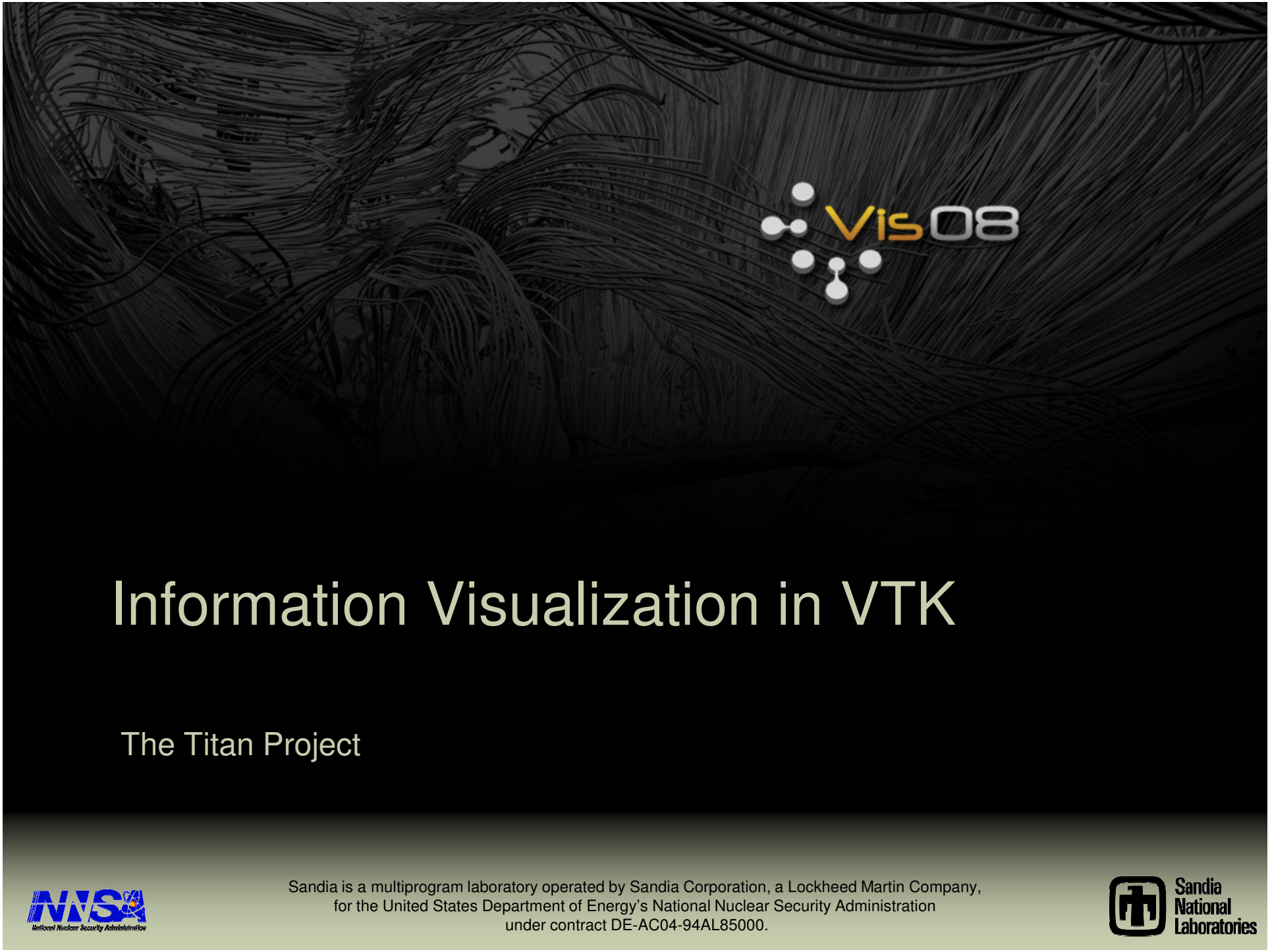


Information Visualization in VTK

The Titan Project

Sandia is a multiprogram 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.

Sandia National Laboratories

Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

Introduction / Motivation

What is Titan? : Led by Sandia National Laboratories, in collaboration with Kitware and Indiana University, the Titan Informatics Project is a substantial expansion of VTK to support informatics and analysis.

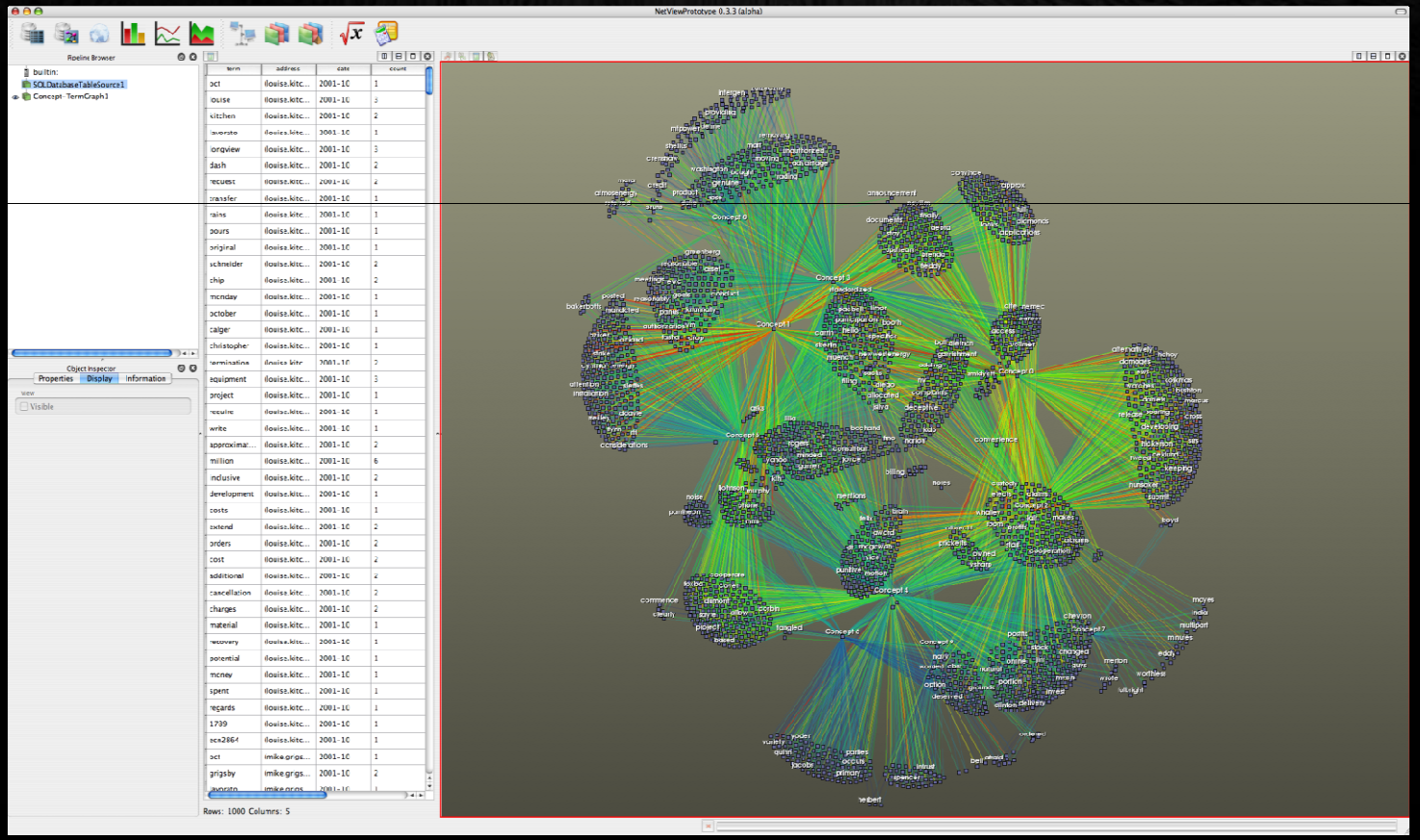
Why Titan? :

- Focused on Algorithms (Graph, Statistics, Algebraic Methods)
- Open, Flexible and Extensible
- Based on Scalable Architecture *(some work done, more coming)*

How do I use it? : In the same way that scientific visualization applications are built with VTK, you can now build information visualization and analysis applications with Titan.

Introduction / Motivation

How do I use it? : Alternatively, you can use the general-purpose OverView client to deploy Titan components …



Project Scope

Vis 08

Data Structures
- Table
- Tree
- DAG
- Directed Graph
- Undirected Graph
- Sparse N-way Array
- Dense N-way Array

Database Drivers
- MySQL
- Postgres
- Oracle
- SQLite
- ODBC
- Netezza

Readers
- Dimacs
- DOT
- GXL
- Chaco
- XML
- Tulip
- DelimitedText
- FixedWidth
- ISI, RIS

Multidimensional Analysis
- TPP / PARAFAC

MTGL Algorithms
- Community Finder
- ST Search
- CSG Search
- Temporal Search

BGL Graph Algorithms
- Breadth First Search
- Connected Components
- Biconnected Components
- Brandes Centrality

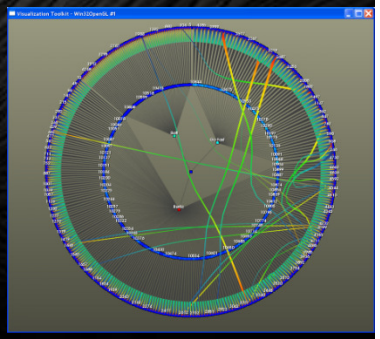
PBGL Integration

MATLAB Integration

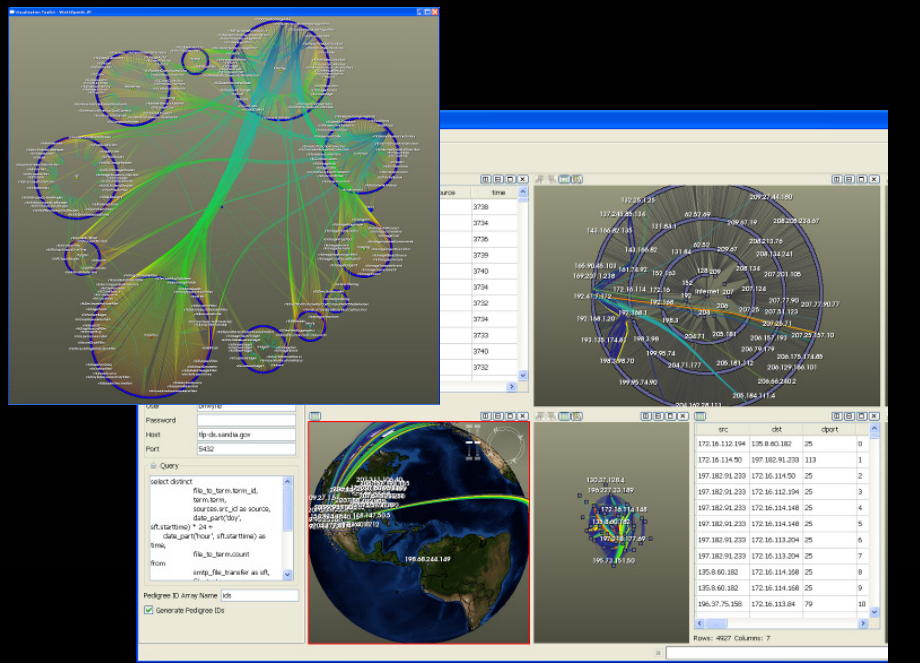
Statistics Algorithms
- Descriptive
- Order
- Correlative
- Contingency

Linear-Time Graph Layouts
- GSpace
- Hierarchical
- Clustered
- Three tree-based variants

Multiple View Types
- Render (3D)
- Graph
- Hierarchical Graph
- Tree
- Treemap
- Georeferenced

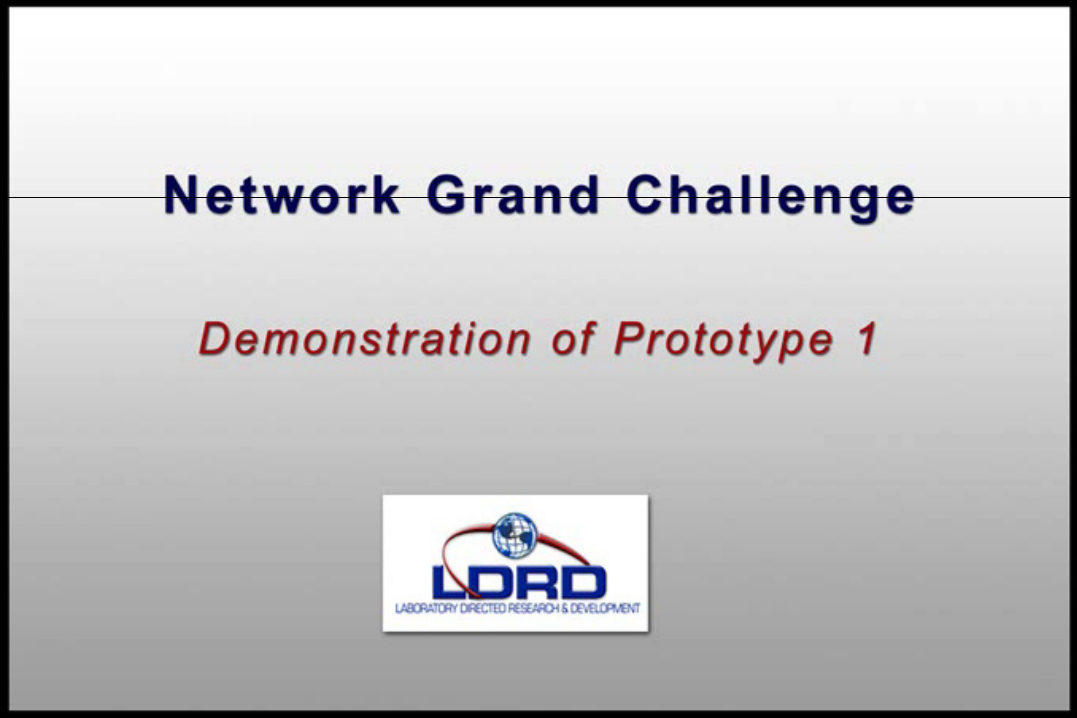


Multiple Platforms / Languages
- Windows, Linux, OSX, HPC
- Write components in C++
- Use with C++, Python, TCL, Java
- Use as OverView “plugins”



Prototype Application

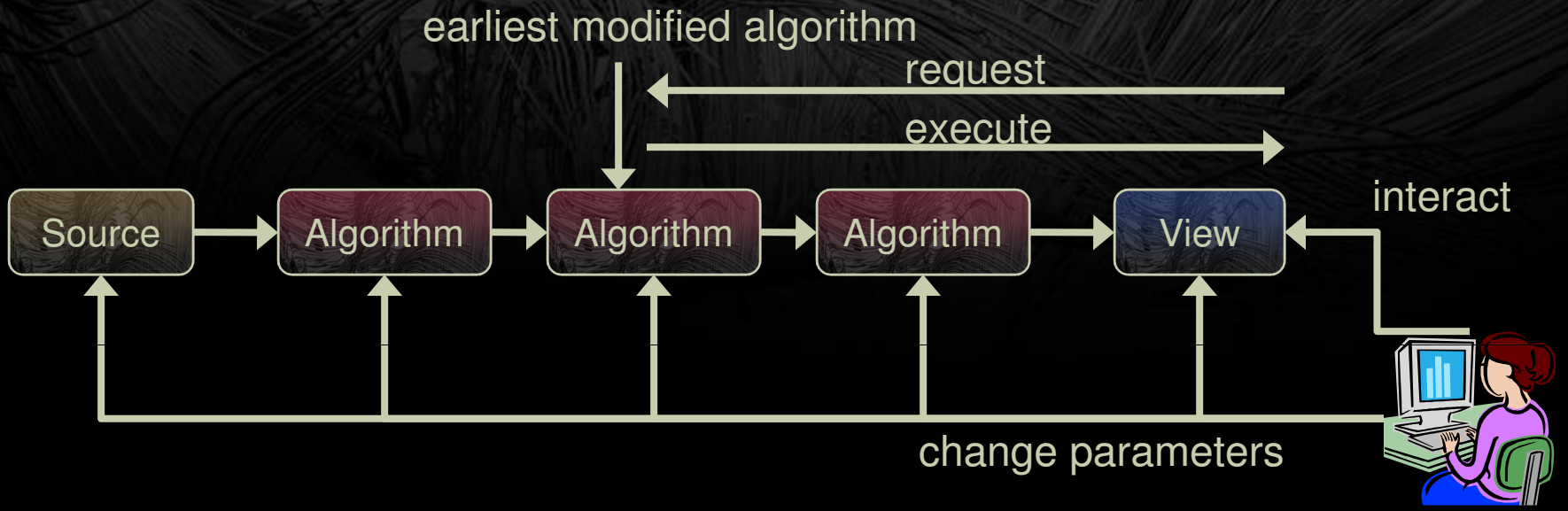
Network Analysis and Cyber Defense: Using network packet captures to detect and track exfiltration events across political boundaries.



Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

VTK Pipeline (Sidebar)

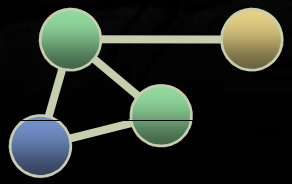


- Demand-driven
- Extensible, component design
- Shallow copy of data

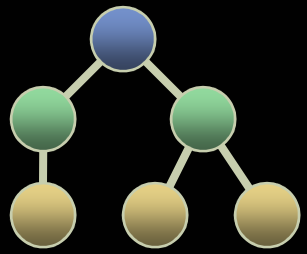
Data Structures

vtkTable

vtkGraph (network)

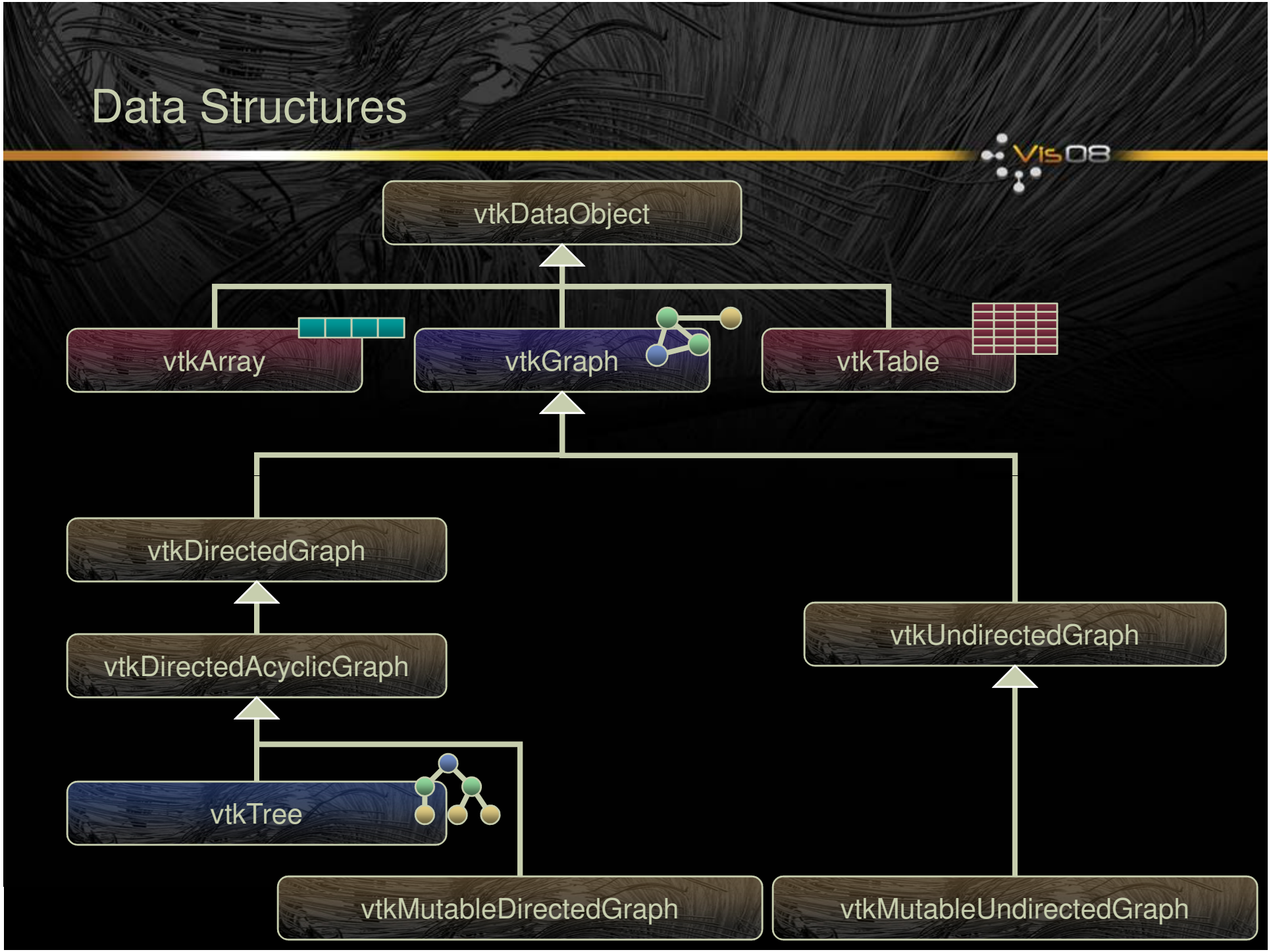


vtkTree (hierarchy)

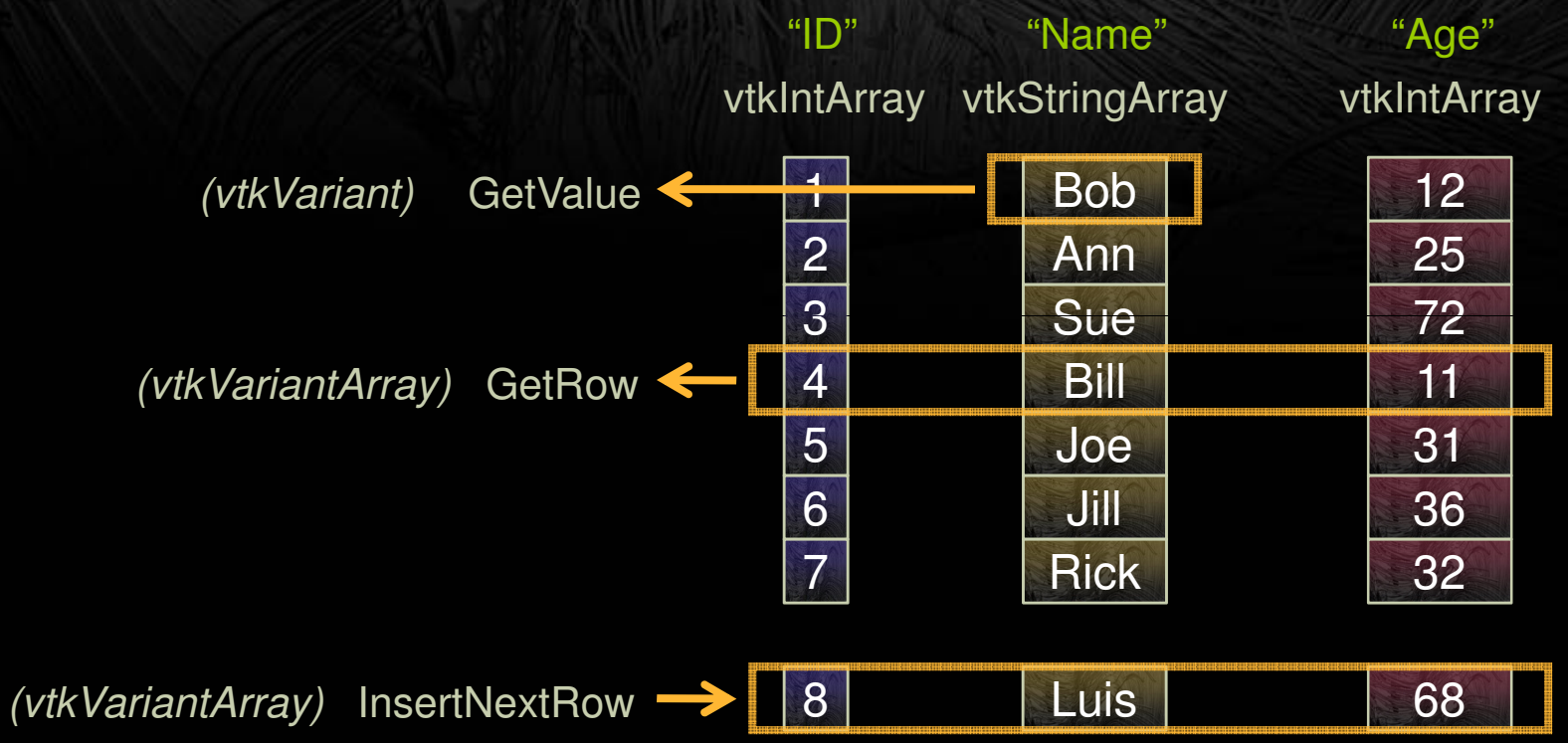


vtkArray

Data Structures



vtkTable

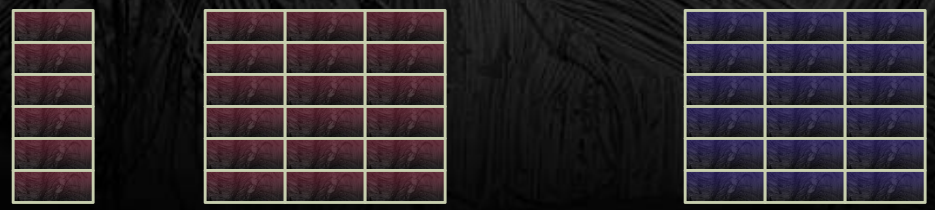

"ID"
vtkIntArray
"Name"
vtkStringArray
"Age"
vtkIntArray
(vtkVariant) GetValue
(vtkVariantArray) GetRow
(vtkVariantArray) InsertNextRow
1
2
3
4
5
6
7
8
Bob
Ann
Sue
Bill
Joe
Jill
Rick
Luis
12
25
72
11
31
36
32
68

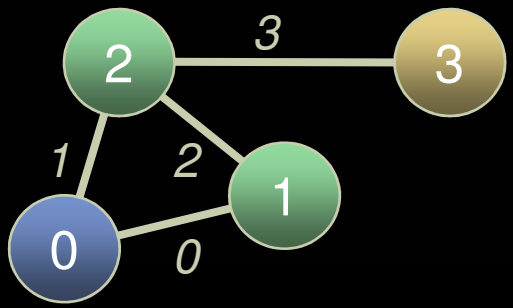
Creating a vtkTable


```
vtkTable* t = vtkTable::New();
vtkIntArray* col1 = vtkIntArray::New();
col1->SetName("ID");
col1->InsertNextValue(0);
col1->InsertNextValue(1);
t->AddColumn(col1);

vtkStringArray* col2 = vtkStringArray::New();
col2->SetName("Name");
col2->InsertNextValue("a");
col2->InsertNextValue("b");
t->AddColumn(col2);
```

