Medical Application Development Start off

Introduction

Presented By: Sohair Ali

Outlines

- Introduction
- Basic APIs(application programming interface)
  - Open inventor
  - VTK
  - ITK
- Rapid Application Development Tools and Libraries (RAD)
  - XIP
  - MeVisLab

Introduction

- Third-party software
  - It is a reusable software component developed to be either freely distributed or sold.
  - Its market increases because many programmers believe that component oriented development improves the efficiency and quality of developing custom applications.

Introduction

- Rapid Application Development(RAD) Tools and Libraries
  - A programming system that enables programmers to quickly build working programs.
  - In general, RAD systems provide a number of tools to help build projects that would normally take a large development effort.
  - Historically, RAD systems have tended to emphasize reducing development time and/or make fast prototypes.

Symbyo TECHNOLOGIES

Basic APIs

Open Inventor

- It is an object oriented 3D visualization toolkit.
- Build over openGL .
- Applications are built with Open Inventor including:
  - Geosciences applications.
  - Engineering design and simulation applications.
  - Medical applications
  - Virtual reality applications.

Open GL Example

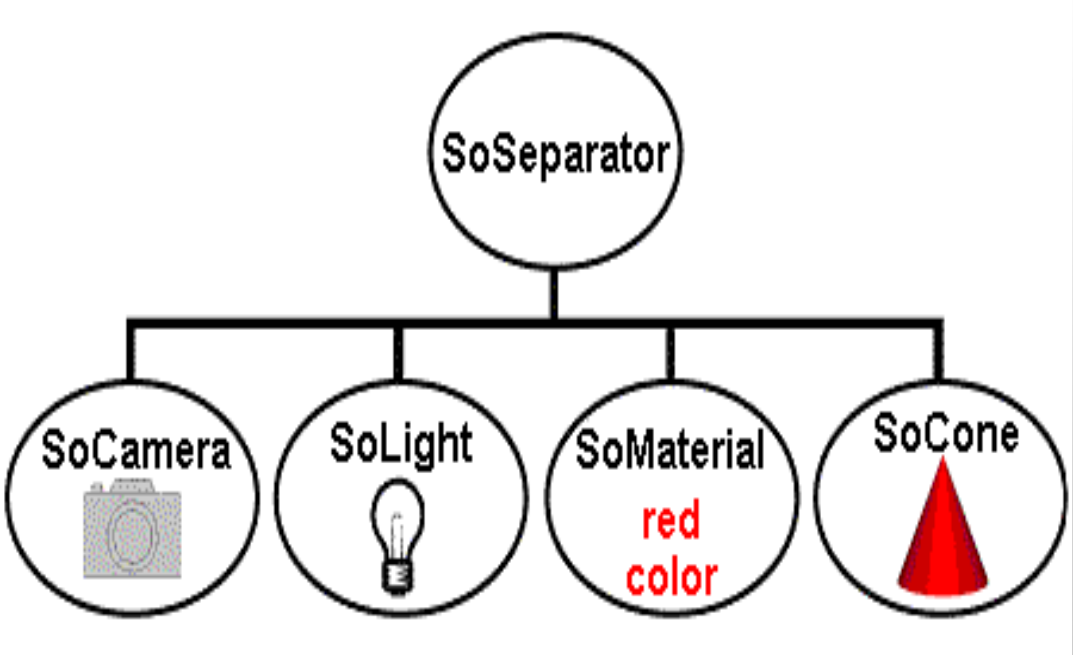
```
glBegin(GL_TRIANGLES);      // Begin Drawing Triangles
glColor3f(1.0f,0.0f,0.0f);      // Set The Color To Red
 // Move Up One Unit From Center (Top Point)
glVertex3f( 0.0f, 1.0f, 0.0f);
 // Left And Down One Unit (Bottom Left)
glVertex3f(-1.0f,-1.0f, 0.0f);
// Right And Down One Unit (Bottom Right)
 glVertex3f( 1.0f,-1.0f, 0.0f);
glEnd();     // Done Drawing A Triangle
```

