Plug-and-Play Macroscopes

Katy Börner
Cyberinfrastructure for Network Science Center, Director
Information Visualization Laboratory, Director
School of Library and Information Science
Indiana University, Bloomington, IN
katy@indiana.edu

With special thanks to the members at the Cyberinfrastructure for Network Science Center, the NWB team, the Sci2 team, the EpiC team, and all other teams that use OSGi/CIShell.

The School of Informatics and Computing Colloquium Series
January 28, 2011

Designing “Dream Tools”

Many of the best micro-, tele-, and macroscopes are designed by scientists keen to observe and comprehend what no one has seen or understood before. Galileo Galilei (1564–1642) recognized the potential of a spyglass for the study of the heavens, ground and polished his own lenses, and used the improved optical instruments to make discoveries like the moons of Jupiter, providing quantitative evidence for the Copernican theory.

Today, scientists repurpose, extend, and invent new hardware and software to create “macrosopes” that may solve both local and global challenges.

The tools I will show you today empower me, my students, colleagues, and 100,000 others that downloaded them.

Macrosopes

Decision making in science, industry, and politics, as well as in daily life, requires that we make sense of data sets representing the structure and dynamics of complex systems. Analysis, navigation, and management of these continuously evolving data sets require a new kind of data-analysis and visualization tool we call a macroscope (from the Greek macros, or “great,” and skopein, or “to observe”) inspired by de Rosnay’s futurist science writings. Macrosopes provide a “vision of the whole,” helping us “synthesize” the related elements and enabling us to detect patterns, trends, and outliers while granting access to myriad details. Rather than make things larger or smaller, macrosopes let us observe what is at once too great, slow, or complex for the human eye and mind to notice and comprehend.
Goal of This Talk

**Inspire computer scientists** to implement software frameworks that **empower domain scientists** to assemble their own continuously evolving macroscopes, adding and upgrading existing (and removing obsolete) plug-ins to arrive at a set that is truly relevant for their work—with little or no help from computer scientists.

While microscopes and telescopes are physical instruments, **macroscopes resemble continuously changing bundles of software plug-ins**. Macrosopes make it easy to select and combine algorithm and tool plug-ins but also interface plug-ins, workflow support, logging, scheduling, and other plug-ins needed for scientifically rigorous yet effective work.

They make it easy to share plug-ins via email, flash drives, or online. To use new plugins, simply copy the files into the plug-in directory, and they appear in the tool menu ready for use. No restart of the tool is necessary. **Sharing algorithm components, tools, or novel interfaces becomes as easy as sharing images on Flickr or videos on YouTube. Assembling custom tools is as quick as compiling your custom music collection.**

Changing Scientific Landscape—Personal Observations

- Different datasets/formats.
- Diverse algorithms/tools written in many programming languages.
Changing Scientific Landscape—General Observations

Science becomes more data driven and computational but also collaborative and interdisciplinary. There is increased demand for tools that are easy to extend, share, and customize:

- **Star scientist —> Research teams.** Traditionally, science was driven by key scientists. Today, science is driven by collaborating co-author teams, often comprising experts from multiple disciplines and geospatial locations.
- **Users —> Contributors.** Web 2.0 technologies empower users to contribute to Wikipedia and exchange images, videos, and code via Fickr, YouTube, and SourceForge.net.
- **Disciplinary —> Cross-disciplinary.** The best tools frequently borrow and synergistically combine methods and techniques from different disciplines of science, empowering interdisciplinary and/or international teams to collectively fine-tune and interpret results;
- **Single specimen —> Data streams.** Microscopes and telescopes were originally used to study a single specimen at a time. Today, many researchers must make sense of massive data streams comprising multiple data types and formats from different origins; and
- **Static instrument —> Evolving cyberinfrastructure.** The importance of hardware instruments that are static and expensive tends to decrease relative to software tools and services that are highly flexible and evolving to meet the needs of different sciences. Some of the most successful tools and services are decentralized, increasing scalability and fault tolerance.

Related Work

Google Code and SourceForge.net provide special means for developing and distributing software

- In August 2009, SourceForge.net hosted more than 230,000 software projects by two million registered users (285,957 in January 2011);
- In August 2009 ProgrammableWeb.com hosted 1,366 application programming interfaces (APIs) and 4,092 mashups (2,699 APIs and 5,493 mashups in January 2011)

Cyberinfrastructures serving large biomedical communities

- Cancer Biomedical Informatics Grid (caBIG) ([http://cabig.nci.nih.gov](http://cabig.nci.nih.gov))
- Biomedical Informatics Research Network (BIRN) ([http://nbirn.net](http://nbirn.net))
- Informatics for Integrating Biology and the Bedside (i2b2) ([https://www.i2b2.org](https://www.i2b2.org))
- HUBzero ([http://hubzero.org](http://hubzero.org)) platform for scientific collaboration uses
- myExperiment ([http://myexperiment.org](http://myexperiment.org)) supports the sharing of scientific workflows and other research objects.

