Chapter

29

The TTCN-SDL Co-Simulator (on UNIX)

This chapter describes the TTCN-SDL Co-simulator in the UNIX version of the TTCN suite. It gives an introduction to the concept of the TTCN-SDL Co-simulator, as well as a guide to its functionality and an introduction to using the TTCN-SDL Co-simulator for controlling real Executable Test Suites.

Note: UNIX version

This is the UNIX version of the chapter. The Windows version is chapter 34, *The TTCN-SDL Co-Simulator (in Windows)*.

The TTCN-SDL Co-Simulator

The TTCN-SDL Co-simulator allows you to:

- Execute a compiled test suite and inspect the results.
- Execute test cases and test groups by stepping through the TTCN lines or executing at full speed with the possibility to set breakpoints at given tables/lines.
- View the actions performed by the various parallel test components (PTCs) in multiple windows if needed when executing a concurrent test suite.

The possibilities to pinpoint the cause of the test result beyond the test verdict *fail* are vast, and will undoubtedly help in improving the time needed for testing in the development phase.

Performing a Co-Simulation

Given a test suite containing a set of tests you want to execute together with a simulated SDL system, the following steps are to be performed:

1. Analyze the test suite and make sure it contains no errors. (This can be done using the TTCN suite *Make* dialog.)
2. Using the TTCN suite *Make* dialog, build an executable test suite with the *Use standard kernel* radio button set, and with the *Co-Simulation* kernel selected.

3. Start the TTCN-SDL Co-simulator by pressing the co-simulation quick-button on the top of the TTCN Browser, or selecting *Start Co-Simulator* from the Browser *Tools* menu.

Setting Up a TTCN-SDL Co-Simulation

A complete test of a system design can be performed prior to building a prototype by designing the system in SDL, and using the TTCN suite to write test specifications in TTCN, or using a tool to automatically or semi-automatically generate test suites from a system description in any language. The actual testing is then a matter of connecting the two simulators.

To achieve the connection, follow these steps:

1. Generate and start the SDL Simulator.
2. Start the TTCN-SDL Co-Simulator by selecting the Co-Simulator quick-button from the tool bar of the Browser.
3. When the TTCN-SDL Co-Simulator has appeared, along with a selection of executable test cases and test groups, enter the command **`start-itex`** in the SDL Simulator.
4. Select one or more test cases or test groups in the TTCN-SDL Co-simulator Setup window, and click the *Run Test* button.
5. In the SDL Co-Simulator, give any of the start commands (**`Forever`**, **`Go`**...).

The loaded SDL system is now being simulated on the SDL suite side, and conformance tested from the TTCN suite side with the TTCN-SDL Co-Simulator.

The TTCN-SDL Co-Simulator User Interface

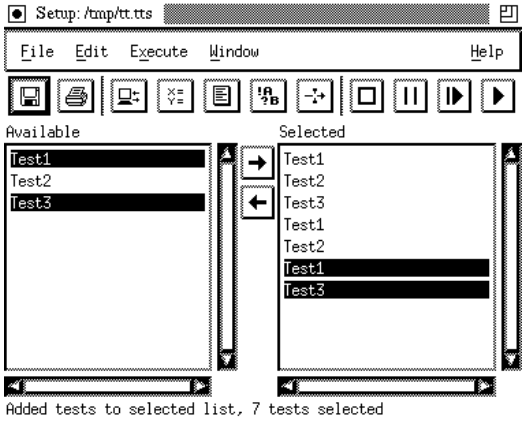

Figure 217: The TTCN-SDL Co-simulator Test Setup window

The Test Setup window is the main window of the TTCN-SDL Co-simulator. It manages an Executable Test Suite and a number of associated documents. The main part of the window is used for test selection. The left list depicts the currently available tests, and the right list depicts a selection of test cases and/or test groups.

Managing Setup Documents

The menu choices in the *File* menu are used for creating and saving documents. For information on the operations not mentioned here, see the general description of a Telelogic Tau *File* menu in "File Menu" on page 8 in chapter 1, *User Interface and Basic Operations*.

File > New

Creates a new Setup document. The new setup inherits the Executable Test Suite from the current Setup.

File > Open

Opens a previously saved Setup file in a new window.

File > Save

Saves the current Setup file. If the Setup was not previously saved, a name will be prompted for. A saved Setup consists of the file names of the associated documents, as well as the ETS name.

File > Save As...

Saves the Setup file with another file name. A running ETS might need to be restarted in order for changes to become effective.