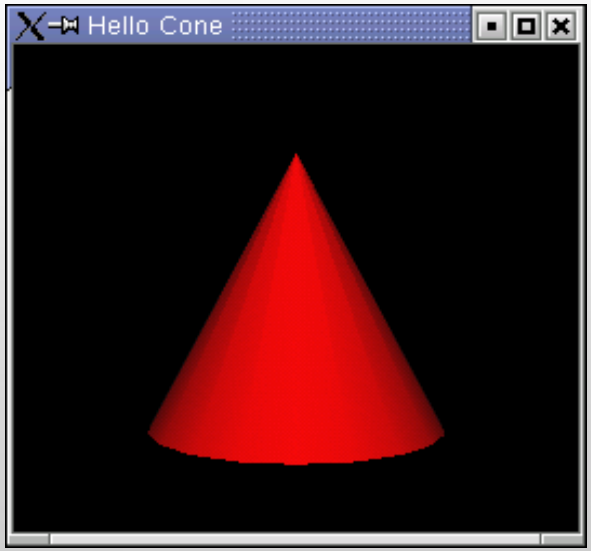
Open Inventor

- Scene graph : Tree of objects called nodes
- Traversed by actions to:
  - Render
  - Pick
  - Search
  - Write
- Open Inventor: Object oriented , High level

Symbyo TECHNOLOGIES

Open Inventor





Symbyo TECHNOLOGIES

Open Inventor

```
#include <Inventor/Qt/SoQt.h>
#include <Inventor/Qt/SoQtRenderArea.h>
#include <Inventor/nodes/SoCone.h>
#include <Inventor/nodes/SoDirectionalLight.h>
#include <Inventor/nodes/SoMaterial.h>
#include <Inventor/nodes/SoPerspectiveCamera.h>
#include <Inventor/nodes/SoSeparator.h>


int main(int , char **argv)
{
  // Init Inventor
  QWidget *window = SoQt::init(argv[0]);
  if (window == NULL) exit(1);

  // create scene root
  SoSeparator *root = new SoSeparator;
  root->ref();

  // create camera
  SoPerspectiveCamera *camera = new SoPerspectiveCamera;
  root->addChild(camera);

  // light
  root->addChild(new SoDirectionalLight);

  // material
  SoMaterial *material = new SoMaterial;
  material->diffuseColor.setValue(1.0, 0.5, 0.0);
  root->addChild(material);

  // cone
  root->addChild(new SoCone);

  // create window
  SoQtRenderArea *renderArea = new SoQtRenderArea(window);

  // set up camera
  camera->viewAll(root, renderArea->getViewportRegion());

  // set up rendering window
  renderArea->setSceneGraph(root);
  renderArea->setTitle("Hello Cone");
  renderArea->show();

  // show window and enter main loop
  SoQt::show(window);
  SoQt::mainLoop();

  // release memory
  delete renderArea;
  root->unref();

  return 0;
}
```

