



OPENTTCN

OpenTTCN GFT Log Viewer Tutorial OpenTTCN Tester 2011

VERSION 4.1.6 – April 25, 2012

Contents

1	OVERVIEW OF OPENTTCN GFT LOG VIEWER IN OPENTTCN TESTER 2011	3
2	FUNCTIONALITY OF OPENTTCN GFT LOG VIEWER	4
3	USING EXAMPLE TEST SUITE	5
3.1	OBTAINING GRAPHICAL LOG FOR THE EXAMPLE TEST CAMPAIGN	5
3.2	DESCRIPTION OF TEST CASES IN THE EXAMPLE TEST SUITE.....	7
3.3	OPENING LOG VIEW PERSPECTIVE FOR THE EXAMPLE TEST CAMPAIGN	8
4	VIEWING EXAMPLE LOGS	9
4.1	MESSAGE-BASED, PROCEDURE-BASED, AND TIMERS	9
4.2	CONCURRENT EXECUTION INVOLVING PARALLEL TEST COMPONENTS	12
5	CONTACTING OPENTTCN	14

1 Overview of OpenTTCN GFT Log Viewer in OpenTTCN Tester 2011

OpenTTCN Tester 2011 is a TTCN-3 test development and execution package built by integrating popular OpenTTCN Tester product and widely used Eclipse framework providing an easy to use TTCN-3 editing, compilation, and execution environment.

Two major new features of OpenTTCN Tester 2011 are TTCN-3 debugger and GFT (Graphical Presentation Format) log viewer.

OpenTTCN Tester 2011 combines new OpenTTCN GFT Log Viewer component that provides the capability to store logs in XML format and view them in graphical, structural, and text form. This tutorial explains the basic features of the GFT Log Viewer and provides examples of its use.

The basic structure of this guide is as follows: first, we demonstrate graphical log for a simple example test case involving message-based communication and timers. Then we demonstrate graphical log for procedure-based communication and parallel execution involving multiple parallel test components.

2 Functionality of OpenTTCN GFT Log Viewer

OpenTTCN GFT Log Viewer allows viewing logs recorded during test campaign execution in graphical format. TTCN-3 test campaigns are primarily supported. Design of the graphical log output takes into account graphical presentation format guidelines set forth by the ETSI ES 201 873-3 V3.2.1 (2007-02) “TTCN-3 Graphical presentation Format (GFT)” standard.

Logs suitable as a data source for graphical rendering performed by the Log Viewer are stored in XML format in a persistent repository in the file system in the Eclipse workspace. Persistent repository is implemented as a collection of plain folders and files for easier processing and sharing of logs. Logs for individual test cases are stored in XML format that complies with ETSI ES 201 873-6 V4.2.1 (2010-07) standard XSD schema definitions as defined in clause 11.4.2 “The TCI-TL interface” and Annex B “XML mapping for TCI TL Provided” of the standard.

Graphical log and related structured log are displayed in the same collection of views of the Log View Eclipse perspective. Correspondence between elements of graphical log and structured log can be established by clicking on elements of the graphical log. Detailed information about selected log event is displayed in the *Log event* view.

This is the first Technical Preview of OpenTTCN GFT Log Viewer. The following graphical elements of the GFT log are implemented in this preview:

- message-based communication: send and receive operations, enqueued message indication;
- timer operations: start timer, stop timer, timer timeout;
- procedure-based communication: call, getcall, reply, getreply, raise, catch;
- verdict operations: setverdict;
- display of ports and elements of port arrays that are actually involved in communication operations;
- display of functional diagrams for individual test components.

3 Using Example Test Suite

We will be using the `gft_example` project included in the OpenTTCN Tester distribution. To access the example, you must first import the TTCN-3 code into the workspace. When first starting OpenTTCN Tester, the welcome screen contains the option to *"Create example workspace"*. Select this option to create the example projects including the `gft_example` project. If you have an existing workspace, either create a new workspace for experiments, or select *Help -> OpenTTCN Tester Welcome Screen* to access the welcome screen again, and create the `gft_example` project.

The `gft_example` project is a simple test suite consisting of one test module. You can run test cases included in the project by selecting the `gft_example` project launch configuration from the list of launch configurations. The test campaign should execute and you should receive pass verdicts.

3.1 Obtaining Graphical Log for the Example Test Campaign

Prior to running test cases, we need to make sure that **XML logging and log storing is enabled for the `gft_example` project launch configuration**. Logs are stored in the log repository as a collection of directories in the file system. Log repository is used by the GFT Log Viewer as a data source for graphical rendering of diagrams and their output.

