

Systems Biology Workbench Standard Modules and Menus

Andrew Finney, Michael Hucka, Herbert Sauro, Hamid Bolouri

`{afinney,mhucka,hsauro,hbolouri}@cds.caltech.edu`
Systems Biology Workbench Development Group
ERATO Kitano Systems Biology Project
Control and Dynamical Systems, MC 107-81
California Institute of Technology, Pasadena, CA 91125, USA
<http://www.cds.caltech.edu/erato>

Principal Investigators: John Doyle and Hiroaki Kitano

July 3, 2002



Contents

- 1 Introduction** **3**
- 2 Installing SBW and SBW applications** **4**
- 3 The SBW program menu** **5**
- 4 Tools for Exploring SBW** **6**
 - 4.1 SBW Inspector 6
 - 4.2 SBW Tray Tool 7
- 5 SBW Modules accessible from JDesigner** **8**
 - 5.1 Standard Simulation Window 8
- References** **9**

1 Introduction

This document describes how to use the basic modules provided in the Systems Biology Workbench (SBW; [Hucka et al., 2001](#)) installation for computers running Microsoft Windows. In particular, this document describes the following:

- How to install SBW;
- The functions of the various items in the “System Biology Workbench” submenu under the Windows “Program” menu; and
- Tools for examining the state of the whole SBW environment.

This document does not describe in detail how to use SBW-enabled applications such as Jarnac ([Sauro, 2000](#)) and JDesigner ([Sauro, 2001](#)). However, the modules supplied with the SBW download and that are accessible via JDesigner are described in Section 5.

SBW itself is not an application, but rather a framework that enables communication between different SBW-enabled applications. Information about programming with SBW is available in the various SBW API language manuals ([Finney et al., 2002a,b](#); [Hucka et al., 2002](#); [Sauro et al., 2002](#)).

2 Installing SBW and SBW applications

To install SBW and SBW applications, visit the following URL:

<http://www.cds.caltech.edu/erato/sbw/software/index.html>

This web page lists several components to download. The rest of this document assumes that these components are installed, although only the SBW and JDK installations are essential for SBW operation. The components list on the web site should be downloaded and installed in the following order:

- Sun's JDK 1.3
- Python Version 2.1
- Numerical Python
- SBW
- Jarnac
- JDesigner

To enable JDesigner and Jarnac to be accessible to other SBW applications, you must run each one, select the "Options/Register with SBW" menu item, then after doing this for both JDesigner and Jarnac, select the the menu item "Programs/Systems Biology Workbench/Utilities/Register New Simulator" from the Windows "Start" menu.

3 The SBW program menu

After installing SBW, the Windows “Start/Programs” menu will have a submenu named “Systems Biology Workbench”. This submenu has the following structure:

- **Modules**

This menu is empty in the basic SBW installation. It is a placeholder for third-party modules.

- **Scripting**

- **Python SBW**

This menu item launches a Python scripting language interpreter window. Python is started in such a way that it loads a custom SBW library allowing direct interaction with SBW. When the Python interpreter first starts up, you will see messages in the window indicating it is contacting SBW.

- **Utilities**

- **Broker Start**

This menu item manually starts the SBW Broker. Normally this is unnecessary because SBW-enabled applications will start the SBW Broker automatically.

- **Broker Stop**

This menu item stops the SBW Broker. This menu item will suspend all links between SBW-enabled applications and modules. This typically means that those applications and modules will terminate. This menu item is useful for terminating orphaned modules and returning the SBW to a known state.

- **Broker Debug**

This menu item starts the SBW Broker in a console window in a mode in which it produces debug output. This is not normally used, but it is provided to help debug possible problems.

- **Uninstall SBW**

This menu item will uninstall all those components installed by the SBW installation.

4 Tools for Exploring SBW

4.1 SBW Inspector

The *Inspector* module is a third-party utility available from the SBW download page.

When installed, this application provides the menu item “Systems Biology Workbench/Utilities/Inspector” in the Windows “Start” menu. The Inspector allows the detailed examination of the state of the SBW including modules and applications not directly visible to users.