ID	Name
0	a
1	b

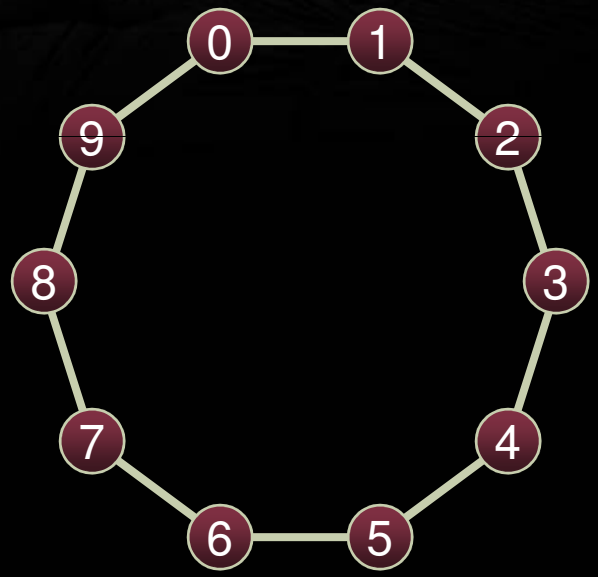
vtkGraph and Subclasses





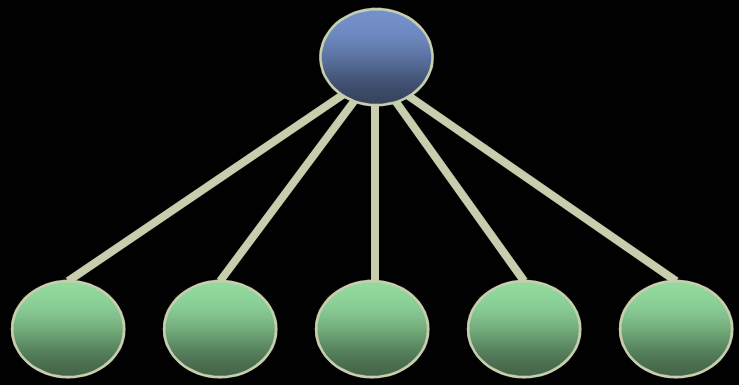
Creating a Graph

```
vtkMutableDirectedGraph* g = vtkMutableDirectedGraph::New();

vtkIntArray* vertId = vtkIntArray::New();
vertId->SetName("id");
g->GetVertexData()->AddArray(vertId);
for (vtkIdType v = 0; v < 10; ++v)
  {
  g->AddVertex();
  vertId->InsertNextValue(v);
  }

for (vtkIdType e = 0; e < 10; ++e)
  {
  g->AddEdge(e, (e+1)%10);
  }
```



Creating a Tree

```
vtkMutableDirectedGraph* g = vtkMutableDirectedGraph::New();

vtkIdType root = g->AddVertex();
for (vtkIdType v = 0; v < 5; ++v)
  {
  g->AddChild(root);
  }

vtkTree* t = vtkTree::New();
t->ShallowCopy(g);
g->Delete();
```

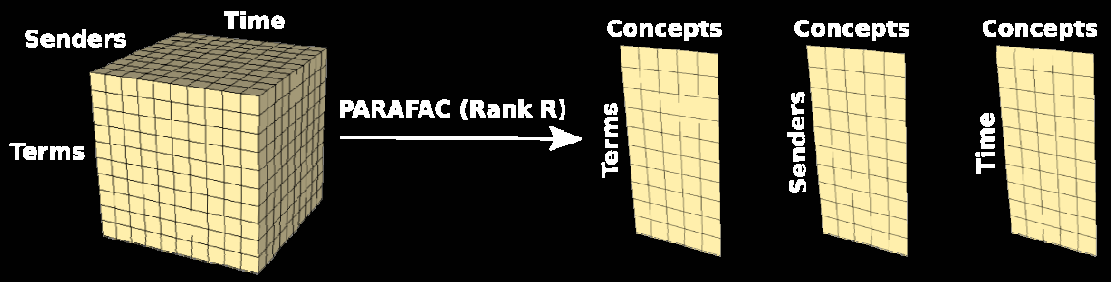


vtkArray and Subclasses

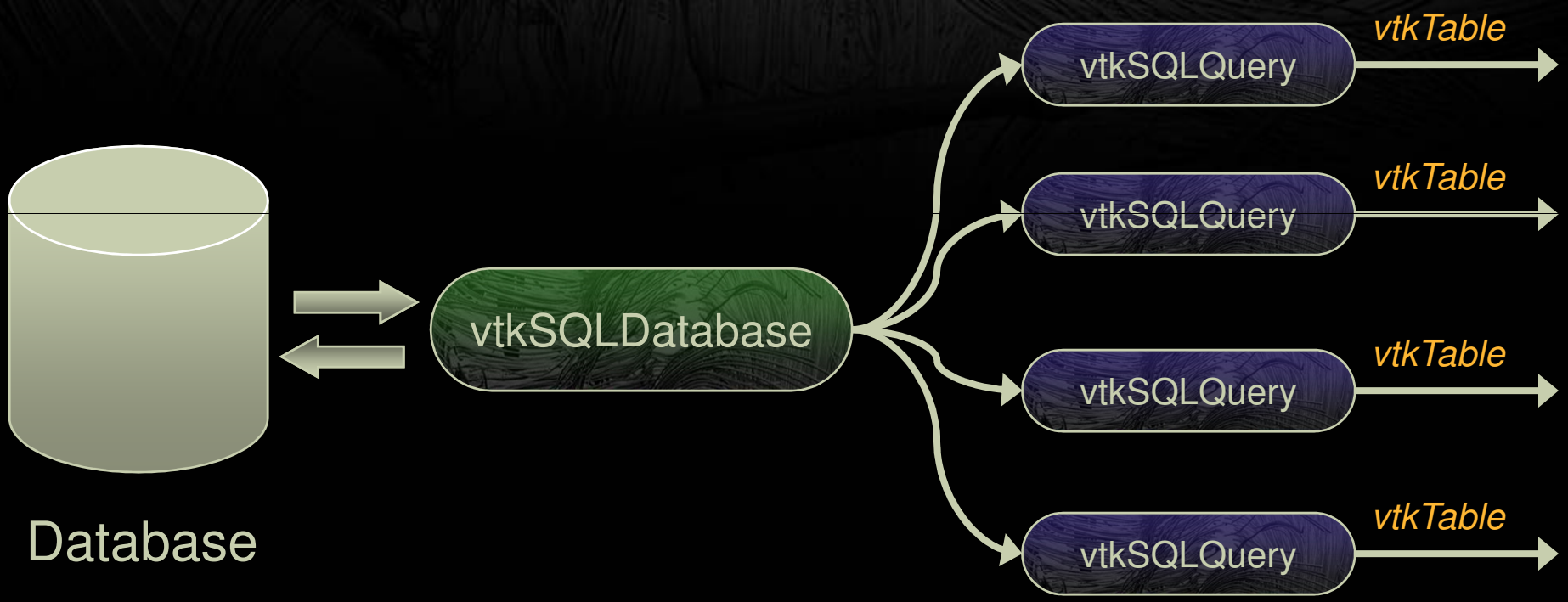
Provides a flexible hierarchy of arbitrary-dimension arrays, including sparse and dense storage, efficient access, and support for custom array types.

There is SO much interesting stuff that we made a separate section. ☺

Using vtkArray with tensor decomposition methods such as PARAFAC:



Database Access In VTK



Creating a Database Connection


```
#include <vtkSQLDatabase.h>
vtkSQLDatabase *db = vtkSQLDatabase::CreateFromURL(
    “mysql://username@dbserver.domain.com/databasename”);
bool openStatus = db->Open(“mypassword”);
```

OR

```
#include <vtkMySQLDatabase.h>
vtkMySQLDatabase *db = vtkMySQLDatabase::New();
    db->SetUserName(“username”);
    db->SetHostName(“dbserver.domain.com”);
    db->SetDataBaseName(“databasename”);
    db->Open(“password”);
```

Querying a Database


```
vtkSQLQuery *query = db->GetQueryInstance();
query->SetQuery("SELECT name FROM employees WHERE salary > 100000");
Bool status = query->Execute();

while (query->NextRow())
{
    cout << query->DataValue(0).ToString() << " is making too much money, hire a
    new PhD."
}

query->Delete();
```

Reading Results the Easy Way

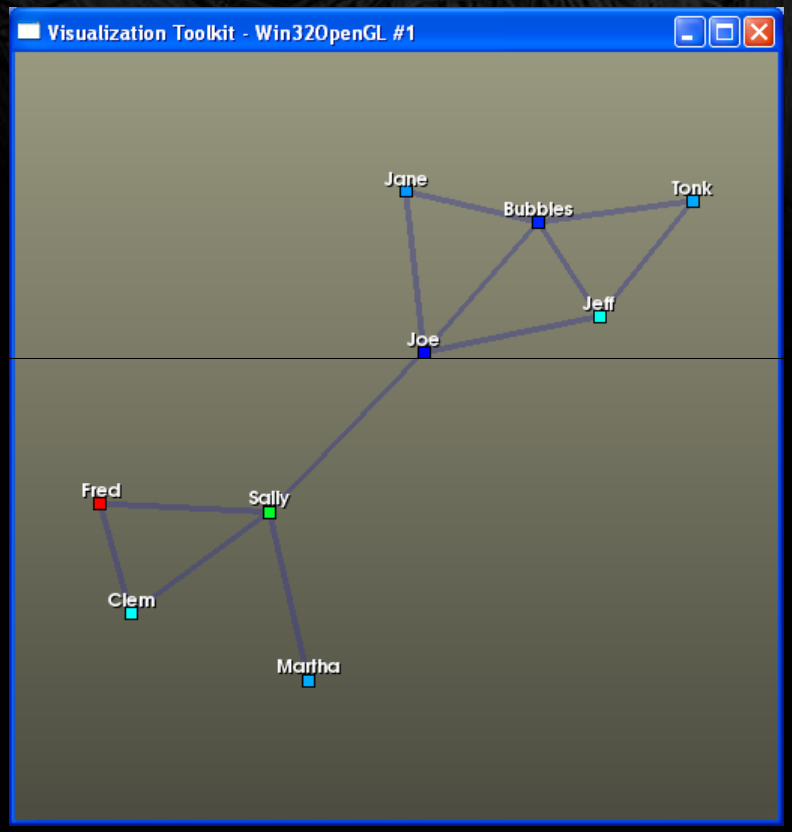

```
vtkRowQueryToTable *tableReader = vtkRowQueryToTable::New();
vtkSQLQuery *query = db->GetQueryInstance();

query->SetQuery("SELECT name, salary, age FROM employees");
tableReader->SetQuery(query);

tableReader->Update(); // will execute query and read the results
                       // into a vtkTable

tableReader->Delete();
query->Delete();
```

Available Database Drivers

- SQLite
  - Public domain - ships with VTK
- PostgreSQL
  - Depends on libpq (part of the Postgres distribution)
- MySQL
  - Depends on libmysqlclient (part of the MySQL distribution)
- ODBC
  - Depends on having ODBC libraries installed
    - Unix (incl. Mac): iODBC, unixodbc
    - Windows: MS ODBC
    - Also requires vendor-specific driver for your particular database

- Add your own! It's simple.
  - Subclass vtkSQLDatabase, vtkSQLQuery and implement abstract methods
  - Add optional support to CreateFromURL()

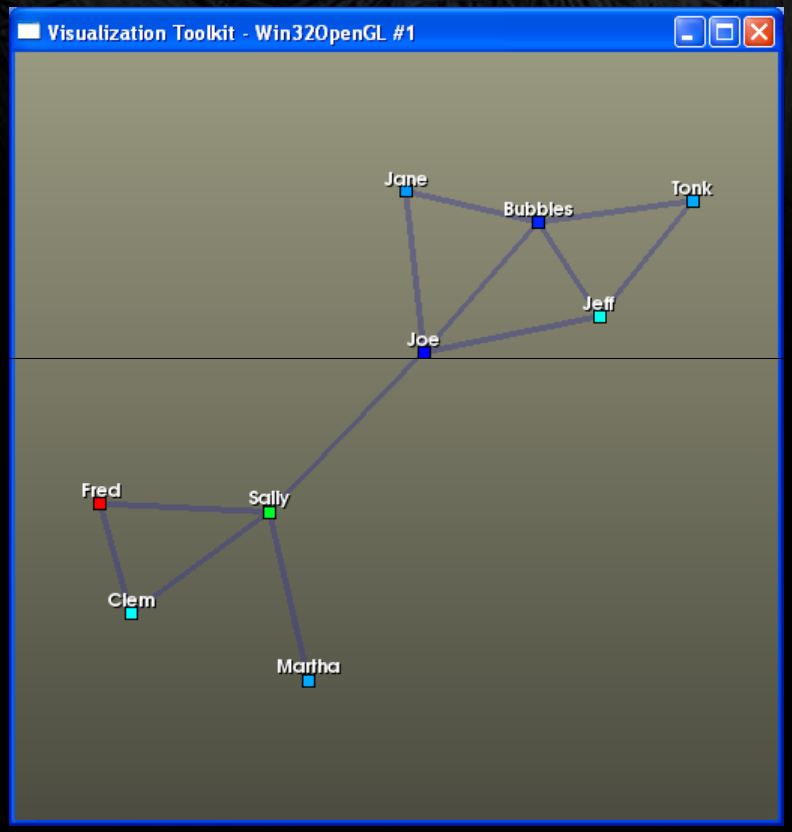
Python Database Example



Using vtkDatabase and vtkRowQueryToTable to hit a database, pull data, and create graphs.

VTK/Examples/Infovis/Python/database.py

Graph Layout Strategies Example



Adding new graph layouts to Titan is a snap!

VTK/Examples/Infovis/Python/database.py

Table and Tree Readers

- vtkDelimitedTextReader
  - Creates a vtkTable from delimited text files, including CSV.
- vtkISIReader
  - Creates a vtkTable from files in the ISI bibliographic citation format.
  - Reference: http://isibasic.com/help/helpprn.html#dialog_export_format
- vtkRISReader
  - Creates a vtkTable from files in the RIS bibliographic citation format.
  - Reference: http://en.wikipedia.org/wiki/RIS_(file_format)
- vtkFixedWidthTextReader
  - Creates a vtkTable from text files with fixed-width fields.
- vtkXMLTreeReader
  - Creates a vtkTree using the structure of any XML file.
  - XML elements, text, and attributes are mapped to vertex attributes in the output graph.

Graph Readers and Sources

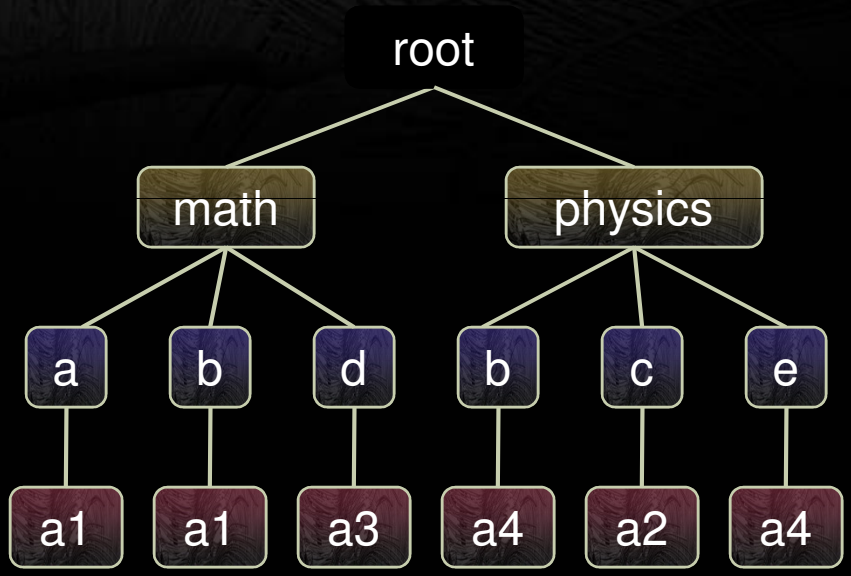
- vtkRandomGraphSource
  - Creates a graph containing a random collection of vertices and edges.
- vtkSQLGraphReader
  - Creates a vtkGraph from a pair of SQL vertex and edge queries.
- vtkDIMACSGraphReader
  - Creates a vtkGraph from files in DIMACS format.
  - Reference: http://www.dis.uniroma1.it/~challenge9/format.shtml
- vtkChacoGraphReader
  - Creates a vtkGraph from files in Chaco format.
  - Reference: http://www.sandia.gov/~bahendr/chaco.html
- vtkTulipReader
  - Creates a vtkGraph from files in Tulip format.
  - Reference: http://tulip.labri.fr/tlpformat.php
- vtkGeoRandomGraphSource
  - Creates a graph containing a random collection of geo-located vertices and edges.

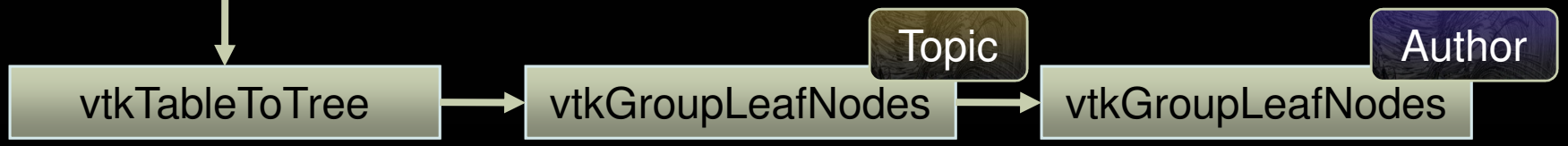
Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

vtkTableToTree – Part I

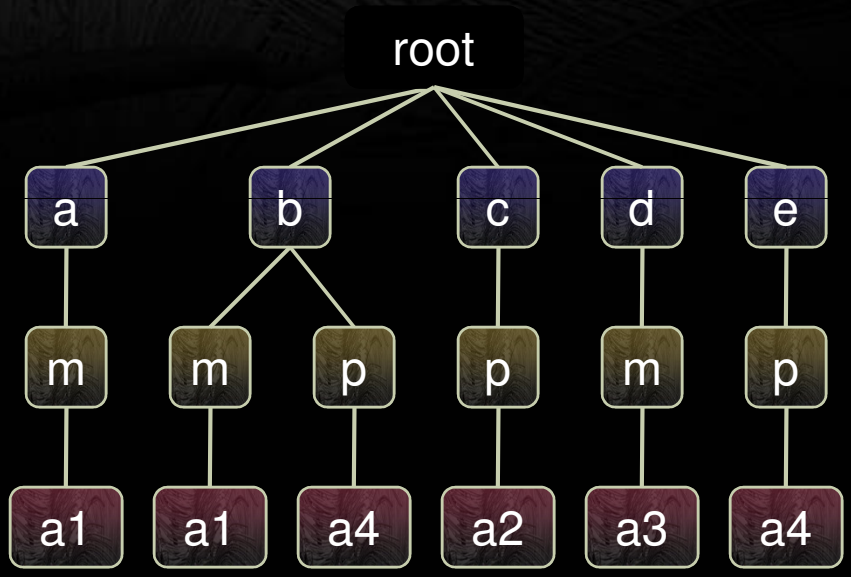
Author	Article	Topic
a	a1	math
b	a1	math
c	a2	physics
d	a3	math
e	a4	physics
b	a4	physics





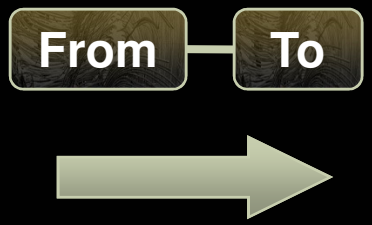
vtkTableToTree – Part II

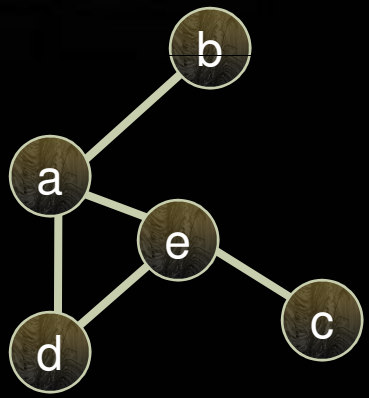
Author	Article	Topic
a	a1	math
b	a1	math
c	a2	physics
d	a3	math
e	a4	physics
b	a4	physics



vtkTableToGraph – Part I

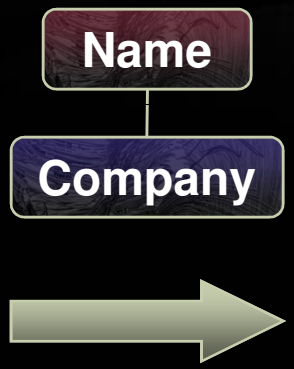
From	To
a	d
a	e
b	a
e	c
e	d

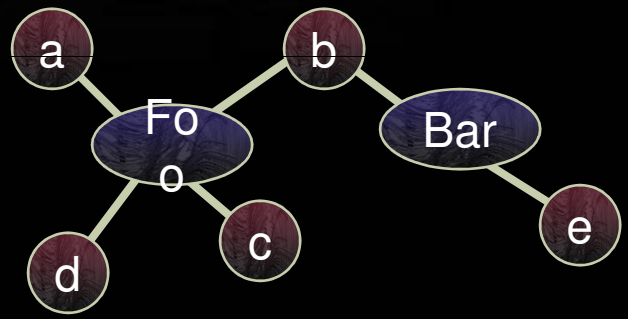




vtkTableToGraph – Part II

Name	Company
a	Foo
b	Bar
c	Foo
d	Bar
e	Bar
b	Foo





vtkTableToGraph – Part III

Author	Article	Topic
a	a1	math
b	a1	math
c	a2	physics
d	a3	math
e	a4	physics
b	a4	physics

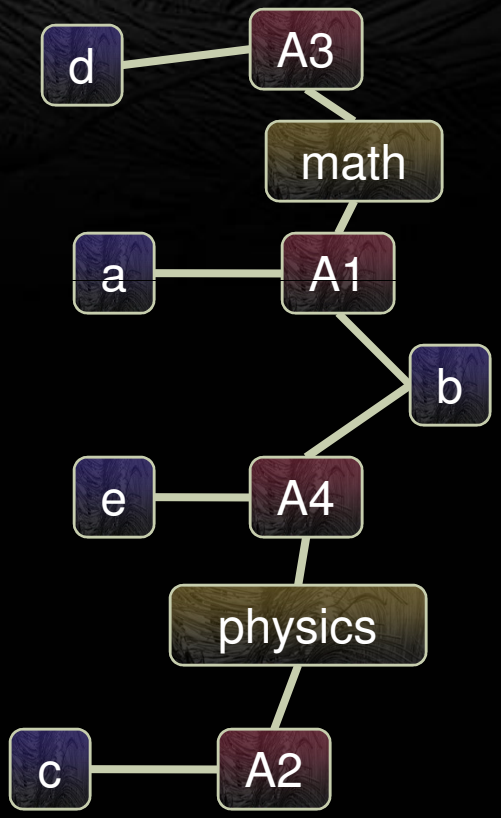
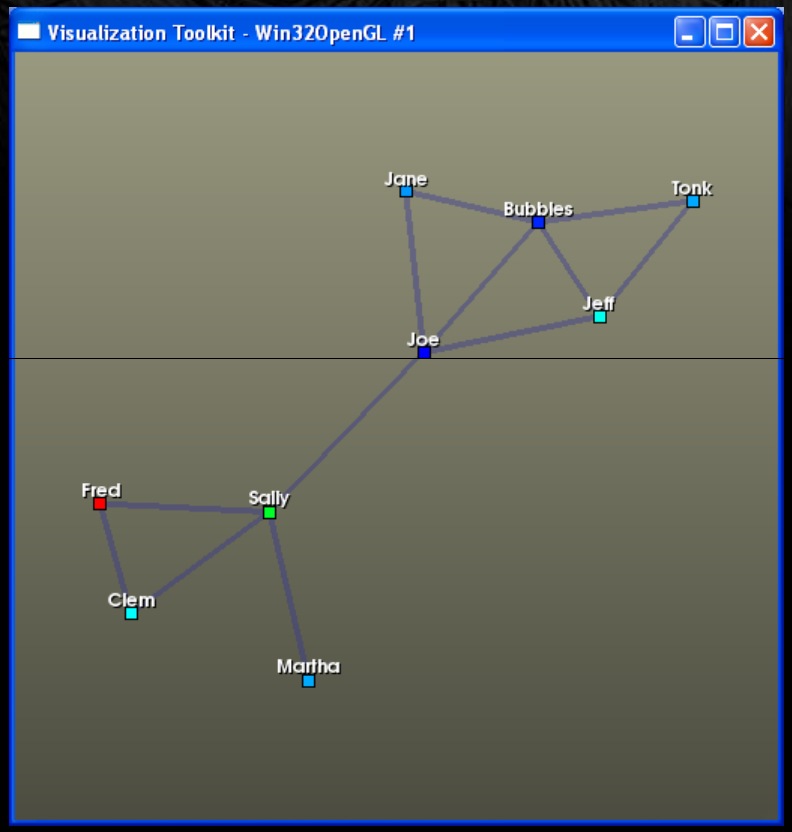


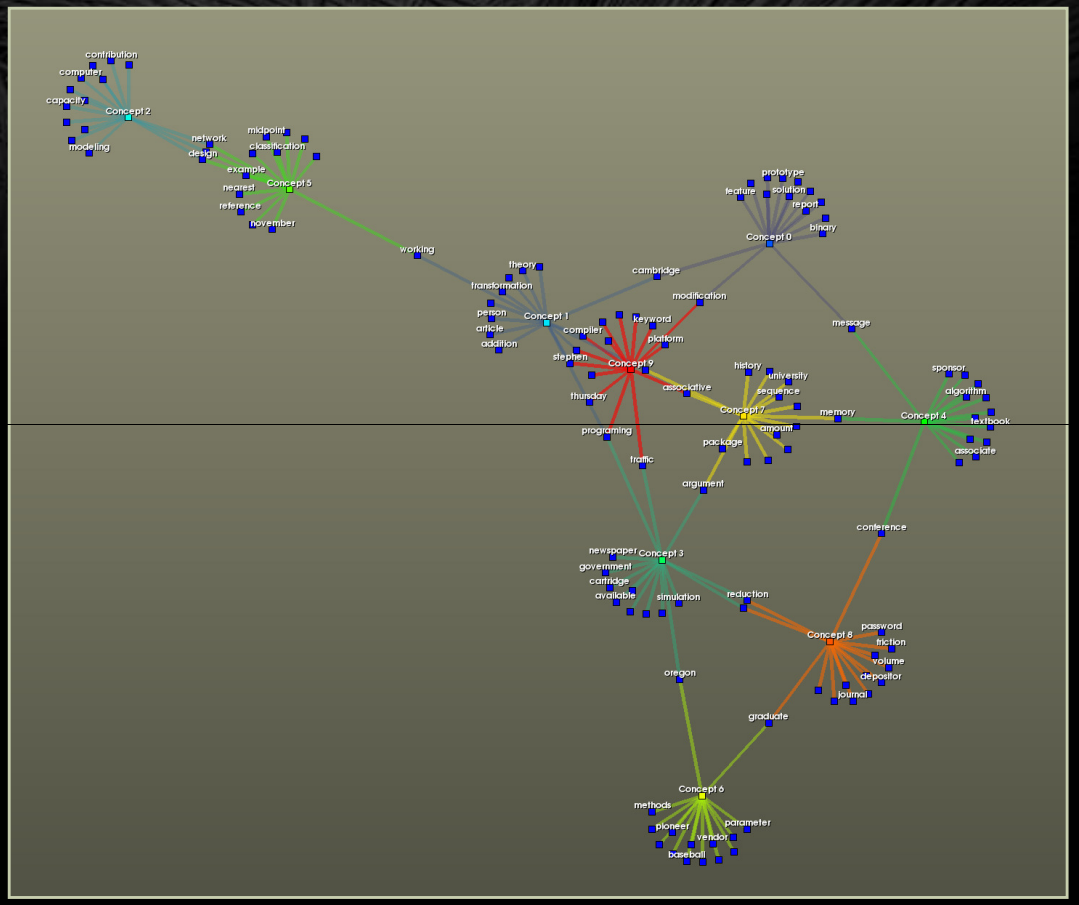
Table to Graph Example



Example that demonstrates the use of vtkTableToGraph filter.

