Comview User's Guide

Version 1.0

July 19, 2000

Contents

1 Introduction

comview is a tool designed for visualizing the message-passing communication that occurs between modules in IPC-based systems. It can serve as a useful tool for debugging and monitoring a IPC session (while the session is active, or for later analysis).

comview uses a log file generated from a session with IPC's central server. For each message, central logs the source and destination modules, the time, the name, and type of the message. This log file is parsed and processed by comview to generate a graphical display showing the communication that occurs between modules over time. The contents of the log file are also shown within a section of the display.

The graphical display consists of widgets that can be selected to see more information about the chosen activity. There is a link between each widget in the graphical display and the message it corresponds to in the log file. Selecting an object in the graphical display will highlight the corresponding line within the log file, and vice versa.

comview is designed to make it easy to search for particular types of message activity, through a number of different mechanisms. A list of all the names of messages sent is maintained, in addition to a list of all modules that have sent or received messages. These lists can be viewed and individual messages from them can be selected. Additionally, the log file itself can be searched using a simple text-search. Configuration files can also be used to suppress the display of particular messages or modules.

2 Starting Comview

Since comview uses Tcl/Tk, the location of these libraries must be specified in the user's environment variables (TCL_LIBRARY and TK_LIBRARY). Currently, comview uses Tcl version 7.4 and Tk version 4.0.

2.1 Logging with central

To create a log file, the IPC central server must be run with at least the following logging options: "-Lmtir -f<logfile>" (m=message traffic, t=time messages are handled, i=ignore logging registration and deregistration messages, and r=log the reference id as well as the message name). You may use any other options except "-Lx" (no logging). Type "central -h" for more information on central's command line options.

In order to be able to view the data associated with messages while using comview you must additionally give the "d" logging option to the central server (e.g. "-Ldmtir").

2.2 Command line options

comview is run as follows: comview [-options] -f <logfile>

```
   lung% comview -h
-h :  Print this message
-v :  Print version information
-l :  Do not display all log file information (only messages)
```

```
-f <logfile>:  Use the logfile
-z <zoom>:  initial zoom value
```

The command line options "-h" and "-v" are self-explanatory. The "-l" option refers to the part of the comview display (the textual log-window) that displays lines directly read in from the IPC log file. The "-l" option indicates that the log-window should contain only lines that correspond to valid messages in the graphic display. Without this option, the log-window will contain all the information that IPC logs, such as the data associated with individual messages.

The "-z" option is for specifying an initial zoom value (a floating point number). For example, a zoom factor of 100 produces an initial canvas width of 1 minutes 10 seconds. The default setting (specified in options.tcl) is 10.0, which results in an initial canvas width of 7 seconds.

3 Using Comview

3.1 Stepping and Running

comview displays information in two places: in a textual log window at the bottom of the display and in the larger graphical trace window called the canvas display (named after the Tk object which is used to create it). Figure 1 shows a snapshot of the entire comview screen.

To visualize message traffic, one can either "Step" through or "Run" through the log file. Stepping will read in a single line from the log file, parse it, and display activity in the log window and canvas display. Running through will continuously step through the log file until "Run" is hit again, or until the end of the log file is reached.

comview can use a log file that is being produced while it is running. If the end of the log file is reached then the "Run" button will change from green (running) to yellow (waiting). If more data is written to the log file, comview will resume running.

The canvas display expands and scrolls horizontally to follow incoming data. The automatic scrolling feature is toggled by clicking on the "Scroll-Lock" button. With "Scroll-Lock" on, the canvas display does not scroll, but the graphics are still updated as messages are read in. The initial setting is for the display to scroll so that newly created activity is visible.

The log window consists of the individual lines from the log file. Unless the "-l" option is given to comview, this consists of all lines from the log file, such as those that contain the data associated with a message, and lines which report on the internal performance of IPC execution. If the "-l" option is given to comview, then the log window will display only lines from the log file that indicate message activity.

When the log file indicates that a new module has connected with central, a label for that module will appear on the left side of the screen. As the log file reports that a module sends or receives messages, the color of the module label changes to indicate what kind of message the module is currently processing or what kind of message that module has just sent.

