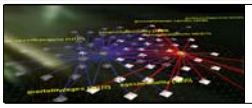
Information Visualization Learning Modules

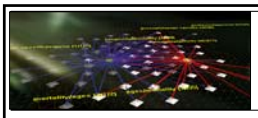
Katy Börner
Information Visualization Lab
School of Library and Information Science
INDIANA UNIVERSITY
BLOOMINGTON

SBC Fellows Forum, May 21st, 2004



Overview

- **The Need for Learning Modules to Teach Information Visualization**
  - Information Visualization Research and Praxis
  - Desirable Teaching Style
- **InfoVis Toolkit**
- **InfoVis Learning Modules**
  - Design
  - Usage
- **Validation: Teaching InfoVis using the Learning Modules**
- **Discussion & Future Work**

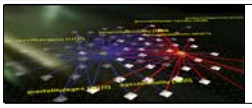


The Need for Learning Modules: Information Visualization Research and Education

Information Visualization (IV) combines aspects of scientific visualization, human-computer interaction, data mining, imaging, and graphics techniques, etc. to transform data that is not inherently spatial (e.g., document collections, network traffic logs, customer behavior, etc.) into a visual form.

Well designed visualizations reduce visual search time, improve understanding of complex data sets, reveal relations otherwise not noticed, enable data sets to be seen from several perspectives simultaneously, facilitate hypothesis formulation, and are effective sources of communication.

There exist a number of excellent textbooks that can be used to teach IV. Several come with accompanying web sites containing screen-sized snapshots of user interfaces as well as animations and movies. However, there exists no toolkit or learning resource that facilitates the exploration, application, evaluation, and comparison of algorithms.



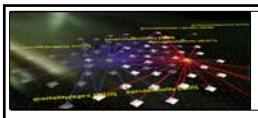
The Need for Learning Modules: Desirable Teaching Style

Since Spring 2001, Börner has been teaching the L579 Information Visualization course at the School of Library and Information Science at Indiana University. The course comprises lecture and lab sections as well as project work.

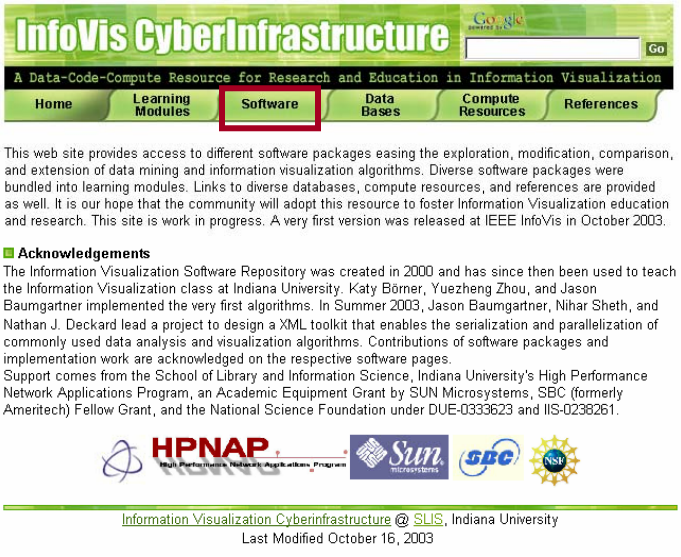
Lectures equip students with working knowledge about visual perception principles, theoretical approaches to IV design, a variety of existing data mining and visualization techniques, algorithms, and systems.

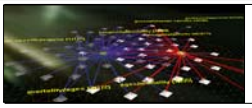
During lab, students run, discuss, and evaluate different information visualizations and gain hands-on experience with diverse IV algorithms.