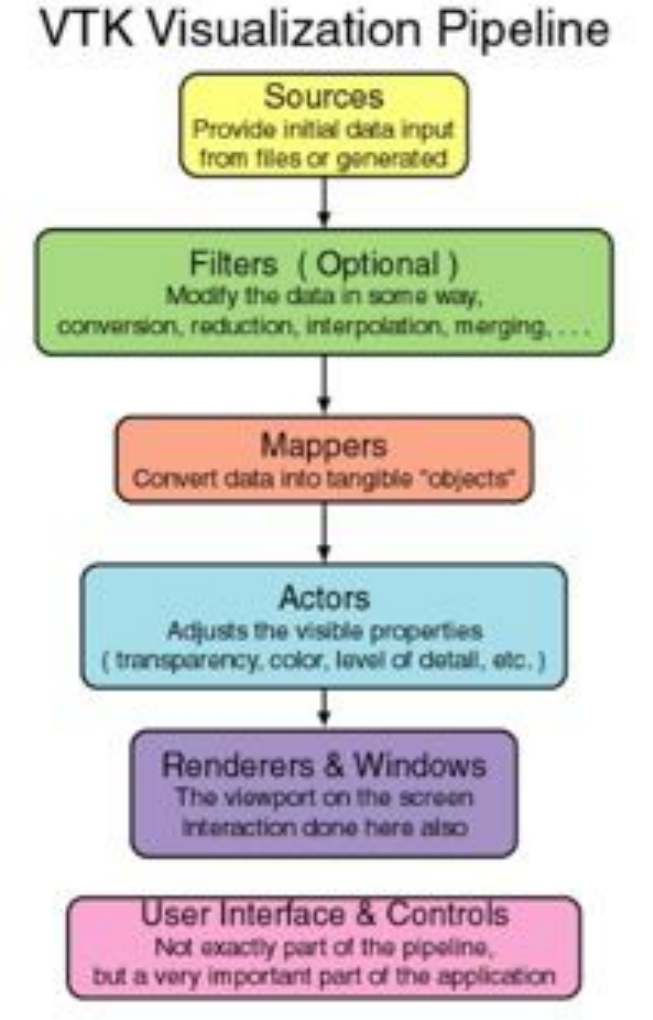
Symbyo TECHNOLOGIES

VTK

- The Visualization Toolkit (VTK) is an open-source, freely available software system for:
  - 3D computer graphics
  - image processing
  - visualization.
- object-oriented design (C++) .
- High-level of abstraction .
- VTK architecture is based on a demand-driven, pipeline architecture (Computation is performed on demand).