File > Restart ETS

Restarts the Executable Test Suite / TTCN-SDL Co-simulator. This may compromise the consistency of running test cases, but might be necessary in order to make changes to the Setup effective.

File > Close

Closes the current view of the Setup document. If there are no more views of the Setup, the ETS will be terminated along with any test cases. If the Setup has been changed, a dialog will prompt for a file name for saving the Setup before it is closed. If the Setup is the only one currently open, the program will quit.

Selecting Test Cases and Groups

This section lists the TTCN-SDL Co-Simulator commands available in the *Edit* menu.

Edit > Documents > Select ETS

Enables you to select another ETS for the test Setup. This might be useful for selecting a new ETS, using the same Setup as the old one.

Edit > Documents > Add Document

Enables you to add a document type to this Setup. A standard set of document types are provided by default at program start-up.

Edit > Show > Test Cases

Sends a command to the ETS such that it lists all Test Cases in the *Available* list.

Edit > Show > Test Groups

Sends a command to the ETS such that it lists the Test Groups.

Edit > Add All

Adds all tests from the set of available tests to the set of selected tests.

Edit > Add

Adds the tests currently selected in the *Available* list, to those present in the *Selected* list. Both test cases and test groups may be present in the *Selected* list at the same time.

Edit > Remove

Removes the selected tests from the *Selected* list.

Edit > Remove All

Removes all tests from the *Selected* list.

Executing a Test

This section lists the TTCN-SDL Co-simulator commands available in the *Execute* menu.

Execute > Test > Cancel

Cancels a test in progress. All information will be lost and the ETS returns to its initial state. This might compromise the consistency of a test since it might leave the IUT in any state.

Execute > Test > Pause

Temporarily pauses the test execution. This might be used to examine the state of the IUT or of the ETS.

Execute > Test > Step

Steps one line of TTCN code. In a simulated environment this is a well-defined operation, as opposed to a real test system. In the latter case, timing might be compromised.

Execute > Test > Run test

Runs/Continues a test at full speed until it is finished or until a breakpoint is reached.

Execute > Breakpoints > Breakpoints

Shows the breakpoint editor, as described in "The TTCN-SDL Co-Simulator Editor" on page 1218. Notice: If the quick button is disabled, this is because no breakpoint document is currently associated with the Set-up. Use this menu item to add such a breakpoint document to the Setup.

Execute > Breakpoints > Enable

Enables all breakpoints (default).

Execute > Breakpoints > Disable

Temporarily disables all breakpoints.

Execute > Monitor > PCOs

Monitors all PCO Queues in a separate monitor window.

Execute > Monitor > Timers

Monitors all Timers in a separate monitor window.

Execute > Monitor > PTCs

Monitors all CPs in a separate monitor window.

Execute > Monitor > CPs

Monitors all CPs in a separate monitor window.

Execute > Enable ITEX Tracking

Highlights the next statement line to be executed in the Table Editor. This is on condition that you are running a TTCN-SDL Co-simulator session and that the TTCN-SDL Co-simulator was started from within the TTCN Suite.

Viewing Documents

This section lists the TTCN-SDL Co-simulator commands available in the *Window* menu.

Window > New Window

Enables you to display a new view of the same Setup, with a possibly different test selection. No new Setup is created with this command.

Window > Execution Trace

Shows a new view of the Test Execution trace. If the item is not accessible, select *Document* and then *Add Trace Document* in the *Edit* menu to add a document of this type to the current Setup.

Window > Conformance Log

Shows a new view of the Test Conformance Log. If the item is not accessible, select *Document* and then *Add Log Document* in the *Edit* menu to add a document of this type to the current Setup.

Window > Configuration Editor

Shows a new editor for the Test Configuration, see "The TTCN-SDL Co-Simulator Editor" on page 1218. If the item is not accessible, select *Document* and then *Add Configuration Document* in the *Edit* menu to add a document of this type to the current Setup. The Test configuration is system dependent and not used for co-simulation.

Window > Parameter Editor

Shows a new editor for the Test Suite Parameters, see "The TTCN-SDL Co-Simulator Editor" on page 1218. If the item is not accessible, select *Document* and then *Add Parameter Document* in the *Edit* menu to add a document of this type to the current Setup.

The TTCN-SDL Co-Simulator Editor

A generic editor is present for editing some of the documents, as for instance the Breakpoints, Parameter and the Configuration. It is of limited interest to those who are only interested in TTCN-SDL co-simulation, but it might be of more interest if you consider using the GUI for running real executable test suites.

