An Introduction to the Network Workbench and Cyberinfrastructure Shell
Introductory Paper

Bruce Herr (bh2@bh2.net)
Weixia Huang (huangb@indiana.edu)
School of Library and Information Science, Indiana University, Bloomington, IN, USA

1 Overview

The Network Workbench (NWB) project will design, evaluate, and operate a unique distributed, shared resources environment for biomedical, social science, and physics research. Investigators are Katy Börner, Santiago Schnell, Alessandro Vespignani, Stanley Wasserman, Eric Wernert (Indiana University), and Albert-László Barabási (Notre Dame University). Major software developers are Weixia Huang, Bruce Herr, Ben Markines, Santo Fortunato (Indiana University), and Cesar Hidalgo (Notre Dame University). Advisory board members comprise Noshir Contractor (NCSA), Craig Alan Stewart (Indiana University), James Hendler (University of Maryland), Jason Leigh (University of Illinois at Chicago), Neo Martinez (Rocky Mountain Biological Lab), Michael Macy (Cornell University), Ulrik Brandes (University of Konstanz), Mark Gerstein (Yale University), Stephen North (AT&T), and Tom Snijders (University of Groningen). The work is supported in part by a NSF IIS-0513650 award.

The NWB data-code-computing resources environment will provide a one-stop online portal for researchers, educators, and practitioners. Users of the NWB will have online access to major network datasets or can upload their own networks. They will be able to perform network analysis with the most effective algorithms available. In addition, they will be able to generate, run, and validate network models to advance their understanding of the structure and dynamics of particular networks. NWB will provide advanced visualization tools to interactively explore and understand specific networks, as well as their interaction with other types of networks.

A major computer science challenge is the development of an algorithm integration framework, called the Cyberinfrastructure Shell (CIShell), that supports the easy integration and dissemination of existing and new algorithms and can deal with the multitude of network data formats in existence today. Another challenge is the design and implementation of an easy to use menu-based, online portal interface for interactive algorithm selection, data manipulation, user and session management.

The NWB will be evaluated in diverse research projects and educational settings in biology, social and behavioral science, and physics research.

This document describes CIShell, the CIShell Reference GUI and the NWB Tool, as well as the NWB Community Wiki.

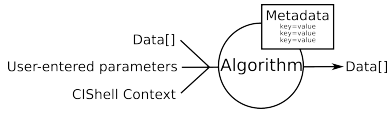
2 Cyberinfrastructure Shell

The Cyberinfrastructure Shell (CIShell) [1] is an open source software specification for the integration and utilization of datasets, algorithms, tools, and computing resources. The Cyberinfrastructure Shell supports (1) algorithm writers to write and disseminate their algorithms in their favorite programming language while retaining their intellectual rights after distribution; (2) data holders to easily disseminate their data for use by others; (3) application writers to design applications from custom sets of algorithms and datasets that interoperate seamlessly; and finally (4) researchers, educators, and practitioners to use existing datasets and algorithms to further science.

Algorithm integration support is built in for Java and most other programming languages. Being Java based, CIShell runs on almost all platforms. The software and specification is released under the Apache 2.0 License.

Subsequently, we describe the bundling, integration, and distribution of datasets and algorithms, the deployment of CIShell algorithms as service pools or applications, as well as the interlinkage of service pools and applications. We then address the issues of automatic data conversion and brandable graphical user interfaces (GUIs).

2.1 Defining Algorithms & Datasets



Using CIShell, datasets and algorithms are bundled as CIShell-defined algorithms, see figure above. An algorithm is a black box that takes in zero or more datasets, some defined user parameters, and a so called CIShell Context. It then processes the inputs and returns an output in the form of zero or more datasets. The CIShell Context provides access to services offered by CIShell such as logging, preferences, GUI creation, and data conversion. The metadata dictionary associated with each algorithm can be used by applications, see section 2.4. The metadata also comprises the author, citations, links to homepages, documentation, run-time complexity, etc. that ensures proper usage and citation.

To give an example, a modeling algorithm takes in no data, some user-entered parameters (e.g., the number of nodes to create a random graph) and returns a single

dataset (e.g., a random graph). An analysis algorithm takes in some data and possibly some user-entered parameters, analyses the graph, then returns a new graph or simply prints out analysis results on the console. A visualization algorithm takes in some data, opens a new window visualizing the data, and typically returns no data. Each dataset is bundled as a dataset provider, i.e., an algorithm that takes in no data or parameters, but returns a dataset that is available to the CIShell system.