Symbyo TECHNOLOGIES

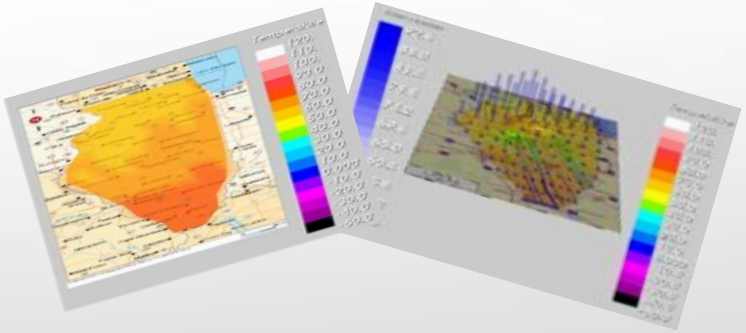
VTK



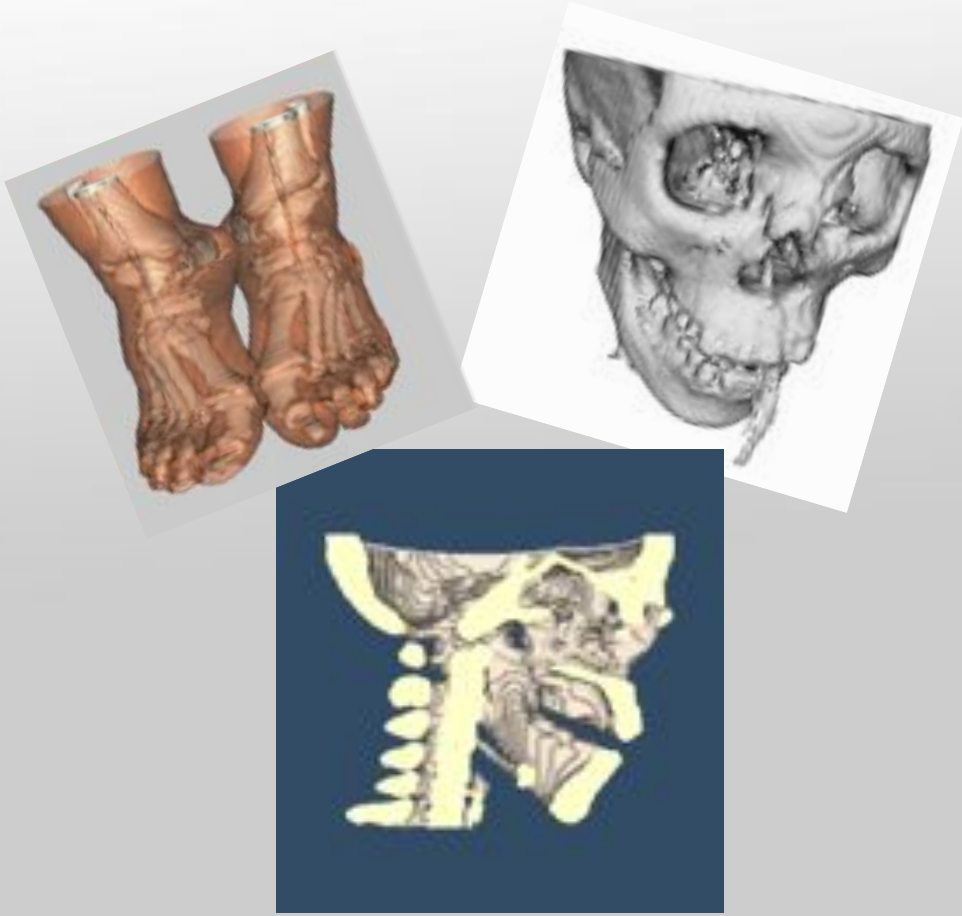
Symbyo TECHNOLOGIES

VTK

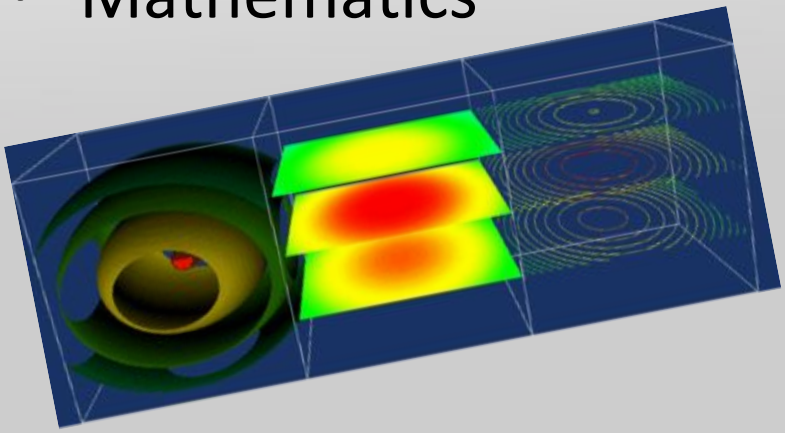
- Weather



- Medical Data

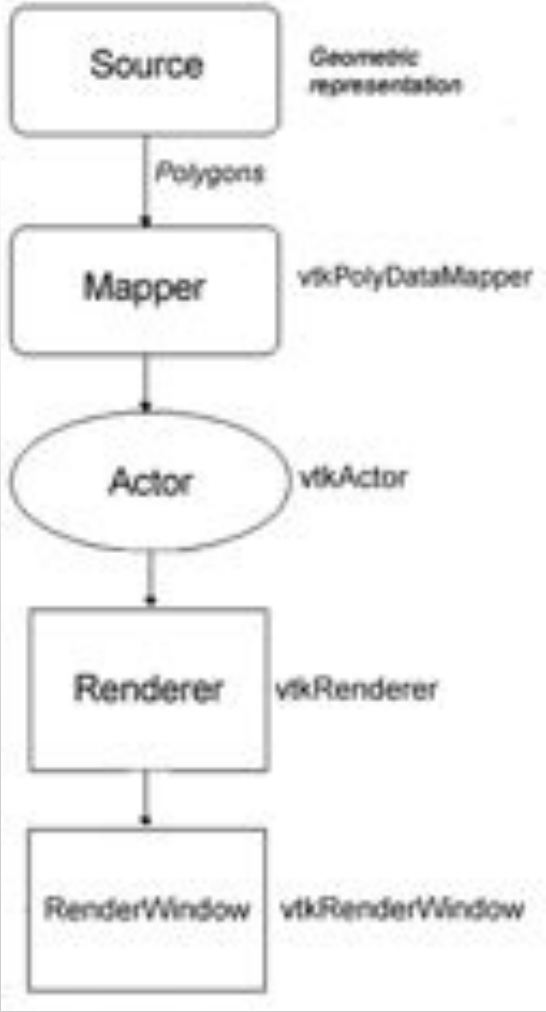


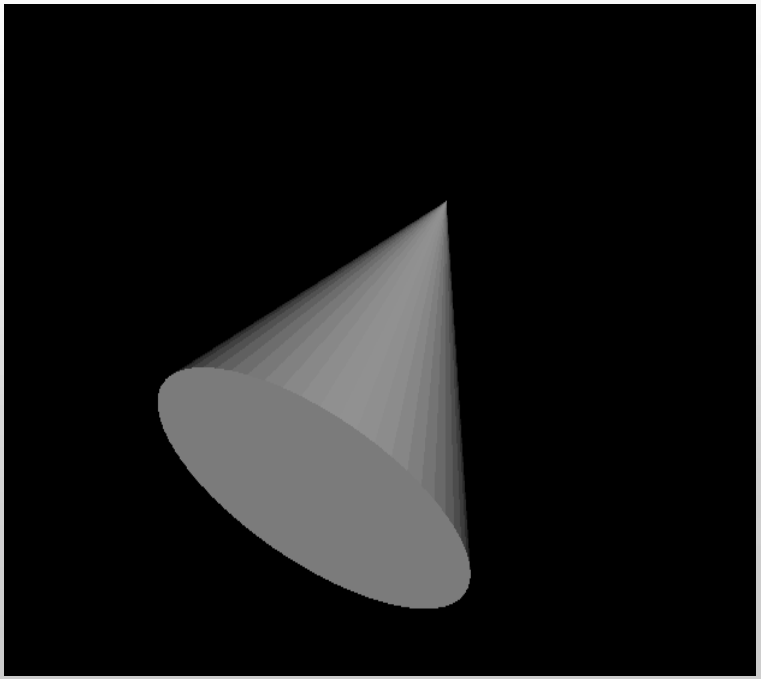
- Mathematics



Symbyo TECHNOLOGIES

VTK





VTK

```
#include <vtkConeSource.h>
#include <vtkPolyData.h>
#include <vtkSmartPointer.h>
#include <vtkPolyDataMapper.h>
#include <vtkActor.h>
#include <vtkRenderWindow.h>
#include <vtkRenderer.h>
#include <vtkRenderWindowInteractor.h>

int main(int, char *[])
{
  //Create a cone
  vtkSmartPointer<vtkConeSource> coneSource =
    vtkSmartPointer<vtkConeSource>::New();
  coneSource->Update();

  //Create a mapper and actor
  vtkSmartPointer<vtkPolyDataMapper> mapper =
    vtkSmartPointer<vtkPolyDataMapper>::New();
  mapper->SetInputConnection(coneSource->GetOutputPort());

  vtkSmartPointer<vtkActor> actor =
    vtkSmartPointer<vtkActor>::New();
  actor->SetMapper(mapper);

  //Create a renderer, render window, and interactor
  vtkSmartPointer<vtkRenderer> renderer =
    vtkSmartPointer<vtkRenderer>::New();
  vtkSmartPointer<vtkRenderWindow> renderWindow =
    vtkSmartPointer<vtkRenderWindow>::New();
  renderWindow->AddRenderer(renderer);
  vtkSmartPointer<vtkRenderWindowInteractor> renderWindowInteractor =
    vtkSmartPointer<vtkRenderWindowInteractor>::New();
  renderWindowInteractor->SetRenderWindow(renderWindow);

  //Add the actors to the scene
  renderer->AddActor(actor);
  renderer->SetBackground(.3, .2, .1); // Background color dark red

  //Render and interact
  renderWindow->Render();
  renderWindowInteractor->Start();

  return EXIT_SUCCESS;
}
```

Symbyo TECHNOLOGIES

ITK

- Open Source
- Written in C++
- Cross-Platform
- Image Processing , Segmentation and Registration
- No Visualization
- ITK has a Data Pipeline architecture , Computation is performed on demand