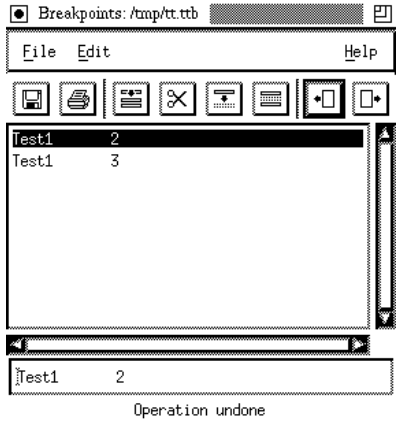

Figure 218: The TTCN-SDL Co-simulator editor for breakpoints

The editor is row oriented and the format of each row is determined by the current ETS. It is up to the ETS to warn the user for syntactic and/or semantic errors in the edited document. Please refer to the documentation of the ETS for more throughout definitions of the expected contents of each row.

Managing Documents in the TTCN-SDL Co-Simulator Editor

The menu choices in the *File* menu are used for creating and saving documents. For more information, see the general description of a Telelogic Tau *File* menu in "File Menu" on page 8 in chapter 1, *User Interface and Basic Operations*.

Editing Documents in the TTCN-SDL Co-Simulator Editor

The following section lists the TTCN-SDL Co-simulator Editor commands available in the *Edit* menu.

Edit > Undo

Undoes the last change of the document. This functionality is disabled when the document is saved or if a new document is opened. There are more than 200 undo steps.

Edit > Redo

Redoes the next undone change to the document. Applying a new operation removes any non-redone steps.

Edit > Cut

Cuts the currently selected row from the document. The row is placed in an internal clipboard.

Edit > Copy

Copies the currently selected row from the document to the internal clipboard.

Edit > Paste

Pastes the contents of the internal clipboard before the currently selected row.

Edit > Add

Inserts an empty row before the currently selected row.

Files Handled by the TTCN-SDL Co-Simulator

A total of 7 file types are handled by the TTCN-SDL Co-simulator user interface. Fortunately, most sessions involve only a subset of those file types. The file types and a short description are listed below:

- ETS file

  Executable Test Suite file. This is a file generated by the TTCN to C compiler. It might in particular be a co-simulator which has a specialized kernel for running in cooperation with an SDL suite simulated system.

- Setup file (`.tts`)

  A Setup file is a collection of the other file types and is used to manage a particular test Setup such that it can easily be re-created.

- Log file (`.log`)

  A Conformance Log File consists of the output from an ETS, as specified in the ISO-9646 standard.

- Trace file (`.ttt`)

  A simplified version of the Log file, more corresponding to the actual TTCN tables contents.

- Test Suite Parameters file (`.ttp`)

  A file for transferring the test suite parameters to the ETS. Comparable to the PICS/PIXIT file of ISO-9646. Please see the ETS kernel implementators user guide for details.

- Configuration file (`.ttc`)

  Test platform dependent file. Please see the ETS kernel implementators user guide for details.

- Breakpoints file (`.ttb`)

  A given set of breakpoints might be saved in the editor for later reuse.

Information Messages

The TTCN-SDL Co-simulator can present messages to the user in three forms:

- Informative messages
- Warning messages
- Error messages

Informative Messages

Informative messages are normally presented at the status bar. These might be for instance progress reports or other simple messages, which are only of a temporary interest (such as the tool tips).

Warnings Messages

Warnings, also known as “non-critical errors”, are events that indicate a problem during the preparation for a co-simulation, or during the actual co-simulation run. These are always presented in standard Motif warning dialogs. Warnings normally does not cause a co-simulation to halt.

Error Messages

Critical errors will halt the co-simulation. They are presented by an error dialog that appears in front of the application. The error dialog can often appear large and intimidating, but it simply gives a more verbose reason to why the error occurred, and what steps should be taken to avoid it.

Using the UI with an ETS

The user interface uses a publicly available format for communication with the ETS. Implementation of a management interface for an ETS is simple since the ETS only uses the standard I/O channels for communication with the GUI. Please request a protocol definition from Telelogic if you consider adapting your own test suite using this GUI. It is currently unlicensed and might be started using this command:

```
$telelogic/itex/bin/${hosttype}bin/isimui -help
```

Type Mappings in Co-Simulation

This section specifies the data type mapping used by the co-simulation of the SDL Simulator and the TTCN-SDL Co-simulator. Specifically it identifies the transfer syntax used in the communication between the SDL Simulator and the TTCN-SDL Co-simulators for each supported data type.