2.2 Algorithm & Dataset Integration Templates

To ease the integration of algorithms and datasets, wizard-driven templates are provided that acquire information from the algorithm writer and then generate the appropriate files and resources. Templates are available to integrate arbitrary file-based datasets, compiled executable code, Java code, and Java libraries. In the case of the Java code template, after running the wizard, only one method, the execute method for the actual algorithm, needs to be filled in. Typically, the integration of executable code does not require writing one line of new code.

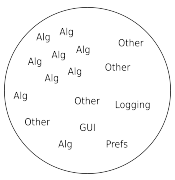
2.3 Algorithm & Dataset Distribution

Distribution of datasets and algorithms is made easy by using Eclipse [5] update sites. Eclipse update sites allow an algorithm writer to package up their algorithms and place them on a webserver. Any algorithm writer can create his or her own private or public update site. Algorithm users can download any subset of datasets and algorithms from any number of update sites. He or she simply needs to select Help/Update from the menu and search for new features or updates of the currently installed features.

The NWB Community Wiki provides an easy interface for algorithm writers and data holders to post links to their algorithms and datasets, explain how to use them, and advertise their update sites, see section 4.

2.4 Creating A Pool

CIShell Algorithms defined in section 2.1 can be located and run by using a service registry that is provided by the underlying Open Services Gateway Initiative (OSGi) technology [4]. This service registry defines a pool where services can be registered, searched for, and retrieved for use. To be a service in OSGi, one must provide an interface, an implementation of the interface, and a dictionary of metadata about the service. For algorithms, the interface is AlgorithmFactory as defined by CIShell and the implementation and metadata as provided by the algorithm writer. All major components of CIShell are defined in terms of services available in the service registry.



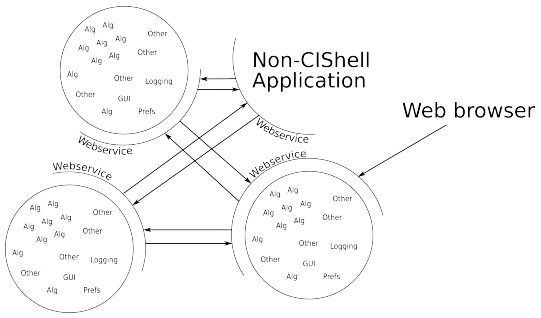
Once the algorithms are in the service pool (see figure above), CIShell applications can easily search for algorithms based on their metadata. A querying mechanism can be used by applications to find subsets of algorithms like all algorithms that belong in the visualization menu, all conversion algorithms, see also section 2.7.

2.5 Creating An Application

Every application that uses CIShell technology utilizes the OSGi Service Registry to locate and run algorithms. For example, a graphical user interface (GUI) application searches for all algorithms that have a menu path and puts them in their appropriate GUI menu; a scripting layer allows scripts to gain access to the algorithms in the service registry, etc. That is, algorithms are independent of their usage.

2.6 Connecting Pools & Applications

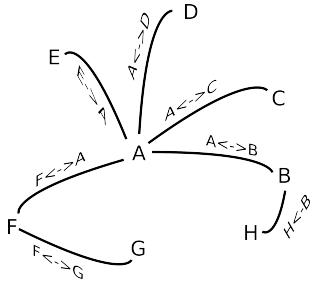
Our future work aims to create CIShell web services and to support the interlinkage of multiple algorithm pools and web services in a peer-to-peer fashion. This inter-pool communication mechanism (see figure below) will also support client-server applications in which computationally demanding algorithms are run on servers and multiple CIShell clients can connect to it to access datasets, algorithms, and computing power.



2.7 Automatic Data Conversion

Algorithms and the service registry can be utilized to solve another problem that arises when dealing with many algorithms written by different people from different scientific communities: the multitude of existing file formats. The data conversion service offered by CIShell supports the seamless conversion of different file formats.

The service searches through the service registry for converter algorithms (type=conversion in the algorithm's metadata) and builds a directed graph based on them. Edges are converters and nodes are file formats (see figure to the right). Edges can be weighted according to the 'losslessness', 'complexity', or other features as reported in a converters' metadata.

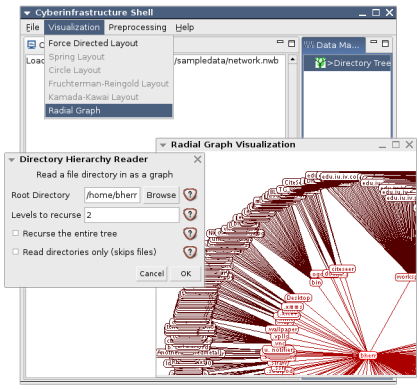


When asked to find a chain of converters from format A to format G, it searches the graph for the shortest (weighted) path and returns a valid chain of converters.