Log storing can be enabled by opening launch configuration settings for the `gft_example` as illustrated by Figure 1 and ticking the corresponding checkbox as illustrated by Figure 2.

Enabling this option is necessary for obtaining graphical logs of test campaigns.

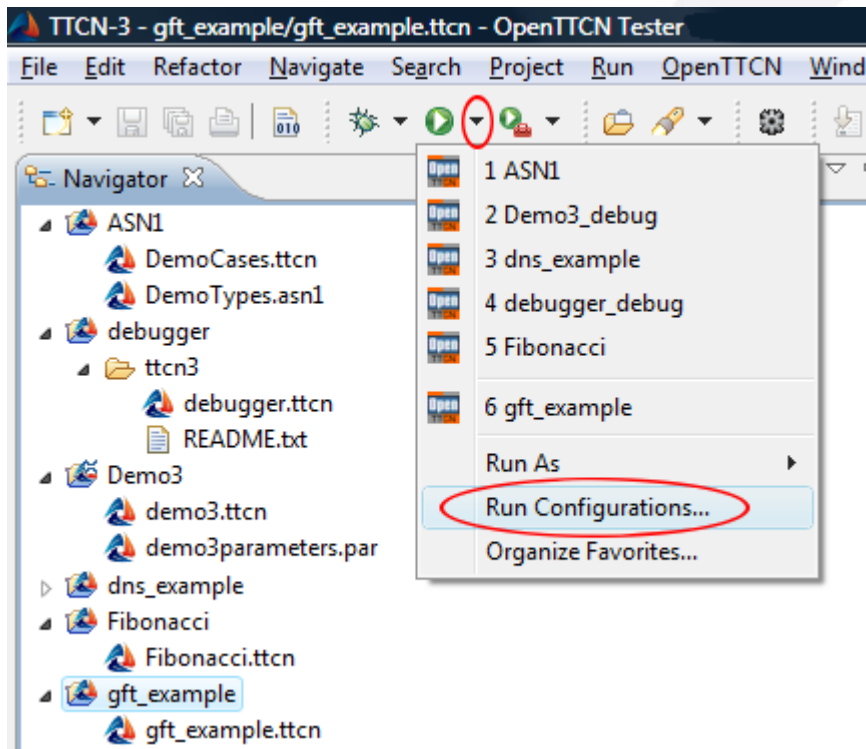


Figure 1. Opening launch configuration for gft_example

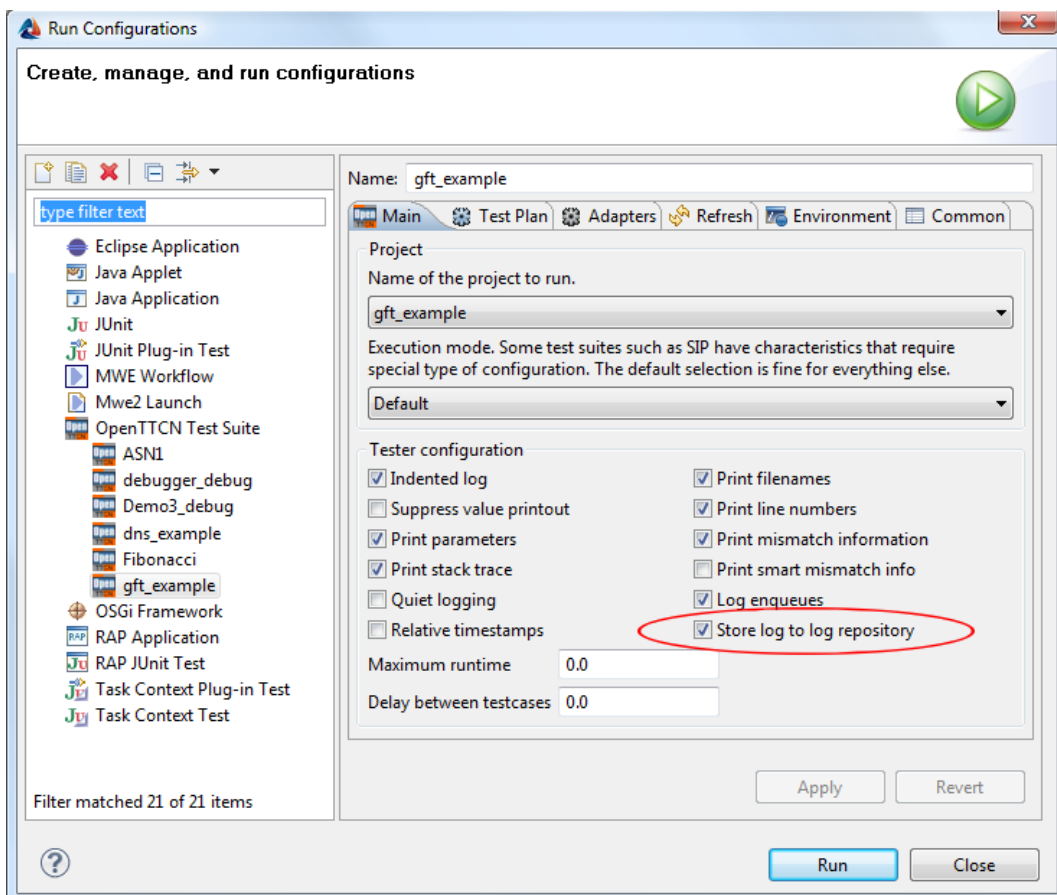


Figure 2. Enabling log storing in the log repository

Now we are ready to launch. Since the `gft_example` comes with a ready-made launch configuration, simply select the `gft_example` launch configuration from the list of launch configurations and run it.

3.2 Description of Test Cases in the Example Test Suite

There are three test cases in the `gft_example` test suite:

- *TC_Simple* demonstrating message-based communication and timer operations;
- *TC_Procedural* demonstrating procedure-based communication;
- *TC_Concurrent* demonstrating parallel execution involving multiple test components.

In *TC_Simple* we:

- start the `T_GUARD` timer (line 65);
- activate the `DefaultAltstep` altstep to catch all kind of unexpected input messages and unexpected timeout events (line 66);
- perform mapping of main test component ports to test system interface ports (lines 68-69);
- send a couple of messages to ports of the loopback adapter (lines 71-73);
- start the `T_wait` timer (line 75);
- wait for receipt of a message on port `p1` (lines 77-88);
- wait for receipt of a message on port `ps[1]` (lines 90-101);
- wait for the `T_wait` timer timeout event (line 103);
- stop the `T_GUARD` timer (line 104).

Notice that message sent on line 72 to port `ps[0]` never comes back, because this port is not mapped to a test system interface port.

In *TC_Procedural*, in addition to basic test case setup operations similar to those described for *TC_Simple*, we:

- perform a procedural call (line 125);
- wait for a procedural call from the loopback adapter (lines 128-139).

In *TC_Concurrent* we:

- create two test components (lines 167, 170);
- connect ports of the main test component to ports of created parallel test components (lines 168, 171);
- start behaviour functions on created test components (lines 173-174);
- send messages to started parallel test components from the main test component (lines 176, 179);
- wait for response from parallel test components (lines 177, 180);
- wait for all parallel test components to terminate (line 182);
- set the pass verdict (line 183).

3.3 Opening Log View Perspective for the Example Test Campaign

To be able to see GFT graphical logs generated during test campaign run, we need to switch to Log View perspective from TTCN-3 perspective in Eclipse. This switching is illustrated by Figure 3.

