Chapter

3

Tutorial: The Editors and the Analyzer

The SDL suite products are used for designing and specifying systems, in particular real-time systems. The SDL suite supports the Specification and Description Language (SDL) as recommended by ITU (the Z.100 recommendation). The SDL suite also supports the definition of Message Sequence Charts (MSCs), as well as parts of the Unified Modeling Language (UML) notation. For full support of the UML language you should rather use the Telelogic Tau UML Suite.

This tutorial assumes that you are already familiar with SDL and have some brief notions about Message Sequence Charts. UML notation is described in the UML Suite documentation.

We will demonstrate, by using a simple SDL system as example, the basic editing and analysis functionality that is available. You will practice various "hands-on" exercises that will get you more familiar with the SDL Editor and the MSC Editor, as well as the SDL Analyzer.

In order to learn how to use these tools, read through this entire chapter. As you read, you should perform the exercises on your computer system as they are described.

Purpose of This Tutorial

The purpose of this tutorial is to make you familiar with the user interface and the essential editing functionality in the SDL suite. This tutorial is designed as a guided tour through the SDL suite, where a number of hands-on exercises should be performed on your computer as you read this chapter.

We have on purpose selected a simple example that should be easy to understand. It is assumed that you have a basic knowledge about SDL — this chapter is **not** a tutorial on SDL.

This tutorial addresses primarily persons with no or little experience of the SDL suite. You may also find it useful if you have experience of earlier versions of the SDL suite and want to learn the new features and user interface concepts in Telelogic Tau 4.5.

Once you have completed the exercises in this tutorial, you may want to continue with the tutorials that are presented in:

- *chapter 4, Tutorial: The SDL Simulator*,
- *chapter 5, Tutorial: The SDL Validator*, and
- chapter 6, *Tutorial: Applying SDL-92 to the DemonGame*.

Note: Platform differences

It is possible to run the tutorials on UNIX as well as on Windows platforms. Should there be any differences between the platforms, this is indicated in the text with the markers “on UNIX”, “Windows only”, etc. This is also indicated in the platform-specific screen shots.

When such platform indicators are found, please pay attention only to the instructions and screen shots that are valid for your platform.

The Demon Game

The example that has been chosen in this tutorial is a simplified version of the “Demon game”, which is a well known example in the SDL community, since it is, among other things, used as example in the SDL recommendation.

The SDL definition of the Demon game may be found in SDL/GR form later in this chapter (see “Appendix A: The Definition of the SDL-88 DemonGame” on page 129). The definition of the behavior of the Demon game is probably not the simplest way of describing the game, but it has been selected since it is good for demonstrating the facilities of simulation and validation.

Behavior of the Demon Game

Seen from the environment, the behavior of the system is as follows. The system accepts four different types of signals, Newgame, Endgame, Probe, and Result, where the first two signals are used to start and end a game. Only one game at a time can be played, that is, Newgame signals will be ignored when a game is in progress and Endgame will be ignored if there is no game in progress.

The game in itself is very simple. A “demon,” which in the system is represented by the process Demon, changes the status of the system every now and then between winning and losing. This is represented by the states Winning and Losing in the process Game. The user is to guess when the status is winning. If the user probes (outputs the signal Probe), when the status is winning, he wins one point. If the user probes when the status is losing he loses one point. The system responds to a Probe signal by either a Win or a Lose signal. To see the current score the user can issue a Result signal, which will be answered by a Score signal containing an integer parameter giving the current score.

Starting the SDL suite

Some Preparatory Work

This tutorial assumes that the SDL suite has been installed correctly, according to the instructions in the *Installation Guide*.

> **Note: Installation directory**
>
> **On UNIX, the Telelogic Tau installation directory is pointed out by the environment variable** `$telelogic`. If this variable is not set in your UNIX environment, you should ask your system manager or the person responsible for the Telelogic Tau environment at your site for instructions on how to set this variable correctly.
>
> **In Windows, the Telelogic Tau installation directory is assumed to be** `C:\Telelogic\SDL_TTCN_Suite4.5` **throughout this tutorial.** If you cannot find this directory on your PC, you should ask your system manager or the person responsible for the Telelogic Tau environment at your site for the correct path to the installation directory.

The directory `$telelogic/sdt/examples/demongame` **(on UNIX)**, or `C:\Telelogic\SDL_TTCN_Suite4.5\sdt\examples\demongame` **(in Windows)**, is created during installation and is the directory that contains the complete example, that you may look at whenever you feel insecure or want to "shortcut" an exercise.

In order **not** to modify these completed example files, you should create a dedicated directory for the purpose of this tutorial.

On UNIX, follow this instruction:

1. Create a new subdirectory in your home directory:

   ```
   mkdir ~/demongame
   ```

   In the remainder of this tutorial, we will assume this name for your personal tutorial directory.

In Windows, follow these instructions:

1. Create a local directory `C:\Telelogic\SDL_TTCN_Suite4.5\work\demongame` on your

PC. In the remainder of this tutorial, we will assume this name for your personal tutorial directory.

> **Note: Do not use space characters in Windows**
>
> **In Windows**, Telelogic Tau does not support file or directory names that contain space characters. Make sure you do not use such names.

Starting the SDL suite

On UNIX, to start the SDL suite environment:

1. Change directory to your `demongame` directory:

   ```
   cd ~/demongame
   ```

2. Now, type:

   ```
   sdt
   ```

> **Note:**
>
> If the command `sdt` is not found, you first have to set up your `$path` variable correctly. Consult your system manager or the person that is responsible for the SDL suite environment at your site.

In Windows, to start the SDL suite in a manner suitable for this tutorial you should create and use a shortcut icon:

1. Locate the executable SDL suite file `sdt.exe` (or just `sdt`) in `C:\Telelogic\SDL_TTCN_Suite4.5\bin\wini386`. (See the note "Installation directory" on page 42 if you cannot find this directory.)
2. Create a shortcut icon to this file on the Windows desktop.
3. From the new icon's popup menu, select *Properties*. In the dialog, select the *Shortcut* tab at the top.
4. In the *Start in* field, enter the path to your new directory, i.e. **`C:\Telelogic\SDL_TTCN_Suite4.5\work\demongame`**
5. Click *OK* to close the dialog.
6. Double-click the shortcut icon.