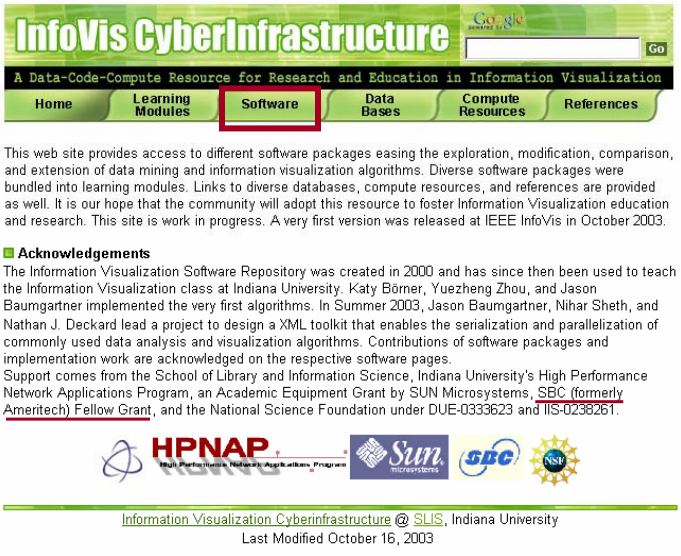
In project work, they constructively apply their knowledge to design novel IVs and develop skills in critiquing and evaluating visualization techniques.

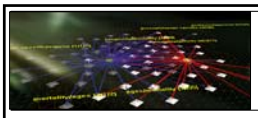


The InfoVis Toolkit









InfoVis Toolkit

The Team

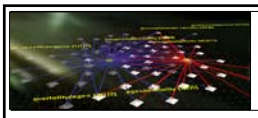
Master Minds/Programmers
Jason Baumgartner, SLIS
Nathan James Deckard, CS
Nihar Sheth, Informatics
Bruce William Herr, CS
Shashikant Penumarthy, CS/SLIS

Graphic Design
Caroline Courtney, Fine Art

Algorithm Development and Integration
Nihar Sanghvi, Informatics
Ning Yu, SLIS
Renee LeBeau, SLIS
Sidharth Thakur, CS
Sriram Raghuraman, Informatics
Todd Holloway, CS
Vivek Agrawal, Summer Intern
Yuezheng Zhou, CS

Project Start

2001



InfoVis Toolkit

Software

- XML Toolkit
- Preprocessing
- Data Mining
- Layout Algorithms
- Social Visualizations
- Interaction Algorithms
- Other Resources

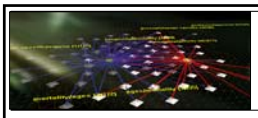
This page provides pointers to commonly used data analysis and visualization algorithms. An XML Toolkit was implement to facilitate the efficient visualization of diverse data sets as well as an easy comparison of visualizations generated by different algorithms. The toolkit provides a unified architecture in which algorithms can be easily incorporated. Many software packages are available in Java and hence can be run on any platform that supports Java 1.4.

Most software packages come with

- Algorithm Description
- Pros & Cons
- Sample Applications
- Implementation Details
- Usage Hints
- References
- Acknowledgements

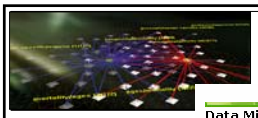
Web Site: http://iv.slis.indiana.edu/sw

Jason Baumgartner, Katy Börner, Nathan J. Deckard, Nihar Sheth. An XML Toolkit for an Information Visualization Software Repository. Poster Compendium, IEEE Information Visualization Conference, pp. 72-73, 2003.



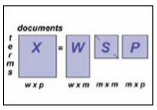
Preprocessing

- Parsers & Converters
- Stop Word Removal
- Porter Stemming Algorithm
- NICE stemmer



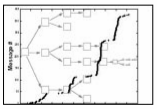
Data Mining

Vector Space Model
Developed by Gerhard Salton
Soon to be in the XML Toolkit

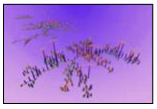


Latent Semantic Analysis
Developed by Tom K. Landauer and Susan Dumais
Code in XML Toolkit
Original code by Michael Berry