The inspector has two tabs:

- **Running Modules**

This tab consists of three parts:

- **List All Modules button:** select this button to activate the other parts of this tab.
- **Module list:** a list of the running module instances. This list consists of the module identification strings of the running module instances. As there can be multiple instances of the same module running there can be more than one entry with the same name in this list. Selecting an entry in this list will display the module instance’s numeric identifier below the list and display the services in the tree to the right of the list module instances. The numeric identifier enables you to distinguish between instances of the same module. The module instance list will always contain an entry for the SBW Broker, “Broker” and at least one entry for the inspector application, “Inspector”.
- **Services and Methods tree:** a tree of the services and methods implemented by the module instance selected in the Module list on the left. The top level or unexpanded level of this tree consists of all the services on the module instance. The next level, exposed by double clicking on a service, consists of the methods implemented by each service. Each method entry consists of two parts: the method name followed by the method signature. A method signature is a string which defines the argument and return types of a given method. The syntax for these signatures is given in detail in the SBW programming manuals ([Finney et al., 2002a,b](#); [Hucka et al., 2002](#); [Sauro et al., 2002](#)).

- **Registered Modules**

This is the second tab of the Inspector application and consists of two parts:

- **List Registered Modules button:** select this button to activate the tree part of this tab.
- **Register Modules Tree:** This tree contains details of all the modules registered with the SBW Broker. The top level or unexpanded level of this tree lists all the module identification strings of the registered modules. Each entry is unique (i.e. not duplicated). The next level, exposed by double clicking on a module, consists of some of the information registered with the module. The attributes displayed, for each module, are:
 - * **DisplayName:** an alternative name for a module which may be more informative than the module’s unique identification string.
 - * **CommandLine:** the command that the SBW Broker uses to launch the module on behalf of SBW applications.
 - * **Help Str:** the help string for the module, which should provide more information about the module’s function

- * **Management Type:** indicates under what conditions SBW will launch a new instance of the module. This attribute can have two values either “SELF_MANAGING” or “UNIQUE”. “UNIQUE” indicates that the SBW Broker will only launch one instance of a module when one or more applications request access to a module instance. “SELF_MANAGING” indicates that the SBW Broker will launch a new instance for each request for access to a module instance.

4.2 SBW Tray Tool

The *TrayTool* module is a third-party utility available from the SBW download page.

When installed, this application provides the menu item “Systems Biology Workbench/Utilities/Tray Tool” in the Windows “Start” menu. When launched, the tool installs a square red icon in the Windows tray tool (the lower right-hand corner of the Windows task bar). The TrayTool application can be invoked by clicking the right mouse button over the icon and selecting “Module List” from the resulting pop up menu.

The tool has two tabs:

- **Current**

This consists of a list of module instances with two columns: the “Module” column displays the module’s unique identification string and the “Id” column displays the module instance’s numeric identifier. If you select one or more module instances in the list and click on the icon in the bottom left hand corner the application will shut down the selected module instances.

- **Log**

This consists of a simple list of module instance start and shutdown events that have occurred in the SBW environment since this application was started.

5 SBW Modules accessible from JDesigner

This section describes those SBW modules, supplied with the SBW download, that are made visible to the user within JDesigner.

JDesigner has a menu called “SBW” which is constructed from all the services registered in the “Analysis” service category. JDesigner applies a selected service in this menu to the currently displayed model.

After all the installations listed in Section 2, JDesigner’s “SBW” menu will contain the following entries:

- **Simulation Service:** launches a GUI which in turn controls an ODE simulation of the current model, implemented by the Jarnac simulator. See section 5.1.
- **Gibson Simulator:** launches a GUI which in turn controls a stochastic simulation of the current model, implemented by the Gibson simulator. See section 5.1.
- **Save to Network Object Model Clipboard:** loads the current model into a model clipboard. For use with the Gillespie simulator. The model clipboard is a unique module, edu.caltech.NOMClipboard, which can be accessed from a scripting environment.
- **Save to MATLAB ODE function file:** enables you to save the current model as a MATLAB ODE function. Selecting this service simply launches a dialog which allows you to specify where the MATLAB function should be stored.
- **Save to MATLAB Simulink function file:** enables you to save the current model as a MATLAB Simulink function. Selecting this service simply launches a dialog which allows you to specify where the MATLAB function should be stored.