The Organizer Window

When you have started the SDL suite, the *Organizer window* is displayed (see Figure 17 and Figure 18). The Organizer is the main tool from which you have access to the tools in the Telelogic Tau environment.

The Organizer also displays the *Welcome* window, where you may read the licensing agreement for Telelogic Tau. The window is always placed on top of the Organizer window and disappears as soon as you perform any action in the Organizer (you may also click the *Continue* button).

You are now ready to start working.

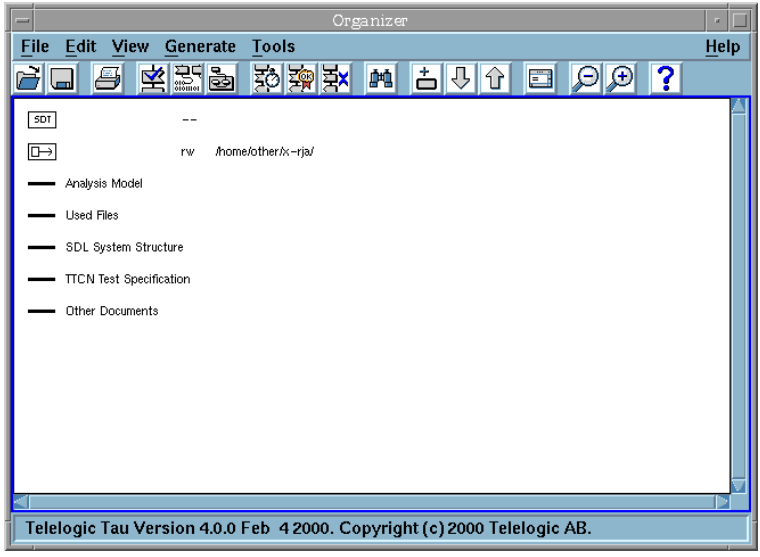


Figure 17: The Organizer window **(on UNIX)**

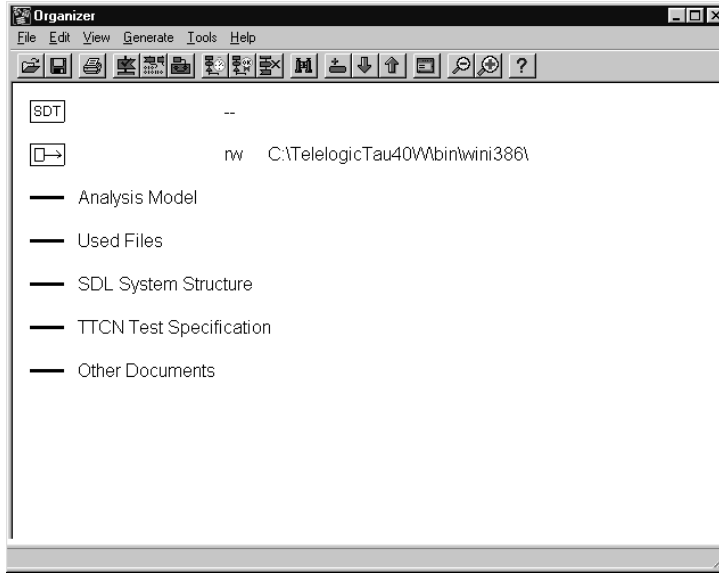


Figure 18: The Organizer window **(in Windows)**

Note: Screen shots

As you can see, screen shots of the Organizer window are shown for each platform, UNIX and Windows. From now on, screen shots will only be shown for one of the platforms, provided they contain the same information for both platforms. This means that **the layout and appearance of screen shots may differ** slightly from what you see on your computer screen.

Only if a screen shot differs in an important aspect between the platforms will two separate screen shots be shown.

Preferences

What You Will Learn

- To set up and save preferences
- The basics of the graphical user interface in the SDL suite, operated from the mouse and the keyboard. You will learn to:
  - Use graphical lists
  - Use pull-down menus
  - Use pop up menus
  - Use quick buttons
  - Use the status bar
  - Use option menus
  - Use text fields
  - Use slide bars
  - Use keyboard accelerators

What Are Preferences for?

Before starting creating your first SDL diagram, you should set up some preferences to match your computer environment. These preferences affect the default behavior of the Telelogic Tau tools and should be adjusted to convenient values in order to have Telelogic Tau function properly (most options in Telelogic Tau may be set as preferences). When Telelogic Tau is installed, the factory settings are used as preference settings. Your system manager may have already prepared the environment for you; good advice is to check this anyway.

At least the following should be checked:

- The help preferences
- The printer preferences
- The drawing area size
- The platform mode.

Displaying and Changing Preferences

To view and possibly change the preferences:

1. From the Organizer's *Tools* menu, select the *Preference Manager* command. The Preference Manager window is displayed:

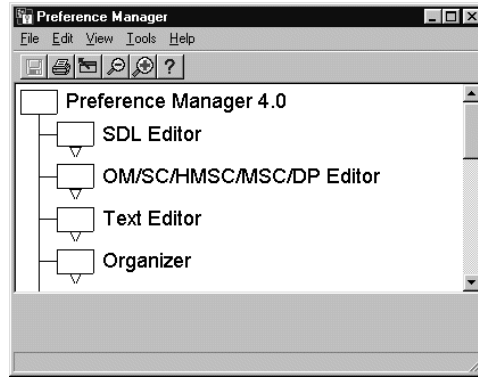


Figure 19: The Preference Manager window

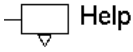
Your next task will be to check and, if required, modify a few preferences.

Help Preferences

Telelogic Tau supports a context-sensitive online help facility that you may use at any moment to request help on a window, on a command, on a dialog etc.