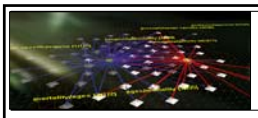
Topics Model
Developed by Tom Griffith & Mark Steyvers
Soon to be available via the XML Toolkit

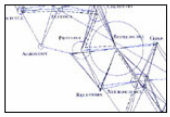


Burst Detection
Developed by Jon Kleinberg
Soon to be available via in XML Toolkit

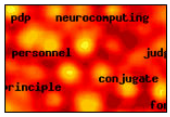


Pathfinder Network Scaling
Developed by Roger Schvaneveldt
KNOT Tools for Pathfinder Network Analysis are available via Interlink Inc.

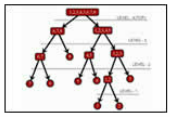




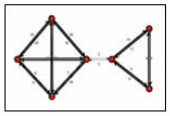
Multidimensional Scaling
Developed by Roger N. Shepard
Fast non-linear MDS algorithm by Matthew Chalmers and Alistair Morrison will soon to be available in the XML Toolkit



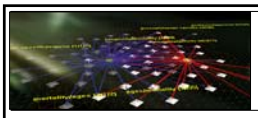
Self Organizing Maps
Developed by Teuvo Kohonen_
Original code from the WEBSOM research group



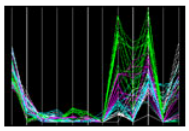
Clustering: Ward's Algorithm
Developed by Ward
Code in XML Toolkit



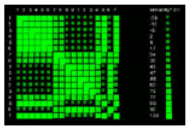
Clustering: Betweenness Centrality
Developed by Ulrik Brandes
Code in XML Toolkit



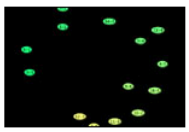
Layout Algorithms



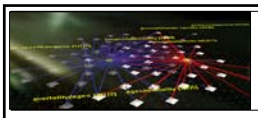
Parallel Coordinates
Developed by A. Inselberg
Soon to be available via in XML Toolkit

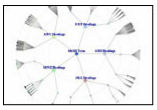


SimVis
Developed by Yuezheng Zhou
Code in XML Toolkit

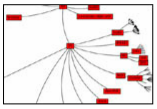


Spring Embedding Algorithm
Developed by Eades
Code in XML Toolkit

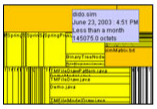




Radial Tree
Code in XML Toolkit

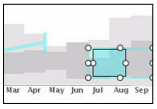


Hyperbolic Tree
Code in XML Toolkit

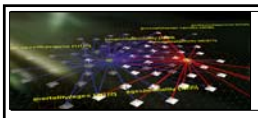


Treemap
Development led by Catherine Plaisant and Ben Shneiderman
Code in XML Toolkit
Treemap code available via HCIL@UMD

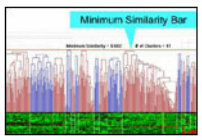
GRIDL
Designed and developed by Anne Rose, David Feldman, and Ben Shneiderman, with software improvements by Harry Hochheiser
Code in XML Toolkit



Time Searcher
Developed by Harry Hochheiser and Ben B. Shneiderman
Original code available via HCIL@UMD

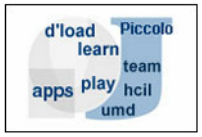


Interaction Algorithms

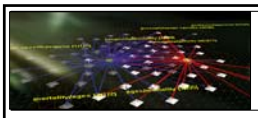


Hierarchical Clustering Explorer
Developed by Jinwook Seo and Ben Shneiderman
Original code available via HCIL@UMD