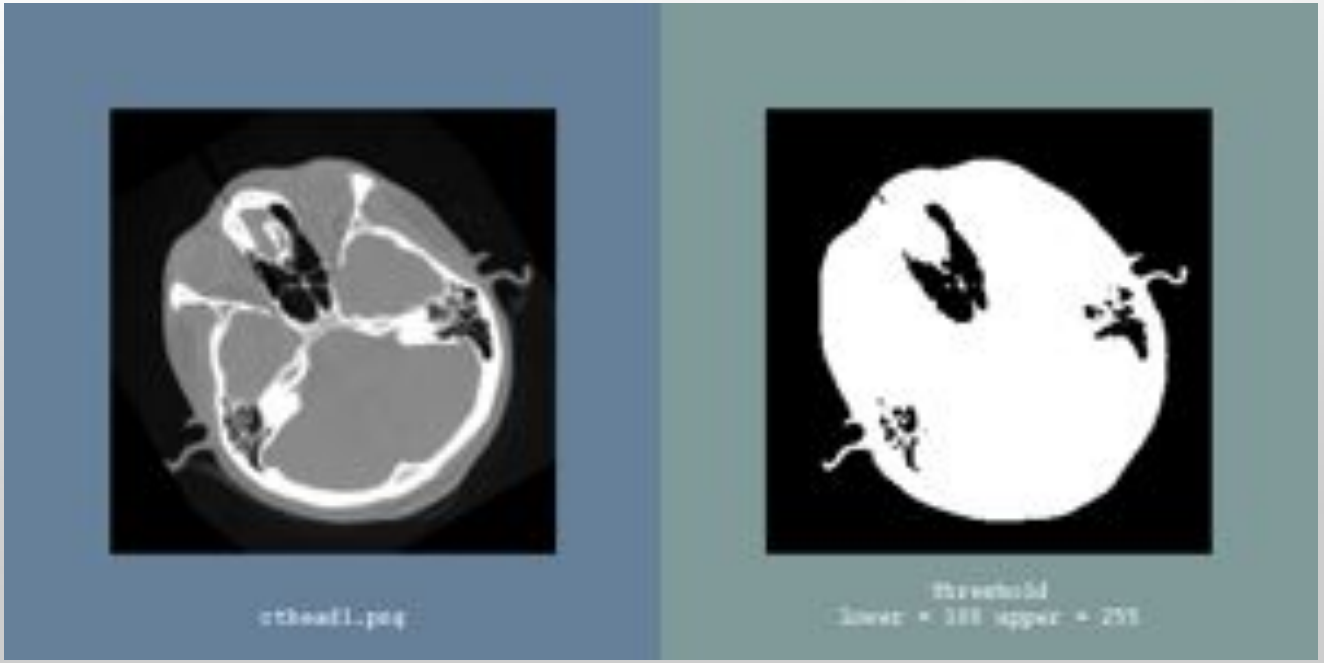
Symbyo TECHNOLOGIES

ITK

```
#include "itkImage.h"
#include "itkBinaryThresholdImageFilter.h"
#include <itkImageFileReader.h>

#include "QuickView.h"

int main(int argc, char *argv[])
{
  if(argc < 2)
    {
    std::cout << "Usage: " << std::endl;
    std::cout << argv[0] << " inputImageFile [lowerThreshold] [upperThreshold]" << std::endl;
    return EXIT_FAILURE;
    }

  int lowerThreshold = 10;
  int upperThreshold = 20;
  if (argc > 2)
    {
    lowerThreshold = atoi(argv[2]);
    }
  if (argc > 3)
    {
    upperThreshold = atoi(argv[3]);
    }

  typedef itk::Image<unsigned char, 2>  ImageType;
  typedef itk::ImageFileReader<ImageType> ReaderType;

  ReaderType::Pointer reader = ReaderType::New();
  reader->SetFileName(argv[1]);

  typedef itk::BinaryThresholdImageFilter <ImageType, ImageType>
    BinaryThresholdImageFilterType;

  BinaryThresholdImageFilterType::Pointer thresholdFilter
    = BinaryThresholdImageFilterType::New();
  thresholdFilter->SetInput(reader->GetOutput());
  thresholdFilter->SetLowerThreshold(lowerThreshold);
  thresholdFilter->SetUpperThreshold(upperThreshold);
  thresholdFilter->SetInsideValue(255);
  thresholdFilter->SetOutsideValue(0);

  QuickView viewer;
  viewer.AddImage<ImageType>(
    reader->GetOutput(), true,
    itkSys::SystemTools::GetFilenameName(argv[1]));
  std::stringstream desc;
  desc << "Threshold[lower = " << lowerThreshold
       << " upper = " << upperThreshold;
  viewer.AddImage<ImageType>(
    thresholdFilter->GetOutput(),
    true,
    desc.str());
  viewer.Visualize();

  return EXIT_SUCCESS;
}
```

Symbyo TECHNOLOGIES

ITK



Rapid Development Tools

- Environment for rapidly developing medical imaging applications from an extensible set of modular elements and libraries .

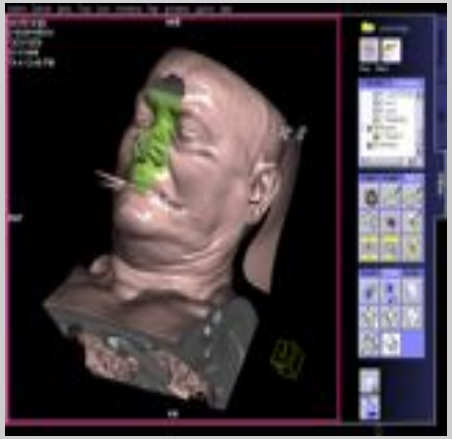
- It allows developers to easily develop, evaluate, and new approaches to medical imaging problems in an early research stage.