The online help is in HTML-format and you may use Netscape Navigator or Internet Explorer as help viewer. You can change the help viewer by setting a preference.

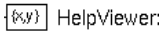
To set up the Help preferences:



1. Locate the icon titled *Help* and double-click it.

   – You can also right-click the icon and select *Expand* from the pop up menu.

   This will expand the list structure below and make the Help preferences visible.



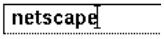
2. Locate the preference *HelpViewer*.

   To the right of the preference icon you can see the current value, the currently saved value and an explanatory text.

3. Change the preference by selecting the *HelpViewer* icon and then selecting a new viewer in the option menu at the bottom of the window. (You may need to ask your system manager if you do not know what viewer to use). The icon will turn gray, which means that the preference has been changed and needs to be saved.

4. Depending on your choice of viewer, you should now check that the command Telelogic Tau uses when starting the help viewer is correct, according to your computer environment:

   – Locate and select the icon titled *NetscapeCommand* or *InternetExplorerCommand*. The current value is shown to the right. If it is not correct, change the text in the text field at the bottom of the window. (You may need to ask your system manager about the correct value.)



5. Collapse the *Help* icon by double-clicking it.

You have now learned how to work with graphical lists in Telelogic Tau. Graphical lists are used extensively throughout the tools; they may hold as many levels as required (the Preference Manager uses three levels of indentation, as seen on the screen).

Some tools also support a vertical tree as an alternative to a graphical list. The functionality is identical, only the presentation differs. You will acquaintance yourself with a graphical tree later in this tutorial.

Setting the Default Printer

In this tutorial, you will learn how to print diagrams. Before you start printing from Telelogic Tau, you should check and, if needed, set up your print preferences in accordance to your computer environment.

To set up the Print preferences:

1. Locate the *Print* icon. Expand it.

2. Locate the *PrinterCommand* preference. Adjust it to an adequate value (if required, ask your system manager). You may specify any suitable operating system command, for instance sending the resulting printouts to a printer queue (the command `lpr`) or previewing a PostScript file in a pre-viewer such as Ghostview[1].

1. Ghostview: A user interface for ghostscript. 1992 Timothy O. Theisen.

3. Locate the *PaperFormat* preference. Telelogic Tau supports a number of predefined paper sizes on the option menu (A4, A3, US Letter and US Legal). You may also specify an arbitrary *UserDefined* value, in which case you also need to specify the preferences *UserDefinedWidth* and *UserDefinedHeight*; these values are expressed in millimeters.

4. Adjust, if required, the preferences *MarginUpper*, *MarginLower*, *MarginLeft* and *MarginRight*. These preferences govern how much space in millimeters will be reserved for the margins on the printed pages; you may use this space for including headers and footers in your printouts.

   – To adjust these preferences, select them and drag the slider for a coarse adjustment. Terminate by clicking left or right of the slide bar or on the arrows to adjust in smaller steps.

5. Adjust, if required, the preference *Landscape* to on or off (this preference specifies the orientation, landscape or portrait).

Setting the Drawing Area Size

When editing SDL diagrams, the pages are assigned a predefined size. You should specify the default size to match the size of the printer pages and the printer margins that you defined in the previous exercise.

1. Locate the *SDL Editor* icon (at the very top), expand it and inspect the preferences *PageWidth* and *PageHeight*.

2. If required, adjust these preferences to suitable values.

Saving the Preferences

You should now save your preference settings for future Telelogic Tau sessions.

1. Select the *Save* command from the *File* menu.

   – Alternatively, you may click the quick button for *Save*. Quick buttons are located in a *tool bar* which may be found immediately beneath the menu bar. Quick buttons are mouse accelera-

tors for frequent commands and are available in all Telelogic Tau tools, not only in the Preference Manager.

– You may “preview” the functionality that a quick button provides by pointing on the quick button; the status bar (situated at the bottom of the window) displays an explanatory text. If you let the mouse pointer rest on the quick button, a short “tool tip” text is also displayed just below the button.

– Another possibility is to type the keyboard accelerator for the *Save* command, by pressing `<Ctrl+S>`. This is indicated immediately to the right of the menu choice *Save*.

2. You will receive a warning that the preferences you have changed will not take effect until the individual tools are restarted (exited and started again). Just click *OK* to acknowledge this.

   Your preferences are now saved on file for the current (and for future) Telelogic Tau sessions.

3. Close the Preferences window by selecting the *Exit* command from the *File* menu.

This concludes your Preference session. You may of course at any moment go back to the Preference Manager and adjust other preferences.

Creating an SDL Structure

You are now ready to create your first SDL diagrams.

What You Will Learn

- To customize the Organizer chapters
- To create an SDL structure
- To add a system root node
- To create a system diagram
- To add a page
- To edit a system diagram
- To save a diagram on file
- To save a diagram structure on a system file
- To work with dialogs (modal and modeless)
- To work with tree structures

Customizing the Organizer Chapters

When the SDL suite is started, the Organizer displays two icons that symbolizes the system file and the source directory for your diagrams. The system file will be explained later. The source directory is where the SDL suite components will look for existing diagrams, and save newly created diagrams. (The source directory can of course be changed.)

The source directory should already be set to the directory that you started the SDL suite from (`~/demongame` **(on UNIX)**, or `C:\Telelogic\SDL_TTCN_Suite4.5\work\demongame` **(in Windows)**).

By default, the Organizer also shows 5 areas in its window:

- *Analysis Model*
- *Used Files*
- *SDL System Structure*
- *TTCN Test Specification*
- *Other Documents*.

These areas are known as *chapters*. You may use the chapters to hold a number of diagrams and documents; the actual use is a matter of personal taste and the default is to be regarded as a suggestion. As you will design a rather simple system, we suggest that you start by removing the chapters *Analysis Model*, *Used Files* and *TTCN Test Specification*.

To remove the chapters:

1. Select the chapter *Analysis Model*.
2. Select the menu choice *Remove* from the *Edit* menu. (You may also press the `<Del>` button on the keyboard.)
   - A dialog opens – confirm the removal by clicking the *Remove* button in the dialog.