As different messages are sent from or received by a module, a colored rectangle is added to the right side of the module's display line. The width of the rectangle is proportional to

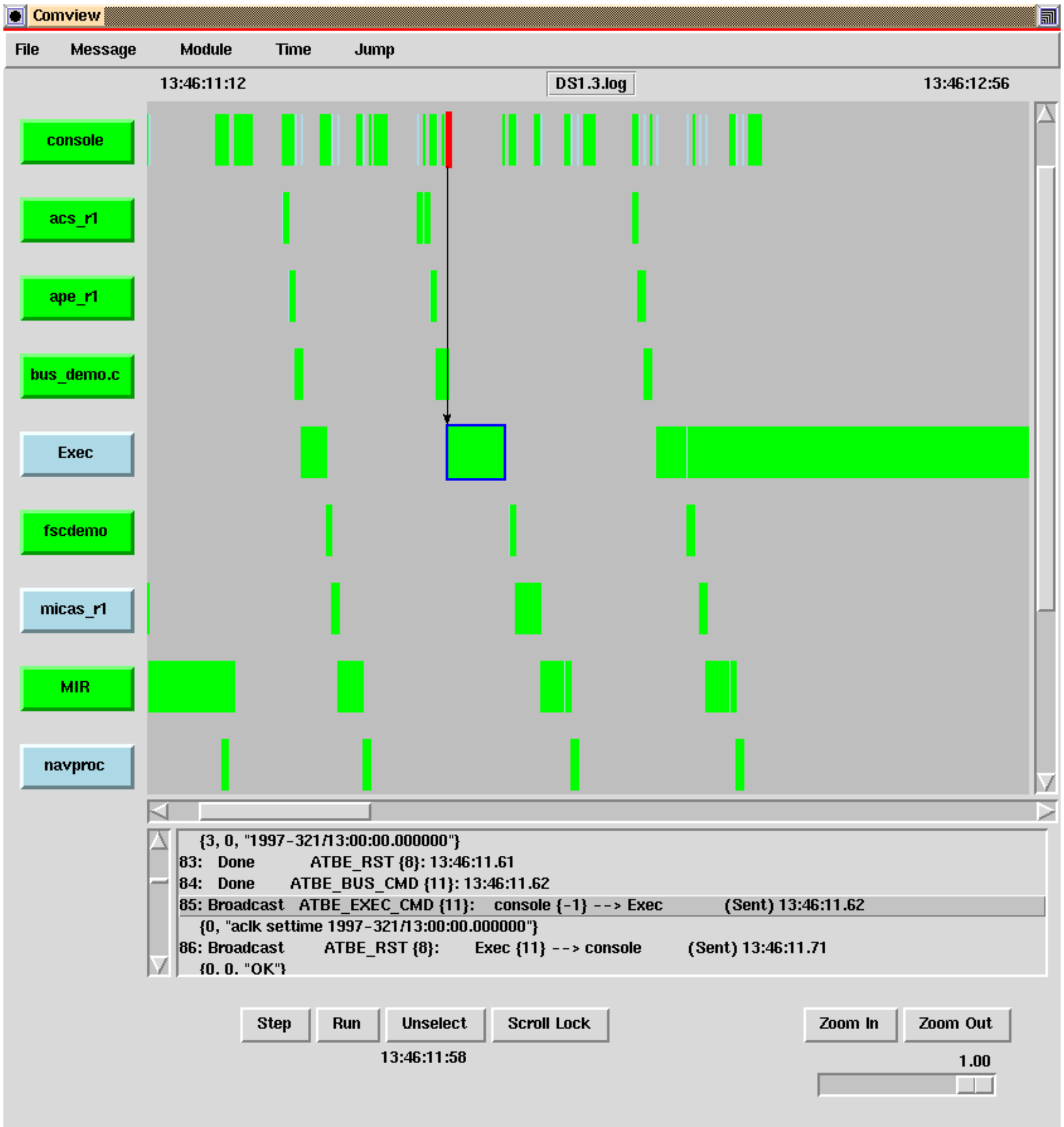

Figure 1: Comview Display

the time it takes for the message to be handled.