The CIShell Reference GUI uses this data conversion service to identify those algorithms that can handle a selected dataset. Only applicable algorithms are selectable and only if the algorithm is selected will the data be converted into the appropriate format. Data format conversion happens invisibly to the user.

2.8 Brandable CIShell Reference GUI

The CIShell project includes a reference GUI that can be used directly or can be customized by application writers. The GUI (see figure below) is a menu-driven interface where all algorithms appear as menu items. Datasets can be loaded or simulated and are listed in the right-hand side data manager. When a user selects a dataset, the GUI determines what algorithms, either directly or through a series of converters, can be applied. Only algorithms that can process the selected dataset are selectable, all others are grayed out. A console is provided to give information about the selected algorithms (authors, implementers, citations, etc.) and any output the algorithm logs when run.



3 Network Workbench Tool

The Network Workbench (NWB) Tool [2] is a large-scale network analysis, modeling, and visualization toolkit for biomedical, social science, and physics research (see page 1). The NWB Tool rebrands the CIShell Reference GUI and provides a custom filling of datasets, algorithms, and converters relevant for the network science community.

The NWB Tool uses the CIShell data conversion service and a set of converters to support the loading and saving of GraphML (.xml), XGMML (.xml), Pajek (.net), and Network Workbench (.nwb) formatted files.

The current version of the NWB Tool gives easy access to 41 algorithms and a few sample datasets. About half of the algorithms were implemented in FORTRAN and integrated by a physicist using CIShell's static executable algorithm integration template. A listing of the currently integrated algorithms follows.

Sampling
- Directory Hierarchy Reader

Analysis
- Node Degree
- Node Indegree
- Node Outdegree
- Undirected Degree Distribution
- Indegree Distribution
- Outdegree Distribution
- Undirected k-Nearest Neighbor
- Directed k-Nearest Neighbor
- One Point Correlations
- Watts Strogatz Clustering Coefficient
- Watts Strogatz Clustering Coefficient Over k
- Average Path Length
- Diameter
- Distribution of shortest path length
- Betweenness Centrality
- Number of Connected Components
- Page Rank
- Attack Tolerance
- Error Tolerance
- k Random-Walk Search
- Random Breadth First Search
- CAN Search
- Chord Search

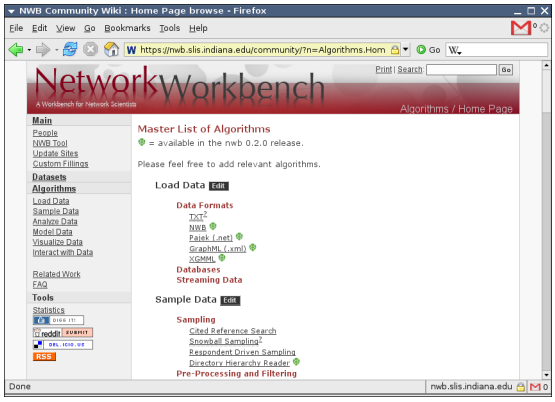
Modeling
- Barabási-Albert Scale-Free Model
- CAN Model
- Chord Model
- Hypergrid Model
- PRU Model
- Erdos-Rényi Random Graph Model
- Watts-Strogatz Small World Model

Visualization
- Circular Layout
- Radial Tree
- Tree Map
- Tree Visualization
- Force Directed
- Spring Layout
- Kamada-Kawai
- Fruchterman-Reingold
- Parallel Coordinates (demo)

Tool
- XMGrace

4 NWB Community Wiki

The Network Workbench Community Wiki [3] is a place for users of the Network Workbench Tool, the Cyberinfrastructure Shell, or any other CIShell-based application to upload, download, and request datasets and algorithms. The site (see figure below) was created so that the network science community can collaboratively create a tool which meets their needs and the needs of the scientific community at large. Users can post want-ads for algorithms and datasets to be integrated into CIShell/NWB, and learn how to use the resources for their own research.



References

[1] CIShell Homepage: http://cishell.org

[2] NWB Homepage: http://nwb.slis.indiana.edu

[3] NWB Community: https://nwb.slis.indiana.edu/community

[4] OSGi: http://www.osgi.org

[5] Eclipse: http://www.eclipse.org