Web Services for Visualization

Dr. Gordon Erlebacher
Florida State University
Contents

- Objectives
- Languages
- Streaming video
- Video creation
- Remote control of visualization package
- Interactive Web Maps
- Conclusions
Issues

- Scientific visualization and information visualization are natural interfaces to complex and large datasets.
- The range and size of display devices is increasing: PDAs, Powerwalls, E-paper, the retina.
- The proliferation of interaction devices make it difficult to reliably and consistently interact with the information displayed.
- The interaction devices are often expensive and non-standard.
• Each user should be able to query the image differently

• Augment image with information from WEB

• Store GUI on PDA

(Picture from www.amiravis.com)
Visualization / Collaboration

- 2D/3D navigation with multiple users
- Synchronization issues
- Interactivity
- Information exchange
- Data analysis / mining
- Feature extraction
Rear projection
16’x8’
Stereo
2 projectors, blended image
Partial Wish List

- User Interaction should be natural, easy to use
- GUIs should combine voice and graphics
- GUIs should minimize unnecessary information
- User interfaces should be configurable by users
- User interface should be portable and transportable
- Users should move freely with respect to the display and themselves
Multiple-PDA Device

- Slice through data
- Query the Internet
- Fuse data
- User-user exchange
Trends and Assumptions
2000-2005

- CPU power of PDA $\rightarrow$ infinity
- Memory of PDA $\rightarrow$ infinity
- Wireless bandwidth $\rightarrow$ infinity
- Wireless bandwidth $\leq 1/10^{th}$ LAN
- Display resolution of PDA $\rightarrow$ infinity
- Size of PDA screen remains limited
Approach

- Conform to standards (Corba, Java, python, XML)
- Allow for extension
- Maximize generality
- Allow for multiple protocols, operating systems, languages

Datasource:
- Supercomputer
- File servers
- Data feeds (i.e., satellites)

- 802.11b standard (2001), 802.11a/g (2002-?)
- Currently, wireless network only serves to get the information from and to the PDA
Proposed Architecture
Heat convection in the Planetary Mantle

Ra = 10^9

Complex Flows
Mining/Extraction/Analysis
Multiple Objectives

- Integrate visualization with Web Services
- Build tools to enhance collaboration through visualization
- Facilitate feature extraction and information sharing between users
- Develop tools to query datasets
- Use of system should be transparent to users (independent of location or hardware)
Approach(es)

- Work with standards
  - Should function on Windows/Linux/Unix

- Develop for ease of maintenance
  - Students only stay for short period

- Insist on web documentation of everything
  - Only partially successful
  - Success is a cascading effect
Languages: C++

- Fast Execution
- Compiled
- Well-supported
- Reasonably portable
- Standard Template Library (not yet fully standardized)
- Operator overloading (very useful)
- Object-oriented
- Strongly typed
- Supported on all platforms
Languages: Python

http://www.python.org

- Fast Prototyping
- Essentially same capabilities as Perl
- Rather well supported
- Nice interface to Java (JPython)
- Weakly typed
- Object-Oriented
- Extremely flexible (sometimes dangerous)
- Extremely easy to use
- Exceptionally clear code (e.g., forces indentation)
- Support on all platforms (including PDA)
Languages: ACE/TAO

http://www.cs.wustl.edu/~schmidt/ACE.html

- ACE: Adaptive Communication Environment
- TAO: The ACE ORB (CORBA Support)
- Totally based on design patterns
- Wraps streaming, sockets, mutexes, etc. in high level classes
- Portable across platforms and operating systems (we ported it to Linux on PDA, support for WinCE)
- Problem: large memory footprint (4-5 Mbytes on PDA)
GRID

User Community

Portal

Services

Hardware

Metadata Database

Information Exchange

Computation

Visualization (wavelets)

Laptops

Desktops

Handhelds

Distributed

Distributed Supercomputers

Fox et al., Grid Services for Earthquake Science, AGES 2001: Special Issue of Concurrency & Practice.
Streaming Video

http://www.csit.fsu.edu/~dongchen

Video Data Source:
- Numerical Results
- Archived Image Data
- Digital Video Camcorder
- ...

