

1 Purpose

The purpose of the exercise is to learn how to make a syntactic and semantic analysis of build SDL system and make a simulation.

2 Analysing the Complete System

The SDL Analyzer's primary task is to check SDL-92 diagrams with respect to syntactic and semantic rules. The Analyzer accepts both SDL/GR and SDL/PR as input, and has the ability to transform SDL/GR diagrams to SDL/PR files and vice-versa.

Syntax and semantics (in Finnish: lauseoppi ja merkitysoppi)

Syntax is the grammar, structure, or order of the elements in a language statement. Semantics is the meaning of these elements.

2.1 Running the Analyser

1. Select the SDL system diagram icon in the Organizer diagram structure. Then, from the Organizer's *Generate* menu, select the *Analyze* command.
2. The Analyzer dialog is opened. Adjust the options and click Analyze.

3 The SDL Simulator

Once you have designed and analyzed a complete SDL system, it is possible to interactively inspect and check its actual behavior.

The SDL Simulator is the tool that you use for testing the behavior of your SDL systems. Typically, simulation means executing the system under user control; stepping, setting breakpoints, examining the system, processes and variables, sending signals and tracing the execution, as you would do with a debugger, but applied on the SDL domain.

To be able to simulate the system, you must first generate an executable simulator and then start the simulator with a suitable user interface.

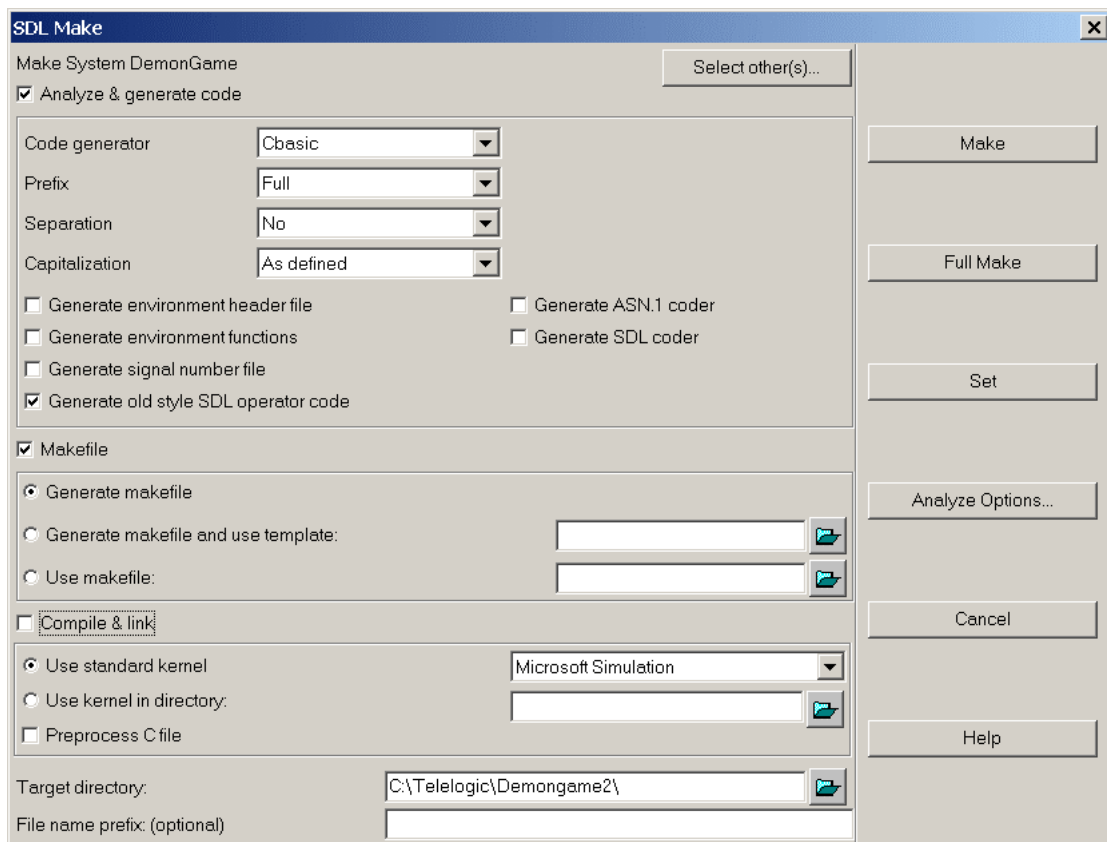
3.1 Generating a Simulator

1. Make sure the system diagram icon is selected in the Organizer.
2. Select the *Make* command from the *Generate* menu. The Make dialog is opened. Adjust the options in accordance to the figure below (uncheck *compile&link* and ensure that you have correct *Target directory*) and click make.
3. Open windows command prompt.

4. Set environment for using Microsoft Visual C/C++ tools by running *VCVARS32.BAT*.

If you got an error message complaining that *VCVARS32.BAT* is not recognized command, switch to following directory and try again:
C:\Program Files\Microsoft Visual Studio\VC98\Bin

5. Compile the system by running *compile.bat* (if it exists) or
nmake /f DemonGame.m SCTDIR=C:\Telelogic\SDL_TTCN_Suite4.2\sdt\sdt\wini386\SCAMDEBCOM



3.2 Starting the Simulator

The generated simulator is now stored on a file called *filename_smc.exe* in the directory from which you started the SDL suite. The simulator contains a monitor system that provides a set of commands which can be used to control and monitor the execution of the simulator.

It is possible for you to execute the simulator directly from an OS prompt, in which case you have to enter all commands to the monitor system textually using a simple command-line interface.

The SDL suite provides a user-friendly graphical interface to the simulator that is started from the Organizer.

1. From the Tools menu, select the sub-menu SDL and the command Simulator UI. The Simulator UI window opened.
2. To start a simulator, select Open from the File menu, or click the Open quick button and select the generated simulator file (demongame_smc.exe).
3. The text area of the Simulator UI greets you with a welcome message to acknowledge that the simulator has been started:

Welcome to SDL SIMULATOR. Simulating system Demongame.

The Simulator UI is now ready to accept commands to the monitor system. Whenever it is possible to enter a command, the prompt Command: is issued in the text area.

4 Tasks

1. Download zipped Demongame file from the course web-site and extract it into your home directory. Open Telelogic TAU and open *demongame.sdt* file. Fix directories (source directory, target directory, change document directory) and save the system file.
2. Perform syntactic and semantic analysis on the Demongame system and fix existing errors (remember to set *Case Sensitive SDL* on).
3. Test the behaviour of the demongame with SDL simulator and try to find an error on it. You can check the behaviour of Demongame from previous exercises.

When the simulator is started, set trace levels and start simulating:

```
> set-trace 6  
> set-gr-trace 1  
> show-next-symbol
```

Remember to compile the simulator after each change to the SDL system.

A tip: To find the error, perform e.g. an use case where the user firstly wins two times and then loses once.