Missing so far is a **common standard** for

- the design of **modular, compatible algorithm and tool plug-ins** (also called “modules” or “components”)
- that can be **easily combined into scientific workflows** (“pipeline” or “composition”),
- and packaged as **custom tools**.
OSGi & CShell

- CShell ([http://cishell.org](http://cishell.org)) is an open source software specification for the integration and utilization of datasets, algorithms, and tools.
- It extends the Open Services Gateway Initiative (OSGi) ([http://osgi.org](http://osgi.org)), a standardized, component oriented, computing environment for networked services widely used in industry since more than 10 years.
- Specifically, CShell provides “sockets” into which existing and new datasets, algorithms, and tools can be plugged using a wizard-driven process.

### CShell Features

**A framework for easy integration of new and existing algorithms written in any programming language** Using CShell, an algorithm writer can fully concentrate on creating their own algorithm in whatever language they are comfortable with. Simple tools are provided to then take their algorithm and integrate it into cishell with no additional coding.

**A well-defined pool of algorithms and datasets** CShell clearly defines how algorithms and datasets are integrated into the system to create a pool of algorithms and data. An application may then query for algorithms in this pool and execute them. Many applications/tools can be built and customized for different user groups by utilizing the same pool of algorithms.

**Leveraging open standards** CShell avoids re-inventing wheels by building on other standards for its specification and reference implementations. It benefits most from the Eclipse family of projects (in particular, the Rich Client Platform and Equinox) and the Open Services Gateway Initiative (OSGi). All CShell algorithms are integrated as OSGi services and can be used by any OSGi compliant system (including any Eclipse 3.0 or newer based products).

**Choose the way you work** CShell offers reference applications that build on the pool of algorithms defined by CShell. Scripting and a Graphical User Interface (GUI) are offered initially with a remoting (peer-to-peer and client-server) architecture, a web front-end, and other interfaces planned. We invite other toolkit developers to build their own applications on top of CShell's algorithm pool.

**Open source, community-driven project** CShell is released under the Apache 2.0 License. Community input is welcome to create a piece of software that advances science and education.
Algorithm Developer's Guide

Overview
The Cyberinfrastructure Shell (CIShell) is an open source, community-driven platform for the integration and utilization of datasets, algorithms, tools, and computing resources. Algorithm integration support is built in for Java and most other programming languages. Being Java based, it will run on almost all platforms. The software and specification is released under an Apache 2.0 License.

This guide attempts to aid algorithm developers in creating algorithms for CIShell (and applications built on CIShell).

This guide tries to contain all the information a new developer needs, but where necessary, it may cite the CShell 1.0 Specification (API) or the OSI, Service Platform Specification, Release 1 (API). While the guide tries to make beginning algorithm development easier, the CShell Specification has the last word on how the CShell Platform works.

Table of Contents
1. CShell Basics
2. Getting Started
   1. Tutorial 0: Setting Up the Development Environment
   2. Tutorial 1: Creating a Hello World Java Algorithm
   3. Tutorial 2: Practical Java Algorithm Development
   4. Tutorial 3: Integrating a Non-Java Program As An Algorithm
   5. Mini-Tutorial: Integrating 3rd-party libraries
   6. Where to Learn More
3. Reference
   2. Accessing the OSI, Console in CShell tools
Network Workbench (NWB) Tool
http://nwb.cns.iu.edu

Science of Science (Sci²) Tool
http://sci2.cns.iu.edu

Epidemics (EpiC) Tool (with Alex Vespignani and Jim Sherman, IU)
In progress

TEXTrend
http://textrend.org

Dyneta
http://www.dynanets.org

Network Workbench Tool
http://nwb.slis.indiana.edu

The Network Workbench (NWB) tool supports researchers, educators, and practitioners interested in the study of biomedical, social and behavioral science, physics, and other networks.

In February 2009, the tool provides more 169 plugins that support the preprocessing, analysis, modeling, and visualization of networks.

More than 50 of these plugins can be applied or were specifically designed for S&T studies.

It has been downloaded more than 65,000 times since December 2006.

