LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Document Type **LIGO-T980014-00 -** **E** Mar. 1998

Alfi - the GUI of the End-to-End model

Matt Evans, Edward Maros, Malik Rahman

and Hiro Yamamoto

Distribution of this draft:

xyz

This is an internal working note
of the LIGO Project.

California Institute of Technology
LIGO Project - MS 51-33
Pasadena CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project - MS 20B-145
Cambridge, MA 01239
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

WWW: http://www.ligo.caltech.edu/

1 ABSTRACT

Alfi (AdLib Friendly Interface) is an X windows application written to aid in the end-to-end modeling.

2 KEYWORDS

Alfi, GUI, End-to-End

3 OVERVIEW

The End to End model simulation program needs to read a file which describes the configuration of the system to simulate. This description file (also called box file in this document) may describe as simple as one digital filter representing an electronic box, or may describe as complex as the full LIGO system. Originally, the files were managed manually using a text editor, but soon it turned out to be to prohibitively complicated to develop and maintain without a program which provides the graphical user interface and other dedicated functionality.

Alfi was designed to offer nice interface to construct and maintain description files. It offers GUI for defining new modules, placing and connecting modules and constructing the configuration files which contains modules and other configuration files.

When dealing with very complex codes, GUI is not necessarily the best choice. E.g., handling a module deeply hidden in other description files is cumbersome using GUI alone. Alfi will offer features which overcome this kind of shortcomings of the GUI, so that it can handle complex systems with the ease of use of GUI without compromising the necessary power.

Internally, Alfi shares some of the C++ code with Adlib, the simulation engine of the End to End model, so that they can tightly integrate each other as much as possible.

Right now, the main focus of Alfi is the preparation of input files for Adlib engine. In the future, it will offer other frontends of the End to End model, including the display of the output data and the interactive manipulation of settings.

4 CONVENTIONS USED IN THIS DOCUMENT

With Unix, there are a variety of command interpreter shells available to the user. Instead of providing an example for each shell, only the C shell and the Bourne shell will have examples as most every other shell is a variation of one of these two. Bourne shell specific commands will be proceeded with the prompt "**bourne-sh>**". C shell specific commands will be proceeded with the prompt "**csh>**". If the command is not specific to either shell, then the prompt will be, "**sh>**".

When "**csh>**" appears at the beginning of a line, then the command that follows is specific to the C shell. When "**bourne-sh>**" appears at the beginning of a line, then the command interpreter is

the bourne shell. When "**sh>**" appears at the beginning of a line, then the command should be valid for any shell that you might be using.

5 FEATURE HISTORY

- Release 2.3.1
  Shift-Left-Click brings up a menu of parent windows.
- Release 2.3.0
  Search by "type of module" or "name of module". Select Edit-Find and fill in the dialog box.

6 GETTING STARTED

Before alfi can be run, several environment variables need to be properly set up.

PATH - This variable specifies the search path for commands to be executed.

csh> setenv PATH /home/e2e/Software/bin:${PATH}

bourne-sh> PATH=/home/e2e/Software/bin:${PATH}
bourne-sh> export PATH

E2E_PATH - This variable specifies the search path for the files that need to be included. The syntax is the same as for the Unix PATH variable where paths are separated by a colon. To search the current directory and /home/e2e/Software/lib:

csh> setenv E2E_PATH .:/home/e2e/Software/lib

bourne-sh> E2E_PATH=.:/home/e2e/Software/lib
bourne-sh> export E2E_PATH

With all of the environment variables properly set, start alfi, by typing "alfi" at the command prompt.

alfi> alfi

Two windows should appear. The larger one should look like figure 1.1 on page 6 and the smaller one like figure 1.2 on page 6.

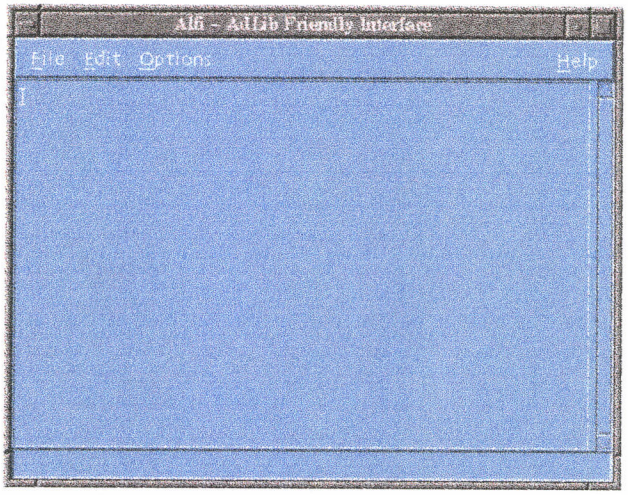

Figure 1.1

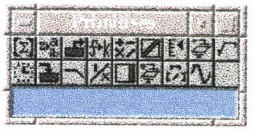

Figure 1.2

7 CREATING A MODULE

7.1. Creating a new Module

From the File menu, choose the Box option.

Type the name of the module in the dialog box that comes up (figure 2 on page 6). The name should not contain any file extension. Alfi will add the appropriate extension when the file is saved.