The various colors in the display have the following meanings:

message type	*sender's color*	*recipient's color*
Query	Yellow	Green
Inform	Green	Dark Green
Command	Light Blue	Blue
Broadcast	Light Purple	Purple
Goal	Light Olive	Olive

These colors are specified (and can be changed) from the file "options.tcl". The colorings allow you to look at a segment of activity for a particular module and be able to identify the state of the module at that time (e.g., waiting for a message, or processing a particular message type).

3.2 Selecting Objects

Any of the objects within the canvas can be selected by clicking in the canvas window with the left mouse button. This will result in arrows being drawn which show the communication pathways of the chosen activity. One can also double-click on a line in the log window and the appropriate activity (if any) within the canvas window will be highlighted.

Blocking and non-blocking messages will result in different types of highlighting. For blocking messages (such as query messages) the module that sent the message is going to wait for a reply. In the log file there will be a reply message. Graphically, highlighting will draw an arrow from the source module to the destination module at the time at which the query was sent. At the time at which the reply was sent an arrow will be drawn from the destination module to the source module. This is shown in figure 2.

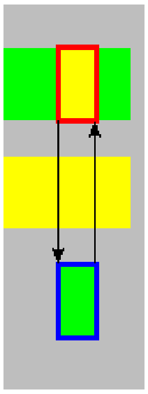

Figure 2: Highlight of Query Message

In contrast, non-blocking messages (such as broadcast or inform messages) are sent and do not wait for a reply. Graphically, this is represented as a short segment of activity (of width 1/100th of a second) for the source module. The destination module becomes active

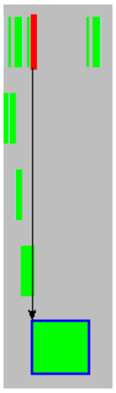

Figure 3: Highlight of Broadcast Message

and will remain active until it completes handling the given message (this is identified within the log file by a "Done" message). Figure 3 shows highlighting for an inform message.

If the central server receives a message for a module that is currently processing another message, it queues these messages until the module has completed handling its current activity before sending the next message. Graphically, this is represented by creating an orange "pending bar" above the other activity for the module (again, the color of the pending bar is reconfigurable). The pending bar will remain until all queued messages have been handled. Unfortunately, `comview` currently has no way to identify when more than one message is pending for a given module.

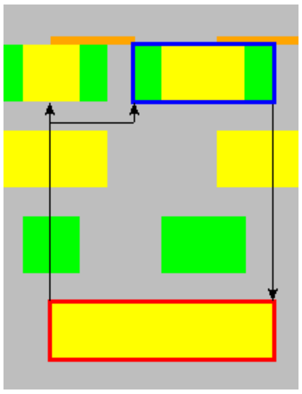

Figure 4: Highlight of a Message That Pends

Figure 4 shows the calling structure for a query message that pends. The first vertical arrow indicates when the message is sent, the horizontal arrow indicates when the message is forwarded to the module to be handled, and the downward pointing arrow corresponds to the reply message. A similar graphical display is created for non-blocking messages (but, without the reply arrow).

3.3 Rearranging Modules

The top-to-bottom order of the modules in the graphic display can be changed (by default, they appear in the order that the connect to central). This is done by pressing the left mouse button on a module name and then dragging it to the new position. This will shift the order of the other modules to fit the moved module to the new location.

3.4 Getting Around the Canvas

`comview` has several mechanisms to obtain different views of the canvas. There is a scroll-bar at the bottom of the canvas for manually scrolling through the canvas. Clicking on a line in the log-window selects the corresponding object, and jumps the canvas window to make the selected object visible. Zooming provides, as expected, a means of scaling the objects on the canvas.

3.4.1 Zoom

The "Zoom In" and "Zoom Out" buttons (Figure 1) will zoom, centered around the currently selected object. If no object is selected, then the zoom will be about the center of the current view of the canvas. If an object is selected and you want to zoom about the center of the canvas, use the "Unselect" button before any zoom.

The zoom slider widget is another way of zooming about the canvas. When set to 1.0 the entire canvas is visible (maximum zoom out). Set to 0.0 the canvas is zoomed in to a fixed scale where 1 pixel corresponds to 1/100th of a second (it is possible to zoom in more by using the "Zoom In" button).

3.5 Time Displays

There are three displays of time on the screen. The clock below the "Unselect" button (Figure 1) shows the time that corresponds to the mouse position when it is over the canvas. Above the canvas on the top-left and top-right are two more clocks that indicate the time that corresponds to the leftmost visible part of the canvas and the rightmost visible part of the canvas.