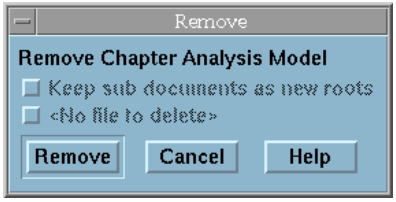


Figure 20: Confirming to remove a chapter

3. Repeat the steps above for the chapters *Used Files* and *TTCN Test Specification*.

You may also rename the remaining two chapters:

1. Select the chapter *SDL System Structure*.
2. Select the *Edit* menu choice from the *Edit* menu. (You may also double-click the chapter.)
3. In the dialog that opens, make sure the option *Edit chapter symbol* is selected and click the *Edit* button.

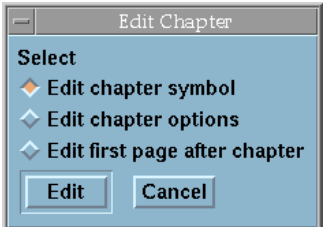


Figure 21: Editing the chapter symbol

4. Change the document name in the opened *Edit* dialog; for instance to `My first SDL system`. Do **not** change the document type indicated by the *Organizer* button and the option menu value *Chapter*.

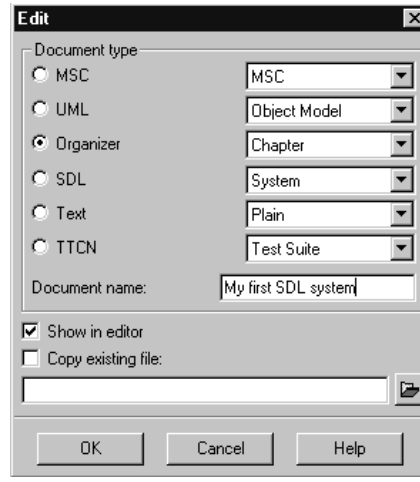


Figure 22: Naming the chapter

   – **On UNIX**, you may note that the cursor changes to the shape of a question mark as soon as it points on the parent Organizer window. This convention has been adopted to indicate that a dialog must be closed before any other operation is allowed to take place in the tool. Dialogs that need to be answered before proceeding further are called *modal* dialogs.

5. **Turn off** (uncheck) the option *Show in editor*.

6. Terminate by clicking the *OK* button.

   – If you like, also rename the *Other Documents* chapter. This chapter will be used later in this tutorial to hold diagrams that are not part of the SDL system but that you will want to keep track of.

Creating a System Diagram

Adding a Root Node

You will now create an SDL system, working in a top-down fashion:

1. Make sure the chapter *My first SDL system* is selected.
2. Select the *Add New* command from the *Edit* menu. The *Add New* dialog opens, prompting you to specify the name and type of diagram to add.

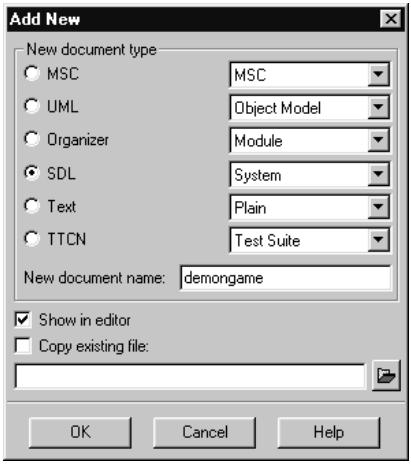


Figure 23: Adding a new diagram

3. Specify the *New document type* as *SDL*, and specify the SDL diagram type as *System*, as depicted above.
   - If the SDL diagram type shows something else than System and thus needs to be changed, click the option menu to adjust it.
4. Specify the *New document name* as `DemonGame` (the default name, *Untitled*, disappears).
   - You may need to point and possibly click with the cursor on the text field to set the focus on it.
5. Make sure the *Show in editor* button is turned **off**.

6. Click the *OK* button.

   – The dialog disappears and the Organizer window is updated with a root node — system DemonGame. Note that the diagram is identified as being [unconnected], meaning that there is no connection to a physical file.

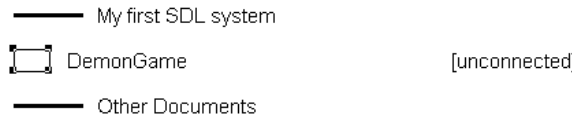


Figure 24: The new root node

Creating the System Diagram

You have so far created an Organizer *diagram structure*, consisting of one *reference* to an SDL system diagram (the referred diagram does however not yet exist).

Your next task is to create the system diagram:

1. Select the DemonGame SDL system diagram icon. See Figure 24.
2. From the *Edit* menu, select the menu choice *Edit*.

   – You may also press the **right** mouse button while pointing on the icon — a popup menu appears — select the sub-menu *Edit* and the menu choice *Edit*.

   – Another way to edit the diagram is to double-click the icon with the left mouse button.

3. The *Edit* dialog opens, suggesting to create a new diagram and to *Show* the diagram *in editor*. (The dialog is very similar to the Add New dialog you just used.) Accept the suggestion by clicking *OK*.

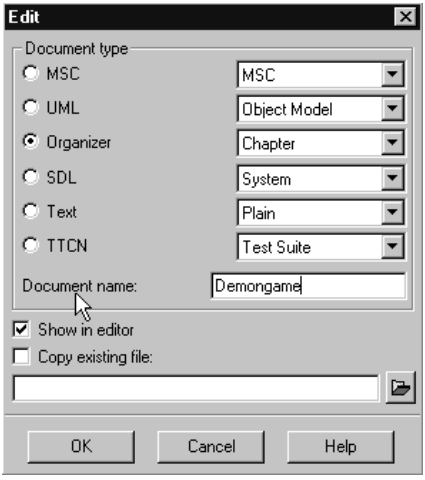


Figure 25: Prompting to create a new diagram

The SDL suite responds by displaying the *SDL Editor window*, showing the upper left corner of page 1 of the system diagram DemonGame.