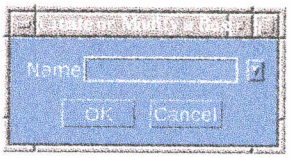

Figure 2

The edit canvas should appear on your desktop. The edit canvas gives either the internal or the external view of the module that is being edited. When the screen first comes up, the edit canvas

show the external representation (figure 3.2 on page 7).This is seen by the module appearing at

Figure 1.1

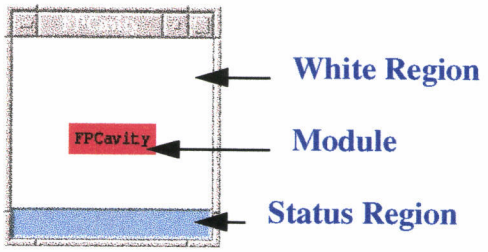

Figure 3.2

the center of the screen.

7.2. Making Ports

Ports are either inflows of information or outflows. They restrict what can be passed into a module and what can be passed out of a module. A module cannot directly access components of a sub-module.

From the external view, there are two ways to add a port. The first is by double clicking the right mouse button while the cursor is over the module. The second method is by bringing up the right mouse button edit menu while the cursor is in any white region (The status region should be blank). Refer to figure 3.2 on page 7 for regions.

Whichever method was used, the "Add Port" dialog box should appear (figure 4 on page 7).

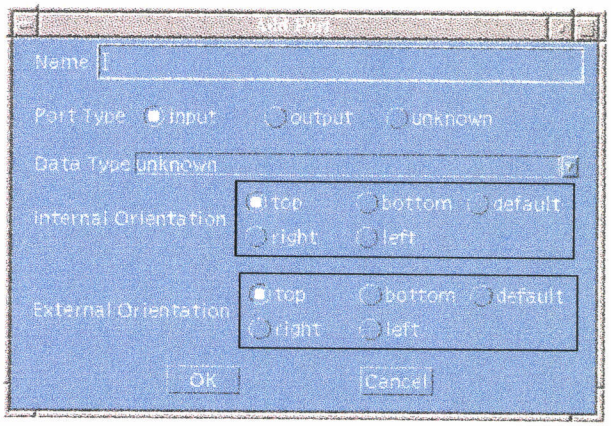

Figure 4

In the "Name" filed, select a name that will be unique to this port type. For instance, if you have an input port known as "0" and an output port known as "1" and you are creating a new input port, you may choose any name other than "0".

The "Port Type" field is a radio button selection with three choices:

Input - Data flows into the module by means of this port.

Output - Data flows out of the module by means of this port.

Unknown - Although this option is available, it should not be used. The designer should know if a port is to be an input or output port.

The "Data Type" field is a combo box from which the designer can choose the format of the data to be passed though this port. Ports that are of type "unknown" can be connected to any data types. All other data types are restricted to being connected to ports of like data types.

The "External Orientation" and "Internal Orientation" fields are radio buttons which specify where the port is to be drawn in respect to the current view. The "default" option allows the system to decide were to place the ports. The current default layout scheme places input ports on the left and the output ports on the right.

Selecting the "Ok" button at the bottom commits the changes, while selecting the "Cancel" button aborts the changes. If the changes were committed, then the view will be redrawn, showing the new port.

7.3. Adding Components

Components can only be added from the internal view. To go from the external view to the internal view, select the "Toggle View" option from the right button edit menu (figure 1.1 on page 7).

To add a component, there are two methods. The first method is to use the tool bar (figure 1.2 on page 6). Simply click on the icon which represents the component to be added. The component will be added to the top-left hand corner. If multiple components are added, they will overlay one another.

The second option is to use the "Primitives" or "Boxes" menus. The "Primitives" menu is activated with the left mouse button in any white area of the internal view. The "Boxes" menu is activated with the control key down and the left mouse button. Modules that would result in a cycle cannot be selected and appear stippled in the menu. Select the item to be inserted by moving the cursor to highlight the choice and then releasing the mouse button. Using this method, the center of the module will be placed at the position of the mouse when the menu was activated.

7.4. Making Connections

Connections are pathways for data to flow from one component to another. Data must flow from a source to a sink. Sources are either input ports of the module being constructed or the output ports of the components contained by the module. Sinks are either output ports of the module being constructed or the input ports of the components contained by the module. The data type of the source and the sink must be compatible.

To construct a connection, start at either a source or a sink and click the left mouse button. The name of the port and its data type shall appear in the status region when the cursor enters the port. Move the mouse in the first direction that connection should go (left or right for vertical and up or down for horizontal). The connection is a series of 90 degree bends connected together. To create an anchor point, click the left mouse button and move in the initial direction of the new segment. If the initial mouse movement was not in the desired direction, press the space-bar to toggle 90 degree bend. (Note: Currently the change doesn't appear until the mouse is moved). At any time prior to completing the connection, the connection can be aborted by pressing the ESC key.

To complete the connection, the cursor must be over an appropriate port. The system performs several test to validate the port. If the port isn't considered valid, an audio prompt is sounded and the line continues to follow the cursor. If the port is validated, then the connection is created and the screen refreshed.

8 SAVING AND RETRIEVING A MODULE

From the "File" menu, select the save option. This writes out all module files used in this instance of the application. If there are changes that are to be saved for one module and not for another, currently there is no facilities to support saving one and not the other. There will be no prompt boxes for additionally information or completion status.

WARNING: If a write request fails for any reason (read-only file, permission denied, system full, etc.) no warnings or errors are produced.

To retrieve a saved module, select the open option from the "File" menu. A file dialog box will appear. Use it to navigate through the file system till the desired file is located. To view the contents, select the module using the box option from the "File" menu.

9 EDITING DRAWINGS

9.1. Components

9.1.1. Renaming Components

Have the cursor on the component to rename. Select the object with the right mouse button. Select "Rename" from the popup menu. In the dialog box that appears, type the new name of the component. The name must be unique to the module.