These three clocks can be displayed in absolute units (the time-stamp that central logs with every message) or relative time units (such that the time at the leftmost point in the canvas will be 00:00:00.00). The "Time" menu has commands for toggling between the units.

4 Menu Items

At the top of the `comview` display is a menu bar with five items: "File", "Message", "Module", "Time" and "Jump". This section describes each of the menu choices, in turn.

4.1 File Menu

4.1.1 Saving canvas to postscript

The canvas can be saved to postscript using the "Print Current View" option under the "File" menu. This will open a window prompting for an output filename. The view saved is the currently visible portion of the canvas window. The view will be saved in landscape orientation.

4.2 Message Menu

Once a log file has been loaded, users will typically need to search through the log file to find particular messages or events. `comview` provides a number of different features that make this possible.

The "Message" menu consists of the names of all the different types of messages that have been sent. This menu item is updated as more of the log file is read in. Selecting a particular message from this menu will bring up a new window that lists all the occurrences of that message type, as well as the number of occurrences of that message (this list is updated as more of the log file is processed). Any particular instance can be selected and the corresponding activity within the graphic display will be highlighted.

For example, Figure 5 shows the sixty-four ATME_RST messages that were sent (in a scrollable window). The highlighted message is line 2 of the log file, and indicates that the message was broadcast from module acs_r1 to the console module at time 14:34:56.21.

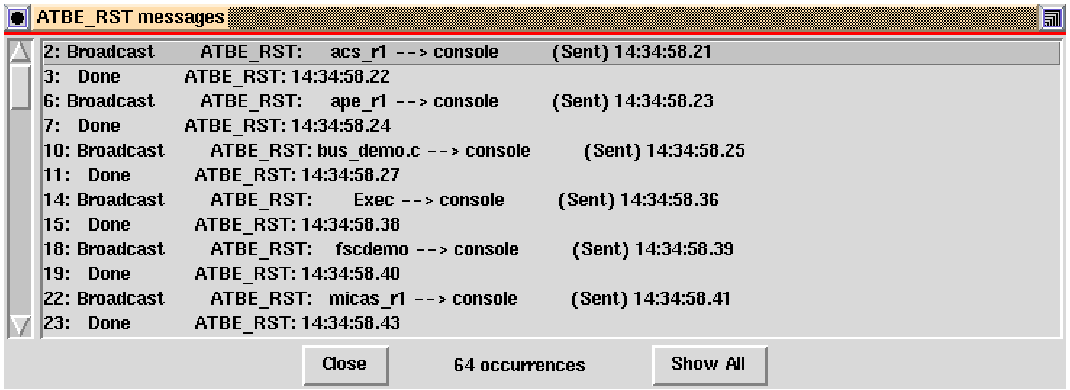

Figure 5: Message List Window

The "Show All" button on the bottom of the message list window highlights all the occurrences of that message name within the canvas. Figure 6 shows a view of the canvas after "Show All" has been selected for the ATBE_RST messages shown in figure 5.

4.3 Module Menu

In a similar manner, the "Module" menu lists all the different modules that have connected to the central server. Selecting a particular module from this menu opens a new window that lists all the messages that have been sent to, or from, the chosen module, as well