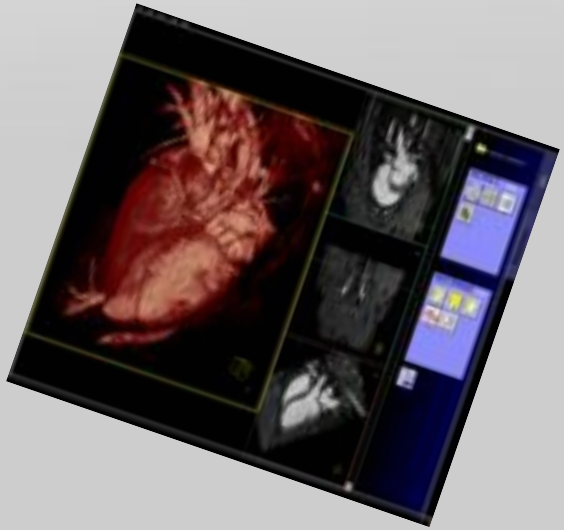
Rapid Development Tools

XIP

- The “eXtensible Imaging Platform” is an open source environment for rapidly developing medical imaging applications .
- The XIP Libraries based on :

  Open Inventor , ITK and VTK libraries






Symbyo
TECHNOLOGIES

XIP


caBIG
cancer Biomedical Informatics Grid
An Initiative of the National Cancer Institute