The SDL Editor is the tool you use when editing the contents of the diagrams. The SDL Editor is also used for building the diagram structure that is displayed in the Organizer window.

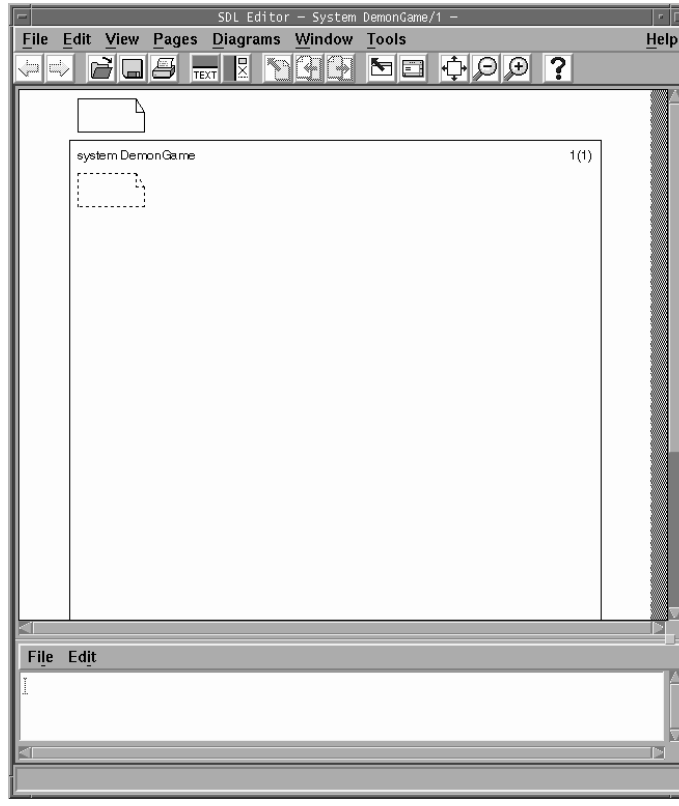


Figure 26: The SDL Editor window **(on UNIX)**

Your next task is to fill in the contents of the diagram. Figure 27 shows the appearance of the diagram when completed and printed on paper. As you can see, the diagram consists of two block reference symbols (GameBlock and DemonBlock), a channel conveying the signals between the blocks (C3) and two channels conveying the signals to and from the environment (C1 and C2). There is also a text symbol where the signal declarations may be found.

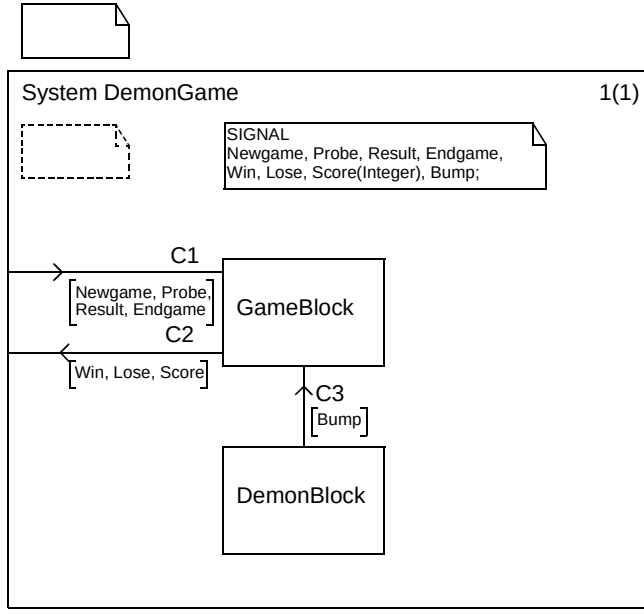


Figure 27: The system diagram

The next pages describe in detail how you proceed to add the symbols and texts to the diagram.

Customizing the SDL Editor Window

Before you start editing, you may want to resize the editor window. You may also hide and show various sub-windows using the command *Window Options* from the *View* menu.

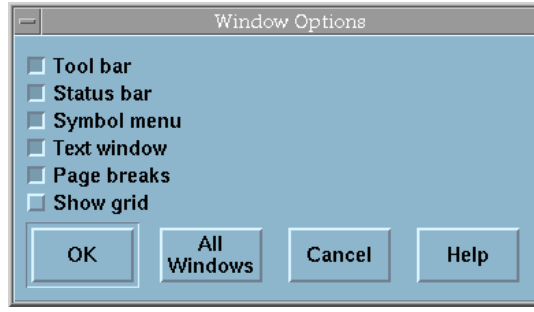
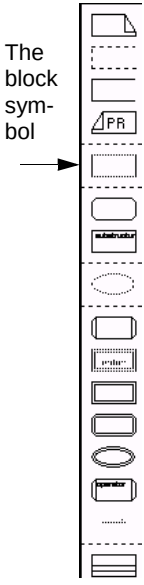


Figure 28: The window options

- You can hide and show the editor *Tool bar*, *Status bar*, printed *Page breaks* and the *grid* points.

- You may also hide and show the *text window* and the *symbol menu* by clicking on the provided quick buttons. You will however need these windows soon.

Placing Block Reference Symbols



1. Start editing the diagram by inserting the two block reference symbols (GameBlock and DemonBlock).

   To place a block reference symbol:

   – Click on the block symbol in the *symbol menu*. **On UNIX**, the symbol menu is located to the extreme right of the window. **In Windows**, the symbol menu is a separate window always placed on top of the SDL Editor window (if the two windows overlap).

   If you are not sure what symbol to use, point to or select a symbol in the symbol menu – its type is displayed in the Status Bar at the bottom of the SDL Editor window.

   – Move the mouse into the drawing area. The symbol “floats” and follows the mouse. Click to position the symbol where you want it to be. **No overlap** between symbols is allowed. (In case symbols are overlapping, an alert sound is emitted and you have to repeat the operation.)