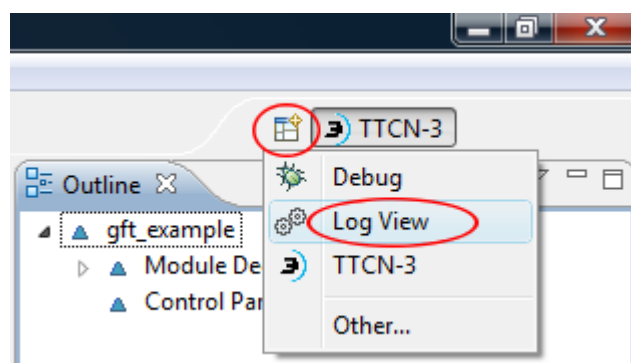
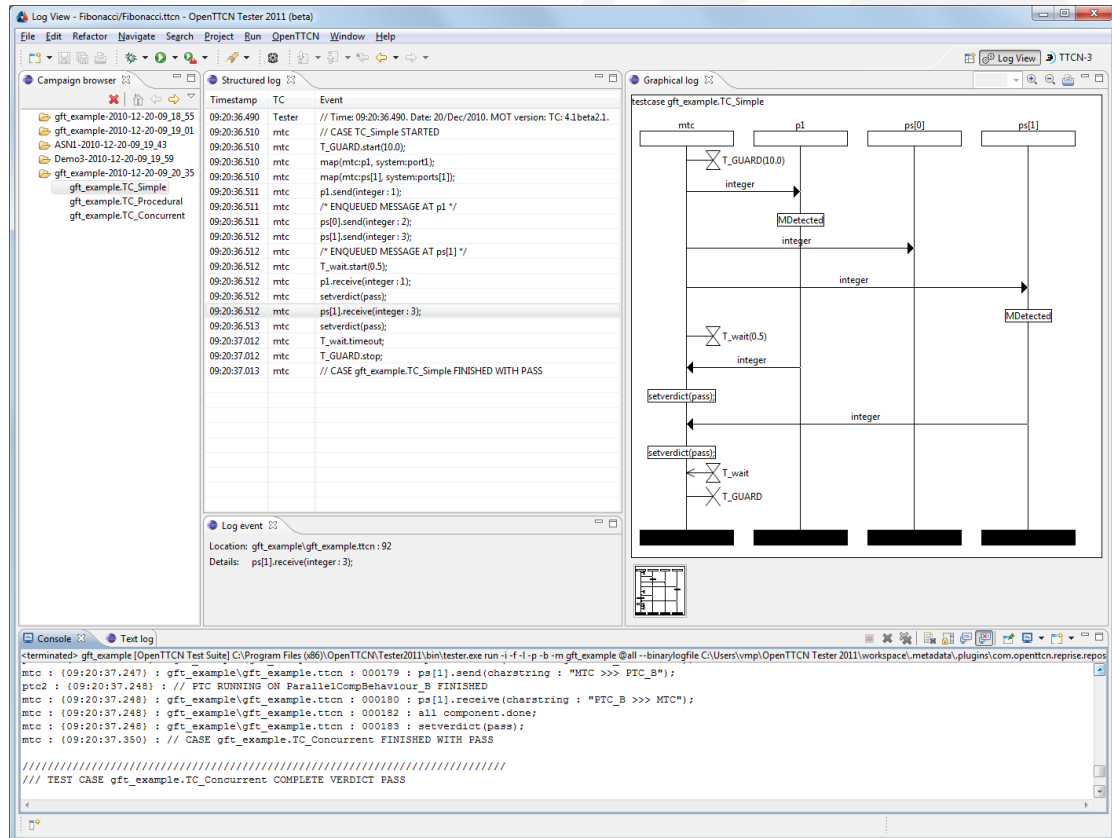


Figure3. Switching from TTCN-3 perspective to Log View perspective

4 Viewing Example Logs

4.1 Message-based, Procedure-based, and Timers

After Log View perspective is selected, you can view campaign logs as illustrated by Figures 4 and 5.