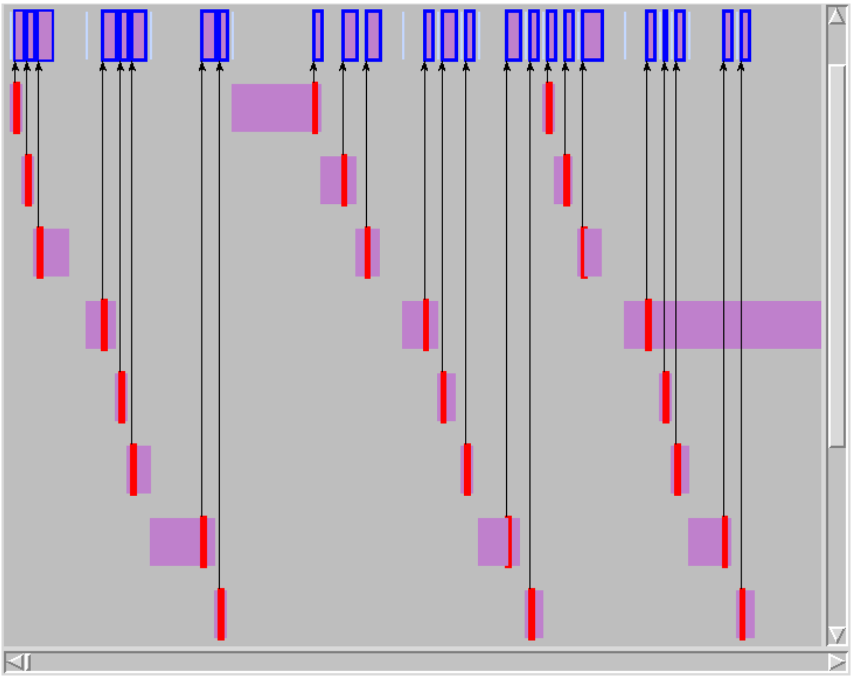

Figure 6: Canvas View After "Show All"

as the number of message instances that involve the chosen module. Again, this list is updated as more of the log file is processed. As with the "Message" menu these messages can be individually selected, or all of them highlighted at once using the "Show All" button. Figure 6 shows the window that is created when the "console" module is selected from the "Module" menu.

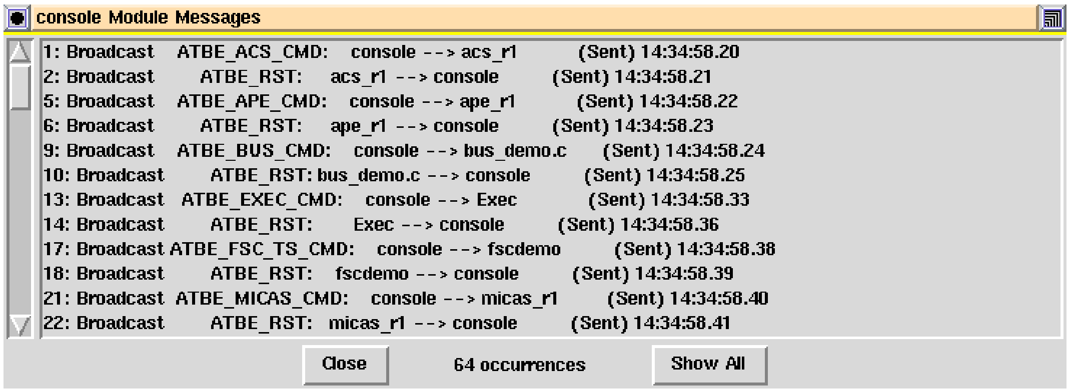

Figure 7: Module List Window

4.4 Time Menu

The first two options in the time menu, "Absolute" and "Relative" toggle the units that are used for the three time displays. "Absolute" selects the units that corresponds to the time-stamp that central logs with every message. Relative units are a measure of time since the first message was logged (such that the time at the leftmost point in the canvas will be 00:00:00.00).

The "Display Visible Time" menu option toggles whether the time at the top-left and top-right positions above the canvas are visible (this may be useful on slower machines).

4.5 Jump Menu

The "Jump" menu is used to easily access particular messages in the log and canvas windows. The "Jump" menu has two features: A selection to jump to a given time, and a selection to jump to a specific message number.

Selecting "Time" from this menu will bring up a new window that shows the time of the first logged message and the time of the last logged message (see Figure 8).

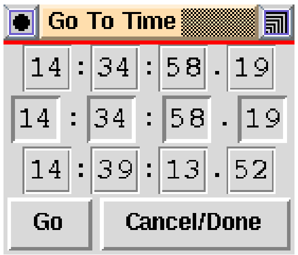

Figure 8: Go-To Time Window

Any time between those two times (inclusive) can be entered. When "Go" is pressed the canvas will be shifted so that it is centered around the chosen time.

The "Message Number" option opens a new window that shows the maximum message number and has an entry in which to type a message number to jump to (see Figure 9). The chosen message will be highlighted and centered in both the canvas window and the log window.

Figure 9: Go To Message Window

Both of these jump commands can be performed using keyboard-only commands (next section).