Note: Aborting and undoing

- To abort the insertion of a symbol after you have moved the mouse into the drawing area, just press `<Esc>`.
- If you happen to perform a command or operation that you wish not had taken place, you should immediately select the *Undo* command from the *Edit* menu.

2. Once you have placed the symbol, type the name of the block: **`GameBlock`** or **`DemonBlock`**
   - You can type in text directly at the cursor's position. The cursor position can be set by clicking on the text in the symbol. However, you cannot select (highlight) text directly in the symbol.
   - Before you have started text editing, the text cursor is not flashing. Pressing `<Delete>` at this stage deletes the whole selected symbol. Once text editing has started, the text cursor is flashing and pressing `<Delete>` only deletes a character.
   - You may see a red underlining appearing in the text, if you enter a name that has incorrect syntax according to SDL. This is the general way to indicate textual syntax errors in the SDL Editor.
   - When you edit text, you may also take advantage of the *text window*, which allows you select text by dragging. **On UNIX**, the text windows is located below the drawing area. **In Windows**, the text windows is a separate window always placed on top of the SDL Editor window (similar to the symbol menu).
   - No matter where you enter the text, the text is always displayed both in the symbol and in the text window.
   - You may also note that the Organizer diagram structure is automatically updated to reflect the insertion of the diagram reference symbol (once the symbol is de-selected).

Moving and Resizing Symbols

To move a block:

- Select and *drag* the block with the mouse to the desired location. (Remember, no overlap with other symbols is allowed).

After you place the blocks where you want them, you may resize them:

- Point to one of the symbol's corners, and drag. You must be fairly close to the corner. If this method fails, first select the symbol with a click and then repeat the procedure while pointing to a selection square.

Drawing Channels between Blocks

To draw a channel from block DemonBlock to block GameBlock:

1. Select the DemonBlock symbol. A "handle" appears.

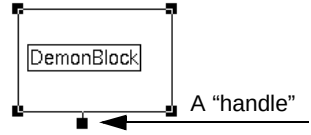


Figure 29: A block symbol's "handle"

2. Drag the handle (i.e. press the mouse button while pointing on the handle, and start moving the mouse while keeping the mouse button pressed).

3. As soon as mouse motion has begun the editor responds by drawing a line; from now on you may release the button while moving the mouse.

4. Move the mouse until it points to the GameBlock symbol. Click the mouse button; the channel is connected at both ends.

   – You may move the channel's endpoints individually by dragging them. Select the channel first if the endpoint is difficult to "hit" with the mouse.

   – You will notice a tiny selection square at the middle of some lines drawn in the SDL Editor. This can be used to create "breakpoints" on the line, thus dividing the line into different line segments. You will not use this feature in this tutorial.

The SDL Editor creates two *text attributes* associated to the channel. These text fields are displayed as *selection rectangles* which you use when entering the name of the channel and the list of the signals the channel is to convey. Initially, when the text attributes are empty, a red underlining is shown to indicate that this is not allowed according to the syntax rules of SDL.

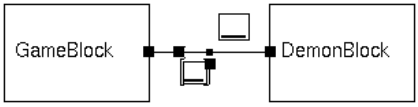


Figure 30: The channel's text attributes with red underling

The two blocks have been aligned horizontally to more easily distinguish the two text attributes.

To fill in the name of the channel C3:

- Type it directly, immediately after the channel has been drawn. (If the channel has become de-selected, select it again.)

To fill in the signal Bump into the signal list text field:

1. Click on the text field surrounded with two brackets '[ ]'. (Click in the space between the brackets.)
2. Type the name of the signal. Note that the brackets are adjusted automatically to fit the size of the text.
3. You may move the text attributes to new locations, if desired. Simply drag them with the mouse.

Drawing Channels to the Environment

To draw a channel from a block to the environment (e.g. C2):

1. Select the block.
2. Start by dragging the handle, and terminate by clicking on the frame symbol (the rectangle that encloses the diagram, see Figure 32 on page 64).

3. Fill in the name and the signals.

   – Note that, as you type the signal list, the red underlining changes and shows if and where the text is not syntactically correct according to SDL. As soon as you have entered the signal names separated with commas, the red underlining disappears and shows that the text now is syntactically correct!

   – The SDL Editor allows you to leave a text containing syntax errors. However, it is not possible to build an SDL system that contains syntax errors.

Drawing a Channel from the Environment

To draw a channel from the environment to a block (e.g. C1):

1. Start by drawing the channel from the block to the environment, as you just learned.
2. Make sure the channel is still selected.
3. Then, select the command *Redirect* from the *Edit* menu. Fill in the name and signals the usual way.

   – You may press `<Return>` to insert line breaks within the signal list, if it becomes too long.

Drawing a Text Symbol

The diagram also contains a text symbol with the required signal declarations.

1. Pick the text symbol in the symbol menu (the top symbol), insert it into the drawing area and fill in the contents as shown in Figure 27. The built-in syntax check is even more evident in this case.
2. When the contents of the text symbol are changed, the editor automatically resizes the text symbol to fit the text. You may resize it by dragging the lower right corner, or toggle between its minimized and maximized sizes by double-clicking the symbol. Try this.

   – The selection squares at the other three corners of the text symbol are gray. This means that the symbol cannot be resized by dragging any of these corners.

Resizing the Text Window

If the text window is too small to bring all the text in view, you may resize it. **In Windows**, this is done in the same way as any normal window. **On UNIX**, this is done by dragging the *sash* up or down; the sash is the small square situated to the right and above the text window menu bar; the text window is a pane of the SDL Editor window.