Figures 4. Graphical log: Message-based communication and timers

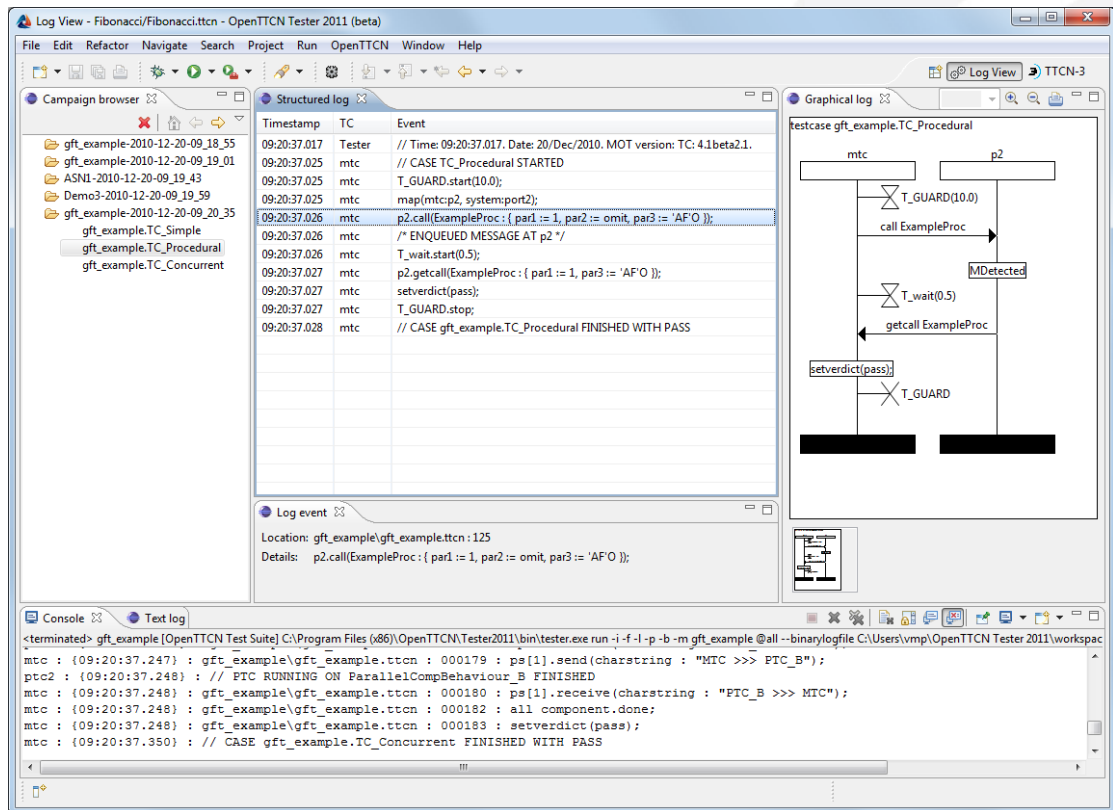
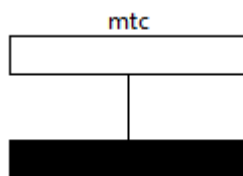


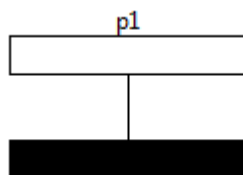
Figure 5. Graphical log: Procedure-based communication

Clicking on an event in the *Graphical log* panel selects corresponding event in the *Structured log* panel and updates the *Log event* panel accordingly.

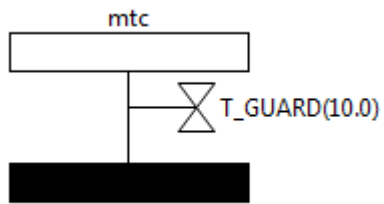
Elements of graphical log diagrams presented in Figures 4 and 5 are shown in detail in Figure 6.