OpenGl. Apps

Video Server

Local Area Network

Receiver

Receiver

Receiver

Receiver

Receiver
Video Streaming with wavelets

Visualization Server

Frame

Wavelet transform

Encode

CORBA/SOAP

GUI Ipaq

Visualization Ipaq

Frame

Wavelet transform

Decode

Color animations at 4 frames/sec on Ipaq (320 x 200) and 802.11b wireless network
Streaming Video

**Active Object:**
- Had its own thread(s)
- Does not block calling method

**Passive Object:**
- Uses parent thread
- Blocks calling method
Streaming Video

Encoder
- CImageBuffers*
- ServerDgram
- Grabber
  - ConvertRgbToYUV()
  - waveletTransform()
  - informSibling()

Grabber
- CImageBuffers*
  - grabImage()
  - informSibling()

ServerDgram
- CImageBuffers*
  - informSibling()

CImageBuffers
- Streaming buffers (2)
- YUV/Wavelet buffers (2)
- RGB image buffers (2)
Online Video Creation

Scientists use visualization tools to analyze and navigate their large datasets.

Videos are created for dissemination and archiving:
- Sequence of frames is stored by viz program.
- Frames are transformed into appropriate video format.
- Video format is converted to one or more additional formats (for easy accessibility).
Online Video Creation

- Generation of videos is usually a manual process
- Videos usually created on Windows or expensive Unix systems
- Visualizations usually created on Linux

Conclusion:
- Need technology to create videos directly on Linux systems
If no input files match patterns in Snakename-Width, an output will be created anyway.

- **File pattern**: (optional)
- **Width/Height**: [value] (optional)
- **Video format**: [value] (optional)
- **Codec**: [value] (optional)
- **Input video**: [value] (optional)
- **Output video**: [value] (optional)
- **Input Width**: [value] (optional)
- **Output Width**: [value] (optional)
- **Input Height**: [value] (optional)
- **Output Height**: [value] (optional)
- **Input bitrate**: [value] (optional)
- **Output bitrate**: [value] (optional)
- **Input sample rate**: [value] (optional)
- **Output sample rate**: [value] (optional)
- **Input frame rate**: [value] (optional)
- **Output frame rate**: [value] (optional)
- **Input quality**: [value] (optional)
- **Output quality**: [value] (optional)

**Codec options**:
- **RIPES**: [description]
- **RIPES**: [description]
- **RIPES**: [description]
- **RIPES**: [description]
- **RIPES**: [description]
- **RIPES**: [description]

**Preset parameter combinations**:
- **Default**: [description]
- **Default**: [description]
- **Default**: [description]
Online Video Creation

Created by J. F. Boisdet

http://vector.csit.fsu.edu:8081/~boisdet/temp/one.py
Amira
http://www.amiravis.com
Amira

- Flowcharts are created interactively by the user.
- Each component has an associated user interface.
- Software capitalizes on graphic hardware (SGI, Onyx, Nvidia, ATI) to achieve good performance.
- Flowcharts, called networks, can be saved for later use.
- Developer version allows users to create their own modules for specialized visualization.
Amira

- Amira is a commercial package
- I don’t necessarily recommend this package
- However,
  - It has nice features, perhaps useful to the visualization of static and time-dependent fluid structures
Amira

- Read in 3D file
- Generate several planar cross-sections
- Generate an iso-surface
- Generate a volumetric plot
- Combine techniques
- Demonstrate data querying (line cut, pointwise, etc.)
Amira Features

- Very Interactive
- Manipulators
  - Interact with the data
- Extensible
  - Users can write own extension modules
  - API is very sophisticated
- Highly advanced algorithms to do
  - Isosurface, volume rendering, vector visualization
  - Combinations of the above
Heat Convection between Two Plates (Amira)