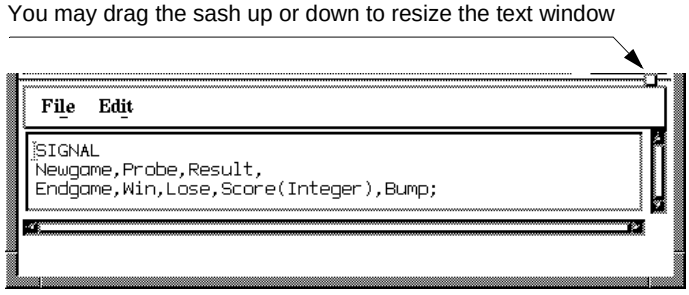


Figure 31: The SDL Editor's sash **(UNIX only)**

Other Items in the System Diagram

Except for SDL symbols, a diagram also contains the following:

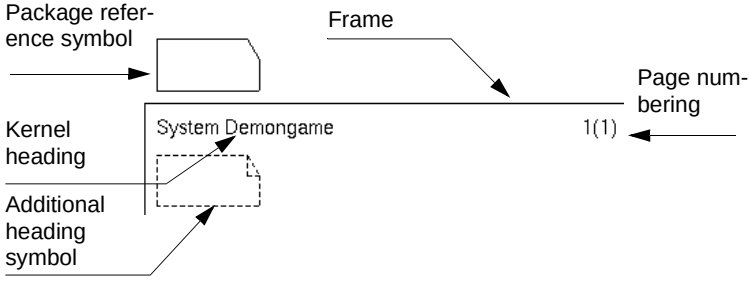


Figure 32: Other symbols

Package Reference Symbol

The package reference is used to refer to included SDL packages. This simple example does not include any packages. Just leave it empty.

The Kernel Heading

The kernel heading is automatically assigned its contents by the editor to reflect the type and the name of the diagram being edited. The kernel heading is editable, but you are not going to alter its contents in this tutorial.

Additional Heading Symbol

The additional heading symbol is not defined further according to Z.100. In the SDL Editor, it looks like a dashed text symbol. The symbol is editable and may be resized the same way as you learned for resizing text symbols, but it cannot be moved. Its intended use in the SDL Editor is, among others, to define inheritance and specialization and to specify formal parameters. You will not use this symbol in this first tutorial.

Frame

The frame surrounds the objects that are contained in your diagram. You may want to resize the frame to create a more compact diagram: simply drag any corner to do this.

> **Note:**
>
> The frame is not the same as the paper border!

Page Numbering

The page numbering is updated automatically, and reflects the name of the page and the total number of pages. It is not editable.

Saving the Newly Created System Diagram

In this exercise, you will learn the commands that store SDL diagrams on files.

1. You should now have two windows on the screen, the Organizer window and the SDL Editor window.

   – To find the Organizer window, you may at any time select the command *Show Organizer* from the SDL Editor's *Tools* menu, or click the *Show Organizer* quick button.

2. Before you save anything, open the Organizer's *View Options* dialog from the *View* menu.

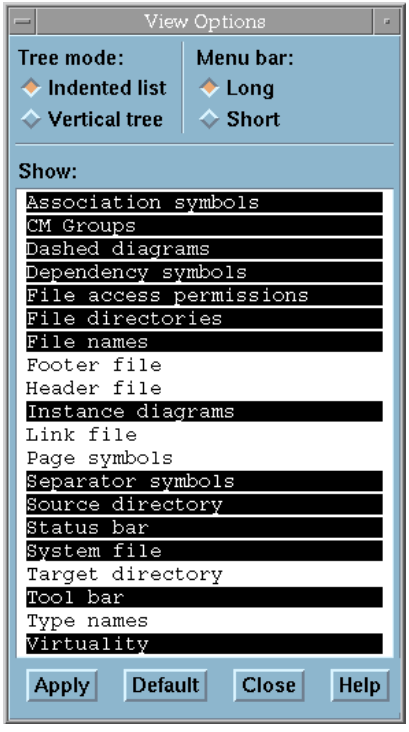


Figure 33: The Organizer's View options

3. Make sure the options are in accordance to Figure 33 and click the *Apply* button. This makes, among other things, the file and directory names visible in the Organizer.

   – The list in the dialog is a *multiple selection list*. When you click on an item in the list **(in Windows while holding down the `<Ctrl>` key)**, its selected state is toggled without affecting any other item. This makes it possible to select any number of items in the list. **In Windows**, if you by mistake click on an item without using the `<Ctrl>` key, you may press *Default* to get back to the default settings.

   – To close the *View Options* dialog, click the *Close* button. This kind of dialog is modeless, meaning that it remains open until you decide it is longer needed and close it. You are not forced to close a modeless dialog to continue working with the tool, in opposite to modal dialogs, such as the *Add New* dialog which you used for creating a new system (see Figure 23 on page 54).

4. Look at the resulting Organizer view. The system diagram icon is drawn with a gray pattern, which shows that the diagram is modified and not saved. The name of the diagram (i.e. DemonGame) is shown in bold face, to indicate that the diagram is currently open in an editor. The text to the right of the icon reads `[unconnected]` which is a convention adopted in Telelogic Tau to show that a diagram has no current binding to a file.

   There are two other diagram icons, which are `[unconnected]` These represent the references to the block diagrams that you added when editing the system diagram.

Figure 34: A modified, unconnected diagram (DemonGame)

5. Now, go back to the SDL Editor and save the SDL diagram by selecting the *Save* menu choice from the *File* menu.

   – To locate the SDL Editor window from the Organizer, you may double-click the icon for the system diagram again, which simply raises the SDL Editor window.

6. A file selection dialog is displayed. This is a generic dialog that Telelogic Tau opens whenever you are prompted to specify a file (to open, to save, etc.). The title of the dialog shows the nature of the operation, *Save* in this case.

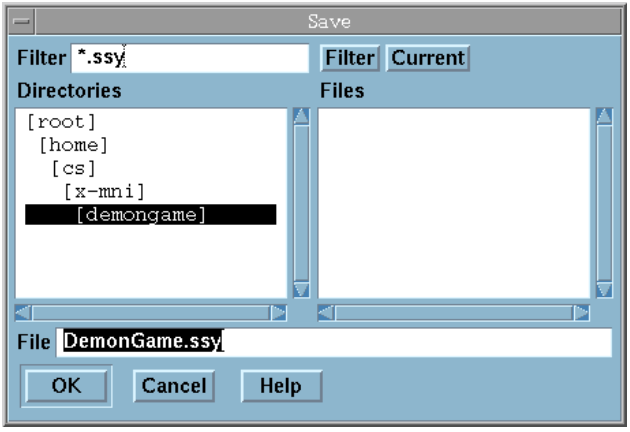


Figure 35: A file selection dialog **(on UNIX)**

On UNIX, this dialog works as follows:

– The *Filter* field is preset to `*.ssy`, which is the default file extension for files that contain SDL system diagrams. To list other files, you have to change the contents of the *Filter* field and click the *Filter* button (but do not do this now).

