GeoVISTA Studio: An Environment for Integrating Geospatial Data Analysis and Visualization Tools

Mark Gahegan … and Masa Takatsuka, Frank Hardisty, Xiping Dai, Junyan Luo, Diansheng Guo, Mike Wheeler, James O’Brien, Isaac Brewer, Dan Haug, Alan MacEachren

GeoVISTA Center, Dept. of Geography, Penn State University
http://www.geovistastudio.psu.edu/
Overview

- Introduction
- The current situation
- Open Systems, Component-Oriented Architecture
- GeoVISTA Studio
- Examples
- Conclusion
Aims

• What we want
  – Ease of use, but with the capability to construct complex analysis applications,
  – Flexibility, with rapid development and modification of applications, minimizing programming requirements,
  – Support for sharing and exchanging of developed applications.
The current situation

- **Diversity of Analysis Tools**
  - Types of tools: GIS, Statistics, Visualization, Machine Learning,
  - Languages: C, C++, AML, Visual Basic, FORTRAN, Java and more
  - Platforms: UNIX, Windows/Dos, Max

- **Need for Better Integration**
  - Integration of independently developed modules:
    - The Geographic Resources Analysis Support System (GRASS) (Byars and Clamons, 1998)
    - Users of ESRI’s Arc-family products can customize (parts of) them or build modules using Microsoft’s Component Object Model (COM).
    - Other scientific visualization applications such as OpenDX (http://www.opendx.org/) and IRIX Explorer (NAG, 2001) allow users to create their own modules using an application’s framework
– cont’d

• Problems
  – Most of the current solutions to seamless integration rely on the use of proprietary interfaces or frameworks (a huge commitment on behalf of all developers).
  – Independent development and deployment of tools is possible only within the particular application environment.
  – Systems acts as a barrier to research, locking users into specific ways of working because of lack of integration of tools.
Revision: Doing GIScience...

Data driven

Exploratory data analysis, data mining

Knowledge construction and learning

Model / process driven

Spatial analysis and map algebra

abduction

induction

deduction

Current GIS

Time
Open Source?

- For peer review / validation
- For teaching and demonstrating good practice in coding
- For utilization (how to connect it to your other components)
- For enhancement (how to modify behavior)
Component-Oriented Architecture

● Benefits of Components
  – Components can be assembled in many different configurations, a *design*, that can be modified ‘on the fly’
  – encapsulation, message passing, inheritance, data abstraction and polymorphism (from Object-Oriented model)
  – Independently deployed and subject to composition / modification by third parties
  – In Java, *introspection* function provides details of method interfaces, without need for source code…
  – …and *serialization* function can save the internal state of components at runtime
  – …and *runtime deployment* can take care of distribution and update of new or modified components.
Studio is designed for geographic visualization and knowledge discovery; to explore data, construct hypotheses, discover, refine and test knowledge, construct models, evaluate results, …, iterate as necessary.
Architecture: application builder

- Studio employs JavaBean technology to construct tools. The JavaBean specification defines a set of standardized Application Programming Interfaces (APIs) for the Java platform.

- From this, the builder automatically constructs a syntactic description of the functionalities and i/o methods of any bean.
Architecture: design box

Studio is a Java-based, visual programming environment providing rapid development of complex applications. Users can drag beans from a palette into the "design box" and link them together to create applications or applets.
Deploying applications on the web

-You hit a button
Studio deployment

web deployment with automatic updates
Components in Studio

• data transformation and statistics (conventional and spatial)
• a visual classifier including color selection (e.g. Munsell)
• Analysis tools: self-organising map (SOM), learning vector quantization, k-means, maximum likelihood, backprop neural networks
• interactive parallel coordinate plot (PCP)
• 2D dynamic map + map matrix
• 3D rendering, including dynamics
• a spreadsheet and a table browser
• a scatterplot matrix
• data mining: subspace clustering, binning, space ordering
• coordination and control tools
Census: DC race & housing
NCI/EPA: Cancer incidence and risk factors
NCI/EPA: Cancer incidence and risk factors
NCI/EPA: Cancer incidence and risk factors

conditioned on >6% black
conditioned on >4% black
Geographic knowledge discovery & clustering

- initial Delaunay triangulation (DT), then Minimum Spanning Tree (MST)
- overcomes single-link effect at different hierarchical levels
- generates spatial cluster ordering to represent a hierarchy of clusters
- ordering provided by spatial clusters serves as a “common” attribute to further non-spatial high-D clustering methods to uncover high-D geographic patterns

parallel coordinate plot (PCP)
Category construction and classification

self-organizing map
Inside the Neural Network

Water Agriculture

[Graph and data visualization related to water and agriculture within a neural network context]
**GeoVISTA Studio: Types of Users**

- **Component Developers**
  They might build JavaBeans components where no current tools are available. These new tools are imported into *Studio* and tested.

- **Application Developers**
  Their role is to construct data analysis and visualization applications that address specific geographical problems then to disseminate their solutions, for instance via the Internet.

- **Application Users**
  Do not use *Studio* directly. Instead they use the standalone applications or applets produced by *Studio*. Applications of this type should prove especially useful for educational purposes, since application developers can define exactly what is exposed to inexperienced users, and effectively remove all other details.
Conclusions

- *Studio* is FREE, you can download it from [www.geovistastudio.psu.edu](http://www.geovistastudio.psu.edu)
- It works for us (mostly!)
- Hopefully, it will not crash too much for you...
- If you plan to develop useful tools, we would be delighted to work with you so that they work effectively within *Studio*.
- There are still a lot of difficulties...
  - How should beans coordinate with one another?
  - What events should be coordinated?
  - How can coordination be achieved without locking all bean developers into our ideas? (just another closed system)
The End

Questions? Comments?