VTK/Examples/Infovis/Python/database.py

Graph/Tree Layout Strategies



vtkDatabase

vtkQuery

vtkTable

vtkTableToGraph

vtkGraph

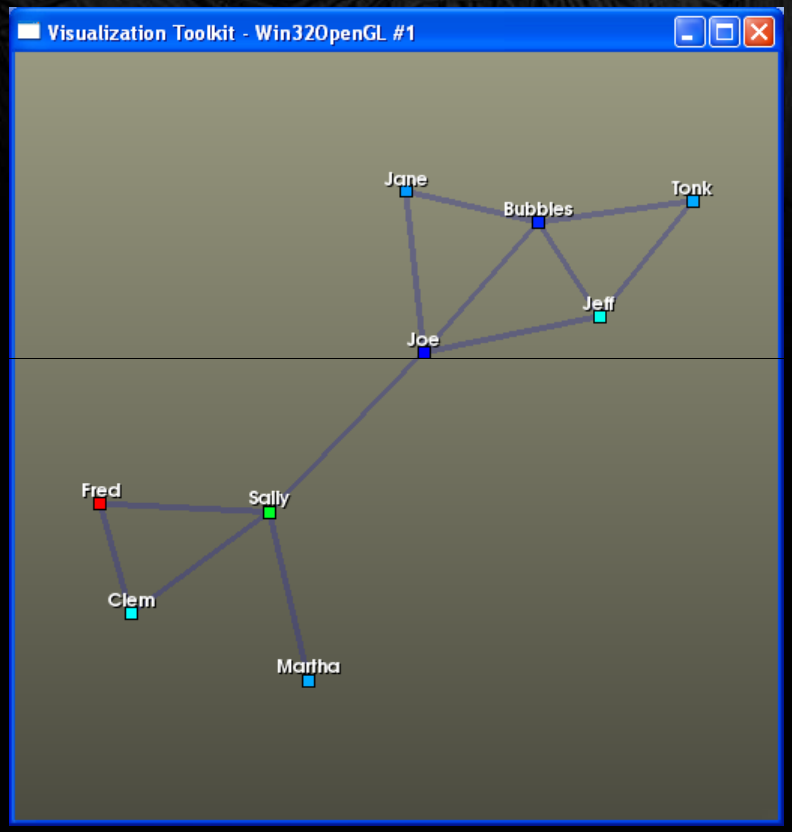
vtkGraphLayout

Layout Strategy

vtkGraph + Coordinates

Rest of Pipeline

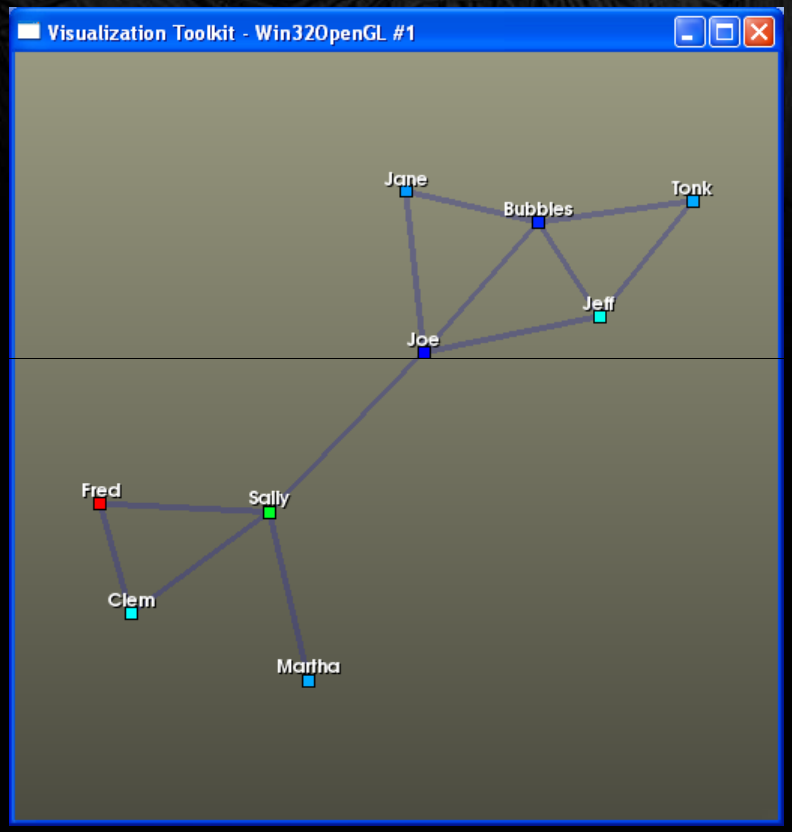
Graph Layout Strategies Example



Adding new graph layouts to Titan is a snap!

VTK/Examples/Infovis/Python/database.py

Making easy… easier…

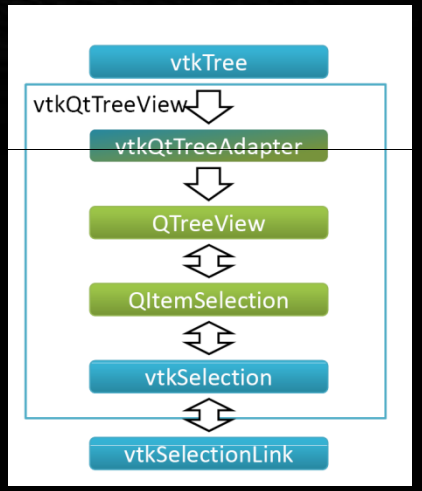


Using vtkSQLDatabaseGraphSouce which combines several classes to simplify pulling graphs from arbitrary databases.

VTK/Examples/Infovis/Python/database2.py

Qt Adapters

- vtkQtAbstractModelAdapter
  - Inherits from QAbstractItemModel, Qt's generic item model for views
  - Provides common infrastructure for converting QModelIndex to VTK ids.
- vtkQtTableModelAdapter
  - Inherits from vtkQtAbstractModelAdapter
  - Adapts underlying vtkTable instance to a Qt model
- vtkQtTreeModelAdapter
  - Inherits from vtkQtAbstractModelAdapter
  - Adapts underlying vtkTree instance to a Qt model
- QTableView, QTreeView
  - Display a QAbstractItemModel
- vtkQtTable/TreeView, vtkQtTable/TreeRepresentation
  - Puts QTableView, QTreeView into VTK view/representation framework using the model adapter classes
  - Supports selection linking with other VTK views.

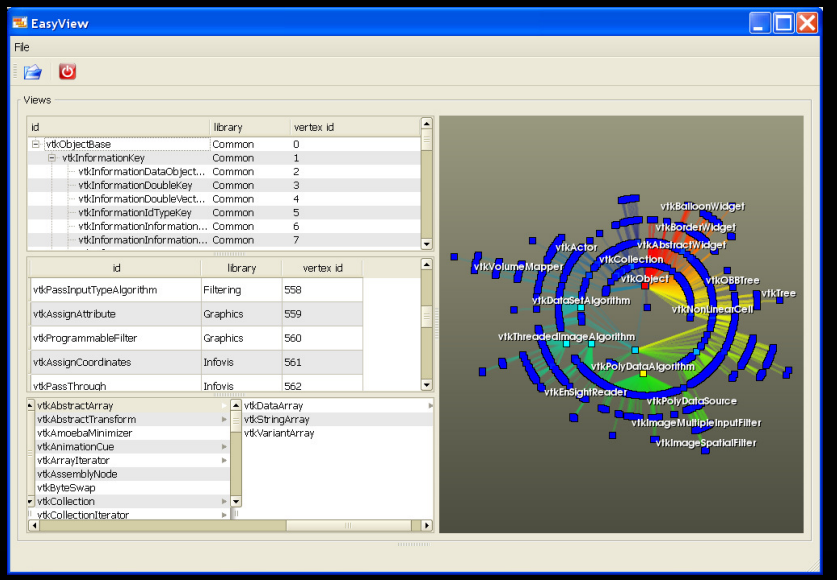


Qt Adapters C++ Example

Qt a reasonable model/view architecture for tables and trees
(specifically shown are QTableView, QTreeView, QColumnView).

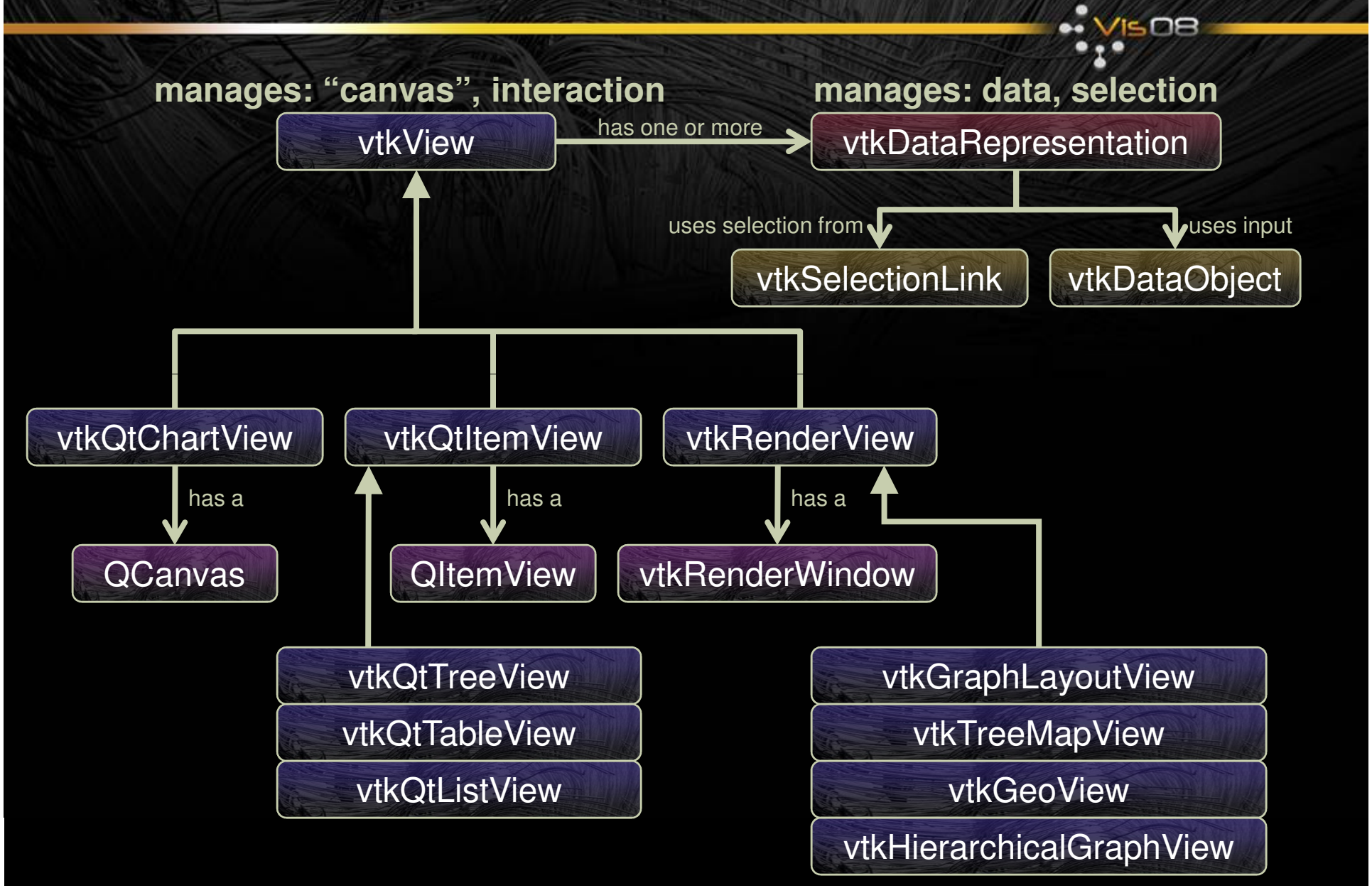
Code “clips” from VTK/Examples/Infovis/Cxx/EasyView

```
…
this->XMLReader    = vtkSmartPointer<vtkXMLTreeReader>::New();
this->TreeView     = vtkSmartPointer<vtkQtTreeView>::New();

// Set widget for the tree view
this->TreeView->SetItemView(this->ui->treeView);
…

// Create xml reader
this->XMLReader->SetFileName( fileName.toAscii() );
…

// Now hand off tree to the tree view
this->TreeView->SetRepresentationFromInputConnection(
    this->XMLReader->GetOutputPort());
…
```



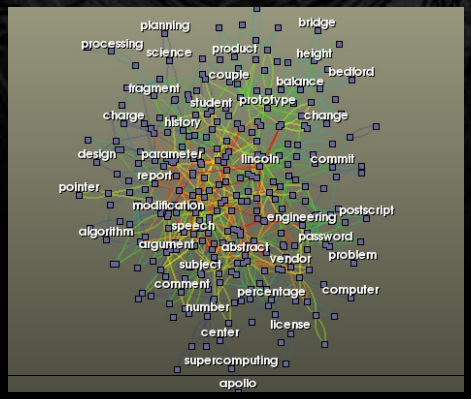
VTK/Examples/Infovis/Cxx/EasyView

Views

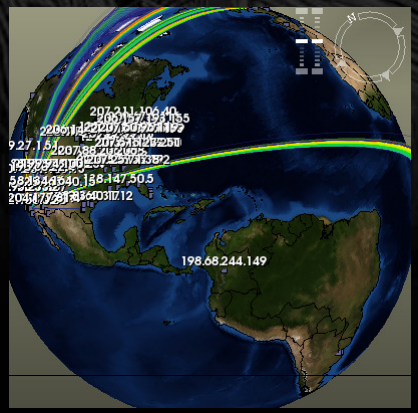

manages: “canvas”, interaction
manages: data, selection
vtkView
has one or more
vtkDataRepresentation
uses selection from
uses input
vtkSelectionLink
vtkDataObject
vtkQtChartView
vtkQtItemView
vtkRenderView
has a
has a
has a
QCanvas
QItemView
vtkRenderWindow
vtkQtTreeView
vtkQtTableView
vtkQtListView
vtkGraphLayoutView
vtkTreeMapView
vtkGeoView
vtkHierarchicalGraphView

Views Tour

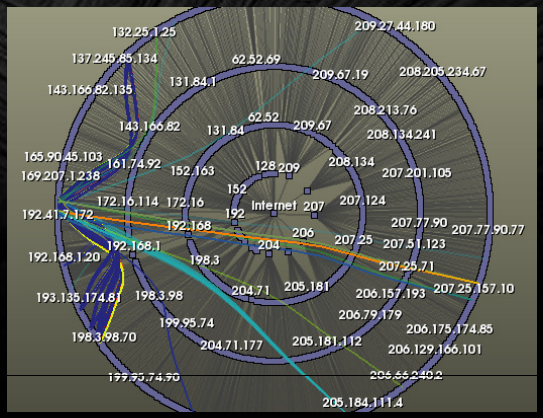
vtkGraphLayoutView



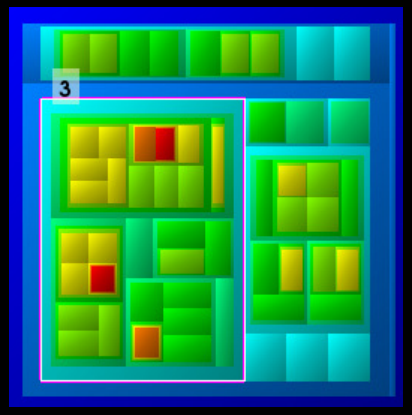
vtkGeoView



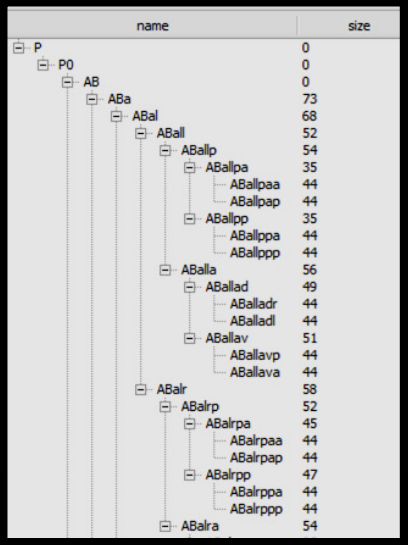
vtkHierarchicalGraphView



vtkTreeMapView



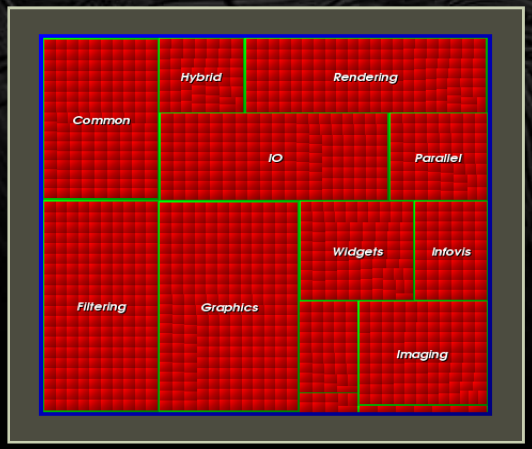
vtkQtTreeView

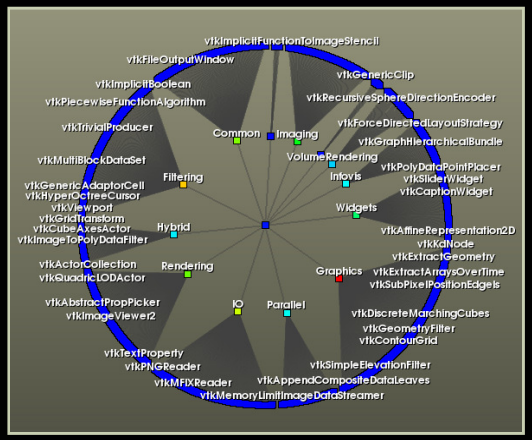


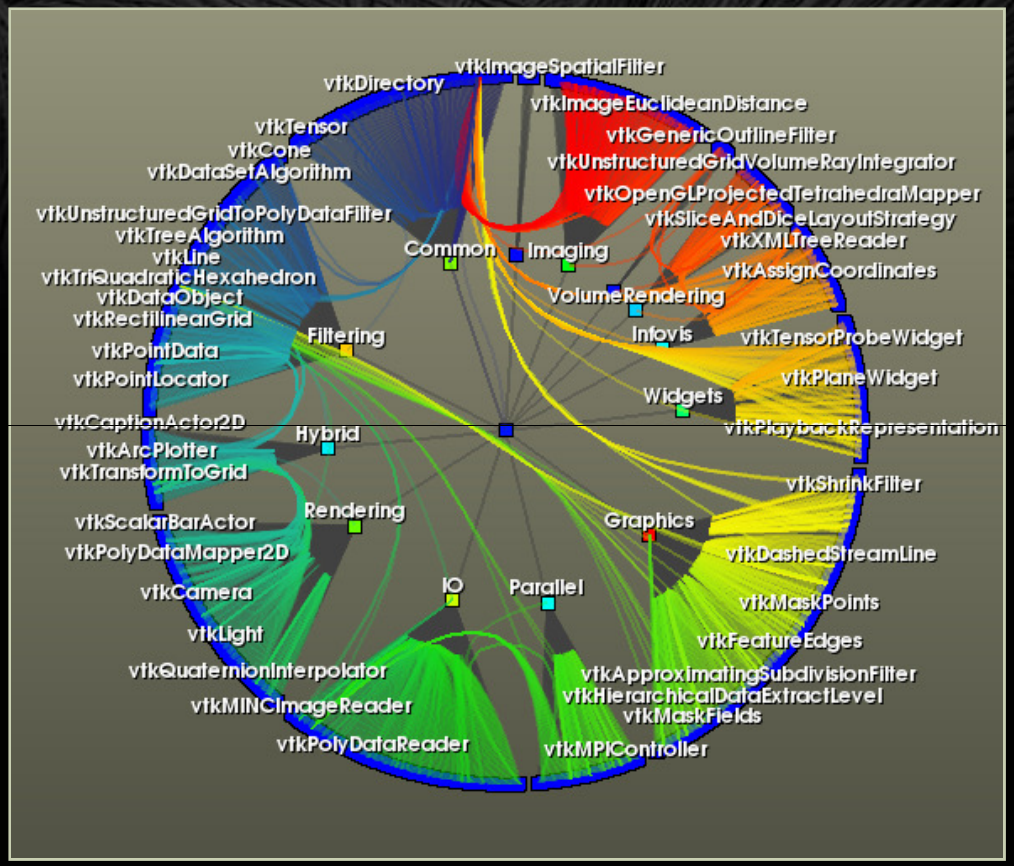
vtkQtTableView

title	year	release_date	num_ratings	average_rating
"18 Wheels of Justice" (2000)	2000	211814438400000	14	4.8
"24: Conspiracy" (2005)	2005	208657814400000	8	4.4
"29 Minutes & Counting" (2004)	2004	208657814400000	0	0
"2gether: The Series" (2000)	2000	211833100800000	17	5.5
"30 Days 'Til I'm Famous" (2006)	2006	208657814400000	0	0
"30 by 30: Kid Flicks" (2001)	2001	208657814400000	0	0
"411, The" (2005)	2005	212006592000000	0	0
"70's House, The" (2005)	2005	211987324800000	12	5.5
"8th & Ocean" (2006)	2006	212008492800000	89	4.9
"A.T.M.: A toda M." (2005)	2005	211989139200000	0	0
"A.U.S.A." (2003)	2003	211911120000000	0	0
"AMC Project, The" (2003)	2003	208657814400000	0	0
"AXN Action TV" (2000)	2000	208657814400000	0	0
"Aardvark" (2000)	2000	211815129600000	0	0
"Abby" (2003)	2003	211908614400000	0	0

Views Python Example



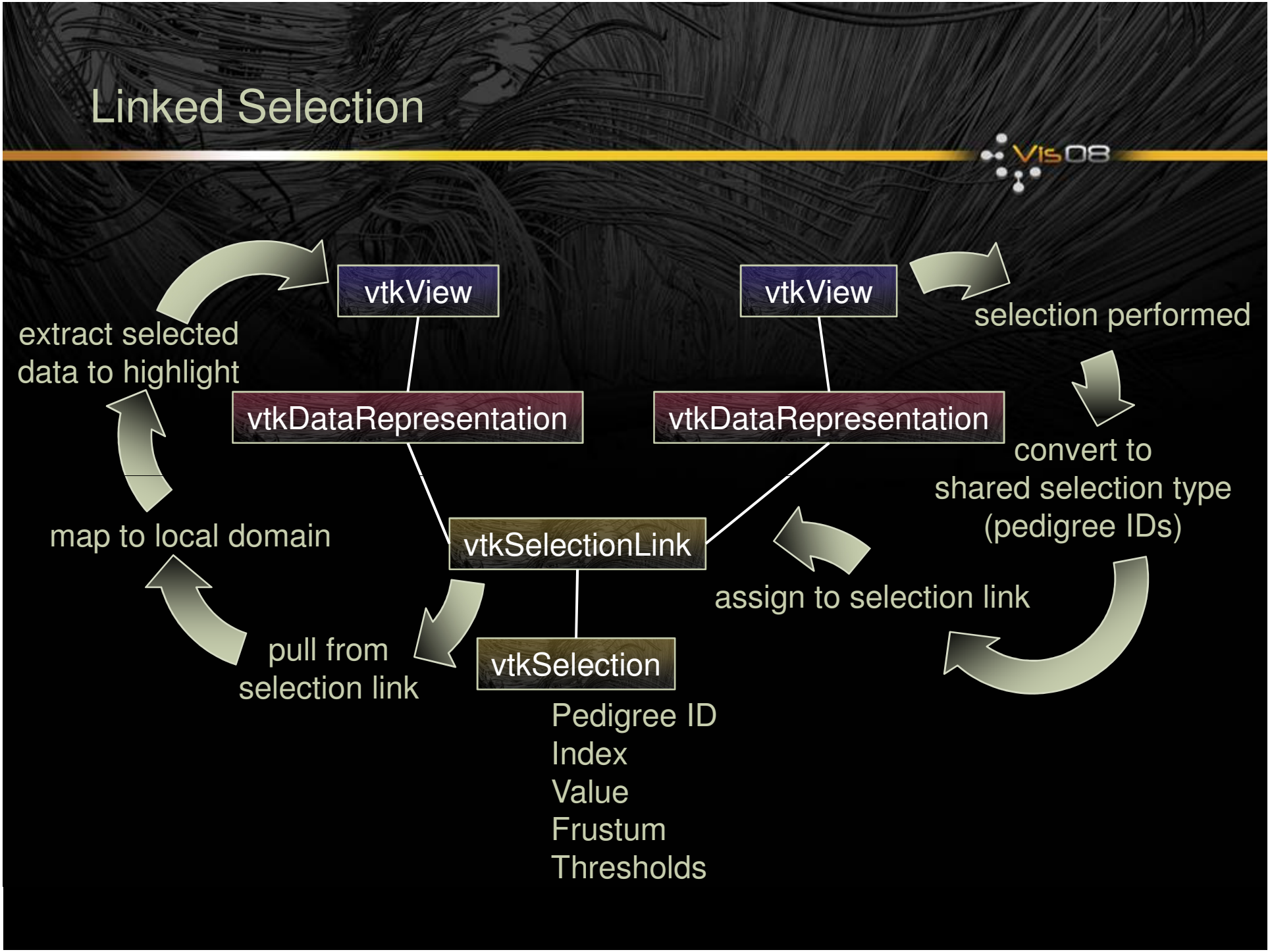




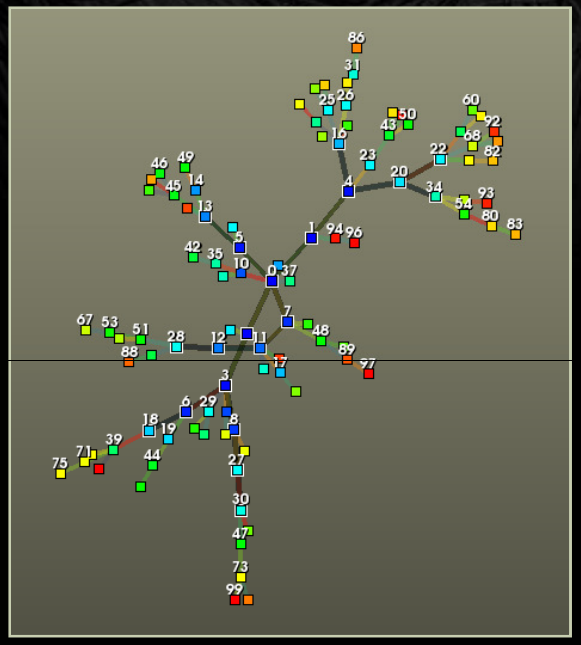
The VTK class hierarchy shown in a vtkTreeMapView, a vtkGraphLayoutView (with tree layout) and a vtkHierarchicalGraphView. The last view also pulls in a graph to show class inheritance relationships.