Computational Proteomics

What relationships exist between protein targets of all drugs and all disease-gene products in the human protein–protein interaction network?


Computational Epidemics

Forecasting (and preventing the effects of) the next pandemic.


## Type of Analysis vs. Level of Analysis

<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>Micro/Individual (1-100 records)</th>
<th>Meso/Local (101–10,000 records)</th>
<th>Macro/Global (10,000 &lt; records)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Statistical</strong></td>
<td>Individual person and their expertise profiles</td>
<td>Larger labs, centers, universities, research domains, or states</td>
<td>All of NSF, all of USA, all of science.</td>
</tr>
<tr>
<td><strong>Analysis/Profiling</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Temporal</strong></td>
<td>Funding portfolio of one individual</td>
<td>Mapping topic bursts in 20-years of PNAS</td>
<td>113 Years of Physics Research</td>
</tr>
<tr>
<td><strong>Analysis (When)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Geospatial</strong></td>
<td>Career trajectory of one individual</td>
<td>Mapping a states intellectual landscape</td>
<td>PNAS Publications</td>
</tr>
<tr>
<td><strong>Analysis (Where)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Topical</strong></td>
<td>Base knowledge from which one grant draws.</td>
<td>Knowledge flows in Chemistry research</td>
<td>VxOrd/Topic maps of NIH funding</td>
</tr>
<tr>
<td><strong>Analysis (What)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Network</strong></td>
<td>NSF Co-PI network of one individual</td>
<td>Co-author network</td>
<td>NSF’s core competency</td>
</tr>
<tr>
<td>**Analysis (With Whom?)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


- Tutorial #01: Science of Science Research
- Tutorial #02: Network Science / Information Visualization
- Tutorial #03: CI/Shell Powered Tools: Network Workbench and Science of Science Tool
- Tutorial #04: Temporal Analysis—Burst Detection
- Tutorial #05: Geospatial Analysis and Mapping
- Tutorial #06: Topical Analysis & Mapping
- Tutorial #07: Tree Analysis and Visualization
- Tutorial #08: Network Analysis and Visualization
- Tutorial #09: Large Network Analysis and Visualization
- Tutorial #10: Using the Scholarly Database at IU
- Tutorial #11: VIVO National Researcher Networking
- Tutorial #12: Future Developments

Geetha Sreedhar (2010) Multidisciplinary Nature of Work With Reference to PIs and ICs Within a Portfolio, PA Group at NIH.

Sci² Tool – “Open Code for S&T Assessment”

OSGi/CIShell powered tool with NWB plugins and many new scientometrics and visualizations plugins.

## Sci² Tool: Algorithms

See [https://nwb.slis.indiana.edu/community](https://nwb.slis.indiana.edu/community)

### Preprocessing
- Extract Top N% Records
- Extract Top N Records
- Normalize Text
- Slice Table by Line
- Extract Top Nodes
- Extract Nodes Above or Below Value
- Delete Isolates
- Extract top Edges
- Extract Edges Above or Below Value
- Trim by Degree
- MST-Pathfinder Network Scaling
- Fast Pathfinder Network Scaling
- Snowball Sampling (in nodes)
- Node Sampling
- Edge Sampling
- Symmetrize
- Dichotomize
- Multipartite Joining
- Geocoder
- Extract ZIP Code

### Modeling
- Random Graph
- Watts-Strogatz
- Small World
- Barabási-Albert Scale-Free
- TARL

### Analysis
- Network Analysis Toolkit (NAT)
- Unweighted & Undirected
  - Node Degree
  - Degree Distribution
  - K-Nearest Neighbor (Java)
  - Watts-Strogatz Clustering Coefficient
  - Watts Strogatz Clustering Coefficient over K
  - Diameter
  - Average Shortest Path
  - Shortest Path Distribution
  - Node Betweenness Centrality
- Weighted & Undirected
  - Clustering Coefficient
  - Nearest Neighbor Degree
  - Strength vs Degree
  - Degree & Strength
  - Average Weight vs End-point Degree
  - Strength Distribution
  - Weight Distribution
  - Randomize Weights
  - Blondel Community Detection
  - HITS
- Unweighted & Directed
  - Node Indegree
  - Node Outdegree
  - Indegree Distribution
  - Outdegree Distribution
  - K-Nearest Neighbor
  - Node In-Out Degree Correlations
- Dyad Reciprocity
- Arc Reciprocity
- Adjacency Transitivity
- Weak Component Clustering
- Strong Component Clustering

