Chapter

7

# *Tutorial: TTCN-SDL Co-Simulator (on UNIX)*

This is an introductory tutorial to the TTCN-SDL Co-simulator on UNIX and an example of a complete test session with Cbasic. The tutorial is divided into sections that describes steps that are performed during a typical test session.

#### Note: UNIX version

This is the UNIX version of the tutorial. The Windows version is <u>chapter 6</u>, *Tutorial: TTCN-SDL Co-Simulator (Windows)*.

# **Purpose of This Tutorial**

The purpose of this tutorial is to make you familiar with the TTCN-SDL Co-simulator. When you have read and practised this tutorial you will know the actions required to execute a simulated test.

In this tutorial it is assumed that you know how to use the SDL Simulator. More information can be found in <u>chapter 4</u>, *Tutorial: The SDL Simulator, in the SDL Suite Getting Started* and in <u>chapter 50</u>, *The SDL Simulator, in the User's Manual*.

# The Steps in a Test Session

When a TTCN test suite and an SDL suite system already exists, this is what you do in a typical test session:

- 1. Generate and start the TTCN-SDL Co-simulator.
- 2. Generate and start the SDL simulator.
- 3. Set up the communication between the TTCN-SDL Co-simulator and SDL simulator.
- 4. Start the simulation.
- 5. Start the TTCN-SDL simulation.
- 6. Perform the simulation.

# The Test System

This section describes the system that is going to be tested and what roles the TTCN-SDL Co-simulator and SDL simulator play in the test.

An SDL system called ABP will be used in this tutorial. The ABP system implements a small OSI stack containing a network layer, a datalink layer and a physical layer. The network component on one side is handled by the TTCN suite and on the other side by an SDL process.

The data-link layer implements a very simple protocol known as the Alternating Bit Protocol (ABP).

The network protocol is very simple protocol, where all messages are echoed back as an acknowledgment that they have been received.

This setup gives us the ability to send messages from the TTCN suite to the network peer simulated by the SDL suite, and to receive acknowledgments in return. If the answer is identical to the message, we have managed to communicate successfully.

What we want to test is the ability of the Data-Link layer's protocol to deliver, even when messages are lost on the way. To enable this, we have introduced an extra channel between the TTCN-SDL Co-simulator and the SDL simulator, on which commands can be sent to make the physical layer lose messages.

The ABP system is stored in the Telelogic Tau installation, in the subdirectory examples/bitprotocol. If you have set up the \$telelogic environment variable, the complete path is \$telelogic/examples/bitprotocol.

# Setting Up a Simulation

### What You Will Learn

- To set up the TTCN-SDL Co-simulator and SDL simulator
- To generate and build simulator executables
- To initialize the communication between the simulators
- To start the simulators

When you have finished with this section, the simulators should be ready to run a TTCN suite test on a simulated SDL system.

### Generating and Starting the TTCN-SDL Co-Simulator

The first thing you should do is to generate and start the TTCN-SDL Co-simulator:

- 1. Start Telelogic Tau.
- 2. In the Organizer, load the system *ABP* from the file abp.sdt.

You can find this file in \$telelogic/examples/bitprotocol (where \$telelogic points to the installation directory).

3. Double-click the *abp* icon in the *TTCN Test Specification* chapter.

This will start the TTCN suite with the abp.mp test suite opened.

4. Select the node named *abp* in the TTCN Browser.

5. Select Make from the Tools menu.

The TTCN Make dialog is opened.

6. Make sure that the *Use standard kernel* option is checked and that the *Simulation* command is selected.

#### Note:

The dialog will show up the first time you build your simulator. In later builds it will remember the settings.

- 7. Select the *Directory for generated files* to point to a location where you have write permission and sufficient space for the generated files.
- 8. Click the *Full Make* button to start the generation and compilation of the TTCN-SDL Co-simulator executable.

The TTCN suite will now generate and build the TTCN-SDL Cosimulator executable.