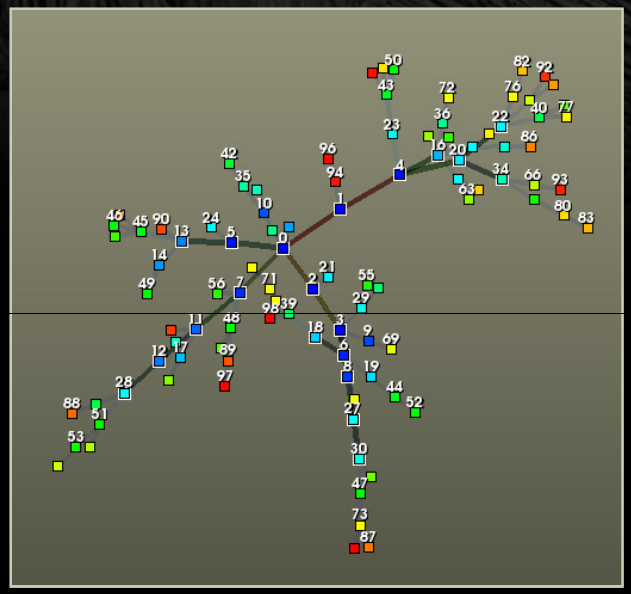
VTK/Examples/Infovis/Python/views.py

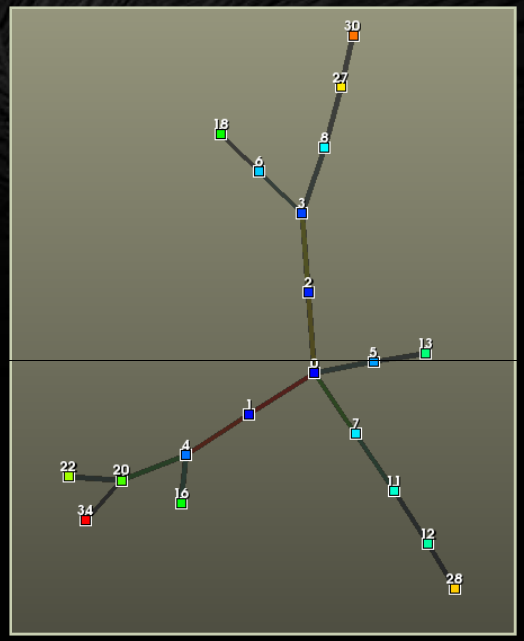
Linked Selection



Selection Python Example





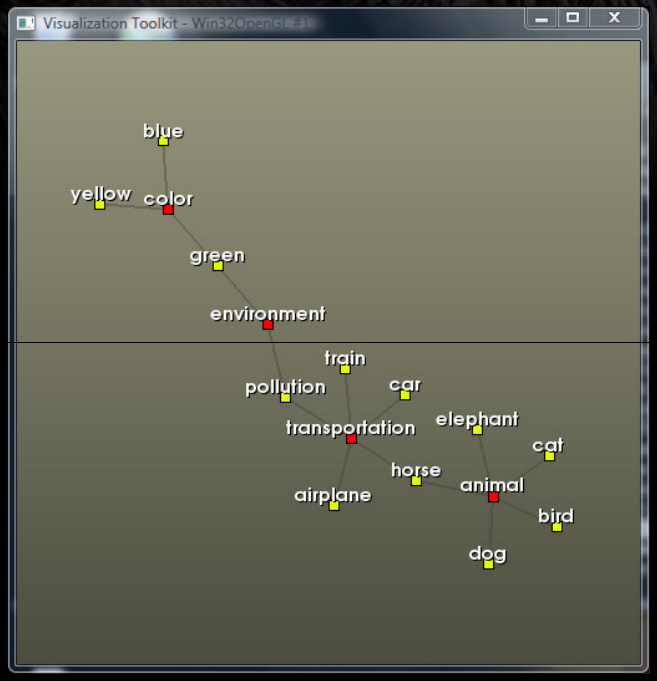


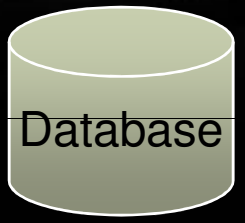
This example demonstrates the use of vtkSelectionLink and vtkSelectionSource. Any vtk view can link it's selection with any other view. vtkSelections are quite flexible and can be used in a variety of ways, here we select edges with high centrality.

VTK/Examples/Infovis/Python/selection.py

Domain Mapping

VTK/Examples/Infovis/Python/selection_convert.py





term | concept

person | document

selected terms

term ↔ document

map

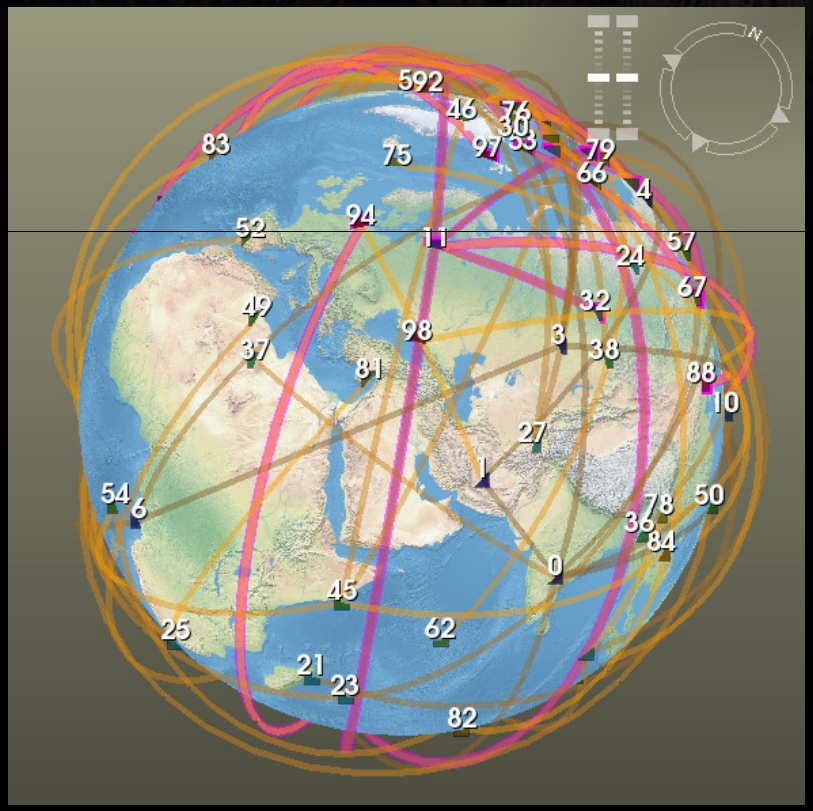
selected documents

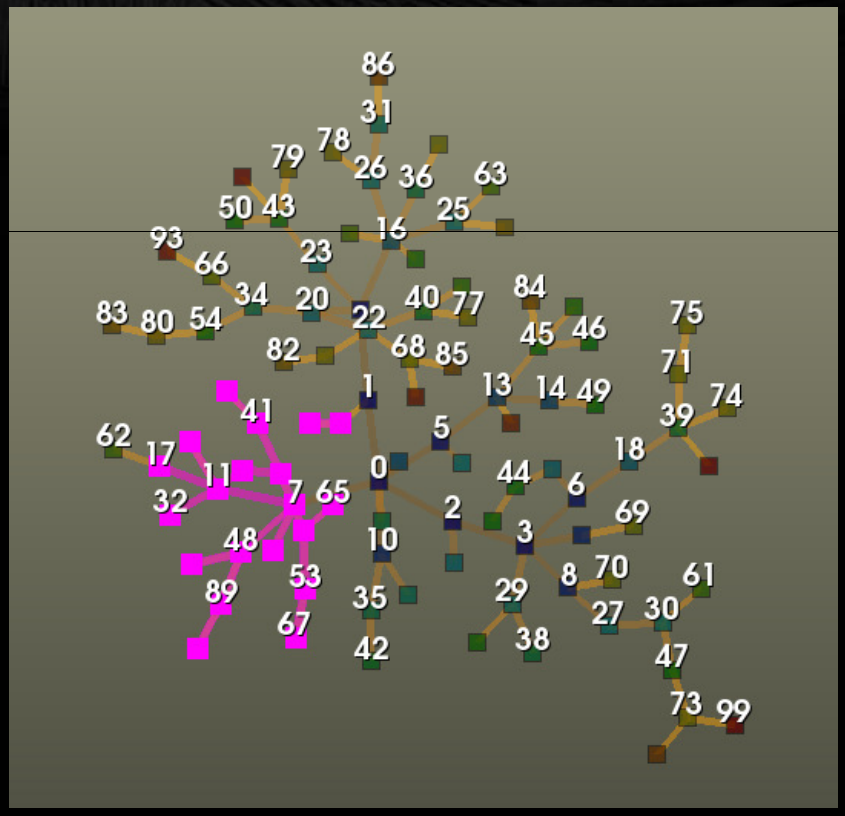
Geographic visualization

- Current features (in VTK now)
  - 3D vtkGeoView
  - Multi-resolution texture and geometry
  - Display placemarks with relationships (i.e. a geolocated graph)
  - Deep integration with other VTK views
    - Takes vtkDataObject input
    - Linked selection with other views
    - Easily embedded into larger applications
- Developing features
  - vtkGeoView2D
  - Multi-texturing overlay images with blending
  - More input sources

3D GeoView Python Example

Uses vtkGeoView and vtkGeoRandomGraphSource, linked with the same graph in a vtkGraphLayoutView.

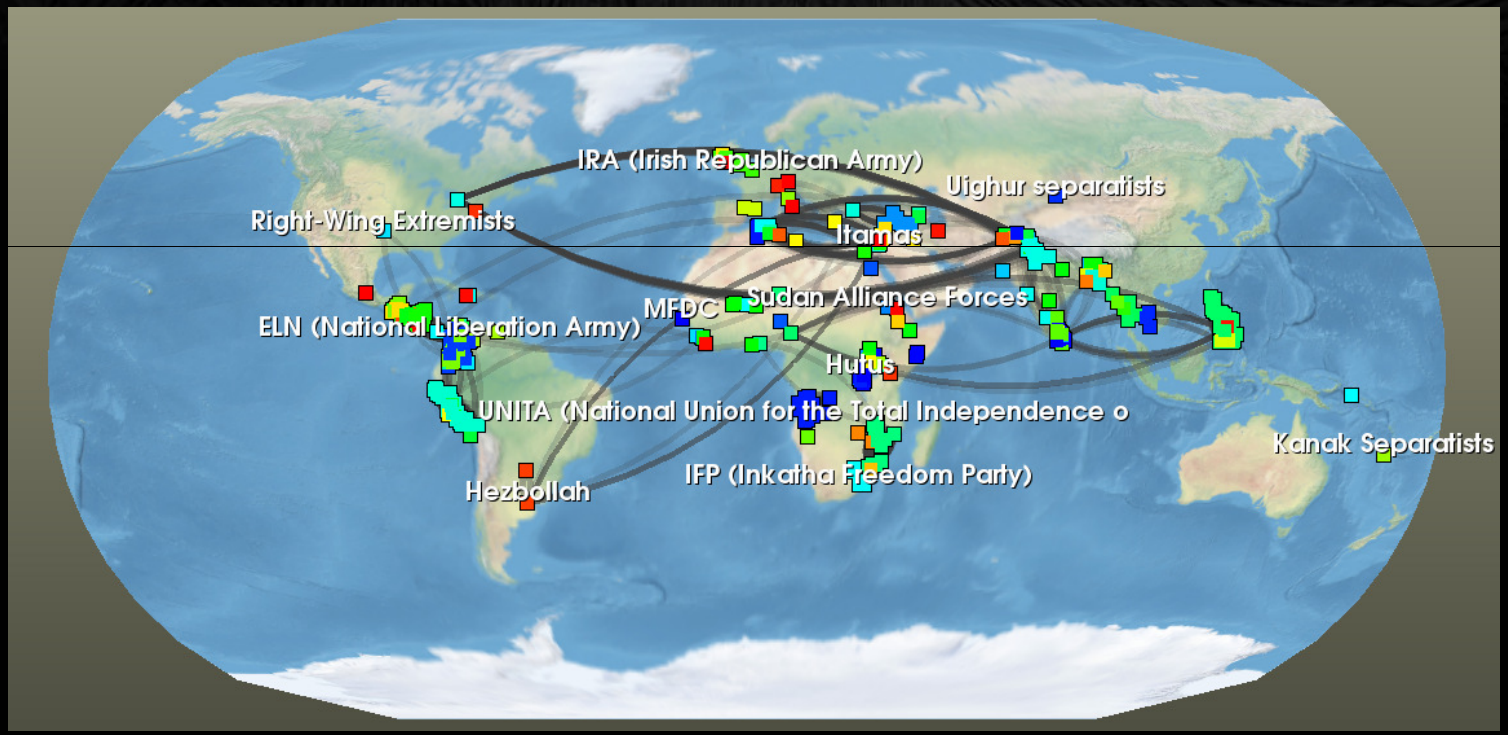




VTK/Examples/Infovis/Python/geovis.py

GeoView Python Examples

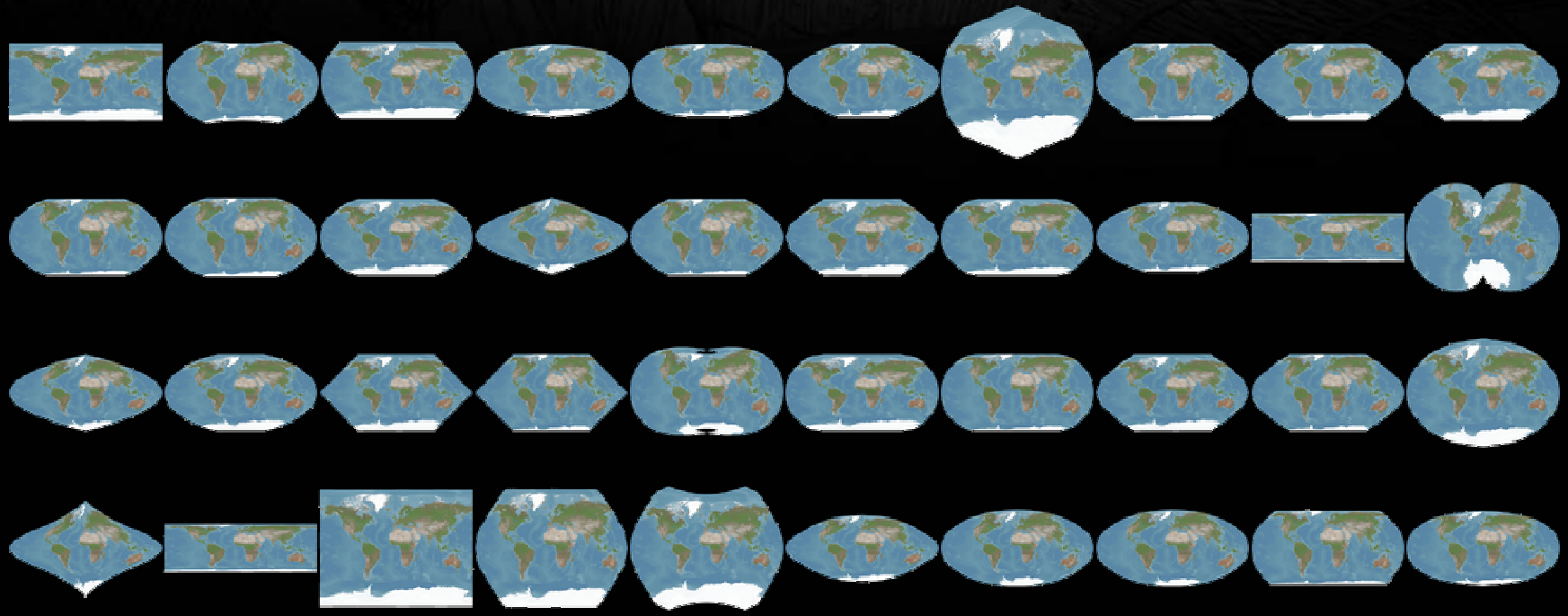
Pulls data from the publicly available GTD (Global Terrorism Database) and uses vtkGeoView2D.



vtkSNL/Examples/Python/Infovis/gtd_geovis_2d.py

Projections

All projections from the open-source Proj.4 projection library are avaliable to vtkGeoView2D.



Break Time!

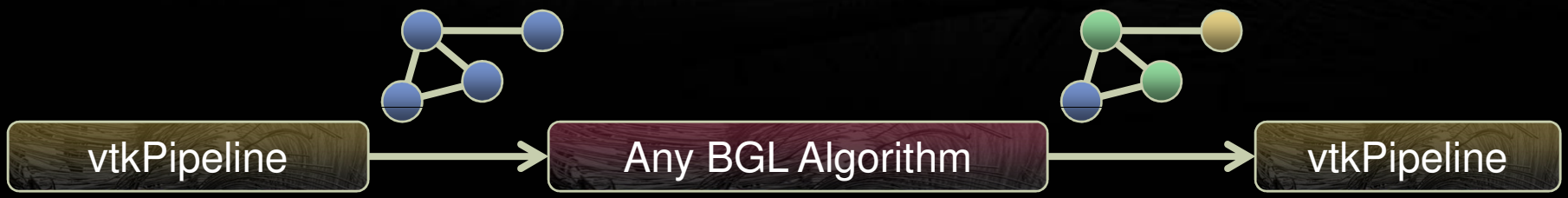
Graph Algorithms, Statistics, and Algebraic Methods are next…

Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

Boost Graph Library (BGL) Adapter

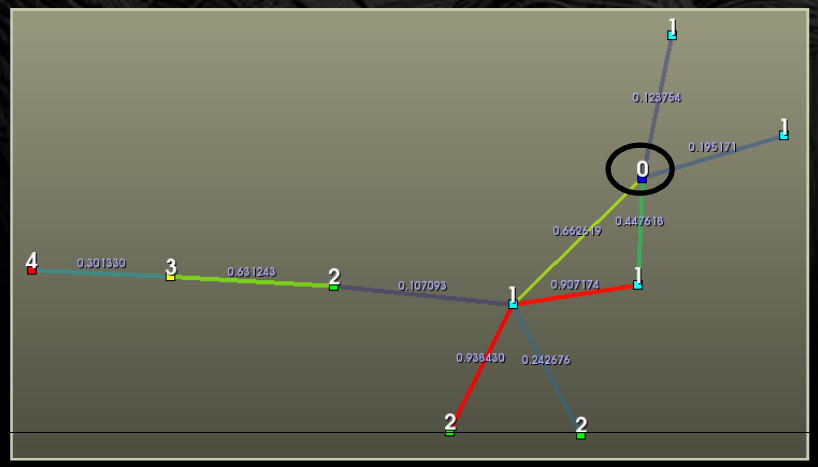
vtkBoostGraphAdapter.h implements the BGL graph concepts for vtkGraph.



vtkBoostBreadthFirstSearch
vtkBoostBreadthFirstSearchTree
vtkBoostBiconnectedComponents
vtkBoostBrandesCentrality
vtkBoostConnectedComponents
vtkBoostKruskalMinimumSpanningTree
vtkBoostPrimMininumSpanningTree

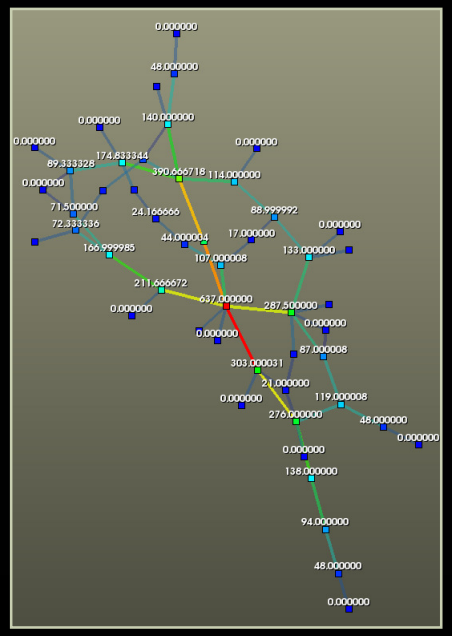
… add your own! ☺

BGL Python Examples



Running vtkBoostBreadthFirstSearch and coloring/labeling the vertices based on the distance from the seed point.

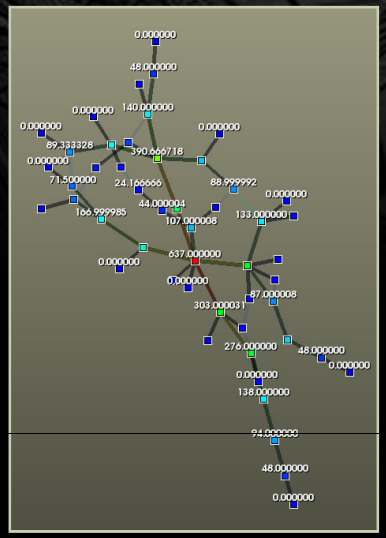
VTK/Examples/Infovis/Python/boost_bfs.py



Running vtkBoostBrandesCentrality and coloring/labeling the edges and vertices based on centrality.

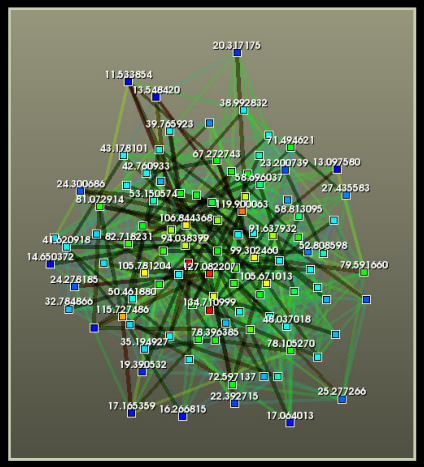
VTK/Examples/Infovis/Python/boost_centrality.py

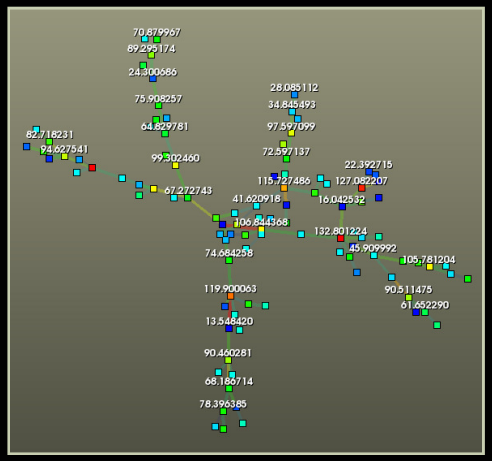
BGL Python Examples



Running vtkBoostBrandesCentrality and then vtkBoostKruskalMinimumSpanningTree to compute a ‘maximal’ spanning tree on high centrality edges.

VTK/Examples/Infovis/Python/boost_mst.py

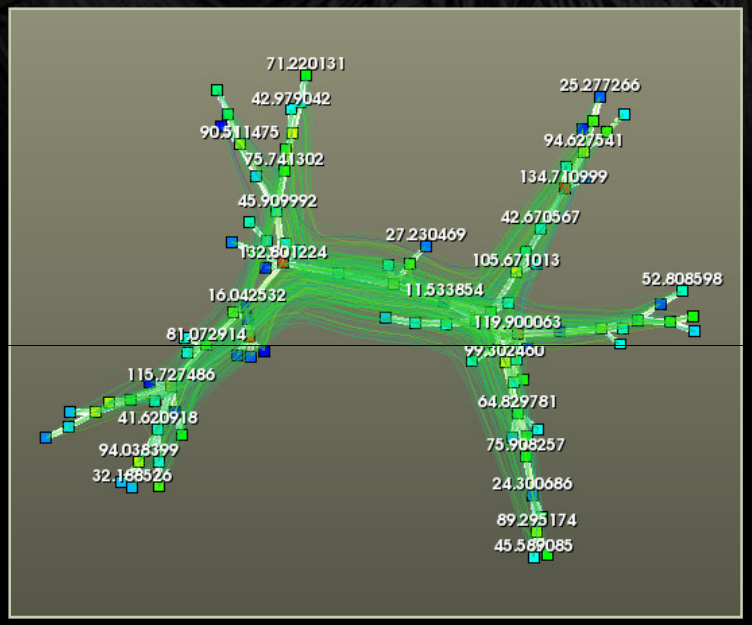


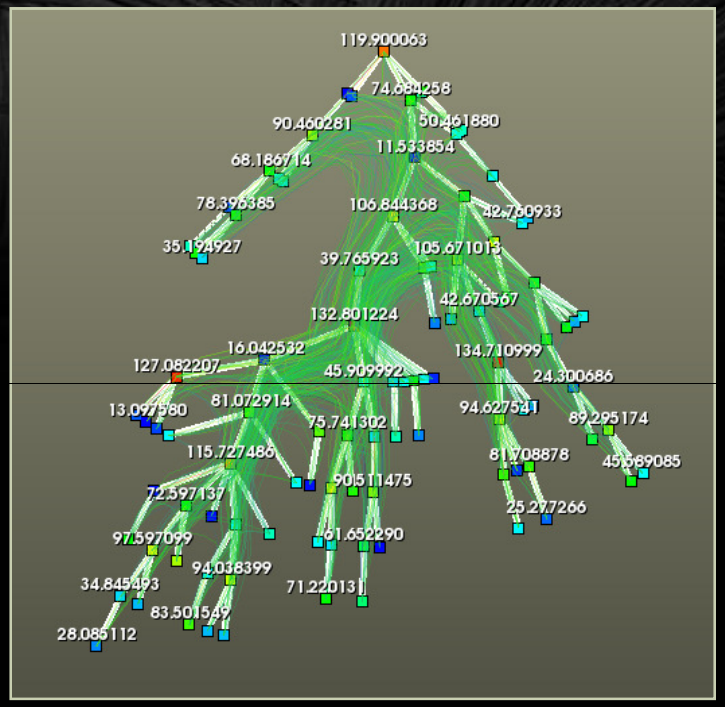


Running the same boost algorithms as above on a more complicated graph and then using vtkExtractSelectedGraph to send the extracted MST to another view.

VTK/Examples/Infovis/Python/boost_mst_extract_graph.py

BGL Python Examples

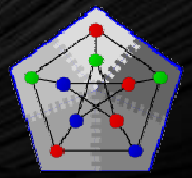




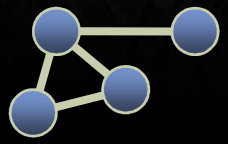
Now showing how the original graph and its computed ‘Maximal’ spanning tree can both be sent to vtkHierarchicalGraphView. The MST is used to drive the hierarchy and layout, the original graph edges are ‘bundled’ by using the hierarchy as control points.

VTK/Examples/Infovis/Python/boost_mst_with_hgv.py

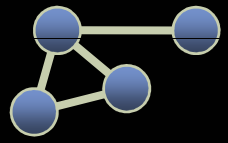
Parallel Boost Graph Library (PBGL) Adapter

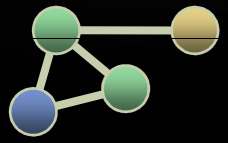


vtkPBGLGraphAdapter.h implements the PBGL graph concepts for a vtkGraph (with associated vtkPBGLDistributedGraphHelper).