257\(^3\) dataset

64\(^3\) subsampling

Data, courtesy David Yeun

Heat flow between two plates at constant temperature
Remote Control Amira
(Yunsong Wang)

- http://vector.csit.fsu.edu:8081/~yunsong/cgi-binbac/remote_amira.py
- Creation or loading of Amira scripts
- Automatic initiation of Amira
- Retrieve bitmap from server
- Working on retrieving x,y,z coordinates
Amira Tcl Scripts

Script Creation

Amira Server

Amira Output

Amira Display
Remote Control Amira
(Yunsong Wang)

http://vector.csit.fsu.edu:8081/~yunsong/cgi-binbac/remote_amira.py
Interactive Web Maps
http://www.csit.fsu.edu/~garbowza/WDI/

- Built by Zachary Garbow
  - (Minnesota Supercomputer Institute)
- Store large datasets on a server
- Clients operate on the dataset
  - Zooming, histograms, mean/avg/stddev
- C++ on the Server
- Java Applet on the Client
- Challenges: balance between computation on client and server, and networking considerations
Interactive Web Maps

[Diagram description]

- C++
- DATA
- CORBA

Java/Java3D

WEB Gateway

Client
- Access point

Client
- Access point

Client
- Access point

Wireless

Wireless

Wireless

Wireless

Handheld
- PCs
- Laptop
- 54 Mbps

Workstation/Supercomputer
- Classroom, electronic publication
- Electronic Survey, E-business, E-laboratory
Temperature field: 2D grid: 3400x500

- Ra = 3 \times 10^7
- Ra = 3 \times 10^8
- Ra = 10^9
- Ra = 10^{10}
Two-way flow of information!!
Interactive Web Maps
http://www.csit.fsu.edu/~garbowza/WDI/

Built by Zachary Garbow (Minnesota Supercomputer Institute, works with D. Yuen)
Visualization Ubiquity

- Collaboration through visualization
- Office walls become visualization displays (E-Ink: thin, pliable medium capable of electronic encoding)
- Exchange of visual data becomes as ubiquitous as exchange of text documents in 2001
An Ideal Visualization System

- Reusable modules
- Flexible
- Ease of use
- Low memory
- Extensible
- Scriptable
- Good debugging
- Portable
- Intelligent defaults
- Changeable defaults
- Interpreted and compiled modes
- Novice and expert modes
- Mathematical text editor
Future trends in Visualization

- Use of Object-Oriented design patterns for reusability
- Plugin technology on distributed systems
- Extensive use of visualization across the network
- Increased intelligence in software
- Insertion of new algorithms without recompilation
Wireless Speeds
Present and Near Future

- **Present: 802.11b**
  - Range: 150 m, 10 Mbit/sec

- **1st quarter 2002: 802.11a**
  - Range: 150 m, 54 Mbit/sec
  - Not compatible with 802.11b

- **3rd quarter 2002: 802.11g**
  - Range: N/A, 54 Mbit/sec
  - Compatible with 802.11b!!
OQO: true mobile computing?
Fall 2002

- Up to 1 GHz
- Crusoe chip
- 256 Mbytes memory
- 10 Gbyte hard disk

- Touchscreen
- USB/Firewire
- Windows XP
- 4” screen
Future Work

- Integrate these packages into consistent framework
  - They are currently developed independently
- Increase support for XML
- (hopefully) integrate some of these packages into frameworks developed in Pervasive Group (do not reinvent the wheel)
- Investigate interoperability Java/Python
- Integrate Streaming with Zope or Java Applets
Conclusions

- We have developed several web-based services related to visualization.
- Objective is to access and manipulate data from remote sites.
- Hope is to allow multiple users to manipulate the same data concurrently.
- Difficulty: integrating multiple languages together without sacrificing efficiency.