### Visualization
- GnuPlot
- GUESS
- Image Viewer
- Radial Tree/Graph (prefuse alpha)
- Radial Tree/Graph with Annotation (prefuse beta)
- Tree View (prefuse beta)
- Tree Map (prefuse beta)
- Force Directed with Annotation (prefuse beta)
- Fruchterman-Reingold with Annotation (prefuse beta)
- DrL (VxOrd)
- Specified (prefuse beta)
- Horizontal Bar Graph
- Circular Hierarchy
- Geo Map (Circle Annotation Style)
- Geo Map (Colored-Region Annotation Style)
- Science Map (Circle Annotation)

### Scientometrics
- Remove ISI Duplicate Records
- Remove Rows with Multitudinous Fields
- Detect Duplicate Nodes
- Update Network by Merging Nodes
- Extract Directed Network
- Extract Paper Citation Network
- Extract Author Paper Network
- Extract Co-Occurrence Network
- Extract Word Co-Occurrence Network
- Extract Co-Author Network
- Extract Reference Co-Occurrence (Bibliographic Coupling) Network
- Extract Document Co-Citation Network

Soon:
- Database support for ISI and NSF data.
### Type of Analysis vs. Level of Analysis

<table>
<thead>
<tr>
<th>Type of Analysis</th>
<th>Micro/Individual (1-100 records)</th>
<th>Meso/Local (101–10,000 records)</th>
<th>Macro/Global (10,000 &lt; records)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statistical Analysis/Profiling</td>
<td>Individual person and their expertise profiles</td>
<td>Larger labs, centers, universities, research domains, states</td>
<td>All of NSB, all of SRS, all of science, all of society</td>
</tr>
<tr>
<td>Temporal Analysis (When)</td>
<td>Funding portfolio of one individual</td>
<td>Mapping topic bursts of PNAS</td>
<td>113 Years of PNAS Research</td>
</tr>
<tr>
<td>Geospatial Analysis (Where)</td>
<td>Career trajectory of one individual</td>
<td>Mapping a states intellectual landscape</td>
<td>PNAS publication intellectual landscape</td>
</tr>
<tr>
<td>Topical Analysis (What)</td>
<td>VxOrd/Topic maps of NIH funding</td>
<td>VxOrd/Topic maps of NIH funding</td>
<td>VxOrd/Topic maps of NIH funding</td>
</tr>
<tr>
<td>Network Analysis (With Whom?)</td>
<td>NSF co-PI network of one individual</td>
<td>Knowledge flows in Chemistry research</td>
<td>NIH’s core competency network</td>
</tr>
</tbody>
</table>

---

**Mapping Indiana’s Intellectual Space**

*Geospatial/Network Analysis*

*2001-2006, BioMed, IN Scope*

*Academic-Industry collaborations and knowledge diffusion*
Mapping Topic Bursts

Co-word space of the top 50 highly frequent and bursty words used in the top 10% most highly cited PNAS publications in 1982-2001.


Spatio-Temporal Information Production and Consumption of Major U.S. Research Institutions

Börner, Katy, Penumarthy, Shashikant, Meiss, Mark and Ke, Weimao. (2006)

Research questions:
1. Does space still matter in the Internet age?
2. Does one still have to study and work at major research institutions in order to have access to high quality data and to produce high quality research?
3. Does the Internet lead to more global citation patterns, i.e., more citation links between papers produced at geographically distant research institutions?

Contributions:
- Answer to Qs 1 + 2 is YES.
- Answer to Qs 3 is NO.
- Novel approach to analyzing the dual role of institutions as information producers and consumers and to study and visualize the diffusion of information among them.
This map highlights the research collaborations of the Chinese Academy of Sciences with locations in China and countries around the world. The large geographic map shows the research collaborations of all CAS institutes. Each smaller geographic map shows the research collaborations by the CAS researchers in one province-level administrative division. Collaborations between CAS researchers are not included in the data. On each map, locations are colored on a logarithmic scale by the number of collaborations from red to yellow. The darkest red is 3,395 collaborations by all of CAS with researchers in Beijing. Also, flow lines are drawn from the location of focus to all locations collaborated with. The width of the flow line is linearly proportional to the number of collaborations with the locations it goes to, with the smallest flow lines representing one collaboration and the largest representing differing amounts on each geographic map.

Geospatial Analysis
World, Chinese Academy of Science
Collaboration and knowledge diffusion via co-author networks
Mapping Transdisciplinary Tobacco Use Research Centers Publications

Compare R01 investigator based funding with TTURC Center awards in terms of number of publications and evolving co-author networks.