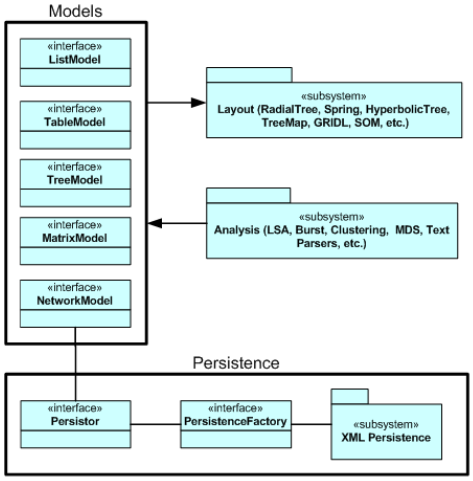
Fisheye Menus
Developed by Ben Bederson
Original code available via HCIL@UMD



Piccolo
Developed by Jesse Grosjean and Ben Bederson
Original code available via HCIL@UMD



InfoVis Toolkit Architecture



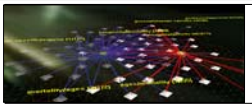
ANALYSIS ALGORITHMS
generate models from parsing other data structures and/or processing on the data

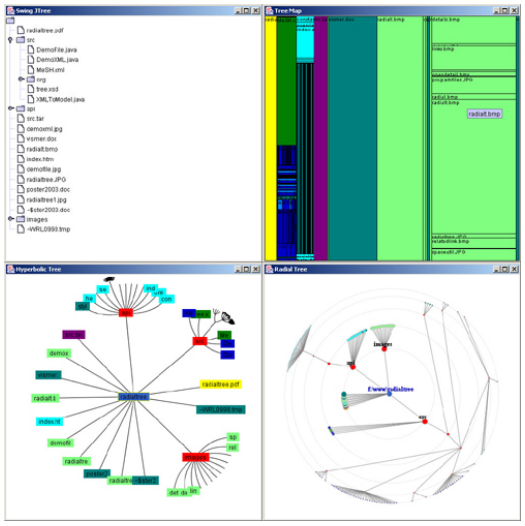
LAYOUT ALGORITHMS
run graphical processes on the appropriate model

DATA MODEL PERSISTENCE
factory to persist a model to a particular data store (i.e. XML format, database)

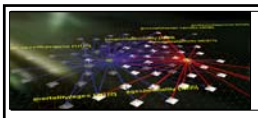
STANDARD MODEL INTERFACES
based on Java 2 Swing standard models

CODE INTEGRATION
new algorithms can be easily integrated by supporting one or more of the models

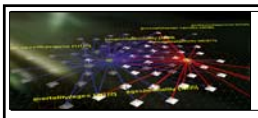




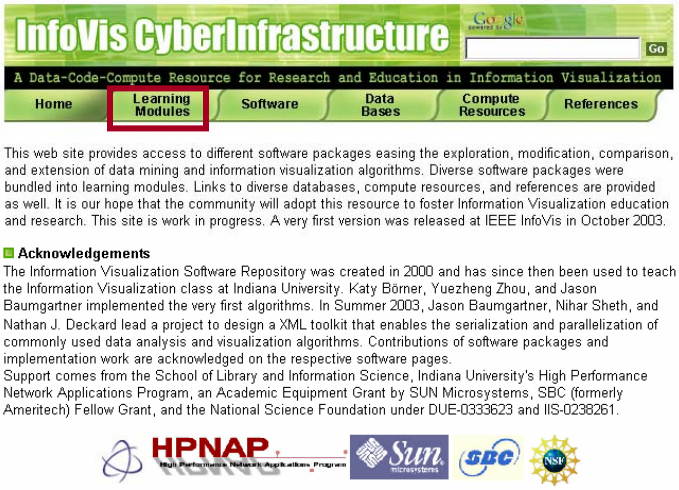
- Framework can run different data analysis and IV algorithms on a standard set of input data formats (tree, matrix, network, table, list).
- Models from the algorithms can be serialized through the persistence layer; and it is generic enough for plugging in various persistence options (XML, SQL database, etc).
- Based on Model-View-Controller (MVC) by focusing on standard data model interfaces for data exchange.