9. Click the *Start Simulator* button on the TTCN suite tool bar.

This will start the TTCN-SDL Co-simulator and open the simulator window. The TTCN suite is now ready to simulate.

### Generating and Starting the SDL Simulator

Now you have to set up the SDL simulator as well:

- 1. In the Organizer, select the system diagram *ABP* from the *SDL System Structure* chapter.
- 2. Click the *Simulate* button in the tool bar.
  - You may also want to check the settings in the *Make* dialog (opened from the *Generate* menu) before you generate and compile the simulator. If you select *Make* or *Full Make* in that dialog, you will have to open the Simulator UI and load the simulator after that.

The SDL simulator generates and builds a simulator executable. The SDL simulator window is displayed with the simulator started. You are now ready to simulate.

Ö

Chapter

### Getting the Simulators to Communicate

The next step you should perform is to initialize the communication between the simulators.

• Enter the command **start-itex** in the SDL simulator window.

This instructs the SDL simulator to communicate with the TTCN-SDL Co-simulator.

# Performing the Simulation

#### What You Will Learn

- To single step a test case
- To run a test case at full speed
- To run a test batch
- To set and delete breakpoints
- To abort a simulation

### **Single Stepping Test Cases**

1. In the TTCN-SDL Co-simulator window, select the test case  $TC_01$  and then click the right arrow that appears.

This selects the test case that we will execute by adding it to the *Selected* box:



Figure 64: Selecting a test case to execute

**2**. Click the *Step Simulation* button

Each time you click this button, the TTCN-SDL Co-simulator executes one step of TTCN code.

A Table Editor is opened. In the first line of TTCN code, there is a colored bar.

#### Note:

This bar indicates which line will be executed the **next** time you click the *Step Simulation* button.

3. Enter the command go-forever in the SDL simulator window.

The SDL simulator is now executing and it is possible to perform a simulation.

4. Click the *Step Simulation* button repeatedly until you reach the end of the test case.

### **Running Test Cases at Full Speed**

1. In the TTCN-SDL Co-simulator window, make sure that the test case *TC\_01* is still in the *Selected* box. If it is not, select the test case again and click the right arrow.

#### Note:

If the test case already is in the *Selected* box, you do not have to select it again. If you select it again, you are actually telling the TTCN-SDL Co-simulator that you want to execute the test case twice.

2. Click the Run Simulation button.

The simulator now starts executing lines of TTCN code without pausing. This will continue until the end of the test case is reached or until you click the *Pause Simulation* button.

As you can see, the colored bar in the Table Editor will change rapidly.

At this point the SDL simulator is running at full speed. This is normal and you can safely ignore the SDL simulator. The cause of this is a timer in the simulated SDL system that fires repeatedly while waiting for input from the TTCN-SDL Co-simulator.

### **Running Test Batches**

It is also possible to execute more than one test case or test group.

- 1. In the TTCN-SDL Co-simulator window, deselect the test case *TC\_01* in the *Available* box. You do this by clicking on the test case.
- 2. Select the test case *TC\_02* and click the right arrow. The test case is added to the *Selected* box after *TC\_01*.

You can perform this operation with as many test cases as you wish. You can add them in any order. The *Selected* box tells you in what order the test cases will be executed.

3. Press the Run Simulation button.

This will cause the TTCN-SDL Co-simulator to execute all the selected test cases in turn without pausing anywhere. Another Table Editor is opened for the second test case.

#### Note:

This requires an SDL suite system that always ends in a state from where you can execute a new test case. The SDL suite system may not require any manual resetting between test cases.

## **Toggling Breakpoints**

1. Make sure that the TTCN-SDL Co-simulator window is active and press <Ctrl+B>.

This will open the *Breakpoints* window. In this window you can add and delete breakpoints.

2. Select Add from the Edit menu.

This adds an empty breakpoint definition to the breakpoint editor.

3. Click in the edit field at the bottom of the *Breakpoints* window and type in TC\_01 3 followed by <Return>.



Figure 65: A breakpoint in the Breakpoints window

- 4. Select *Save* from the *File* menu.
- 5. Type in a filename with the extension .ttb and select OK.