*Zoss & Börner, forthcoming.*

---

**Temporal/Network Analysis**

1998-2009, US, Tobacco research scope
Evolving co-author networks

---

**MEDLINE Publication Output by The National Institutes of Health (NIH)**

Using Nine Years of ExPORTER Data

Katy Börner, Nianli Ma, Joseph R. Biberstine, Cyberinfrastructure for Network Science Center, SLIS, Indiana University, Robin M. Wagner, Rediet Berhane, Hong Jiang, Susan E. Ivey, Katrina Pearson and Carl McCabe, Reporting Branch, Division of Information Services, Office of Research Information Systems, Office of Extramural Research, Office of the Director, National Institutes of Health (NIH), Bethesda, MD.

---

**Topical Analysis**

2001-2009, US, All NIH funding and associated papers
Changes in publication patterns
Changing Scientific Landscape—Personal Observations Cont.

- Common algorithm/tool pool
- Easy way to share new algorithms
- Workflow design logs
- Custom tools

Converters, Sci2, NWB, TexTrend, EpiC, Converters

- IS
- CS
- Bio
- SNA
- Phys
Desirable Features and Key Decisions

- **Division of labor:** The design/documentation of the “core architecture” requires extensive computer science expertise and a close collaboration with domain experts. Data set and algorithm plug-ins—the “filling”—are typically provided by domain experts most invested in the data and most knowledgeable about the inner workings and utility of different algorithms. The design/documentation of “custom tools” is best performed by domain experts, as only they have the expertise needed to bundle different plug-ins relevant for diverse workflows.

```
34

Custom Tool 1
Tool 2

Domain Experts

Core + Algs & Tools
Workflow

Domain Experts

Core Architecture
Workflow

Alg
Alg
Alg
Alg
Alg
Alg

Workflow

Workflow

Workflow

Workflow

Workflow

Workflow

Workflow

Workflow

Workflow
```

Desirable Features and Key Decisions Cont.

- **Ease of use.** Non-computer scientists must be able to contribute, share, and use plug-ins without having to write new code. Wizard-driven integration of algorithms and data sets, sharing of plug-ins through email and online sites, deploying plug-ins by adding them to the “plug-in directory,” and running them via a menu-driven user interface work well;

- **Core vs. plug-ins.** Determining whether the graphical user interface, interface menu, scheduler, and data manager should be part of the core or its filling depends on the types of tools and services to be delivered;

- **Plug-in content and interfaces.** Should a plug-in be a single algorithm or an entire tool? What about data converters needed to make the output of one algorithm compatible with the input of another algorithm? Which general interfaces are needed to communicate parameter settings, input, and output data? Answers are domain-specific, depending on existing tools and practices and the problems domain experts aim to solve;

- **Supported (central) data models.** Some tools (such as Cytoscape) use a central data model to which all algorithms conform. Others (such as NWB and Sci2) support many internal data models and provide an extensive set of data converters. The former often speeds execution and visual rendering, and the latter eases integration of new algorithms. In addition, most tools support an extensive set of input and output formats, since a tool that cannot read or write a desired data format is usually of little use by domain experts;

- **Supported platforms. Standalone tools vs. Web services.** Domain specific answers.
Empowering “A Million Minds”

To design flexible, scalable software that can be used by many flexibly and freely.

- **Modularity.** Software modules with well-defined functionality that can accept contributions from multiple users reduce costs and increase flexibility in tool development, augmentation, and customization;
- **Standardization.** Standards accelerate development, as existing code is leveraged, helping pool resources, support interoperability, and ease migration from research code to production code and hence the transfer of research results into industry applications and products; and
- **Open data and open code.** The practice of making data sets and code freely available allows users to check, improve, or repurpose data and code, easing replication of scientific studies.

Acknowledgements

- Micah Linnemeier and Russell J. Duhon Bruce W. Herr II, George Kampis, Gregory J. E. Rawlins, Geoffrey Fox, Shawn Hampton, Carol Goble, Mike Smoot, Yanbo Han for stimulating discussions and comments.
- Software development benefits greatly from the open-source community. Full software credits are distributed with the source, but I would especially like to acknowledge Jython, JUNG, Prefuse, GUESS, GnuPlot, and OSGi, as well as Apache Derby, used in the Sci2 tool.

This research and development is based on work supported by National Science Foundation grants SBE-0738111, IIS-0513650, IIS-0534909 and National Institutes of Health grants R21DA024259 and 5R01MH079068.
References


All papers, maps, cyberinfrastructures, talks, press are linked from http://ens.slis.indiana.edu