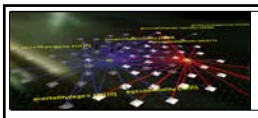


Demo InfoVis Toolkit



InfoVis Learning Modules





InfoVis Learning Modules: Design

InfoVis CyberInfrastructure

A Data-Code-Compute Resource for Research and Education in Information Visualization

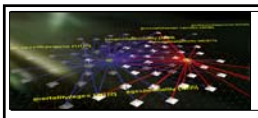
Home | Learning Modules | Software | Data Bases | Compute Resources | References

Learning Modules

Most information visualizations are highly interactive. While a number of excellent textbooks exist, the two-dimensional printouts on paper often cannot convey their true visual appearance and interactive performance. Several textbooks come with accompanying web sites that contain snapshots of user interfaces as well as animations and movies. However, none of them facilitates the exploration, application, evaluation, and comparison of algorithms.

This web page will provide access to a number of learning modules. Each learning module comes with an:

- Description of the data analysis and visualization task
- Usage hints on how to run and use a particular algorithm or tool
- Learning task - a challenging scenario to use an algorithm or to analyze and/or visualize a data set.
- Discussion of the results, and
- References to research papers, online demos, (commercial) applications.
- Acknowledgements



Visualizing Tree Data

http://iv.slis.indiana.edu/lm/lm-trees.html

InfoVis CyberInfrastructure

A Data-Code-Compute Resource for Research and Education in Information Visualization

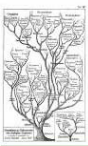
Home | Learning Modules | Software | Data Bases | Compute Resources | References

Learning Modules > Visualizing Tree Data

Description | Usage Hints | Learning Task | Discussion | References | Acknowledgments

Description

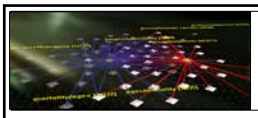
Many data sets come in tree format. There are family trees, organizational charts, classification hierarchies, and directory structures. The figure below shows an inheritance tree by Ernst Haeckel ('Stammbaum' in German). Read also To Draw a Tree by Pat Hanrahan.



Click image for larger version

A tree graph is a set of straight line segments (edges) connected at their ends containing no closed loops (cycles). You can also call it a simple, undirected, connected, acyclic graph (or, equivalently, a connected forest). A tree with n nodes has n-1 graph edges. All trees are bipartite graphs.

Many trees have a root node and are called rooted trees. Trees without a root node are called free trees. Subsequently, we will only consider rooted trees. In rootes trees, all nodes except the root node have only one parent node. Nodes which have no children are called leave nodes. All other nodes are referred to as intermediate nodes.



Student's Project Results

Visualizing and Evaluation of Tree Data Layouts

- Visualizing the structure of IU's Decision Support System
- Visualizing the co-occurences of keywords in DLib Magazine articles.
- Visualization of the Java API
- Visualizing the Library of Congress Classification System to retrieve legal materials in a library.

See Handin pages at
http://ella.slis.indiana.edu/~katy/handin/L579-S04/cgi/handinlogin.cgi

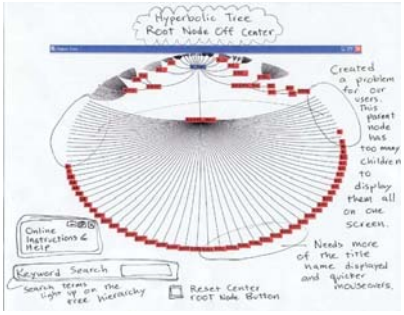
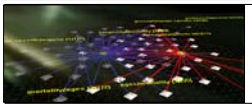


