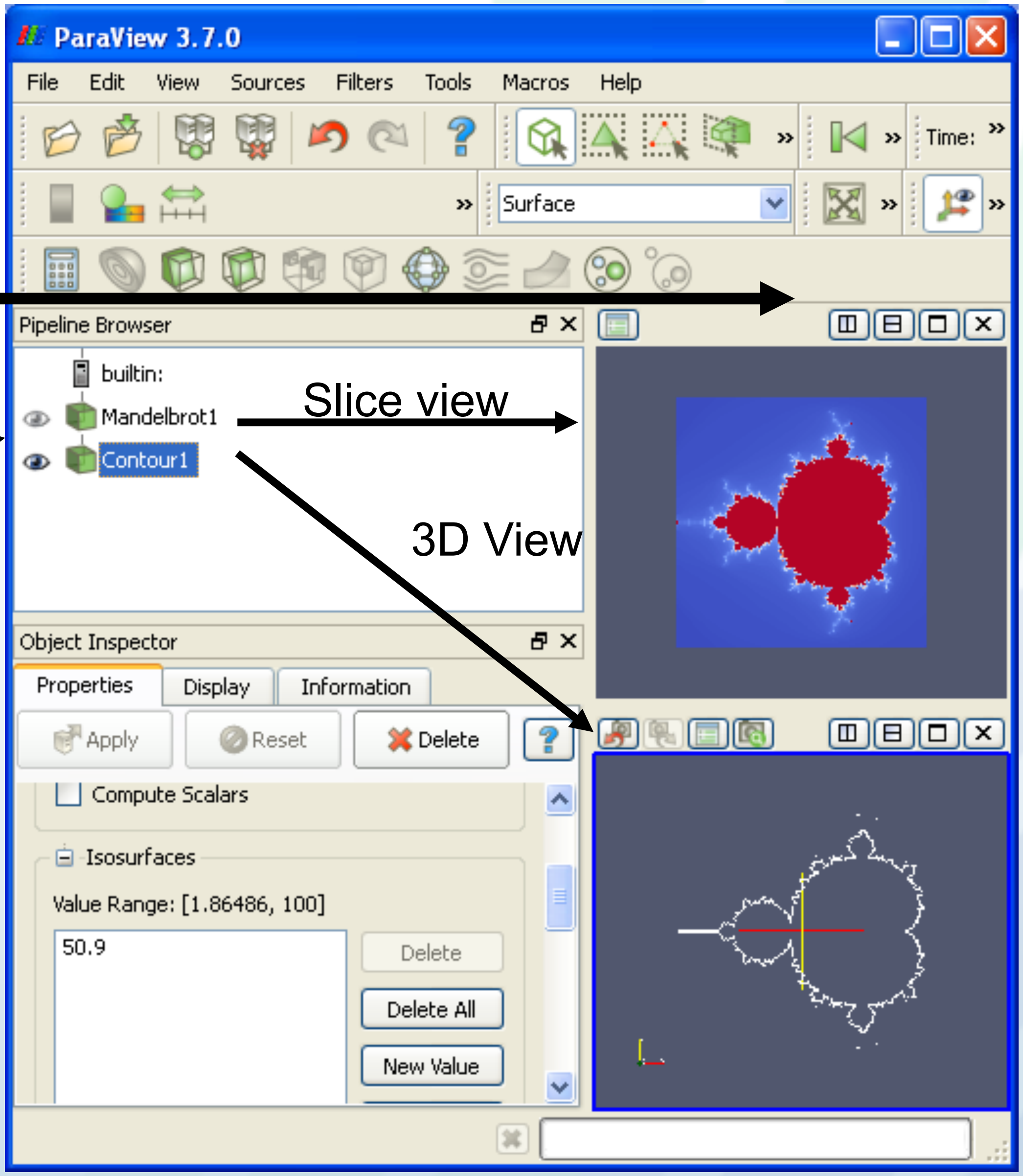
Default View Types

for server,

for particular filters

Display Pipeline
Creation for each filter

etc



Kitware

Simplified Application Construction

A minimal Qt application just above ServerManager layer is now:

CMakeLists.txt

```
SET (SOURCE_FILES  DemoApp0.cxx)
INCLUDE_DIRECTORIES(
 ${CMAKE_CURRENT_SOURCE_DIR}
 ${CMAKE_CURRENT_BINARY_DIR})
ADD_EXECUTABLE(DemoApp0 $
     {SOURCE_FILES}
      ${MOC_SRCS} $
     {UI_BUILT_SOURCES})
TARGET_LINK_LIBRARIES(DemoApp0
     pqCore ${QT_LIBRARIES} )
```

DemoApp0.cxx

```
#include <QApplication>
#include "pqApplicationCore.h"
#include <QMainWindow.h>
int main(int argc, char** argv) {
  QApplication app(argc, argv);
  pqApplicationCore appCore(argc,
     argv);
  QMainWindow window;
  window.show();
  return app.exec();
}
```

Branded Applications

- Instead of copying and editing a few thousand lines of code, ask the macro to put together the major components you need
- Supply arguments like
  - Title
  - Splash image
  - Proxy defining xml files
  - Source filenames
- Macro builds up the required glue

```
build_paraview_client(paraview_revised_2
  TITLE "ParaView (ReVisEd)"
  ORGANIZATION  "Kitware Inc."
  VERSION_MAJOR 1
  VERSION_MINOR 1
  VERSION_PATCH 1
  SPLASH_IMAGE
    "${CSD}/RSplash.png"
  PVMAIN_WINDOW myMainWindow
  PVMAIN_WINDOW_INCLUDE
     myMainWindow.h
  GUI_CONFIGURATION_XMLS    ${CSD}/
     ParaViewSources.xml   ${CSD}/
     ParaViewFilters.xml    ${CSD}/
     ParaViewReaders.xml    ${CSD}/
     ParaViewWriters.xml
  SOURCES $
     {ParaView_SOURCE_FILES}
)
```

Kitware

Reactions

- Think of a centralized notification service
- Allows application components to stand alone
  - don't need direct signal->slot connections
- Encapsulates logic for enable state of widgets
- Ex:

```
new pqLoadDataReaction(ui.actionLoadData);
pqHelpReaction::showHelp(   QString("qthelp://
   paraview.org/paraview/%1.html").arg(proxyname));
```

Kitware

Builders

- Functions that populate application GUI with reactions

  pqParaViewMenuBuilders::buildFileMenu(menu_File);

  - Creates reactions for quit, open file, save data, etc

  pqParaViewMenuBuilders::buildFiltersMenu(menu_Filters);

  - Makes reactions that populate filters menu from contents of GUI_CONFIGURATION_XMLS: ${CSD}/ParaViewFilters.xml

  pqParaViewMenuBuilders::buildToolbars(this);

  - Build the standard set of toolbars

- Easily clone standard ones (as above does)
- Easily subset by copying individual reactions out of pqParaViewMenuBuilders

Behaviors

- Encode the way the application acts
- Centralized notifications
  - Helps breaks apart widget dependencies too
  - Assumptions no longer hardcoded throughout
  - Makes it easy to modify assumptions made
- Easily clone the standards ones

```
new pqParaViewBehaviours(this);
```

- Or easily pick and choose

```
new pqDefaultViewBehaviour(this);
new pqAlwaysConnectedBehavior(this);
```

Example 1 : Minimal Vis App

- DemoApp1
  minimal 3D Visualization capability
  like VTK RenderWindow
  built on top of ServerManager

  ~50 lines of code for app

Example 2 : Special Purpose App

- SpreadSheet

  Data centric app with none of ParaView’s workflow

Example 3 : Exact Clone

- Exact clone of ParaView app in 218 lines (most of which are comments)

Example 4 : Subset Clone

- An clone application that gets rid of large portions of ParaView functionality
  - Pipeline Browser
  - Most ToolBars
  - Most File Menu Reactions
  - Most readers, sources and filters

Try it!

- Not yet available in ParaView CVS
- Planned for 3.8
- Until then:
  - git::/github.com/utkarshayachit/ParaView.git

Kitware