= vtkGraph.SetDistributedHelper(PBGL);



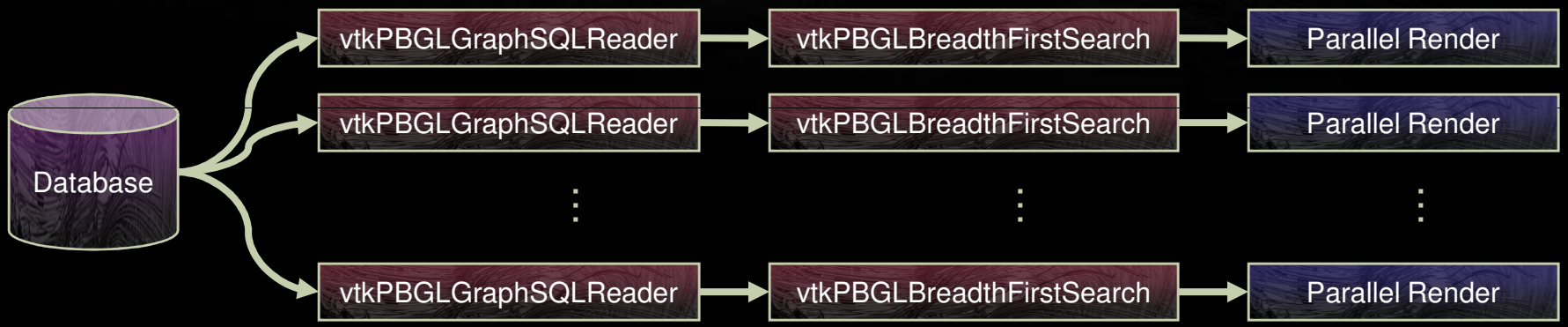


vtkPipeline → Any PBGL Algorithm → vtkPipeline

vtkPBGLShortestPaths
vtkPBGLRMATGraphSource
vtkPBGLMinimumSpanningTree
vtkPBGLGraphSQLReader
vtkPBGLConnectedComponents
vtkPBGLVertexColoring
vtkPBGLBreadthFirstSearch
vtkPBGLRandomGraphSource

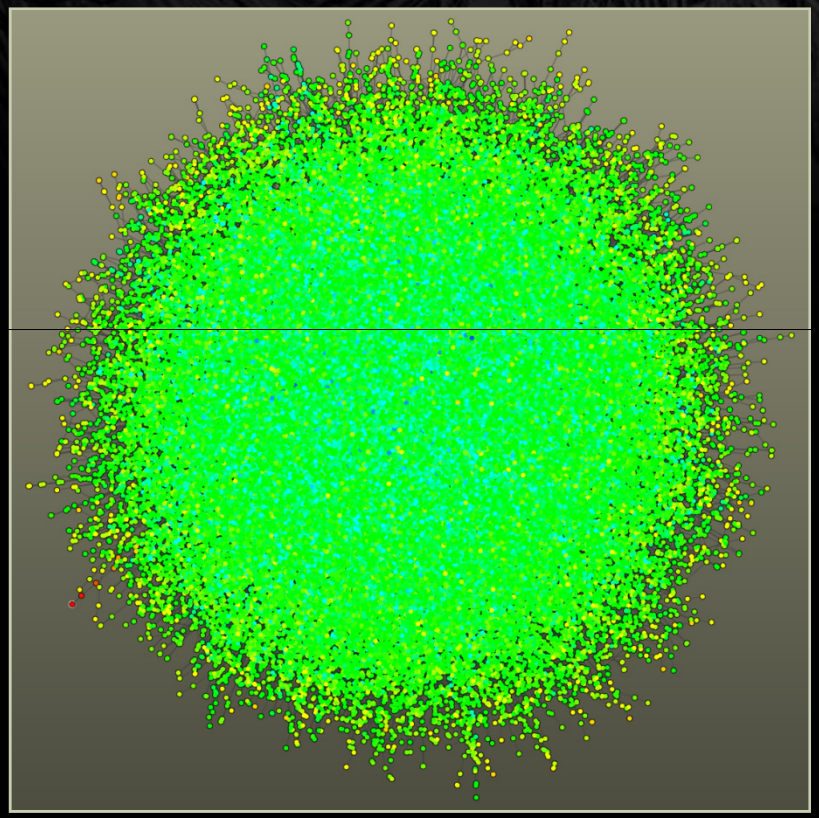
Parallel Graph Analysis – PBGL *(work in progress)*

PBGL: Parallel Boost Graph Library – http://www.osl.iu.edu/research/pbgl
Andrew Lumsdaine, Douglas Gregor (Indiana University)



Currently in the “Hello World” stage:
Running a BFS on a random graph containing 50M vertices and 500M edges on 80 nodes.

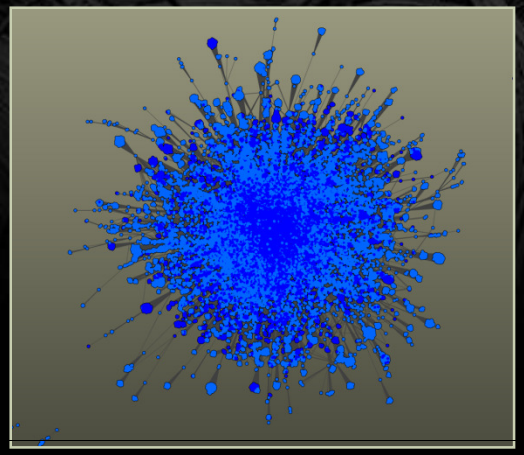
PBGL Examples



Performing BFS on a random graph with 100K vertices and 100K edges in parallel, collecting the graph and viewing it in graph layout view.

VTK/Examples/Infovis/Cxx/ParallelBFS.cxx

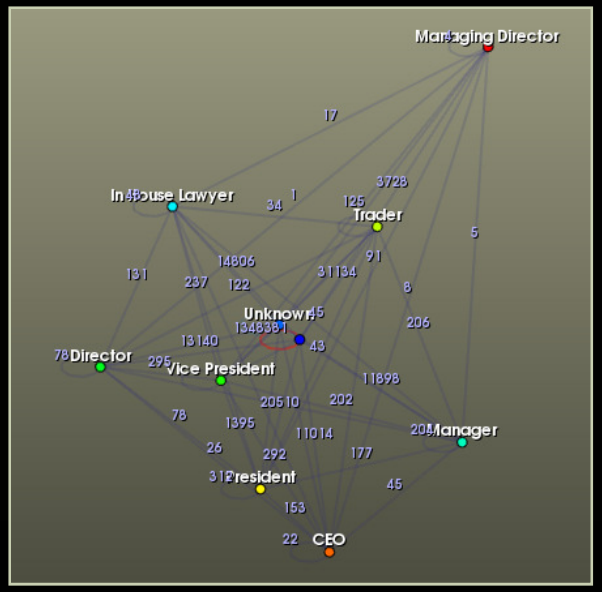
PBGL Examples



The Enron email corpus graph, containing 75K email accounts and 2M email communications.

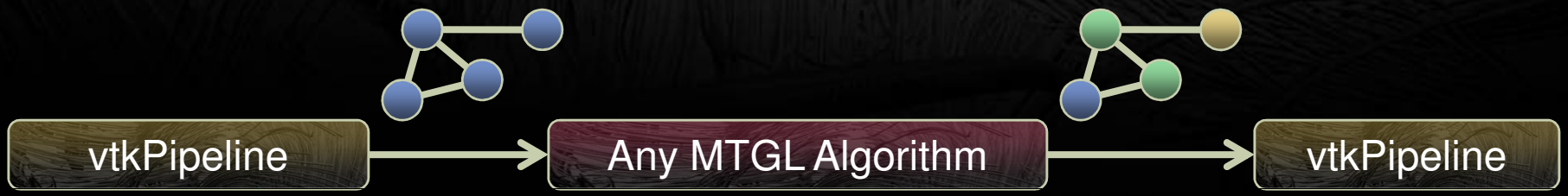
VTK/Parallel/Testing/Cxx/TestPBGLGraphSQLReader.cxx

Using a parallel pipeline to extract summary information of how people with different job titles interact.



Multi-Threaded Graph Library (MTGL) Adapter

vtkMTGLGraphAdapter.h implements the MTGL graph concepts for vtkGraph.

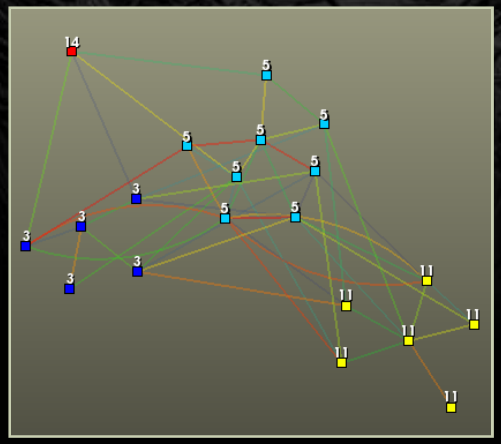


vtkMTGLCommunityFinder
vtkMTGLHierarchicalCommunityFinder
vtkMTGLSearchEdgeTime
vtkMTGLSearchSSSPDeltastepping
vtkMTGLSelectionFilterCSG
vtkMTGLSelectionFilterST

… list is growing…

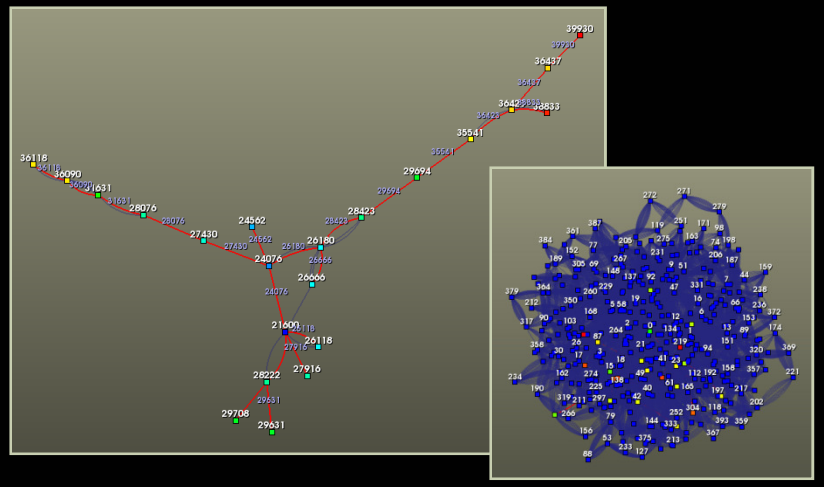
Cray XMT: Massively multithreaded platform, great for graph algorithms. ☺

MTGL Python Examples *(work in progress)*



Running vtkMTGLCommunityFinding and coloring/labeling the vertices based on the community.

vtkSNL/Examples/Python/Infovis/mtgl_community.py



Running vtkBoostTemporalSearchFwd and coloring/labeling the edges and vertices based on earliest ‘reachability’.

vtkSNL/Examples/Python/Infovis/temporal_search_test.py

Titan Statistics Functionality

The statistics engines can be run in “Learn” (calculate model statistics from a data set) and/or “Assess” (given model statistics – from the same or another data set -- mark each datum) options.

Univariate statistics:

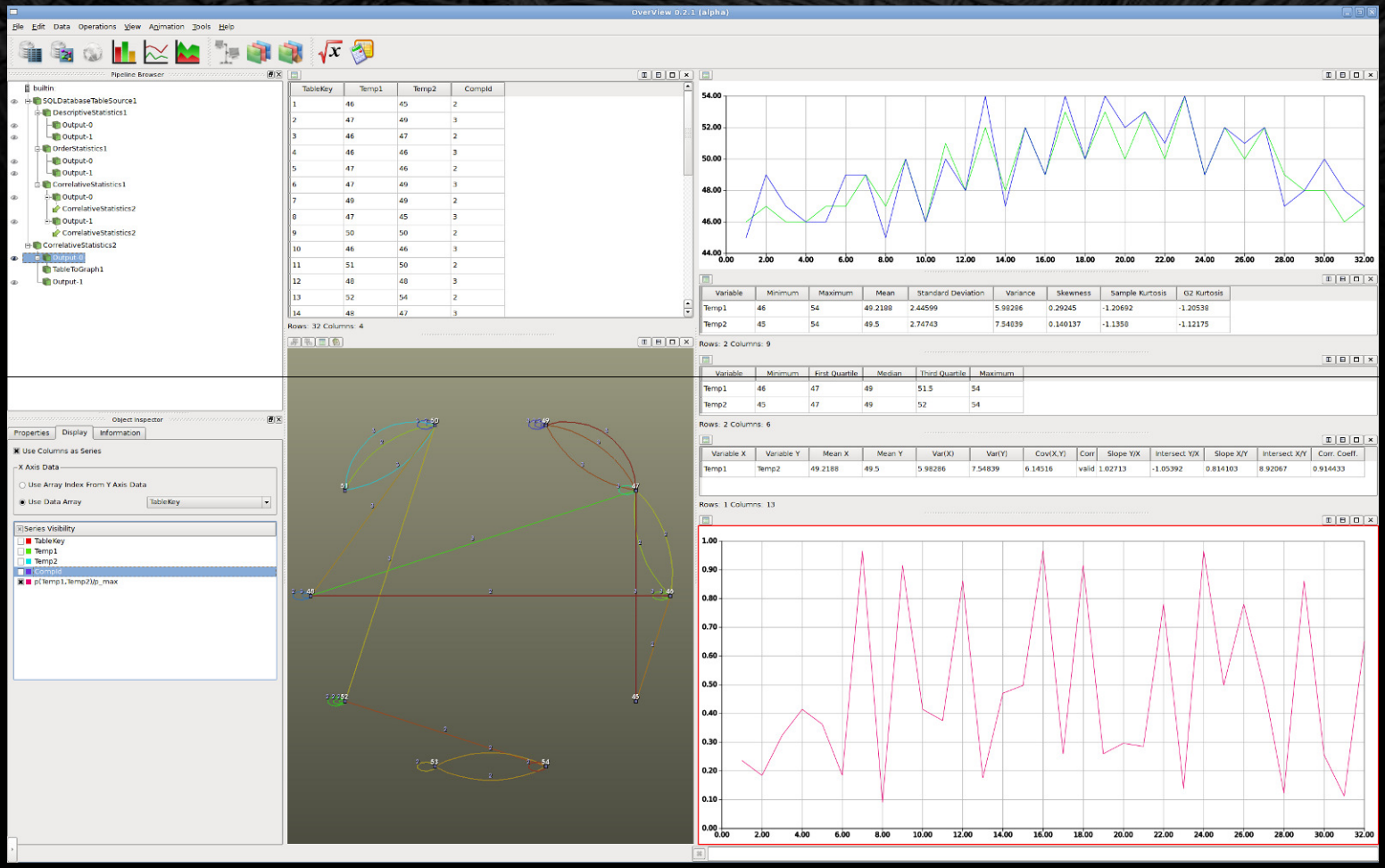
- Descriptive statistics:
  - “Learn”: minimum, maximum, mean, variance, skewness, kurtosis (various estimators)
  - “Assess”: mark with number or relative deviations (specified means and deviation)
- Order statistics:
  - “Learn”: arbitrary quartiles (in particular, "5-point" statistics (quartiles) and box plots, deciles, percentiles, etc.), histogram.
  - “Assess”: mark with deviation from specified box

Bivariate statistics:

- Correlative statistics:
  - “Learn”: bivariate mean, variance/covariance matrix, Pearson regression
  - “Assess”: mark with relative probability w.r.t. to specified means and covariance matrix
- Contingency statistics:
  - “Learn”: contingency table and joint probabilities
  - “Assess”: mark with conditional PDF values (X|Y and Y|X) and with joint PDF value. Also, calculate information entropies to decide which conditioning is the most informative.

Contingency statistics can be performed on any (categorical) type of variables; the other engines take numerical variables as inputs.

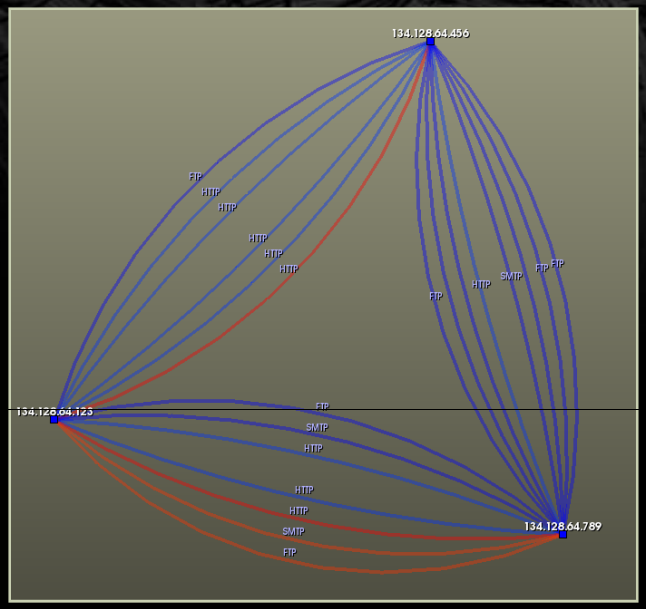
Descriptive, Order and Correlative Statistics



BTW: Linux Screenshot

Awesomeness is free ☺

Contingency Statistics Example



Running contingency statistics on network transfers illuminates protocols going over non-standard network ports.

VTK/Examples/Infovis/Python/contingency_port_protocol.py

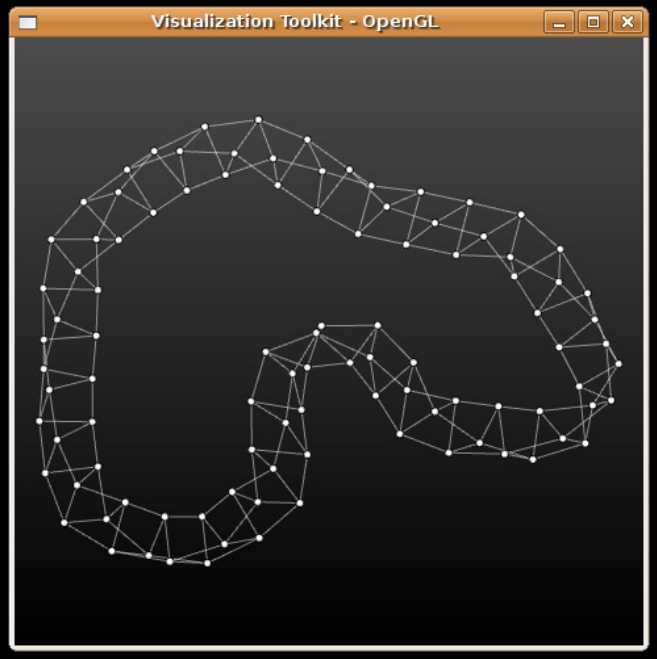
Demonstrates a conditional probability calculation p(port | protocol).

MATLAB® Titan Toolbox *(work in progress)*

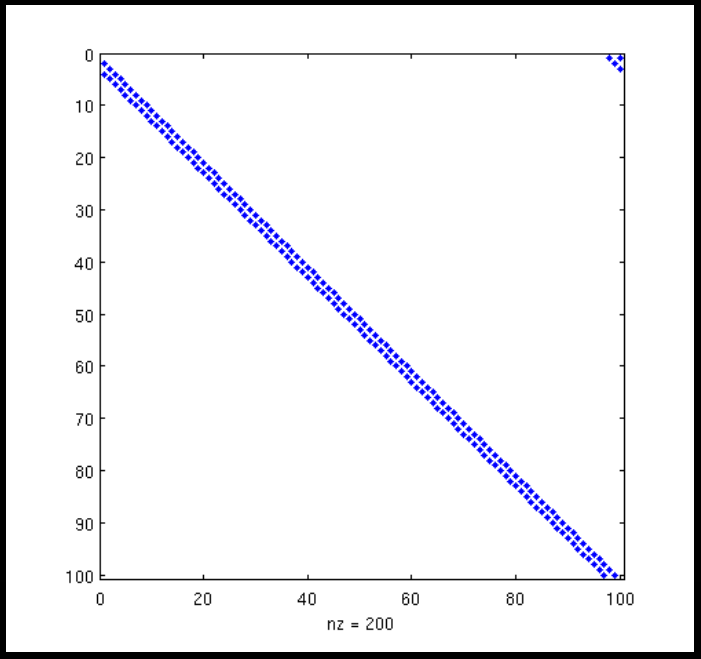
- The MATLAB® Titan Toolbox allows Titan functionality to be accessed from the MATLAB® command line.

```
>> g = graph;
>> for i=0:99 g.addvertex; end
>> for i=0:99 g.addedge(i,mod(i+1, 100)); g.addedge(i,mod(i+3, 100)); end
```

```
>> g.view;
```

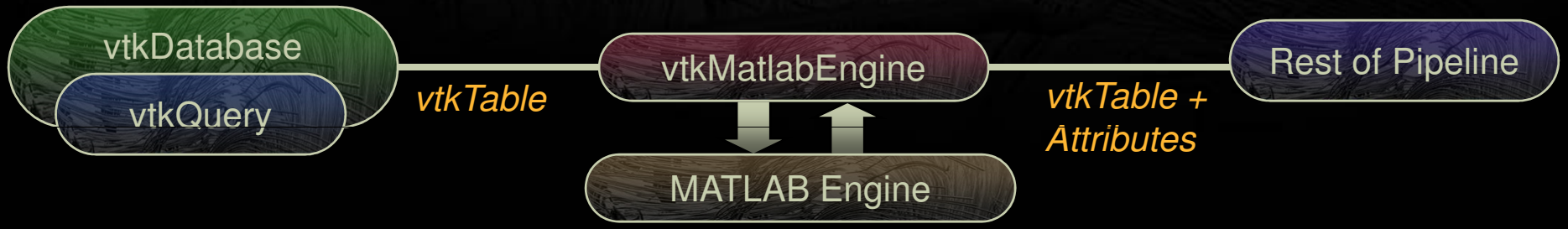



```
>> spy(g.tomatrix);
```



MATLAB® Interface *(work in progress)*

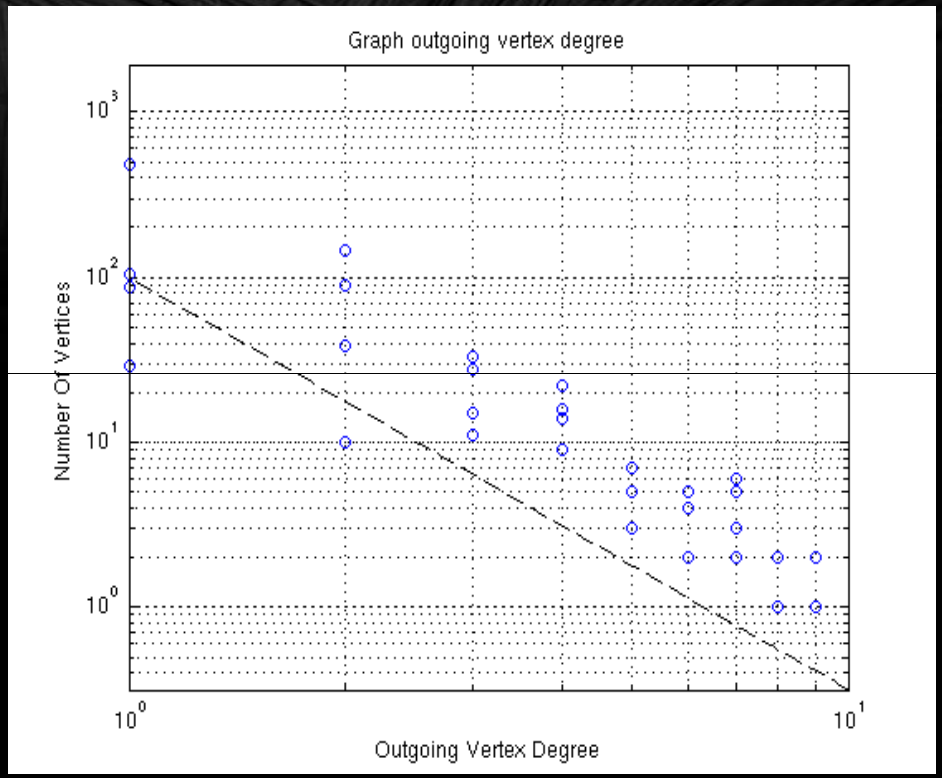
- vtkMatlabEngineFilter
  - Interact with a MATLAB® engine process
  - Execute MATLAB® commands from VTK
  - Push VTK pipeline data to MATLAB®
  - Pull MATLAB® data into the VTK pipeline



- vtkMatlabProgrammableFilter
  - Call a compiled MATLAB® M-file function from VTK
  - Filter handles the necessary data conversions
  - Allows fast prototyping through M-file scripting
  - Requires MATLAB® MCC compiler product

MATLAB® Examples *(work in progress)*

Uses algorithm by **Volchenkov** and **Blanchard** (*An algorithm generating random graphs with power law degree distributions, Physica A, Volume 315, Number 3, 1 December 2002 , pp. 677-690(14)*) to produce four random graphs with a power law degree distribution on graph vertex out-degree.



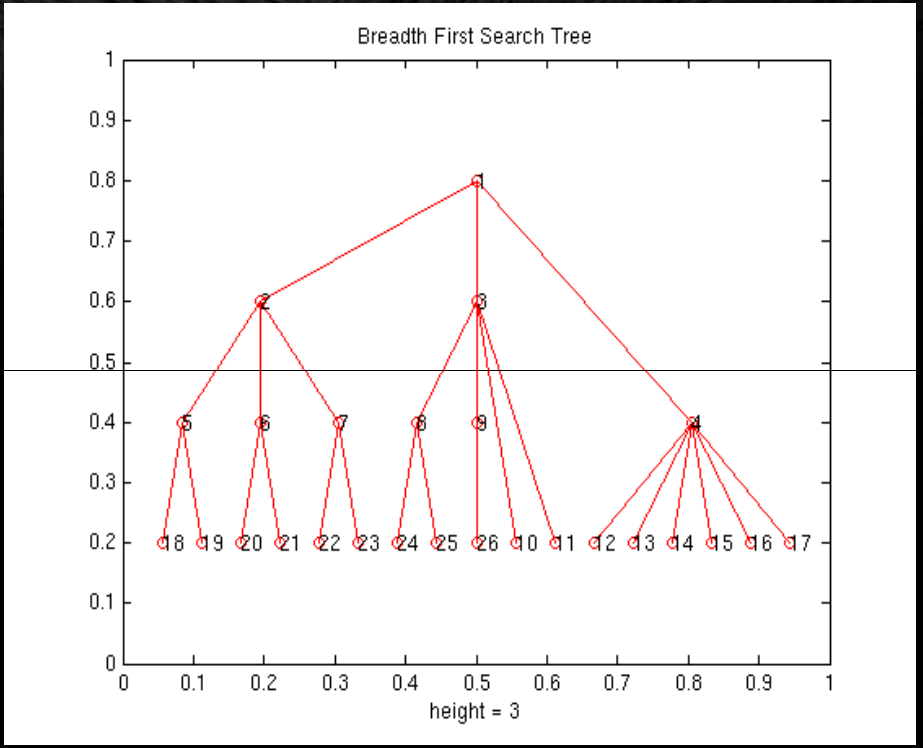
A log plot of vertex out-degree (with blue circles) from the four random graphs shows the expected linear relationship between vertex fraction and outgoing vertex degree.

vtkSNL/Examples/MatlabTitanToolbox/powerlawgraph.m

MATLAB® Examples *(work in progress)*

Generate a random power law graph with 50 vertices.

Perform a Breadth First Search on the graph and display the resulting search tree.