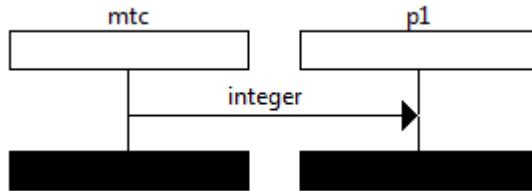
(a) Main test component instance



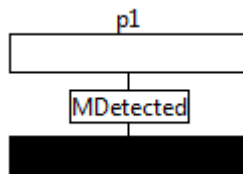
(b) Port instance



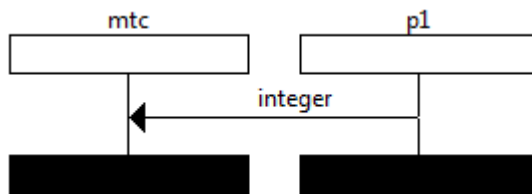
(c) The start timer operation



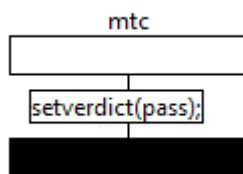
(d) The send operation



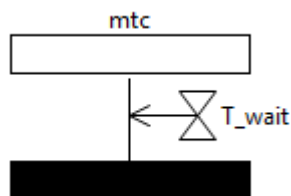
(e) The enqueued message indication



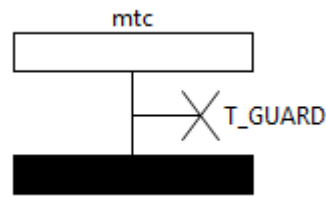
(f) The receive operation



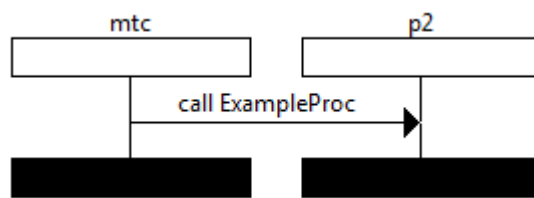
(g) The setverdict operation



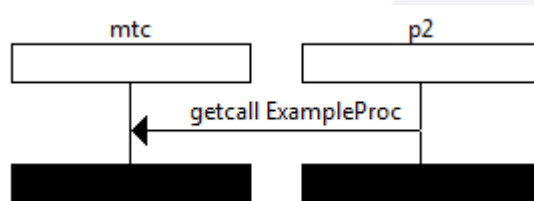
(h) The timeout operation



(i) The stop timer operation



(j) The call operation



(k) The getcall operation

Figure 6. Magnified view of elements of graphical log diagrams

4.2 Concurrent Execution Involving Parallel Test Components

To obtain a graphical log diagram for a particular test component in a test case run involving multiple parallel test components, select the corresponding test case in the *Campaign browser* panel, then expand the test case to show the list of test components created during the test case execution, then select the test component in question to show its graphical log representation as illustrated by Figure 7.

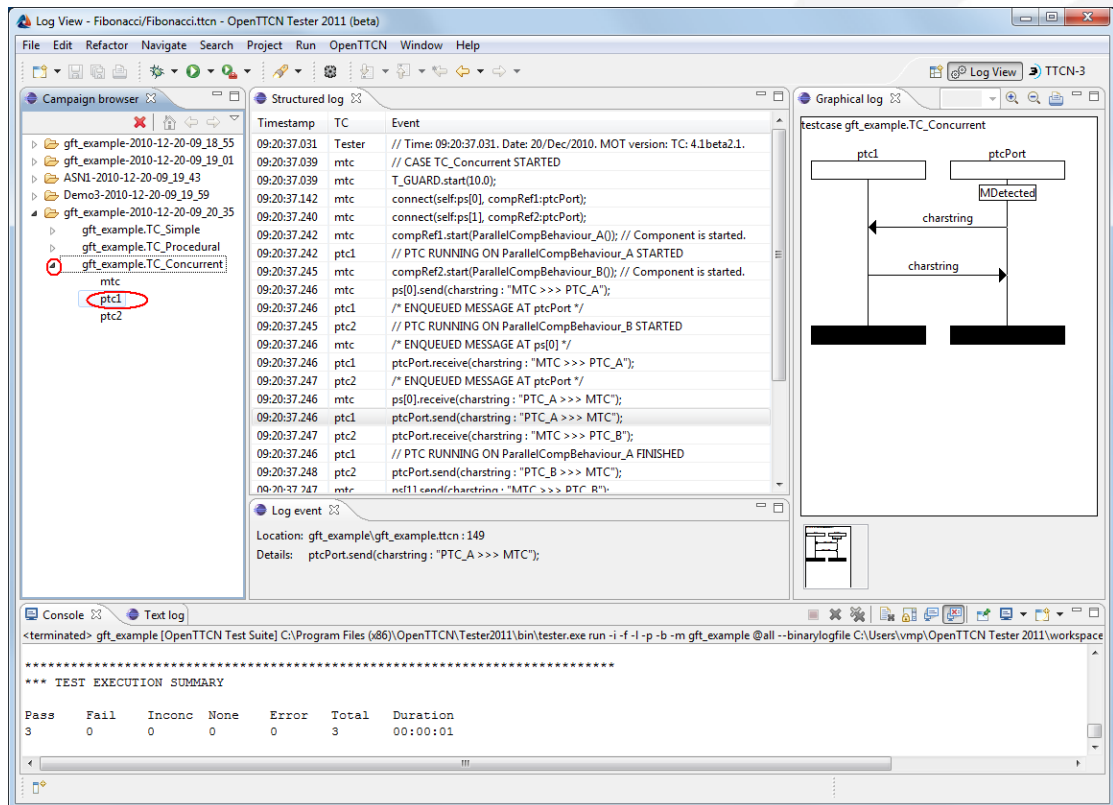


Figure 7. Obtaining graphical log diagram for a parallel test component

5 Contacting OpenTTCN

Please give us your feedback about OpenTTCN GFT Log Viewer and OpenTTCN Tester 2011 in general by sending your questions using support@openttcn.fi e-mail address.

Latest user guides and training course materials can be found from:

<http://www.openttcn.com/support/user-guides>

Tutorials and articles about OpenTTCN use and programming can be found from:

<http://wiki.openttcn.com>

You can contact OpenTTCN sales by sales@openttcn.fi e-mail.

OpenTTCN wishes you a good time using OpenTTCN Tester 2011!