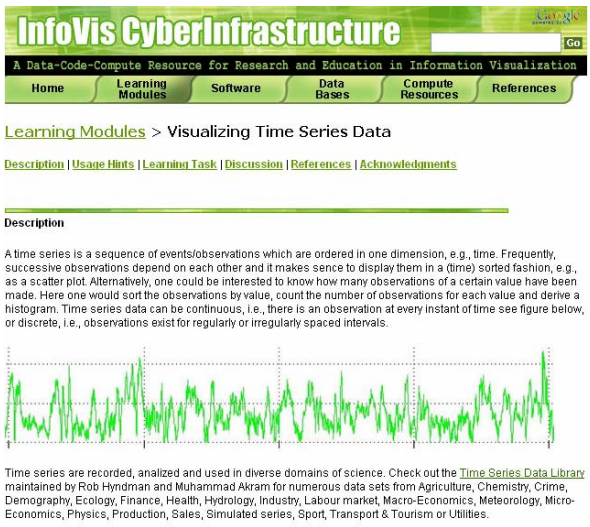
Image by Peter Hook and Rongke Gao

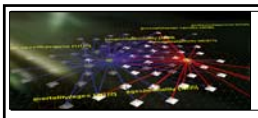


Validation: Teaching InfoVis using the Learning Modules

Time Series Analysis & Visualization

http://iv.slis.indiana.edu/lm/lm-time-series.html





Student's Project Results

Time Series Analysis & Visualization

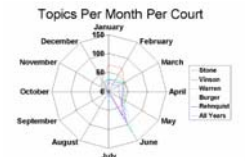
- Using Timesearcher and the Burst Detection Algorithm to Analyze the Stock Market from 1925 to 1945
- Applying Burst and TimeSearcher to Chat Data
- Lab Access Trends
- Quest Atlantis Chat Log Data

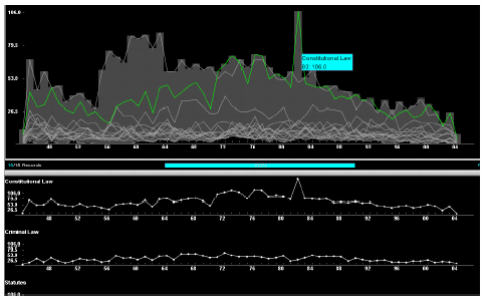
See Handin pages at
http://ella.slis.indiana.edu/~katy/handin/L579-S04/cgi/handinlogin.cgi

SBC Fellows Forum, May 21st, 2004

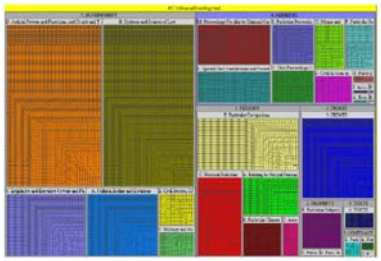
Visualizing the Work of the United States Supreme Court Based on Time Data and Top Level West Topics