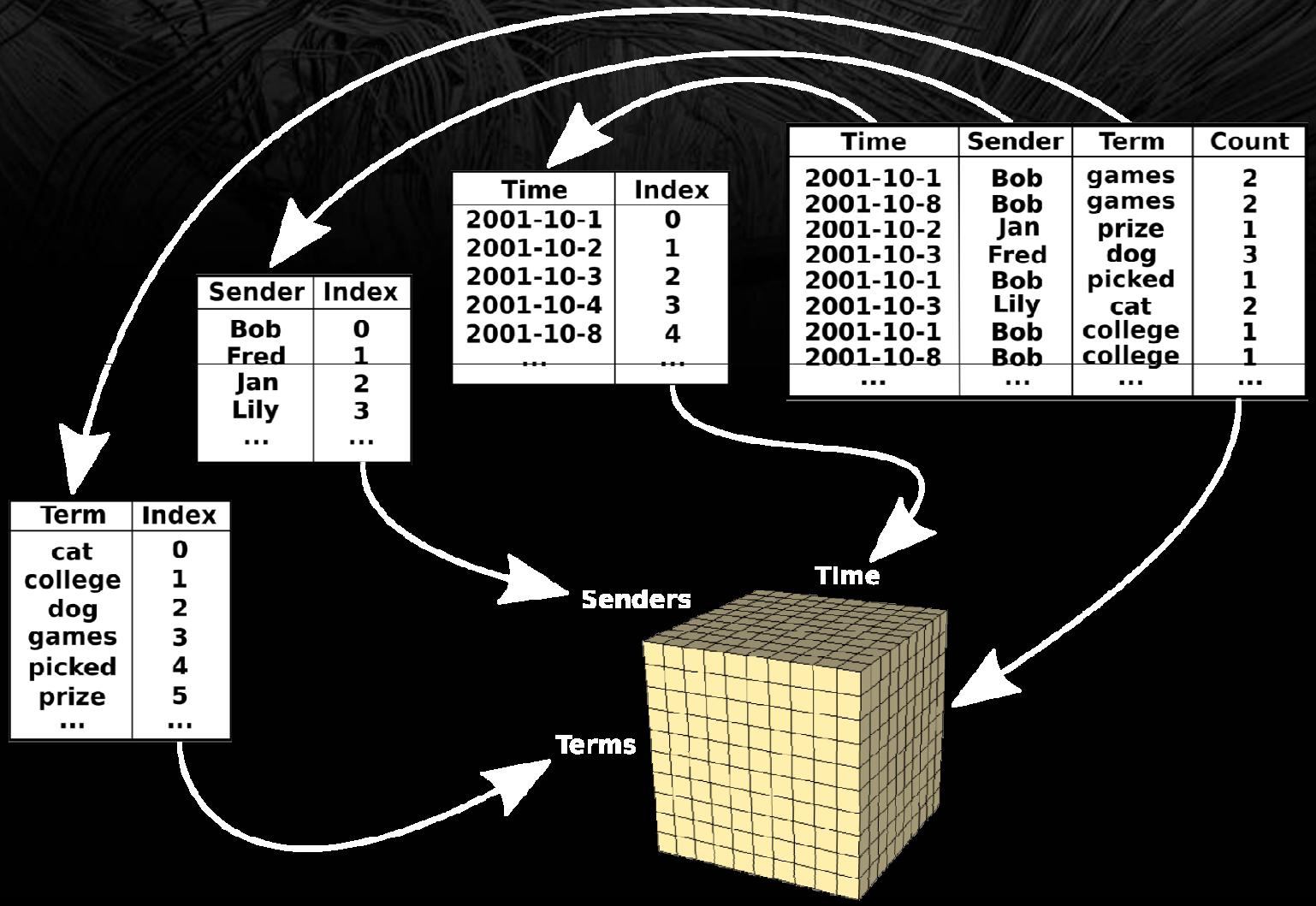


vtkSNL/Examples/MatlabTitanToolbox/bfstree.m

Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

Algebraic Methods - Motivating Example

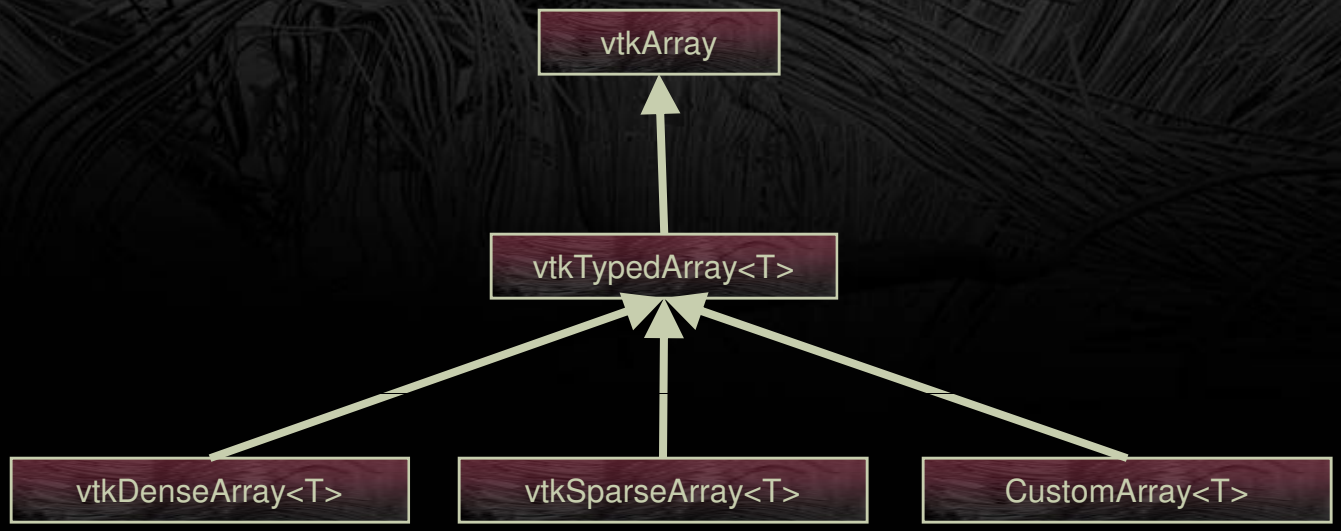


Data Structures for Algebraic Methods

- What We Wanted To Do:
  - Integrate algebraic methods into the pipeline, including tensors.
    - Note: mathematical tensors, not tensor-fields!
    - SVD / LSA
    - PARAFAC / TUCKER / DEDICOM
  - Provide dense and sparse storage.
  - Efficiently represent graphs as adjacency matrices.
  - *Possibly* provide a future replacement for vtkAbstractArray and friends.

- What We *Didn't* Want To Do:
  - Invent another linear algebra package.
- Caveats:
  - Currently, the array classes aren't wrapped / aren't accessible from scripting languages (because they're class templates).
  - You can still use array *algorithms* for scripting.

vtkArray Hierarchy



- vtkArray: Generic arrays of unknown type.
- vtkTypedArray: Generic arrays of known type.
- vtkDenseArray: Contiguous-storage arrays of known type.
- vtkSparseArray: Coordinate-storage arrays of known type.
- Custom Arrays: The array API allows for "custom" arrays - imagine compressed-row storage, upper / lower diagonal storage, etc.

Creating N-Way Arrays - *VTK/Examples/Array/Cxx/ArrayBasics.cxx*


```
// Creating a dense array of 10 integers:
vtkDenseArray<vtkIdType>* array = vtkDenseArray<vtkIdType>::New();
array->Resize(10);
array->Fill(1);

// Creating a dense 10 x 20 matrix:
vtkDenseArray<double>* matrix = vtkDenseArray<double>::New();
matrix->Resize(10, 20);
matrix->Fill(0.0);

// Creating a sparse 10 x 20 x 30 x 40 tensor:
vtkArrayExtents extents;
Extents.SetDimensions(4);
extents[0] = 10;
extents[1] = 20;
extents[2] = 30;
extents[3] = 40;
vtkSparseArray<vtkIdType>* tensor = vtkSparseArray<vtkIdType>::New();
tensor->Resize(extents);
```

Manipulating Values - *VTK/Examples/Array/Cxx/ArrayBasics.cxx*


```
// Assign array value [5]:
array->SetValue(5, 42);

// Assign matrix value [4, 3]:
matrix->SetValue(4, 3, 1970);

// Assign tensor value [3, 7, 1, 2]:
vtkArrayCoordinates coordinates;
coordinates.SetDimensions(4);
coordinates[0] = 3;
coordinates[1] = 7;
coordinates[2] = 1;
coordinates[3] = 2;
tensor->SetValue(coordinates, 38);
```

```
// Access array value [5]:
array->GetValue(5);

// Access matrix value [4, 3]:
matrix->GetValue(4, 3);

// Access tensor value [3, 7, 1, 2]:
tensor->GetValue(coordinates);
```

Populating Sparse Arrays

- vtkSparseArray<T>::AddValue() appends unchecked values to a sparse array.
  - Note: it's up to the caller to avoid calling AddValue() more than once with the same set of coordinates!
- vtkSparseArray<T>::ResizeToContents() updates the array extents to match the current array contents.
  - Handy once you're done calling AddValue().
- Example: *VTK/Examples/Array/Cxx/IdentityMatrix.cxx*

Array Value Iteration

- Iteration provides unordered access in O(1) time:
  - Eliminates the cost of lookups when getting / setting sparse array values.
  - Only visits non-null values in sparse arrays.
  - Provides a consistent interface across dense and sparse arrays.
  - Is completely dimension-independent.
  - Example: *VTK/Examples/Array/Cxx/ArrayIteration.cxx*

```
// Set the n-th value in an array:
void vtkTypedArray<T>::SetValueN(vtkIdType n, const T& value);

// Return the n-th value in an array:
const T& vtkTypedArray<T>::GetValueN(vtkIdType n);

// Return the coordinates for the n-th value in an array:
void vtkArray::GetCoordinatesN(vtkIdType n, vtkArrayCoordinates& coordinates);
```

Array Sources

- vtkDiagonalMatrixSource
  - Produces sparse or dense matrices of arbitrary size, with user-assigned values for the diagonal, superdiagonal, and subdiagonal, e.g:
- vtkBoostRandomSparseArraySource
  - Produces sparse matrices with arbitrary size and number of dimensions.
  - Separate controls for random values and random sparsity pattern.
- vtkTableToSparseArray
  - Converts a vtkTable containing coordinates & values into a sparse array of arbitrary dimensions.

Array Algorithms

- vtkAdjacencyMatrixToEdgeTable
  - Converts a dense matrix into a vtkTable suitable for use with vtkTableToGraph.
  - Dimension labels in the input matrix are mapped to column names in the output table.
- vtkArrayVectorNorm
  - Computes an L-norm for each column-vector in a sparse double matrix.
- vtkCosineSimilarity
  - Treats each row or column in a matrix as a vector, and computes the dot-product similarity between each pair of vectors, producing a vtkTable suitable for use with vtkTableToGraph as output.

More Array Algorithms

- vtkBoostLogWeighting
  - Replaces each value p in an array with the natural logarithm of p+1.
  - Good example of a filter that works with any array, containing any number of dimensions.
- vtkMatricizeArray
  - Converts sparse double arrays of arbitrary dimension to sparse matrices.
  - For example, an i x j x k tensor can be converted into an i x jk, j x ik, or ij x k matrix.
- vtkNormalizeMatrixVectors
  - Normalizes either row vectors or column vectors in a matrix.
  - Good example of a filter that works efficiently with both sparse and dense input matrices.
  - Good example of a filter that works with either row or column vectors.
- vtkTransposeMatrix
  - Does what it says …

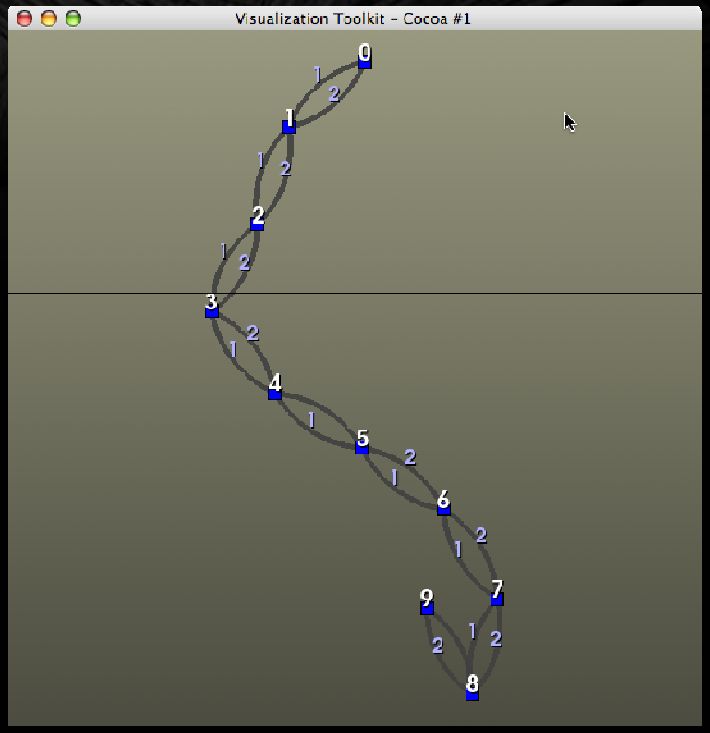
Example - VTK/Examples/Array/Cxx/AdjacencyMatrix.cxx


```
// Create a matrix with non-zero super- and sub-diagonals
vtkDiagonalMatrixSource* source =
    vtkDiagonalMatrixSource::New();
source->SetExtents(10);
source->SetDiagonal(0);
source->SetSuperDiagonal(1);
source->SetSubDiagonal(2);
```

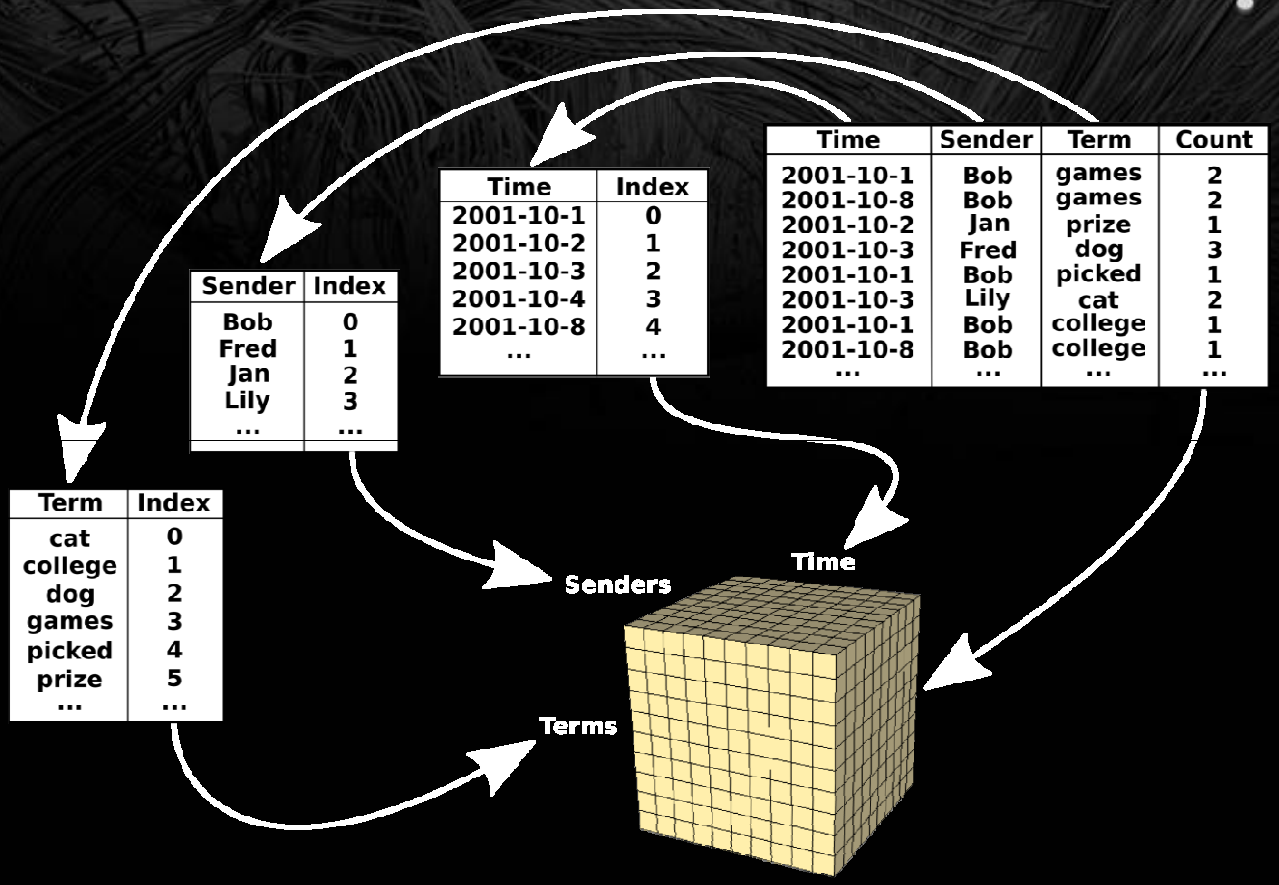
```
0 1 0 0 0 0 0 0 0 0
2 0 1 0 0 0 0 0 0 0
0 2 0 1 0 0 0 0 0 0
0 0 2 0 1 0 0 0 0 0
0 0 0 2 0 1 0 0 0 0
0 0 0 0 2 0 1 0 0 0
0 0 0 0 0 2 0 1 0 0
0 0 0 0 0 0 2 0 1 0
0 0 0 0 0 0 0 2 0 1
0 0 0 0 0 0 0 0 2 0
```

Example - VTK/Examples/Array/Cxx/AdjacencyMatrix.cxx

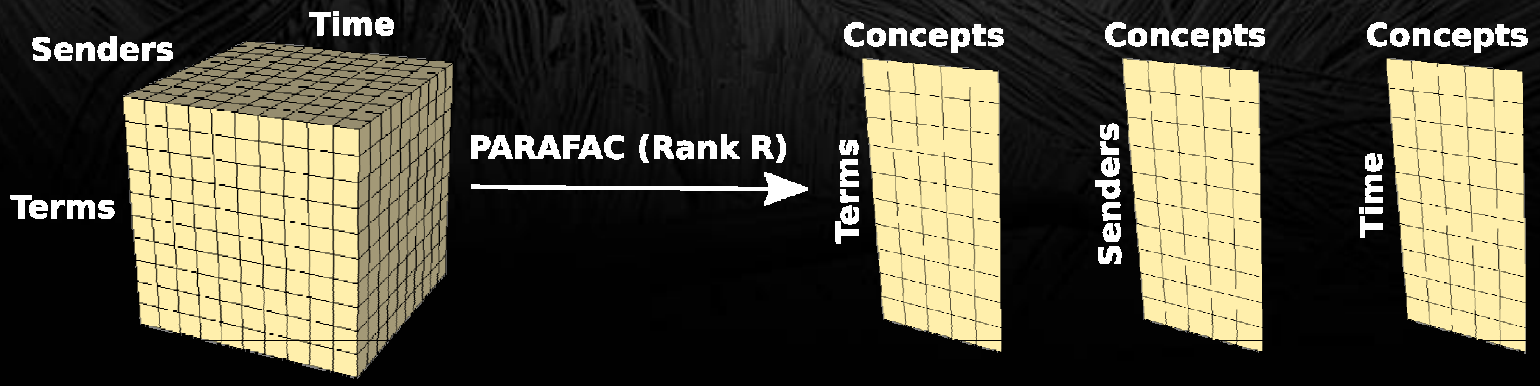
```
// Convert an adjacency matrix to an edge table
vtkAdjacencyMatrixToEdgeTable* edges =
      vtkAdjacencyMatrixToEdgeTable::New();
edges->SetInputConnection(source->GetOutputPort());

// Convert the edge table to a graph
vtkTableToGraph* graph = vtkTableToGraph::New();
graph->SetInputConnection(edges->GetOutputPort());
graph->AddLinkVertex("rows", “eid", false);
graph->AddLinkVertex("columns", “eid", false);
graph->AddLinkEdge("rows", "columns");
```

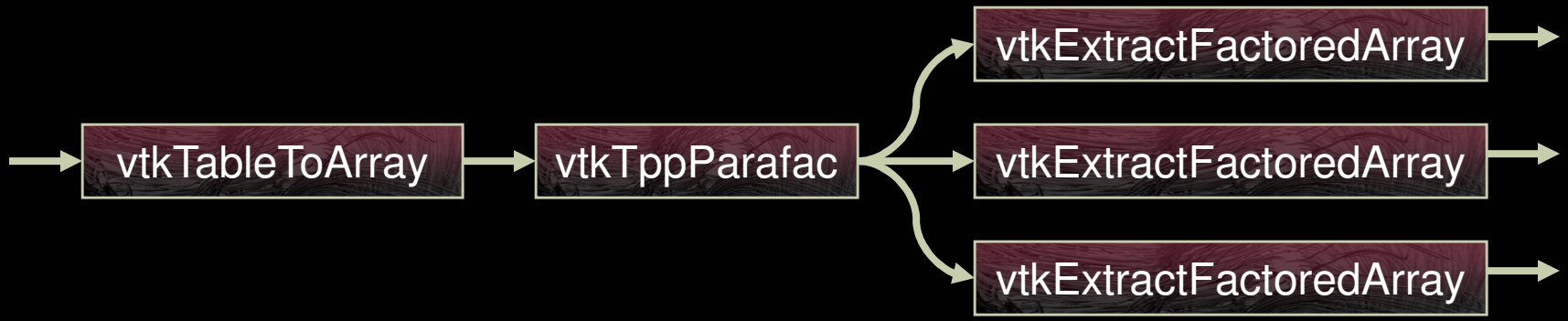


Tensor Creation

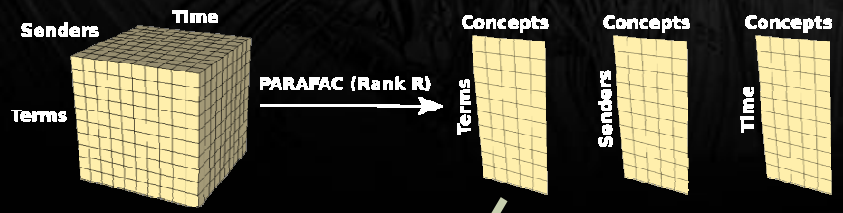


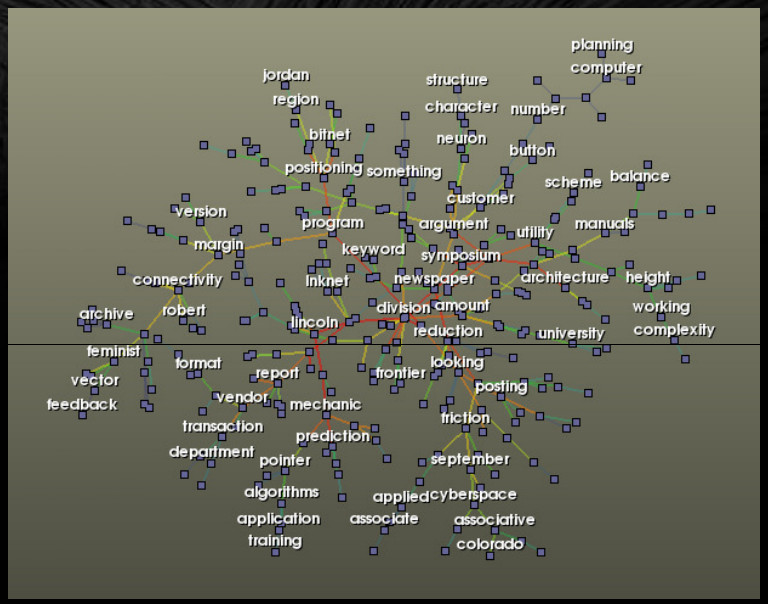
Tensor Reduction



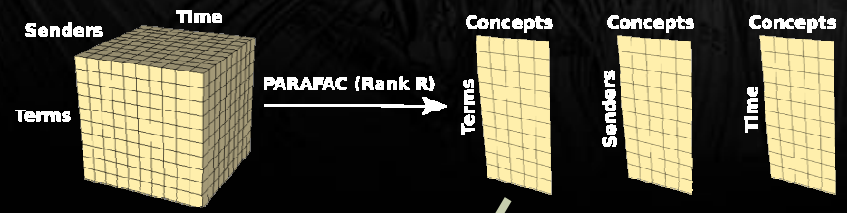


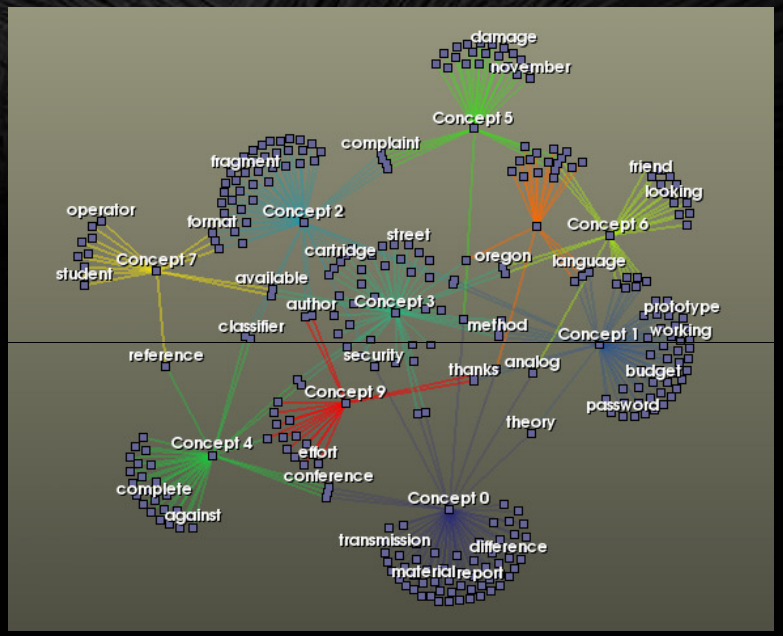
Term-term Similarity Graphs





Concept-term Similarity Graphs

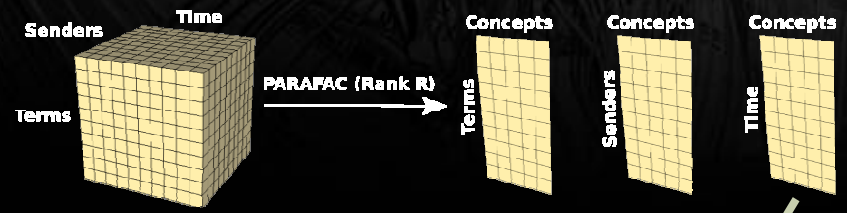


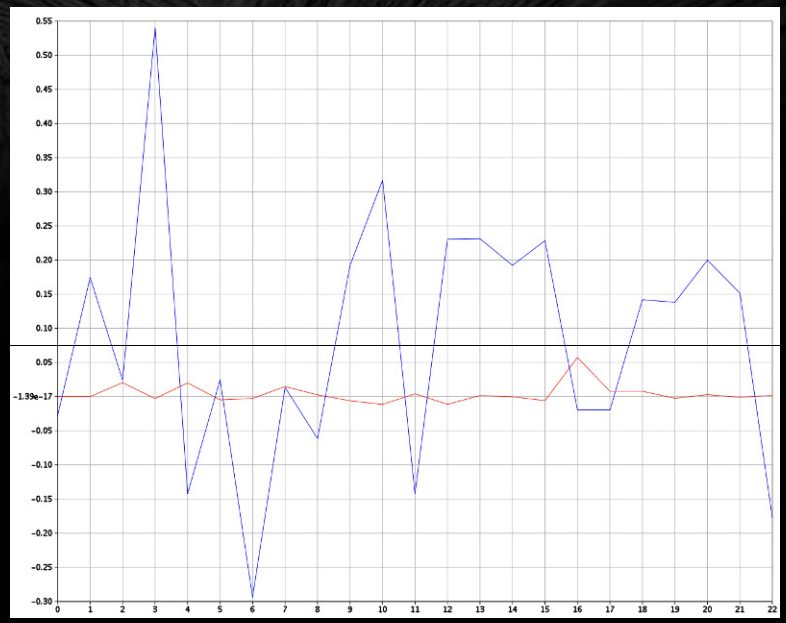


vtkExtractFactoredArray

vtkAdjacencyMatrixToEdgeTable → vtkTableToGraph → vtkGraphView

Concept-Activity Over Time





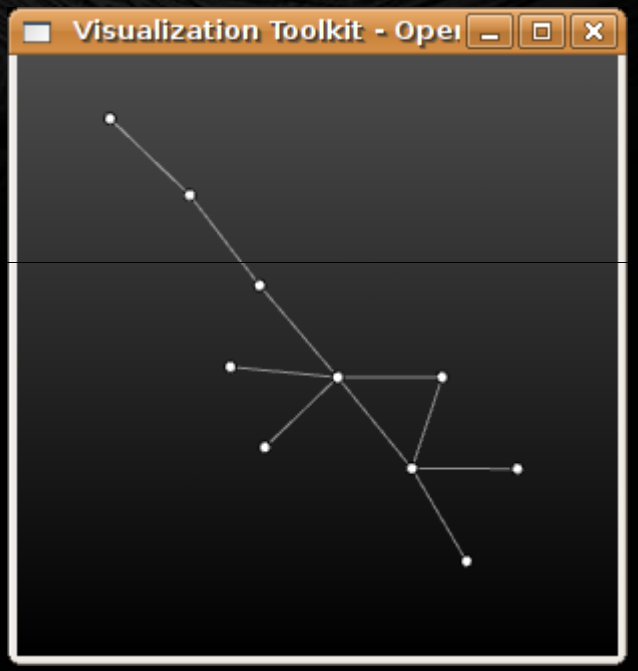
vtkExtractFactoredArray → vtkArrayToTable → vtkQtChartView

Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

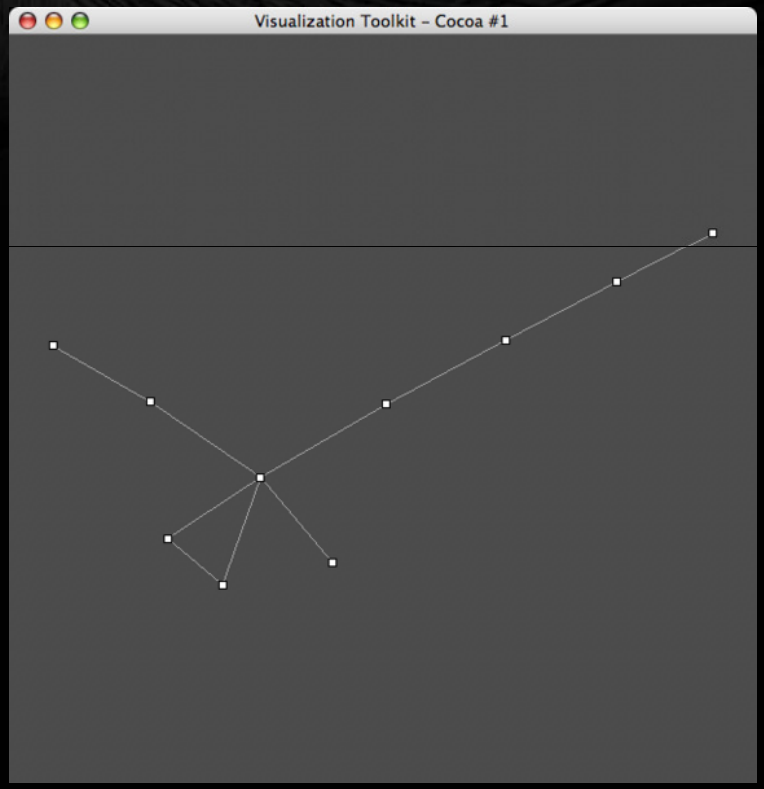
Titan Tcl/Tk Interface

```
package require vtk

vtkRandomGraphSource src

vtkGraphLayoutView view
view AddRepresentationFromInputConnection
  [src GetOutputPort]

vtkRenderWindow window
view SetupRenderWindow window
[window GetInteractor] Start

wm withdraw .
```



Titan Python Interface

```
from vtk import *

source = vtkRandomGraphSource()

view = vtkGraphLayoutView()
view.AddRepresentationFromInputConn
ection(source.GetOutputPort())

window = vtkRenderWindow()
window.SetSize(600, 600)
view.SetupRenderWindow(window)
window.GetInteractor().Start()
```

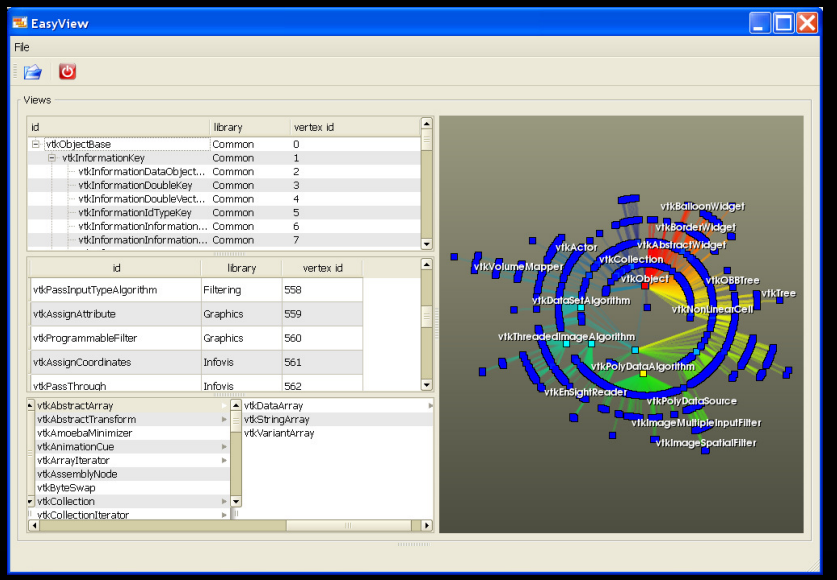


C++ Example Application

Qt has model/view classes for tables and trees *(specifically shown are QTableView, QTreeView, QColumnView).*

Code “clips” from VTK/Examples/Infovis/Cxx/EasyView

```
…
this->XMLReader    = vtkSmartPointer<vtkXMLTreeReader>::New();
this->TreeView     = vtkSmartPointer<vtkQtTreeView>::New();

// Set widget for the tree view
this->TreeView->SetItemView(this->ui->treeView);
…

// Create xml reader
this->XMLReader->SetFileName( fileName.toAscii() );
…

// Now hand off tree to the tree view
this->TreeView->SetRepresentationFromInputConnection(
    this->XMLReader->GetOutputPort());
…
```

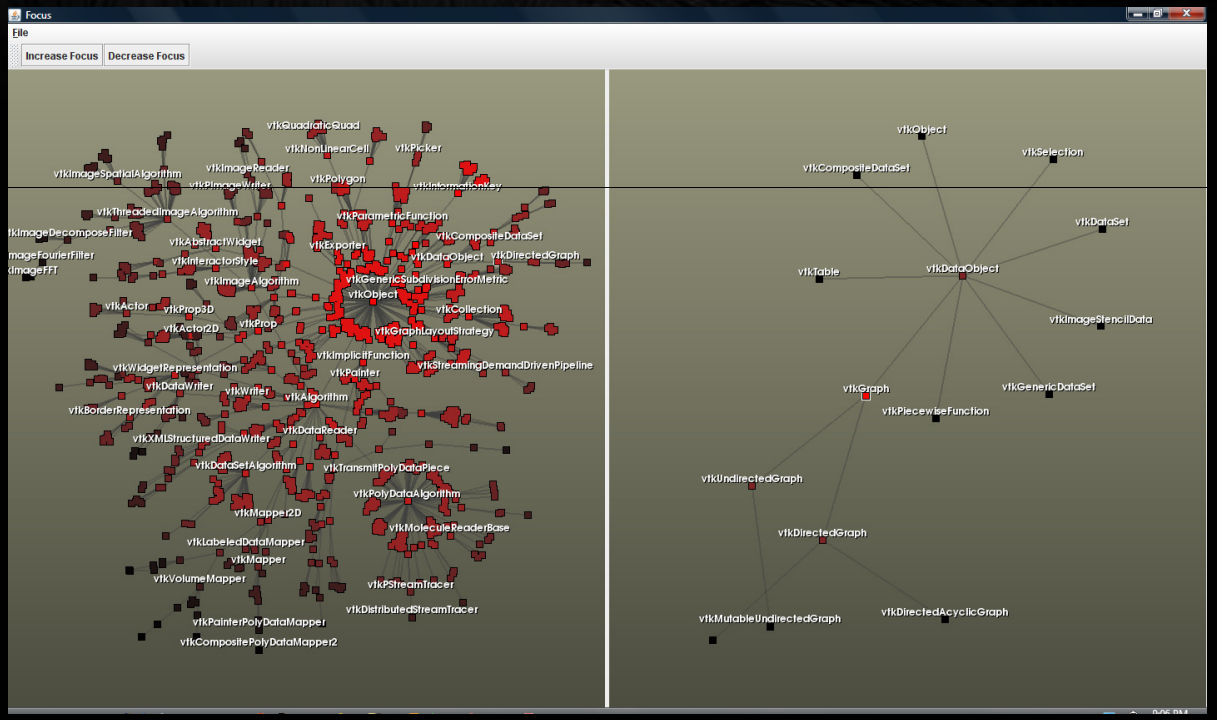


VTK/Examples/Infovis/Cxx/EasyView

Java Example Application

VTK/Examples/Infovis/Java/Focus.java

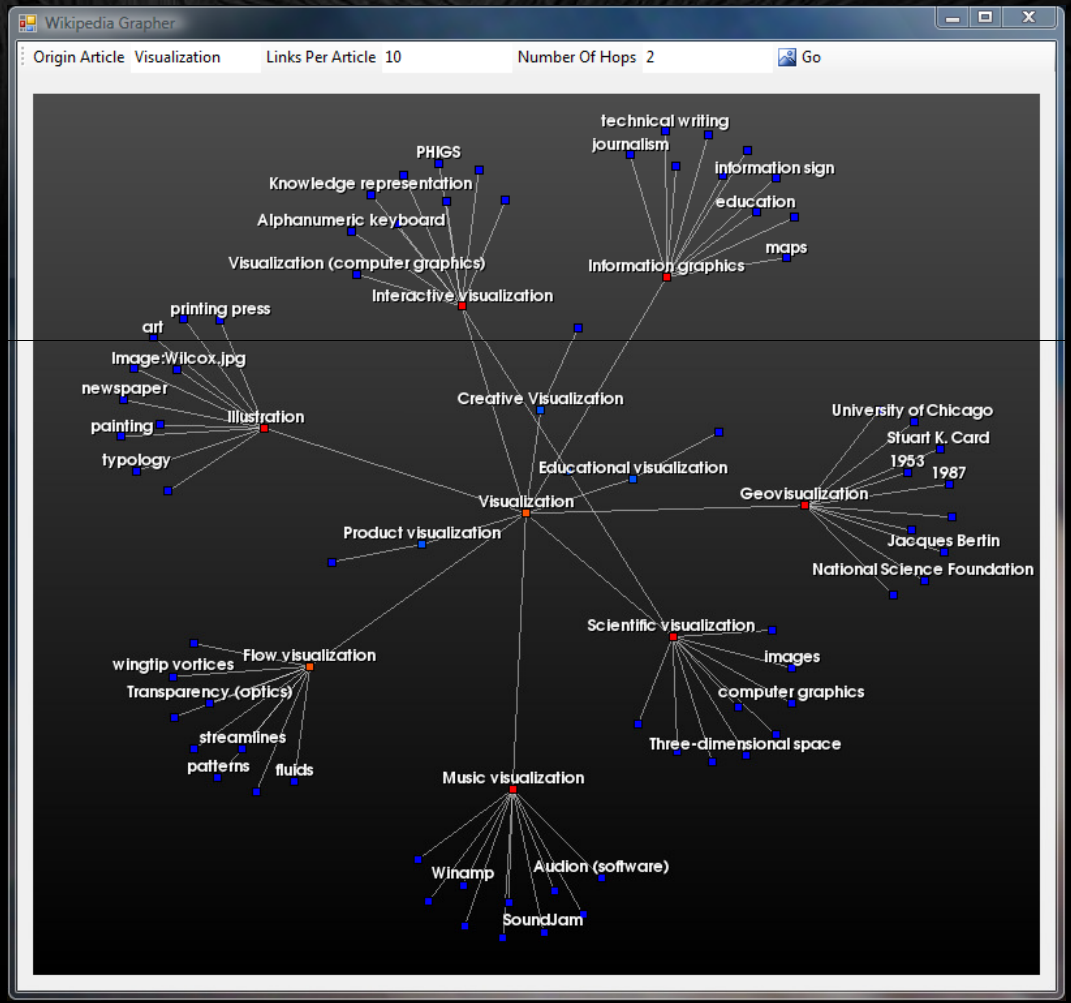
Display all data, along with focused selection using breadth first search. Uses Java Swing components.



.Net Example Application: Wikipedia Browsing (C#)

http://www.kitware.com/products/activiz.html

Application for browsing wikipedia connectivity using C# wrappers. Uses Windows GUI components.

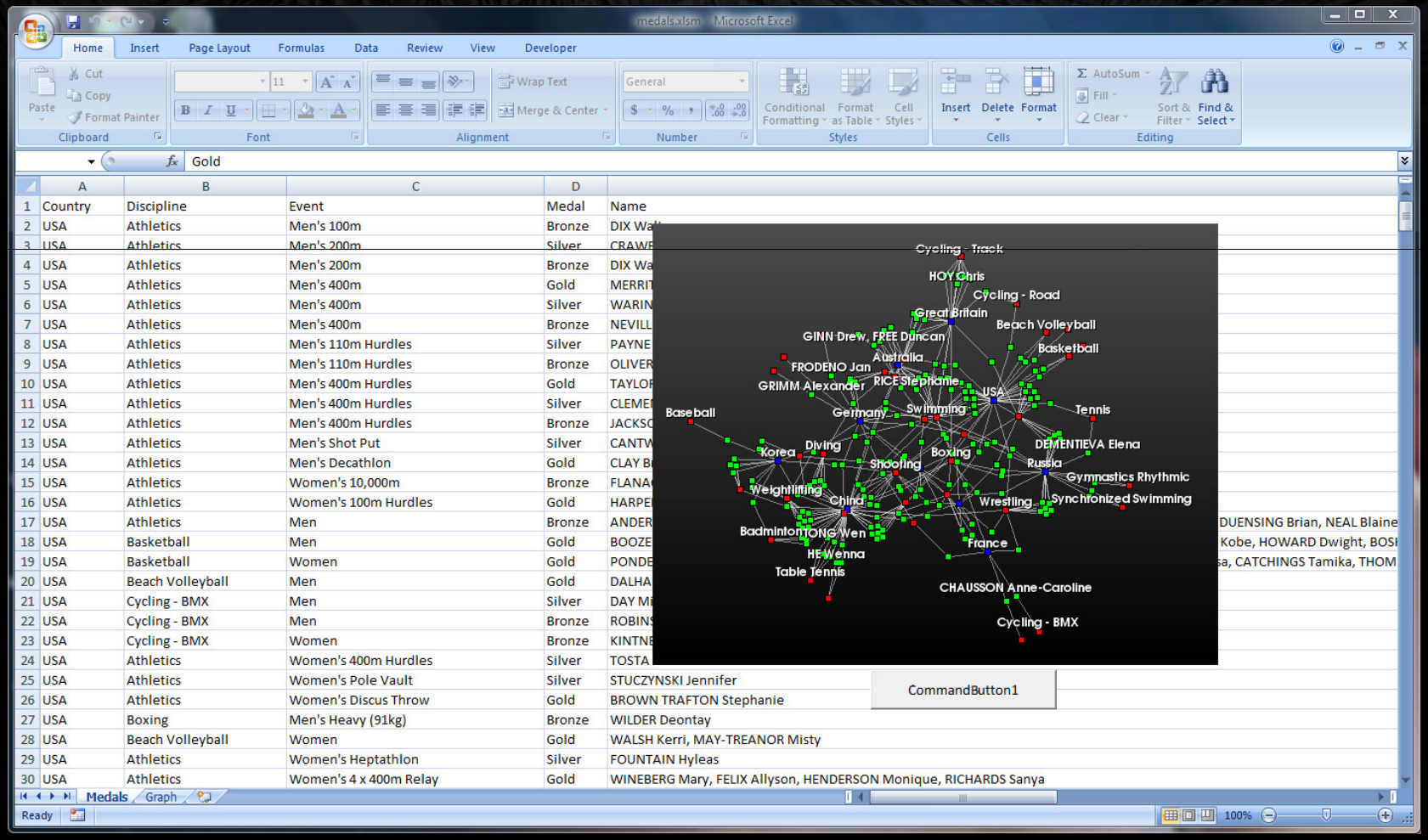


Wikipedia Browsing (C# code)


```
…
using Kitware.VTK;
…
private void addLinks(Kitware.VTK.vtkMutableDirectedGraph g,
    string lookupValue, int hops)
{
// Fetch XML from  Wikipedia
System.Net.HttpWebRequest webRequest = …
… // Parse XML to get links to other articles
// If the new vertex is not already there add it
int v = label.LookupValue(substring);
if (v < 0)
{
    v = g.AddVertex();
    label.InsertNextValue(substring);
}
}
```

Olympic Medals (Visual Basic embedded in Excel)

vtkRenderWindow COM ActiveX control shows connections between Countries, Athletes, and Events at Beijing 2008.



	A	B	C	D	E
1	Country	Discipline	Event	Medal	Name
2	USA	Athletics	Men's 100m	Bronze	DIX Wa
3	USA	Athletics	Men's 200m	Silver	CRAWF
4	USA	Athletics	Men's 200m	Bronze	DIX Wa
5	USA	Athletics	Men's 400m	Gold	MERRIT
6	USA	Athletics	Men's 400m	Silver	WARIN
7	USA	Athletics	Men's 400m	Bronze	NEVILL
8	USA	Athletics	Men's 110m Hurdles	Silver	PAYNE
9	USA	Athletics	Men's 110m Hurdles	Bronze	OLIVER
10	USA	Athletics	Men's 400m Hurdles	Gold	TAYLOF
11	USA	Athletics	Men's 400m Hurdles	Silver	CLEMEI
12	USA	Athletics	Men's 400m Hurdles	Bronze	JACKSO
13	USA	Athletics	Men's Shot Put	Silver	CANTW
14	USA	Athletics	Men's Decathlon	Gold	CLAY B
15	USA	Athletics	Women's 10,000m	Bronze	FLANA
16	USA	Athletics	Women's 100m Hurdles	Gold	HARPE
17	USA	Athletics	Men	Bronze	ANDER
18	USA	Basketball	Men	Gold	BOOZE
19	USA	Basketball	Women	Gold	PONDE
20	USA	Beach Volleyball	Men	Gold	DALHA
21	USA	Cycling - BMX	Men	Silver	DAY Mi
22	USA	Cycling - BMX	Men	Bronze	ROBIN
23	USA	Cycling - BMX	Women	Bronze	KINTN
24	USA	Athletics	Women's 400m Hurdles	Silver	TOSTA
25	USA	Athletics	Women's Pole Vault	Silver	STUCZYNSKI Jennifer
26	USA	Athletics	Women's Discus Throw	Gold	BROWN TRAFTON Stephanie
27	USA	Boxing	Men's Heavy (91kg)	Bronze	WILDER Deontay
28	USA	Beach Volleyball	Women	Gold	WALSH Kerri, MAY-TREANOR Misty
29	USA	Athletics	Women's Heptathlon	Silver	FOUNTAIN Hyleas
30	USA	Athletics	Women's 4 x 400m Relay	Gold	WINEBERG Mary, FELIX Allyson, HENDERSON Monique, RICHARDS Sanya

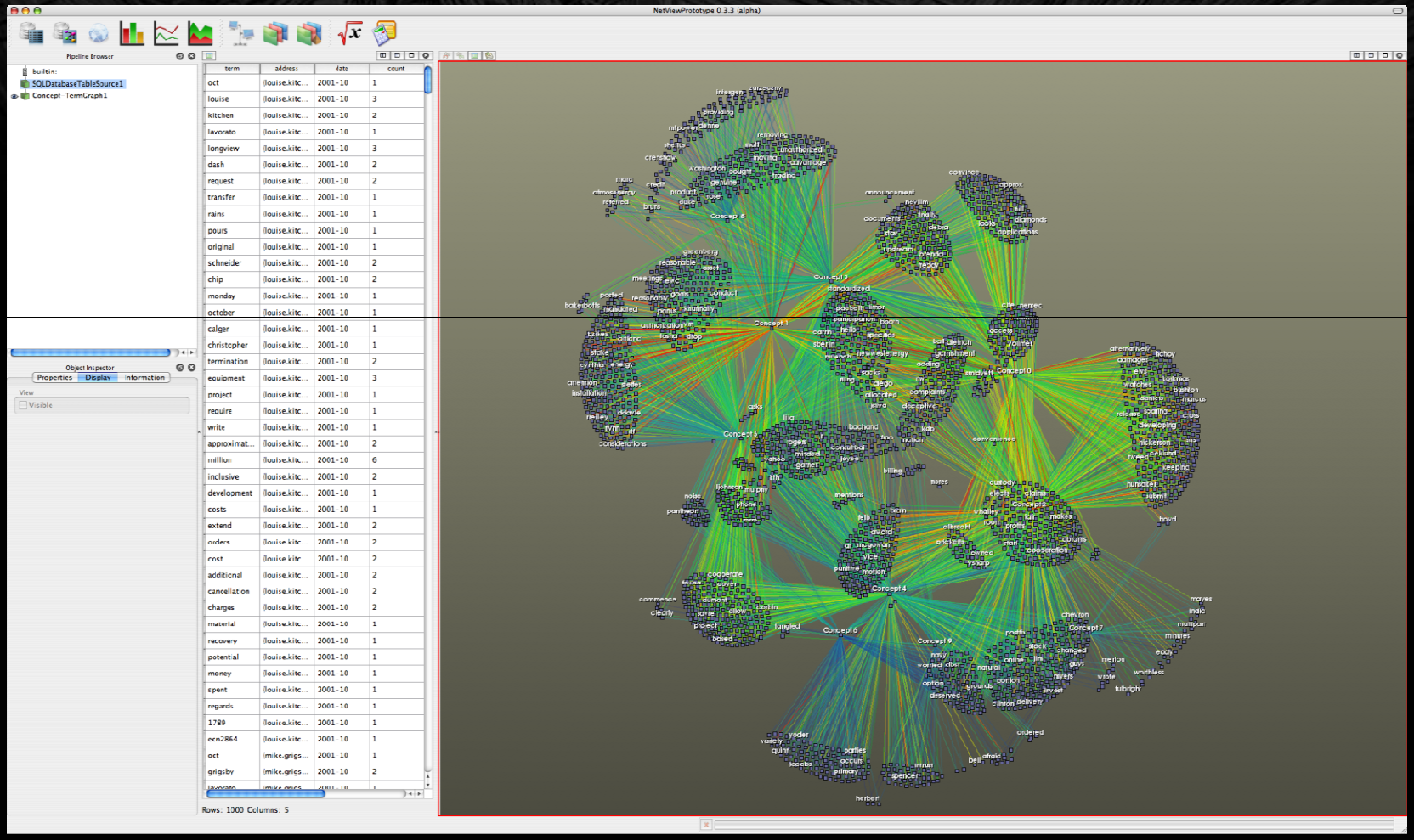
Olympic Medals (VB code)


```
Private Sub CommandButton1_Click()

… // Create a vtkTable by looking up Excel cells

… // Use vtkTableToGraph to make a graph

Set win = vtkRenderWindowControl1.GetRenderWindow

Set v = New vtkGraphLayoutView
v.AddRepresentationFromInput cat.GetOutput
v.SetupRenderWindow win
v.SetLayoutStrategyToSimple2D
v.SetVertexLabelArrayName "value"
v.VertexLabelVisibilityOn
v.SetVertexColorArrayName "category"
v.ColorVerticesOn
v.Update

End Sub
```

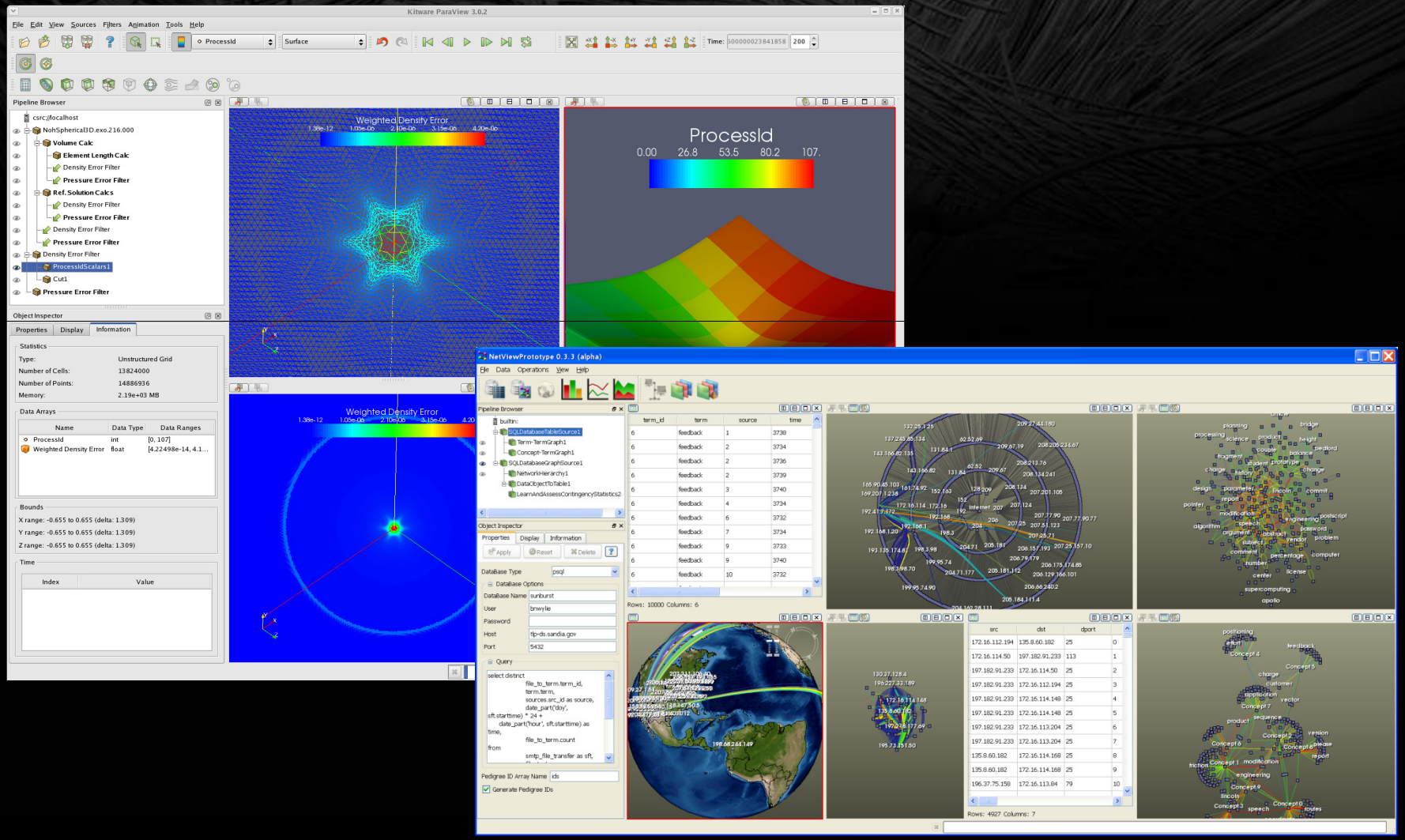
Tutorial Outline

- Introduction and Motivation (15 minutes)
  - Project Background and Scope
  - Use Case: Cyber Defense
- VTK InfoVis Data Structures (30 minutes)
  - Data Structures
    - Tables
    - Trees
    - Graphs
  - Database Access
  - Table, Tree and Graph Readers
- VTK InfoVis Components (45 minutes)
  - Data Conversions
  - Graph/Tree Layout Strategies
  - Qt Model Adapters
  - Views/Displays
  - Selection
  - Geographic Visualization
- *Short Break*
- Analysis Capabilities (50 minutes)
  - Graph Algorithms
  - Statistics
  - MATLAB interface
- Algebraic Methods (20 minutes)
- Building Applications (30 minutes)
  - Script Language Support
  - C++
  - Java
  - .Net/Com Interfaces
- OverView (30 minutes)
  - Application
  - Plugins