TTCN Types

This subsection describes the mapping from TTCN types to SDL types. For each TTCN type the corresponding SDL type and some examples of the transfer syntax is given.

Predefined Types

TTCN Type	SDL Type	Transfer Syntax Example
INTEGER	integer	1 -3
BOOLEAN	boolean	true false
BITSTRING	bit_string	'0101'B
HEXSTRING	-	
OCTETSTRING	octet_string	'4BA0'O
NumericString	NumericString	'3295' (0..9 + space)
PrintableString	PrintableString	'ask38-'
TeletexString	-	
VideotextString	-	
VisibleString	VisibleString	'xy'
IA5String	charstring	'123 abc'
GraphicString	-	
GeneralString	-	

References Types

TTCN Type	SDL Type	Transfer Syntax Example
SimpleType	syntype	same as referenced type
Struct	struct	(. 1, true .)
PDU	struct	(. 1, true .)
ASP	- (ASPs maps to signals)	

ASN.1 Types

This subsection describes the mapping from ASN.1 types to SDL types. For each ASN.1 type the corresponding SDL type and some examples of the transfer syntax is given.

ASN.1 Type	SDL Type	Transfer Syntax Example
BOOLEAN	boolean	true false
INTEGER	integer	1 0 -55
ENUMERATED	enumeration types	The enumerated values
REAL	- (Not supported in the TTCN to C compiler)	
BIT STRING	bit_string	‘0100’B
OCTET STRING	octet_string	‘4BA0’O
NULL	null	Null
SEQUENCE	struct	(. 1, true .)
SEQUENCE OF	array, string	(: 1, 2 :)
SET	-	
SET OF	bag, powerset	[1, 2, 2]
CHOICE	choice	<name> : <value>
ANY	-	
OBJECT IDENTIFIER	-	

ASN.1 Type	SDL Type	Transfer Syntax Example
Selection type	-	
Tagged type	-	
SubType	syntype	same as referenced type

SDL Types

This subsection defines the mapping from SDL to TTCN and to ASN.1. For each SDL type first the TTCN type is given, then the ASN.1 type and finally an example of transfer syntax.

Predefined Sorts

SDL Type	TTCN Type	ASN.1 Type	Transfer Syntax Example
`Bit`	`BITSTRING`	`BIT STRING (SIZE (1))`	'1'B
`Bit_String`	`BITSTRING`	`BIT STRING`	'1010'B
`Boolean`	`BOOLEAN`	`BOOLEAN`	true false
`Character`	`IA5String`	`IA5String (SIZE (1))`	'a'
`Charstring`	`IA5String`	`IA5String`	'123 abc'
`IA5String`	`IA5String`	`IA5String`	'123 abc'
`NumericString`	`NumericString`	`NumericString`	'123'
`PrintableString`	`PrintableString`	`PrintableString`	'ask38-'
`VisibleString`	`VisibleString`	`VisibleString`	'xy'
`Duration`	-	-	
`Time`	-	-	
`Integer`	`INTEGER`	`INTEGER`	0 4 -66
`Natural`	`INTEGER`	`INTEGER`	0 55
`Null`	-	`NULL`	Null

SDL Type	TTCN Type	ASN.1 Type	Transfer Syntax Example
Object_Identifier	-	-	
Octet	OCTETSTRING	OCTET STRING (SIZE (1))	‘FE’O
Octet_String	OCTETSTRING	OCTET STRING	‘F0E2’O
PId	-	-	
Real	-	-	

User Defined Sorts

SDL Type	TTCN Type	ASN.1 Type	Transfer Syntax Example
Syntypes	SimpleTypes	SubType	same as referenced type
Enumeration Sorts	-	ENUMERATED	the enumerated values
Struct	struct	SEQUENCE	(. 1, true .)
Bit Fields	-	-	(. 2, 3, 0 .)
Optional	-	-	
Choice	-	CHOICE	C1 : 1

Predefined Generators

SDL Type	TTCN Type	ASN.1 Type	Transfer Syntax Example
Array	-	SEQUENCE OF	(: 1, 4, 7 :)
String	-	SEQUENCE OF	(: 1, 4, 7 :)
Powerset	-	SET OF	[1, 6, 8]
Bag	-	SET OF	[1, 6, 6]