by Peter A. Hook & Rongke Gao

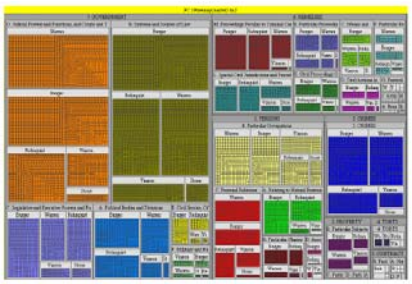




Top fifteen most occurring topics from 1944 to 2004 in Timesearcher



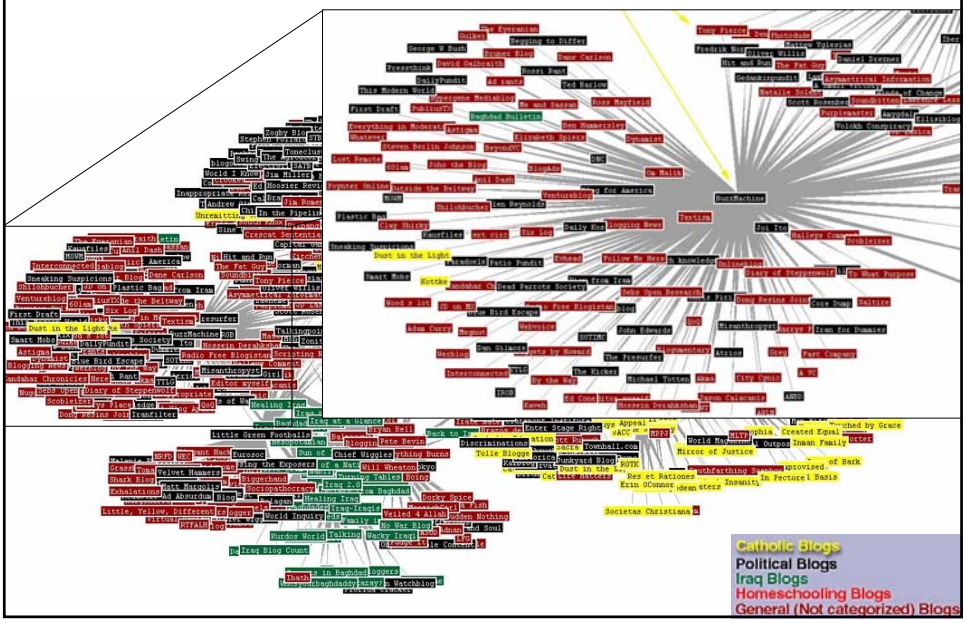
All topics grouped by West Category and Sub-Category grouped over the entire lengths of the data set

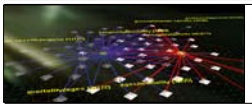


All topics by West Category and Sub-Category grouped corresponding to the five chief justices

Visualizing Niches of the Blog Universe

BY Mike Tyworth and Elijah Wright





Discussion

The Learning Modules are currently used in training students to master large scale data mining, modeling and visualization projects

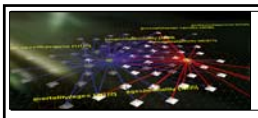
L597 Structural Data Mining and Modeling
Fall 2004 (http://ella.slis.indiana.edu/~katy/L597)

L579 Information Visualization
Spring 2004 and 2005 (http://ella.slis.indiana.edu/~katy/L579)

Since Fall 2003, the IVR was downloaded from about 50 institutions, organizations and companies in US, 14 institutions in Europe and 16 unidentifiable units.

Please consider using them in your classes!

SBC Fellows Forum, May 21st, 2004



Future Work

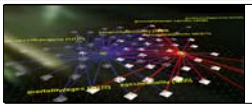
This summer, six data modeling, several data analysis and some new visualization algorithms will be integrated into the InfoVis Toolkit.

Implement programmer-friendly Java API that allows researchers to pipeline data between analysis algorithms and visualization tools within and outside the IVR.

Learning Modules will be updated and expanded.

There will be Tutorials on the InfoVis CyberInfrastructure and associated Learning Modules at the

- InfoVis Conference in London, UK, July 14-16, 2004.
- IEEE Visualization 2004 (Vis04) conference in Austin, Texas.



Acknowledgements

The Information Visualization Software Repository was created in 2000 and has since then been used to teach the Information Visualization class at Indiana University. Katy Börner, Yuezheng Zhou, and Jason Baumgartner implemented the very first algorithms. In Summer 2003, Jason Baumgartner, Nihar Sheth, and Nathan J. Deckard lead a project to design a XML toolkit that enables the serialization and parallelization of commonly used data analysis and visualization algorithms. Contributions of software packages and implementation work are acknowledged on the respective software pages.
Support comes from the School of Library and Information Science, Indiana University's High Performance Network Applications Program, an Academic Equipment Grant by SUN Microsystems, SBC (formerly Ameritech) Fellow Grant, and the National Science Foundation under DUE-0333623 and IIS-0238261.

Craig A. Stewart, Stephanie Burks, Mary Papakhian, Anurag Shankar all UITS generously made the Research Database Complex available for this project and provided very insightful comments and Oracle administration support.

SBC Fellows Forum, May 21st, 2004