The DeVIDE Runtime Environment
An Instantly-deployable Python-VTK-ITK lab

Charl Botha, MedVis TU Delft; Dept. Radiology LUMC, Leiden.
15-10-2009
DeVIDE

- *Delft Visualisation and Image processing Development Environment*, since 2002
- **Cross-platform turn-key rapid prototyping** environment for medical visualisation and image processing techniques
- Supports **visual programming**
Demo?

- Loading the superquadric.
- Introspection, CodeRunner with SphereSource.
- DICOMBrowser, DICOMReader, Slicer.
DeVIDE

- **Pervasive interaction** down to code-level at run-time!
- VTK, ITK, numpy, matplotlib, statistics, the kitchen sink, all out of the box.
- **Now also self-contained DeVIDE Runtime Environment!**
- **Off-line mode** for large-scale processing, can be used as black-box by coordination framework, e.g. Nimrod
  - Parameter sweeps
  - Large scale processing (many datasets)
  - Use in production workflow
- **Same software is used for all stages**: algorithm prototyping, large-scale processing, and post-process visual analysis.
- Lovingly dubbed "Not Responding" by students (32 vs 64)
Similar solutions

- AVS, OpenDX
- Amira
- SCIRun
- MeVisLab
- VisTrails

Why DeVIDE?
- Made for medical vis + ip
- Introspection
- Ease of integration
- Prototyping
- Python.
- Hybrid Scheduling
- BSD open source.
- Runtime environment.
- Close to libraries.
What happened since Vis2007

- Open source announcement at the BOF in 2007.
- Hybrid scheduling + OSS paper [1]
- Releases:
  - 8.2: Hybrid Scheduling, first OSS release.
  - 8.5: DICOMBrowser
  - 9.8: DeVIDE Runtime Environment (DRE)

Scheduling

- 2 major approaches: demand vs event
- Memory vs. simplicity
- e.g.: MeVisLab, Amira, SCIRun, VTK/ITK, VisTrails
Event-driven Scheduling

- Central
- Simple
- $K \times$ data
- e.g. $5 \times 500$ MB = 1.5G
Hybrid Scheduling

- Streamable subsets
- \((1 + K \times \frac{1}{N})\) data
- e.g. \(500 + 5 \times 25 = 650\) MB
- Interactive (1)
Hybrid Scheduling

- Adaptive
- Re-analyses topology at each execution
- Simplicity + efficiency!
The DRE

- DeVIDE is now a Python distribution including the following:
  - Python 2.6.2 final-0, numpy 1.3.0, matplotlib 0.98.5.3, wxPython 2.8.10.1, VTK 5.4.2 with special sauce, ITK 3.14.0, gdcmtk 2.0.12, dcmtk
  - Swig 1.3.38, WrapITK 0.3.0 !!
- Single downloadable installer: Linux 32/64, Win 32/64
- DRE Application Modules == DREAMs
  - DeVIDE
  - ipython, python-wx shell
  - Any other Python script relying on any combination of above libs
  - This means: “dre your_python_script.py” just works.
- Full SDK included, just add compiler.*
Future work, Info

• Integrating more out-of-the-box libraries, 700MB installed is not enough 😊
• Slice-by-slice segmentation tool.
• Incremental update tool.
• More momentum.

• Website:
  http://visualization.tudelft.nl/Projects/DeVIDE
• Help:
  http://code.google.com/p/devide/wiki/HelpDRE
• Master-level workshop-format MedVis course:
  http://visualisation.tudelft.nl/Courses/in4307
Detour: TUD MedVis Overview

- Faculty EEMCS, Dept. Mediamatics, Computer Graphics, Visualisation, Medical Visualisation
- 1 associate prof., 1 assistant prof., 6 PhD students, 5 M.Sc. students.
- Medical Visualisation application areas focused on:
  - medical research (neuro, longitudinal, molecular)
  - diagnosis and treatment planning
- Cooperating with academic hospitals in NL (especially LUMC), Philips Medical Systems, Medis, TU/e
- Medical Delta: TUD, LUMC, EMC
MIA lifecycle

Figure 1. Information flow among phases of the MIA lifecycle, from development to clinical deployment, and the systems involved: DeVIDE, Nimrod, AMC-DWMS

Some Use Cases

- In progress: recumbent, recumbent lat, sitting MRI for cord-sac spine analysis; CT processing for instrumented prosthesis analysis
- Medical image analysis problem solving on the grid (Maheshwari 2007)
- MIA development workflow (Olabarriaga 2007)
- Retrobulbar fat analysis (ARVO 2005, ASCI 2005)
- Parts of Visible Orbit reconstruction (ARVO 2005b)
- Pre-op planning (SimVis 2006)
- Chorionic villi visualisation
- Shoulder segmentation