The following sections describe the GUIs of the above modules in more detail.

5.1 Standard Simulation Window

This section describes how to use the window for controlling simulations that is used to control the Jarnac and Gibson simulators. This window allows you to specify the following aspects of one more simulations of the selected model:

- the time range and temporal detail of the simulation
- the set of variables which you wish to visualize
- how you want to visualize those variables

The simulation window allows you to create several 2D visualizations of simulation plot data. This window has the following components, here ordered as you typically interact with them:

- **Start Time** This field allows you to set the modeled start time of the next simulation
- **End Time** This field allows you to set the modeled end time of the next simulation
- **Number of Points** This field allows you to specify the number of sets of variable values that should be generated by the simulator during the next simulation.
- **Hidden Variables list** This list contains the set of variables that won’t be output by the next simulation. None of the variables in this list can be visualized. Select those variables that you wish to visualize.
- **<< Button** This button transfers the variables in the *Hidden Variables list* to the *Output Variables list*. Use this to visualize the variables selected in the *Hidden Variables list*.

- **Output Variables list** This list contains the set of variables that will be output by the next simulation. The variables in this list can be visualized.
- **>> Button** This button transfers the variables in the *Output Variables list* to the *Hidden Output Variables list*. Controlling the size of the *Output Variables list* will affect the performance of simulations especially when interacting with remote simulators.
- **X-Axis list** This list contains all the variables in the *Output Variables list* and allows you to select either an output variable or time to be on the x axis on the next data visualization window.
- **Y-Axis list** This list contains all the variables in the *Output Variables list* and allows you to select a set of output variables to be on the y axis on the next data visualization window.
- **Start Button** This button starts a simulation on the selected model using the information in the following fields and List:
 - Start Time
 - End Time
 - Number of Points
 - Output Variables

This button will not be selectable if the the above fields do not have consistent values or if a simulation is currently running.

- **Plot Button** This button creates a new visualization window using the output from the current simulation. The visualization is based on the selected items in the x and y axis list. Any number of visualization windows can be created. This button is will not be selectable until a simulation has generated at least one data value and items are selected in the x and y axis lists.
- **Stop Button** This button causes the current simulation to be stopped. This button will not be selectable when there is no simulation running.
- **Pause Button** This button causes the current simulation to be suspended. This button will not be selectable when there is no simulation running.
- **Restart Button** This button causes the current suspended simulation to be resumed. This button will not be selectable when there is no suspended simulation.
- **Close Button** This button closes this window and shuts down the simulator.
- **State** This field, which can't be edited, shows the state of the simulator.

References

Finney, A., Hucka, M., Sauro, H. M., and Bolouri, H. (2002a). Systems Biology Workbench C programmer's manual. Available via the World Wide Web at <http://www.cds.caltech.edu/erato/sbw/docs/>.

Finney, A., Hucka, M., Sauro, H. M., and Bolouri, H. (2002b). Systems Biology Workbench C++ programmer's manual. Available via the World Wide Web at <http://www.cds.caltech.edu/erato/sbw/docs/>.

Hucka, M., Finney, A., Sauro, H. M., and Bolouri, H. (2001). Introduction to the Systems Biology Workbench. Available via the World Wide Web at <http://www.cds.caltech.edu/erato/sbw/docs/intro>.

Hucka, M., Finney, A., Sauro, H. M., and Bolouri, H. (2002). Systems Biology Workbench Java™ programmer's manual. Available via the World Wide Web at <http://www.cds.caltech.edu/erato/sbw/docs>.

Sauro, H. M. (2000). Jarnac: A system for interactive metabolic analysis. In Hofmeyr, J.-H. S., Rohwer, J. M., and Snoep, J. L., editors, *Animating the Cellular Map: Proceedings of the 9th International Meeting on BioThermoKinetics*. Stellenbosch University Press.

Sauro, H. M., Hucka, M., Finney, A., and Bolouri, H. (2002). Systems Biology Workbench Delphi/Kylix programmer's manual. Available via the World Wide Web at <http://www.cds.caltech.edu/erato/sbw/docs>.

Sauro, H. S. (2001). JDesigner: A simple biochemical network designer. Available via the World Wide Web at <http://members.tripod.co.uk/sauro/biotech.htm>.