VisTrails: Enabling Interactive, Multiple-View Visualizations

Louis Bavoil  
Steven P. Callahan  
Juliana Freire  
Claudio T. Silva  
Carlos Scheidegger  
Huy T. Vo  

University of Utah

Patricia Crossno  

Sandia
Motivation

➤ Develop a system to enable the creation and maintenance of a large number of complex visualizations
  • Ability to create complex pipelines
  • Ability to track changes to visualizations
  • Need to enable comparisons:
    • Across multiple versions of the same visualization
    • Same visualization on different sets of data
  • Efficient, easy to use, portable, and simple to implement
Motivating Example: CORIE

➤ CORIE is an Environmental Observation and Forecasting Systems (EOFS) that combines real-time sensor measurements with advanced computer models to increase reliability of complex, dynamic environmental systems

➤ Thousands of visualizations daily
  • No management infrastructure

➤ http://www.ccalmr.ogi.edu/CORIE/
VisTrails

A new system that enables interactive, multiple-view visualizations

• Simplify the creation and maintenance of a large number of visualizations
• Detailed provenance of visualization results
• Separation between pipeline specification and execution instances
• Optimization of execution through caching
Many previous dataflow visualization systems
  • Paraview, Opendx, SCIRun, IRIS Explorer, many others
Kreuseler et al.: tree history for exploratory data mining
  • Readily applicable to exploratory visualization
  • Tree provides provenance of the process
Brodlie et al.: extension of IRIS Explorer over Grid resources
Jankun-Kelly and Ma use a spreadsheet-like interface to aid exploration
Jankun-Kelly et al provide a model for the visualization exploration process
See paper for details
“everything” in XML
- Open standards allow easy interoperability
- Vistrails can be queried

The Cache Manager mediates calls to the player

Visualization API is unaware of the infrastructure
- Extensible to other API's, and possibly to multi-API systems
➤ “everything” in XML
  • Open standards allow easy interoperability
  • Vistrails can be queried
➤ The Cache Manager mediates calls to the player
➤ Visualization API is unaware of the infrastructure
  • Extensible to other API's, and possibly to multi-API systems
“everything” in XML
- Open standards allow easy interoperability
- Vistrails can be queried

The Cache Manager mediates calls to the player

Visualization API is unaware of the infrastructure
- Extensible to other API's, and possibly to multi-API systems
➤ “everything” in XML
  • Open standards allow easy interoperability
  • Vistrails can be queried
➤ The Cache Manager mediates calls to the player
➤ Visualization API is unaware of the infrastructure
  • Extensible to other API's, and possibly to multi-API systems
Some stats:
- 15,000 lines of code
- 80 files
- C++, bash, CMake, Qt, OpenGL, VTK, xerces-c, graphviz, swig
- Windows, Linux, Mac OS X

Stay tuned for updates: http://www.sci.utah.edu/~vgc
The Vistrail model

- Vistrail: sequence of operations used to generate a visualization
- Parameter settings are distinguished from the dataflow
- Modules represent filters in the dataflow network, and connections determine the dependencies
➤ VisTrails stores visualizations in a Vistrail Collection
  • version tree given from metadata
➤ Exporting a visualization result ensures reproducibility
Vistrail Builder

- User adds new modules and connections by dragging and dropping appropriate classes
- The builder invokes the spreadsheet directly
Vistrail Builder

- User adds new modules and connections by dragging and dropping appropriate classes
- The builder invokes the spreadsheet directly
Vistrail Builder

- User adds new modules and connections by dragging and dropping appropriate classes.
- The builder invokes the spreadsheet directly.

![Vistrail Builder Interface]

- VTK Classes:
  - `vtkObjectBase`
  - `vtkObject`
  - `vtkAlgorithm`
    - `vtkAbstractMapper`
    - `vtkAbstractMapper3D`
    - `vtkMapper`
      - `vtkDataSetMapper`
    - `vtkDataObjectAlgorithm`
    - `vtkDataSetToObjectDataObjectFilter`
    - `vtkDataReader`
    - `vtkDataSetReader`
    - `vtkDataSetAlgorithm`
    - `vtkDataToObjectDataSetFilter`
    - `vtkGenericDataSetAlgorithm`
    - `vtkHierarchicalDataSetAlgorithm`
The Visualization Spreadsheet

- User can compare a large number of visualizations in the spreadsheet
- Views can be linked
The Cache Manager determines pipeline sharing

- Each module is broken into a series of subnetworks
- Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters
- Results are linked to the ID, and only computed if missing in the cache
➤ The Cache Manager determines pipeline sharing
➤ Each module is broken into a series of subnetworks
➤ Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters
➤ Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing

Each module is broken into a series of subnetworks

Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters

Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing

Each module is broken into a series of subnetworks

Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters

Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing

Each module is broken into a series of subnetworks

Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters

Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing
Each module is broken into a series of subnetworks
Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters
Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing
Each module is broken into a series of subnetworks
Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters
Results are linked to the ID, and only computed if missing in the cache
The Cache Manager determines pipeline sharing

- Each module is broken into a series of subnetworks
- Each subnetwork receives a unique ID, comprising its modules, connectivity and parameters
- Results are linked to the ID, and only computed if missing in the cache
Discussion and conclusions

- VisTrails is a system that allows interactive multiple-view visualizations
- Leverages formal specification of pipelines to increase efficiency
- Allows fast exploration of parameter space with the Visualization Spreadsheet
- Provides detailed provenance of visualization results
VisTrails: Demo

(Check out http://www.sci.utah.edu/~vgc for updates and code)

We’ll be at the VTK BOF tonight
Future Work

➤ Changeset orientation really defines an algebra of pipelines
  • Checking commutativity: move actions around

➤ Parallelism
  • Execution in a grid environment

➤ Graph layout of time-varying graphs

➤ Deployment
  • CORIE Vis’03 paper
Acknowledgments

- Antonio Baptista
- NSF grants IIS-0513692, CCF-0401498, EIA-0323604, CNS-0541560, and OISE-0405402
- DOE VIEWS and MICS
- Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94AL85000
- Emanuele Santos, John Schreiner, Wayne Tyler
- AT&T, Kitware, Trolltech, SWIG project
- Bruno Notrosso, Gordon Kindlmann