5 Keyboard Shortcuts

There are a number of keyboard shortcuts that allow you to step through the log file and execute several of the same commands that you can select via the menus. The following table describes them:

Keyboard Control

function	*keypress*
highlight next message	control-n
highlight previous message	control-p
highlight first message	meta-p
highlight last message	meta-n
scroll forward one screen on canvas	spacebar
scroll backward one screen on canvas	delete
run	meta-r
step	meta-s
zoom in	ctrl-i
zoom out	ctrl-o
goto message number	meta-g
goto time	meta-t
search log window	ctrl-s
reverse search log window	ctrl-r
mini-buffer quit	ctrl-g

6 Searching the Log Window

`comview` includes a feature for textually searching through the processed log file. This feature is accessible only via the keyboard (there is no menu option for it). One can search the log file from the current position to the end. This is invoked by pressing Control-S, and it operates in a manner similar to an emacs mini-buffer (see Figure 10).

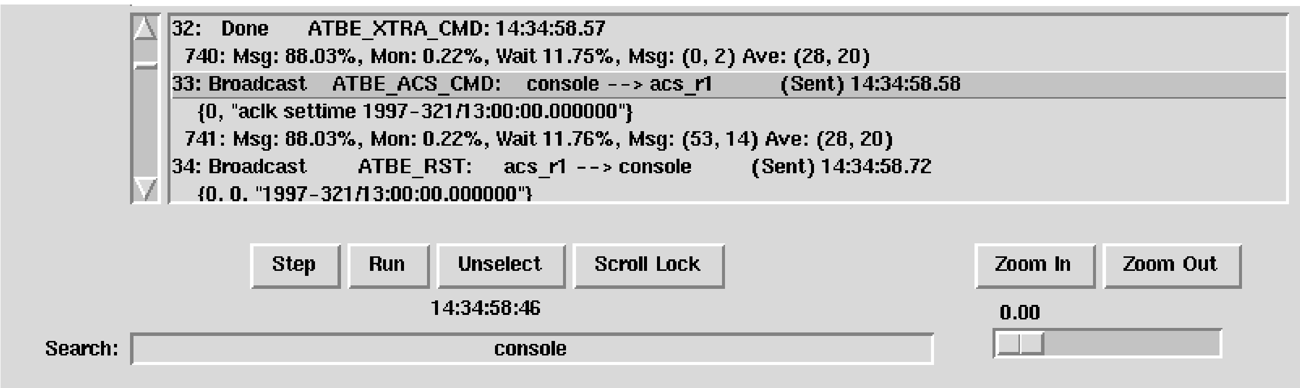

Figure 10: Searching the Log Window

If the search fails, the mini-buffer will blink. If the search is successful, the highlight-bar in the log file will move to the line that contains the text which matches the search string. Pressing Control-S again searches from the new position to the end of the log file. Control-G terminates the search.

In a similar manner, Control-R will search the log window in the reverse direction.

Meta-P moves the current log window selection line to the first message in the log window. This highlights the first message and shifts the canvas so that the appropriate graphic is visible in the canvas window. Any search will now begin from the top of the log file. To jump to the end of the log file, press Meta-N.

7 Settings File

Certain features of `comview` can be specified in a settings file. This file must have the same filename as the log file, except with the extension ".set" instead of ".log" (i.e. "test.log" would have a settings file "test.set"). At present, the settings file can indicate to ignore displaying of certain messages and/or certain modules.

7.1 Filtering Out Specific Messages

`comview` has support for filtering out specific message names for a given log file. The messages to filter out are listed in separate lines in the settings file. The settings file should have lines of the form:

ignore_message message_name1
ignore_message message_name2

`comview` will then ignore any messages with the name message_name1, message_name2, etc. These messages will neither appear in the log window nor will they result in the creation of any graphical activity in the canvas. Note that the settings file must exist at the time `comview` is started, and any changes after `comview` has been started will not be recognized.

7.2 Ignoring Modules

In a manner similar to the way messages can be ignored, messages that involve a specific module can also be filtered out. All messages that are sent to or from the selected module will be completely ignored (they will not result in any graphical activity, and will not appear in the log window).

The settings file should contain lines of the form:

ignore_module module_name1
ignore_module module_name2