This will set a breakpoint on line 3.

- 6. In the simulator window, make sure that only the test case  $TC_01$  is selected.
- 7. Press the Run Simulation button.

The TTCN-SDL Co-simulator will run without pausing between lines of TTCN code. When it reaches the line where the breakpoint was set, it will stop, as you can see in the Table Editor.

- 8. Select the breakpoint line we just added in the *Breakpoints* window.
- 9. Select *Delete* from the *Edit* menu.

This will remove the breakpoint from line 3.

- 10. Select Save from the File menu.
- 11. Select Close from the File menu.
- 12. Click the Run Simulation button.

The simulator now continues to run until it reaches the end of the test case.

Chapter

# **Ending a Simulation**

It is possible to end a simulation in the middle of an execution

1. Click the *Step Simulation* button twice.

This will place the current line on line 2 in the test case.

You now realize that the execution was a mistake and you want to abort it.

- 2. Click the Abort Simulation button.

This aborts the execution of the test case for the TTCN-SDL Cosimulator.

3. In the SDL simulator window, select Restart in the File menu.

This restarts the SDL simulator. This step is necessary because otherwise the SDL Simulator and the TTCN-SDL Co-simulator would be out of sync.

To continue co-simulation, you will need to enter the command start-itex in the SDL simulator, start running the test in the TTCN-SDL co-simulator, and then enter the command go-forev-er in the SDL simulator.

# **The Execution Trace**

Every execution of the TTCN-SDL Co-simulator produces a trace. This trace is output in the trace window. Every line in the test case that is executed will be present in the trace window.

This makes it possible to examine a trace of the execution after it has finished. This is valuable when running the test cases in a batch.

• To see the execution trace, select *Execution Trace* in the *Window* menu of the TTCN-SDL Co-simulator.

The trace contains the test component, table name, line within the table, behaviour, constraint and verdict.

MTC	TC_01:4	C1 ? ToNetworkLayer1	RecieveC(Message4)	PASS
MTC	TC_01:1	C1 ! FromNetworkLayer1	SendC(Message1)	
MTC	TC_01:2	C1 ? ToNetworkLayer1	RecieveC(Message1)	
MTC	TC_01:3	C1 ! FromNetworkLayer1	SendC(Message4)	
MTC	TC_01:4	C1 ? ToNetworkLayer1	RecieveC(Message4)	PASS
MTC	TC_02:1	C1 ! FromNetworkLayer1	SendC(Message1)	
MTC	TC_02:2	C1 ? ToNetworkLayer1	RecieveC(Message1)	
MTC	TC_02:3	C2 ! LoseNextFrame		
MTC	TC_02:4	C1 ! FromNetworkLayer1	SendC(Message4)	
MTC	TC_02:5	C1 ? ToNetworkLayer1	RecieveC(Message4)	PASS

Figure 66: An execution trace

# The Conformance Log

Every test case produces a conformance log. This log contains detailed information about the tests performed. It is very useful when determining why a test case failed or succeeded.

• To see the conformance log, select *Conformance Log* in the *Window* menu of the TTCN-SDL Co-simulator.

[StartDEF]	Start Default:	[2] OTHERWISE_FAIL Tab: TC_0
[NOMatch]	Line not matched:	1 Tab: OTHERWISE_FAIL
[StopDEF]	Default ended:	[2] OTHERWISE_FAIL Tab: TC_0
[MatchValue]	Succeded	Attempting to match SEQUENCE/SEQUENC
[MatchValue]	Succeded	Attempting to match SEQUENCE/SEQUENC
[MatchValue]	Succeded	Matching 1 against 1
[MatchValue]	Succeded	Matching "Connect" against "Connect"
[MatchValue]	Succeded	Matching SEQUENCE/SEQUENCE_OF
[MatchValue]	Succeded	Matching SEQUENCE/SEQUENCE_OF
[StopTC]	Test Case ended:	TC_01

Figure 67: A conformance log