OverView - A general-purpose informatics tool

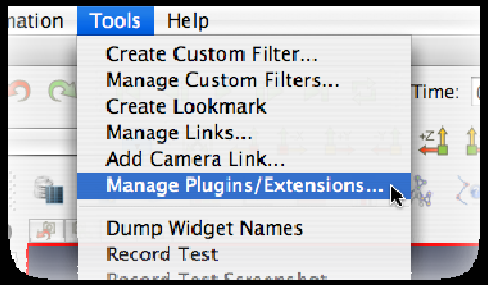


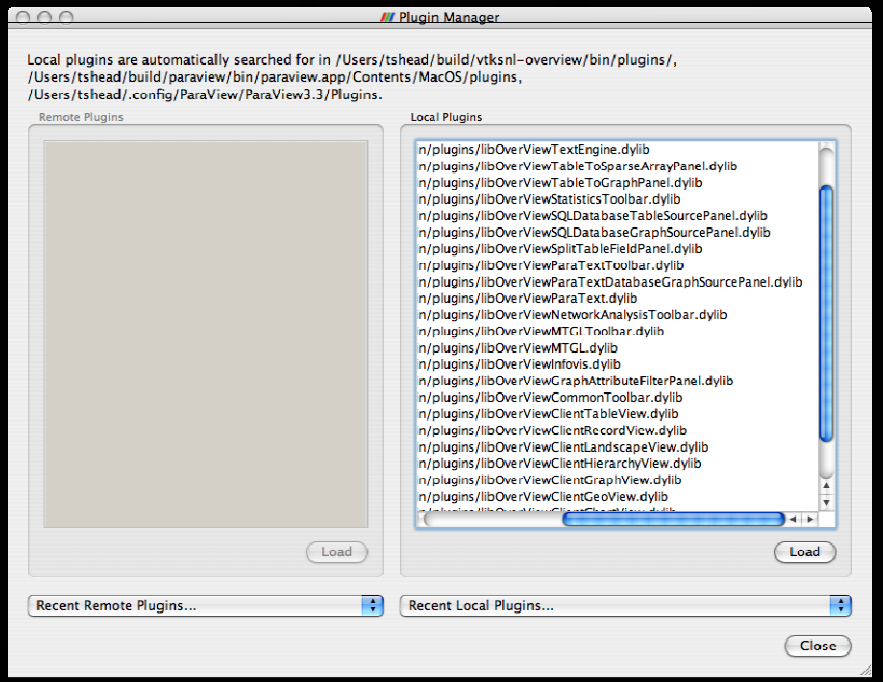
OverView - based on the ParaView architecture



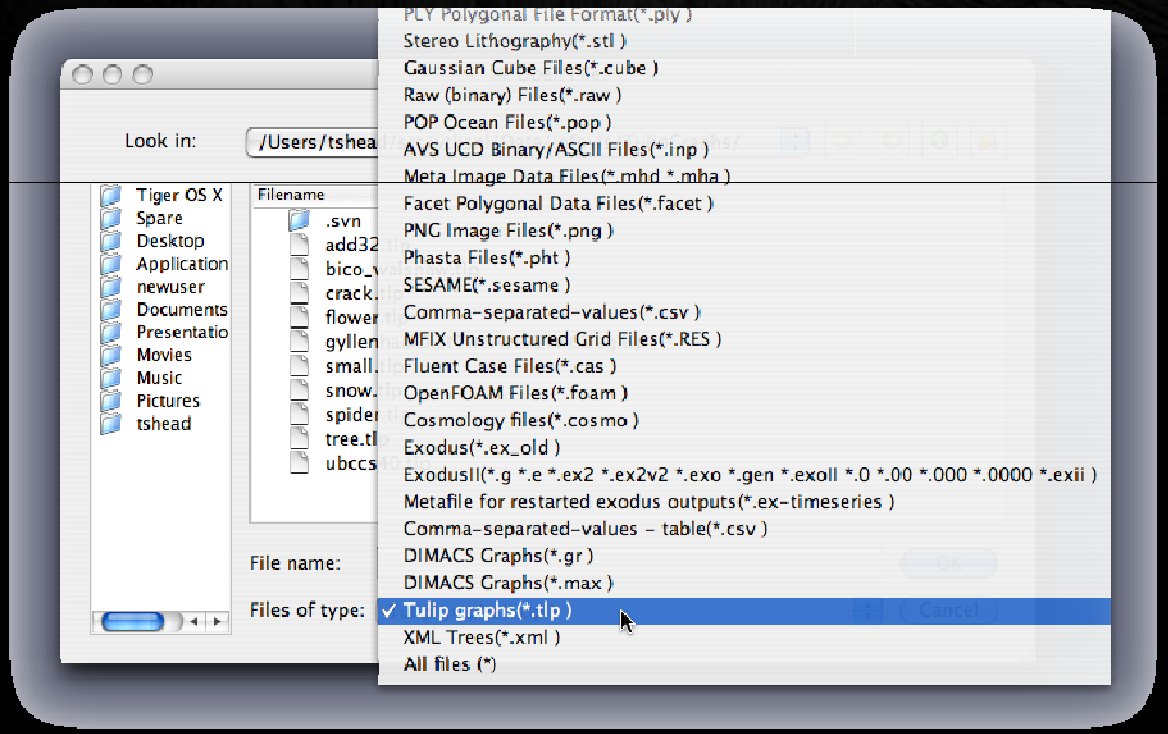
OverView Plugins

- Extends the collection of readers, writers, and filters at runtime.
- Shared libraries containing new filters are dynamically-linked into the working-set at runtime.
- Plugins can be loaded automatically at startup from known locations, locations specified via environment variable (PV_PLUGIN_PATH) or manually loaded via the plugin manager GUI:

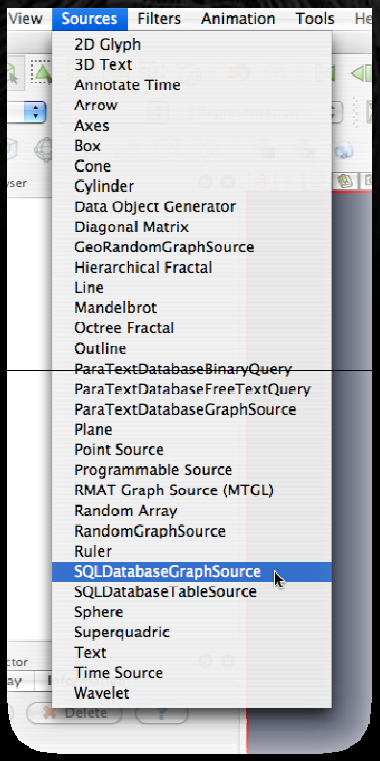


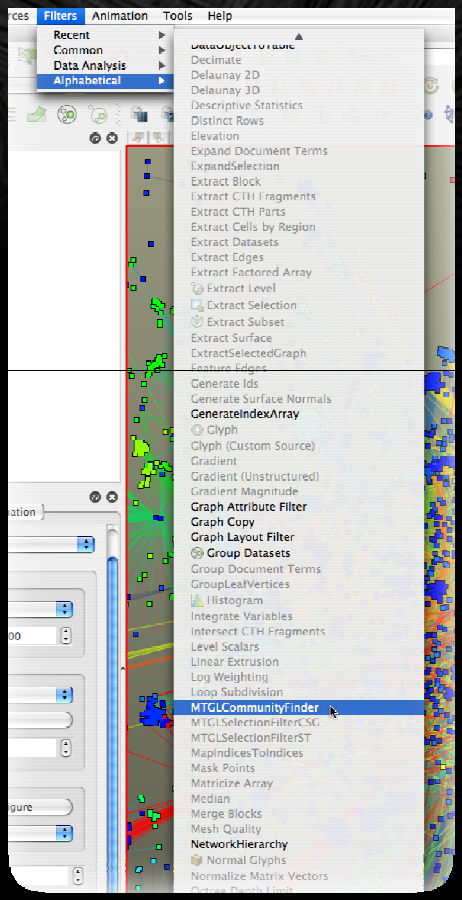


Plugin Types - Readers & Writers



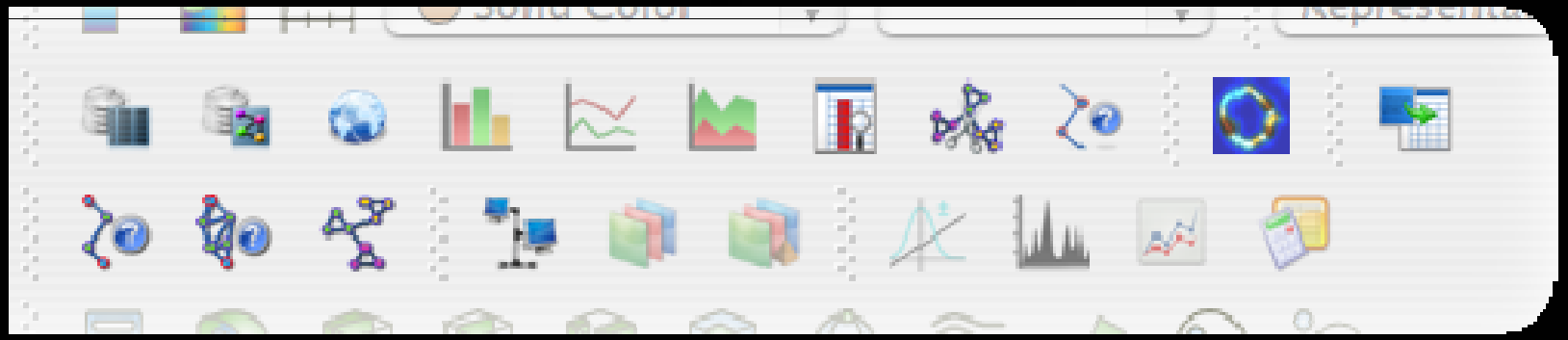
Plugin Types - General Filters





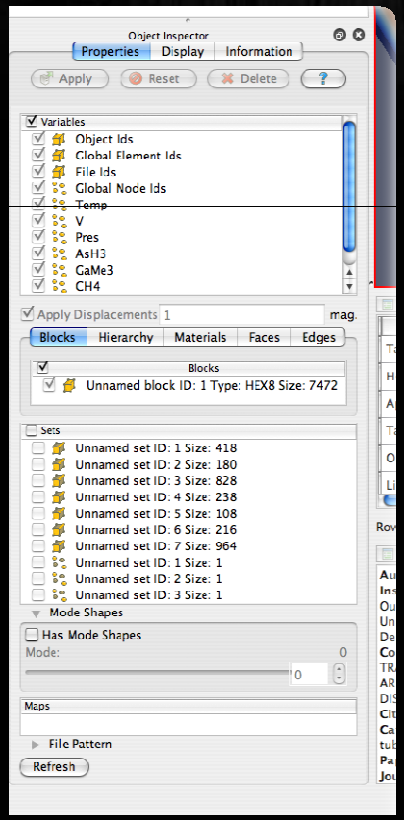
Plugin Types - Custom Toolbars

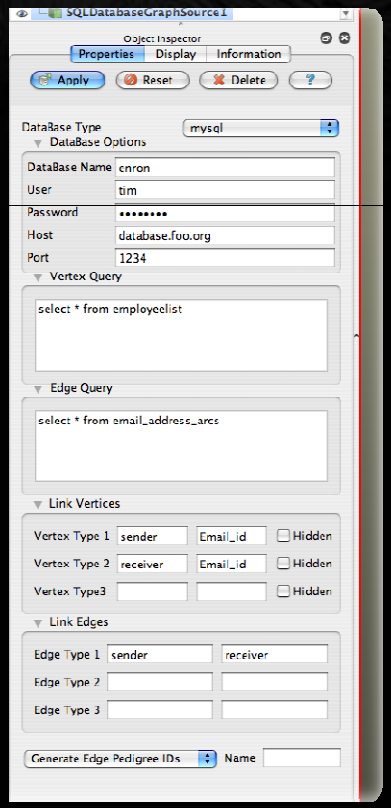
Useful for automating setup of a complex pipeline, an arbitrary view configuration, etc.



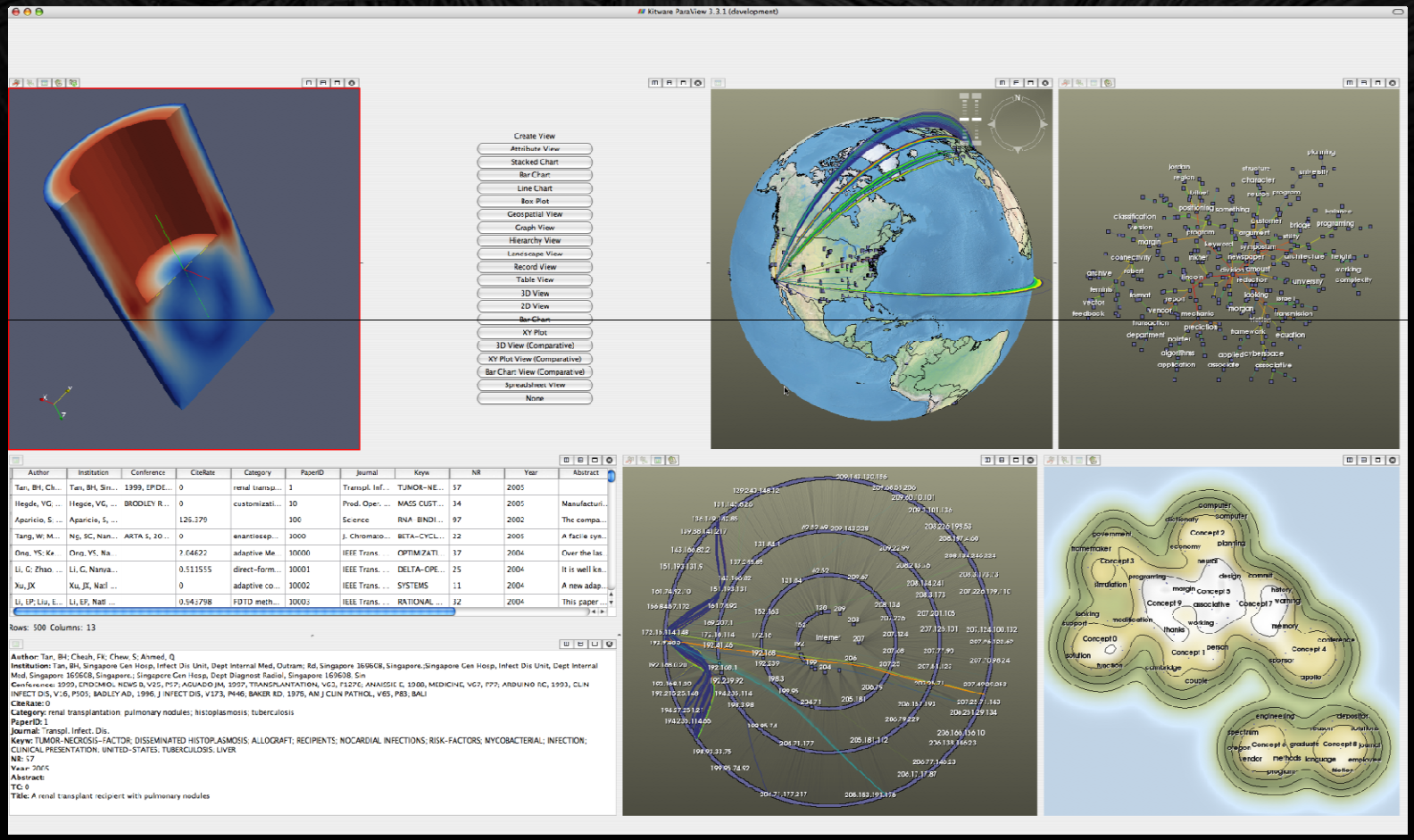
Plugin Types - Custom Panels

Provides a filter-specific user interface panel - useful with complex filters where the auto-generated GUI is insufficient.

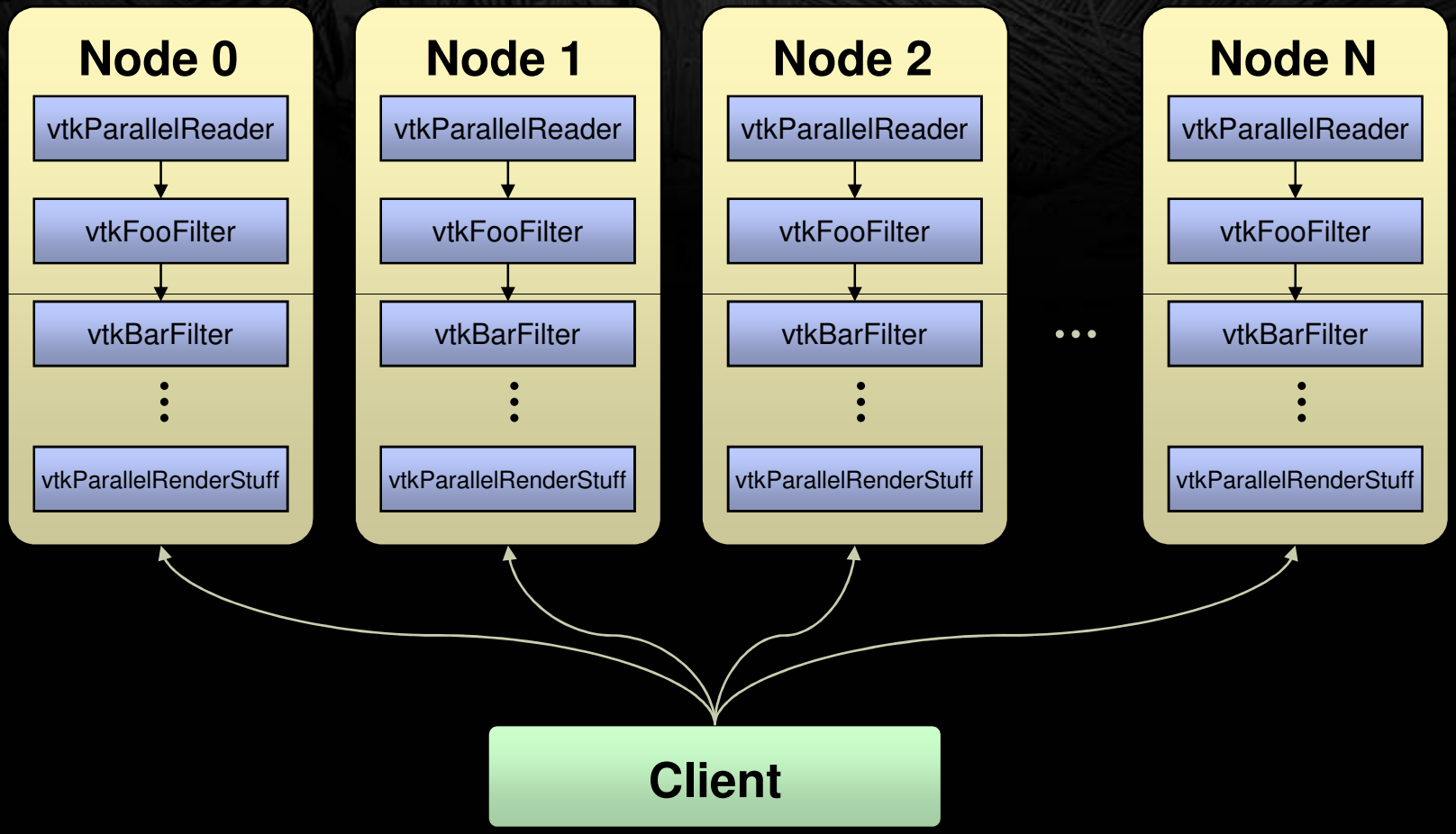




Plugin Types - Custom Views



Plugin Technical Challenges



Creating Plugins

- "Server Manager XML" is used to "wrap" a Titan filter so it can be used in OverView:

```
<ServerManagerConfiguration>
  <ProxyGroup name="filters">
    <SourceProxy name="FooFilter" class="vtkFooFilter">
      <InputProperty name="Input" … />
      <IntVectorProperty name="FooCount" … />
    </SourceProxy>
    <!-- More proxies in this group … -->
  </ProxyGroup>
  <!-- More groups in this plugin … -->
</ServerManagerConfiguration>
```

- The XML is linked into the plugin binary, and parsed by OverView when the plugin is loaded.
- The XML is used to auto-generate a graphical user interface for the plugin.

Sample Reader Plugin XML


```
<ServerManagerConfiguration>
  <ProxyGroup name="sources">
    <SourceProxy name="TulipReader" class="vtkTulipReader">
      <StringVectorProperty name="FileName"
        command="SetFileName" number_of_elements="1">
        <FileListDomain name="files"/>
      </StringVectorProperty>
      <Hints>
        <View type="ClientGraphView"/>
      </Hints>
    </SourceProxy>
  </ProxyGroup>
</ServerManagerConfiguration>
```

Useful Plugin References

- "Advanced ParaView Visualization" Tutorial, tomorrow!
- General Plugin Information
  - http://paraview.org/Wiki/Plugin_HowTo
  - The ParaView Guide, Version 3, chapter 19.
- Server Manager XML
  - The `ParaView/Servers/ServerManager/Resources/` directory.
  - The ParaView Guide, Version 3, section 18.6, pp 262 - 273.
- Sample Plugins
  - The ParaView/Plugins directory.
  - The ParaView/Examples/Plugins directory.

More Info…

Interested in Using?

Sandia website: www.sandia.gov/Titan and www.sandia.gov/OverView

Kitware Wiki: www.kitware.com/InfovisWiki

Source code: Download the VTK repository (instructions at www.vtk.org).

Questions/Issues: vtkusers@vtk.org

Interested in Contributing?

We are actively pursuing collaborators to join Sandia, Kitware and Indiana University in our efforts to grow and refine the capabilities.

Contacts:

Brian Wylie (bnwylie@sandia.gov)

Timothy Shead (tshead@sandia.gov)

Jeff Baumes (jeff.baumes@kitware.com)

End

Questions/Comments?

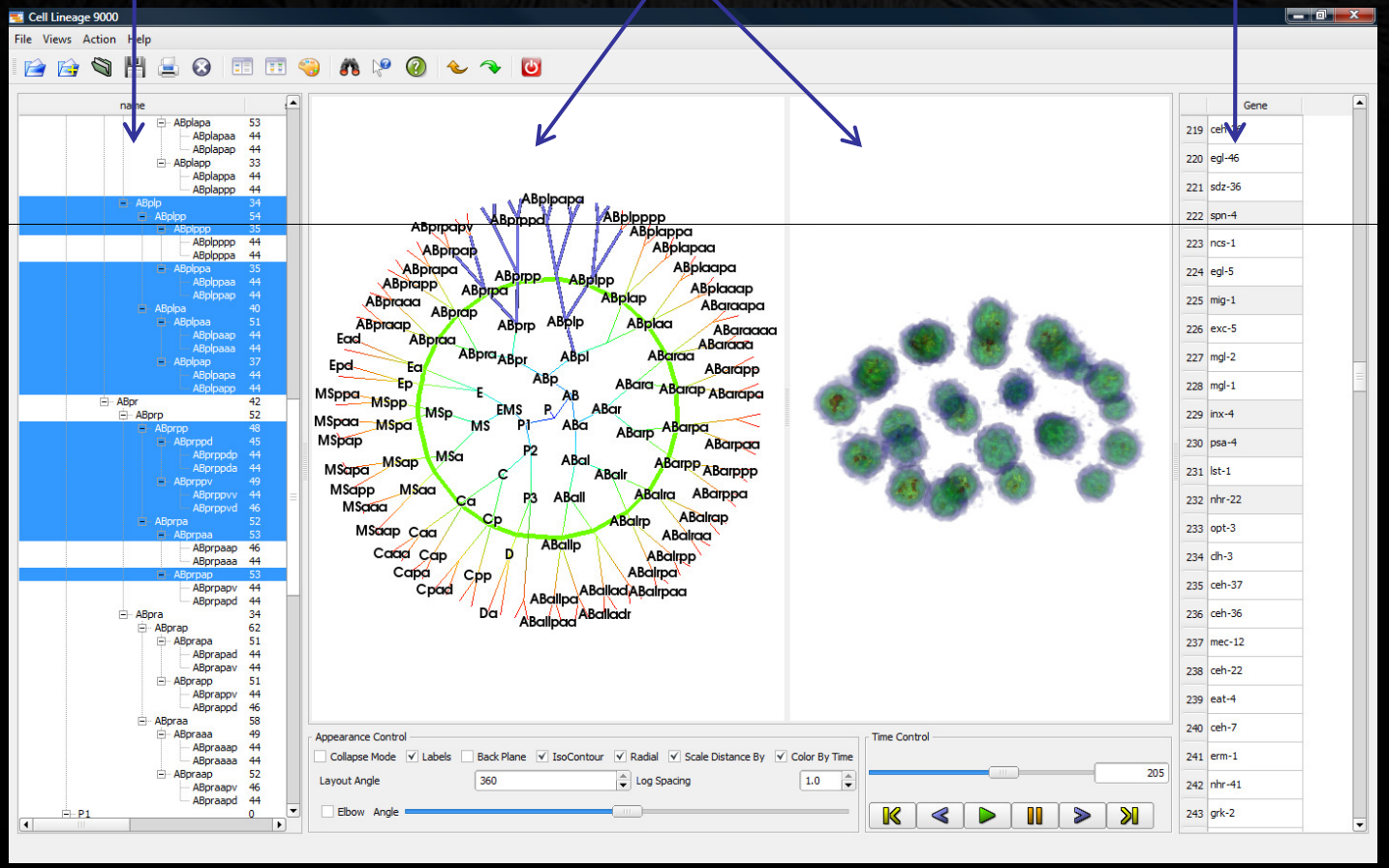
Storage Slides

Qt Interface

QTreeView

QVTKWidget

QTableView



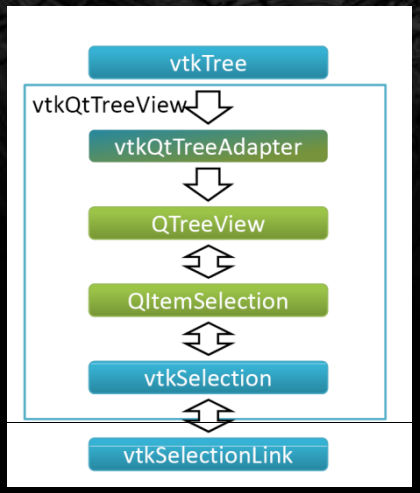
Motivation: Project Goals

Unified Toolkit: Scientific Visualization and Information Visualization together at last!

Scalable Toolkit: Sandia's use of VTK/ParaView has provided scalability on some of the world's largest simulation results *(Billions of cells/Terabytes on disk).*

Flexible Toolkit: Component based pipeline architecture provides a development model that allows expansion, agility and domain specific application construction.

Qt Adapters

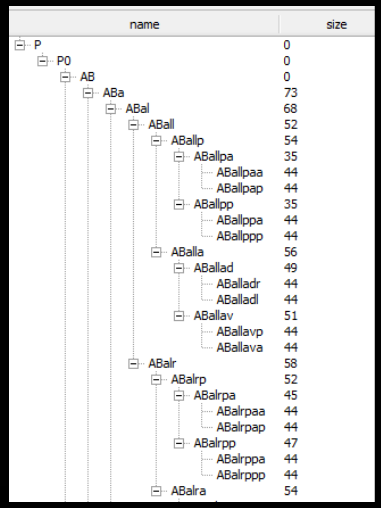


The Titan toolkit can be used by any 'front-end' GUI (TCL/TK, Java, Python and Qt).

Specific adapters have be written for vtkTree and vtkTable that implement the QtAbstractItemModel interface.

title	year	release_date	num_ratings	average_rating
"18 Wheels of Justice" (2000)	2000	211814438400000	14	4.8
"24: Conspiracy" (2005)	2005	208657814400000	8	4.4
"29 Minutes & Counting" (2004)	2004	208657814400000	0	0
"2gether: The Series" (2000)	2000	211833100800000	17	5.5
"30 Days 'Til I'm Famous" (2006)	2006	208657814400000	0	0
"30 by 30: Kid Flicks" (2001)	2001	208657814400000	0	0
"411, The" (2005)	2005	212006592000000	0	0
"70's House, The" (2005)	2005	211987324800000	12	5.5
"8th & Ocean" (2006)	2006	212008492800000	89	4.9
"A.T.M.: A toda M." (2005)	2005	211989139200000	0	0
"A.U.S.A." (2003)	2003	211911120000000	0	0
"AMC Project, The" (2003)	2003	208657814400000	0	0
"AXN Action TV" (2000)	2000	208657814400000	0	0
"Aardvark" (2000)	2000	211815129600000	0	0
"Abby" (2003)	2003	211908614400000	0	0

vtkQtTableItemModelAdapter



vtkQtTreeItemModelAdapter