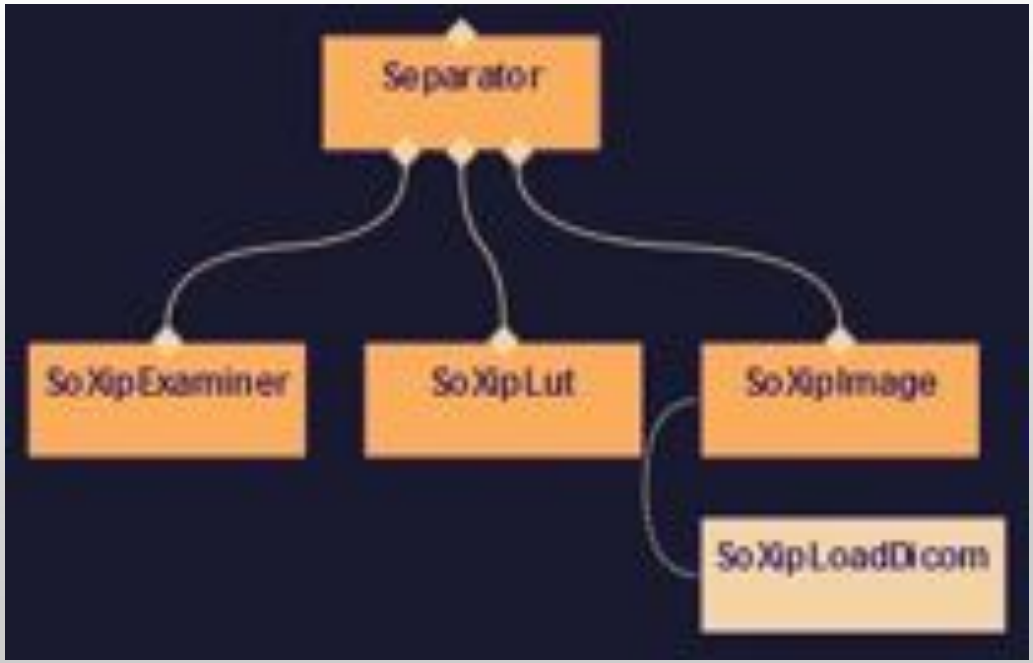

MIR
Mallinckrodt Institute of Radiology


Washington University in St. Louis
School of Medicine


SIEMENS
corporate research

Symbyo TECHNOLOGIES

XIP



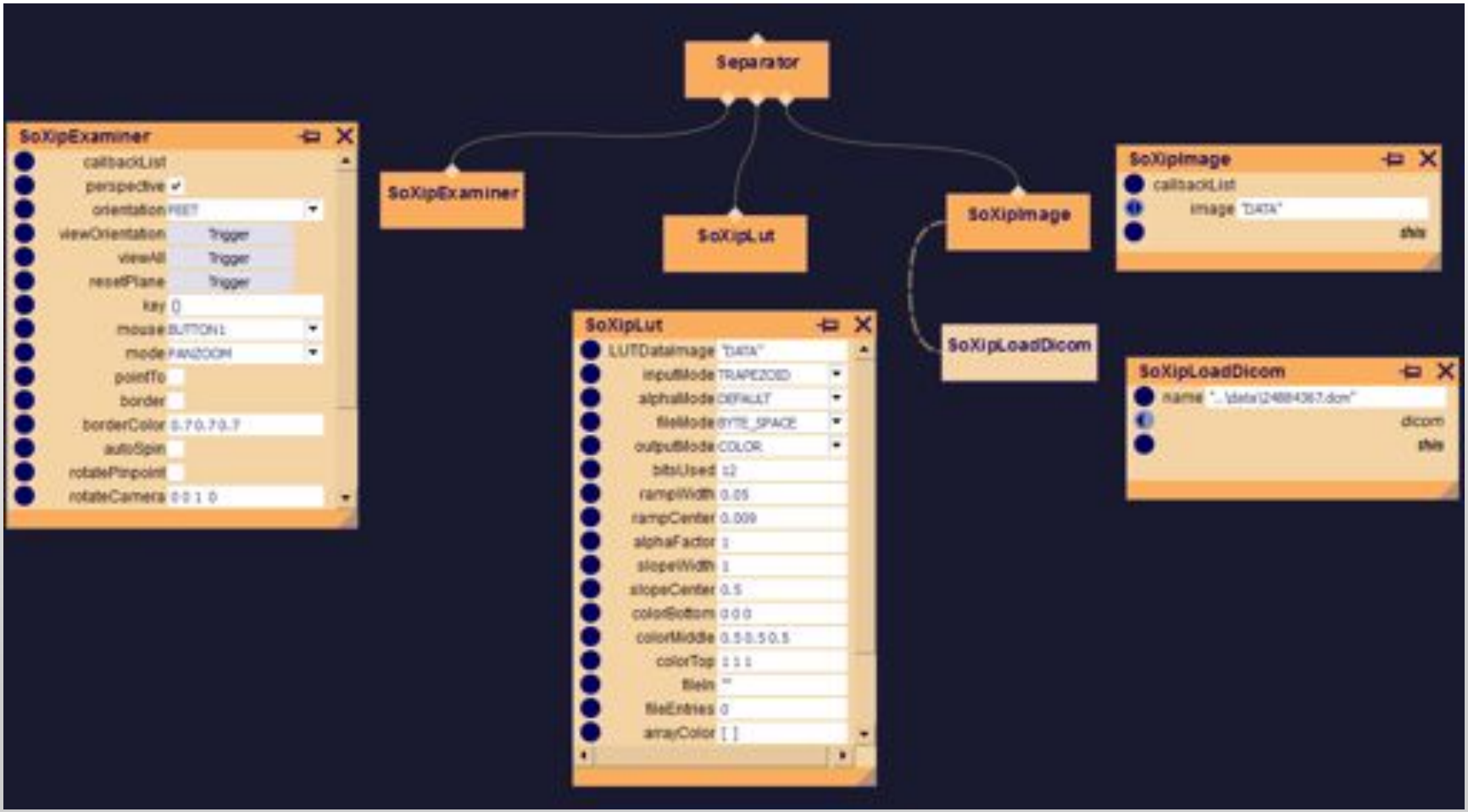
Symbyo TECHNOLOGIES

XIP

Node Name	Description
SoXipLoadDicom	Loading of single DICOM image
SoXipImage	Contains the image
SoXipLut	Computes look up table
SoXipExaminer	It is used to control the orientation of the image. It is also used to panning, rotating and scaling (zoom-in or zoom-out) the image.

Symbyo TECHNOLOGIES

XIP



DEMO

How To Select

- Performance
- Data size / Memory consideration
- Extensibility
- License .
- Portability (Cross platform).
- Ease of use.
- Performance .
- Quality and quantity of the documentation.

QUESTIONS?

Symbyo
TECHNOLOGIES

Keep in Touch with us

At:

/Symbyo

/Symbyo