– The right list shows a list of files that match the file filter. It should be empty since you have not created any diagrams yet.

– The left list shows the directory structure from the root node of the file system down to the current directory. You may double-click here in order to navigate in your directory structure (do not use this list now).

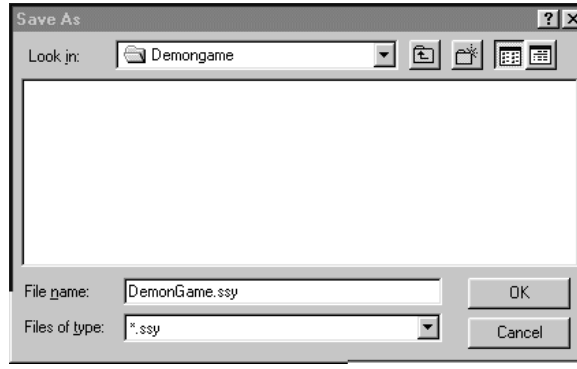


Figure 36: A file selection dialog **(in Windows)**

In Windows, this dialog works as follows:

– The *Files of type* field is preset to `*.ssy`, which is the default file extension for files that contain SDL system diagrams. To list other files, you have to change the contents of the *File name* field and click the *OK* button (but do not do this now).

– The list shows a list of files that match the file filter. It should be empty since you have not created any diagrams yet.

– The *Look in* list shows the directory structure from the root node of the file system down to the current directory. You may click here in order to navigate in your directory structure (do not use this list now).

7. The SDL Editor suggests a file name to store the diagram on: `DemonGame.ssy`. You may change to any file name; we assume however in this tutorial that you accept the suggested file name.

   So, simply click the *OK* button to accept the file name. The diagram is now stored on file. If you look at the title of the SDL Editor window, you may see that the diagram has been saved on file.

Figure 37: The SDL Editor window title

This information is also available in the Organizer structure, where the file name has changed from `[unconnected]` to `DemonGame.ssy`.

Figure 38: The diagram structure after saving the diagram

Saving the Diagram Structure

You have, so far, saved the system diagram. You should also save the Organizer's view options and diagram structure for future sessions. If you look at the Organizer's window title, you notice an ending asterisk. This asterisk denotes that the Organizer's view or structure information has been modified and needs to be saved.

Figure 39: The Organizer window title

The System File

The Organizer saves its view, along with a number of options, on a dedicated file called the system file[1]. System files are used as a means to maintain the consistency of an SDL structure and provide immediate access to the diagrams that are defined in the structure.

The system file is represented by its own icon at the top of the Organizer view, a rectangle with "SDT" in it. Even though the system file has not yet been saved, the Organizer has assigned a file name for it.

1. A system file may contain information related to any kind of SDL structure, not necessarily an SDL **system**. The term system file is a general term.

To save the system file:

1. Select the *Save* command from the Organizer's *File* menu. The Organizer responds by issuing the Save dialog.

Figure 40: The Organizer's Save dialog

2. The tool suggests a file name to store the information on: `demongame.sdt` (system files are by default assigned the extension `.sdt`). Accept the suggestion by clicking the *Save* button.

Once a system file has been created, the diagram structure and the Organizer options are saved for future sessions. You *Open* an existing system file from the Organizer's *File* menu.

More About Saving

For the purpose of this tutorial, you have learned how to save individual diagrams and how to save the system file. There are however other handy ways to save everything with one single command. Two of these methods are listed below.

- You may click the *Save All* button in the Organizer's Save dialog (see Figure 40).

- You may click the quick button for *Save* on the Organizer's tool bar. This button orders a global and silent save of all diagrams (no prompting will be issued unless special cases need your attention), including the diagram structure. (The SDL Editor's quick button for *Save* saves the current diagram only.)

Printing the System Diagram

What You Will Learn

- To print one SDL diagram
- To adjust print options
- To scale a printout

How to Print

You have now drawn your first SDL diagram. It may be convenient to print the diagram before proceeding with the remaining exercises. **On UNIX**, we assume that your computer environment includes a PostScript printer. If not, you may skip this exercise.

To print the diagram:

1. Raise the SDL Editor window.
2. Select the *Print* command from the *File* menu. The *Print* dialog is opened.

   – You may instead click the quick button for *Print*.

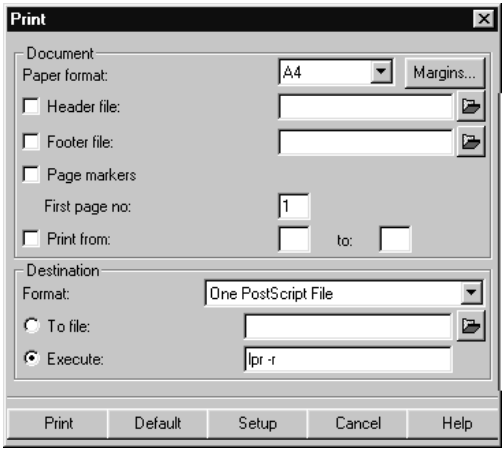


Figure 41: The Print dialog

3. If you have set up your preferences adequately, the dialog should be preset with the correct options. If not, you need to check and possibly modify at least the:

   – *Paper format* option menu

   – *Destination Format*: if you do not have access to a PostScript printer **in Windows**, you may select *MSW Print*.

   – *Execute* command (governed by the preference *PrinterCommand*).

4. Once the settings look OK, order the printout by clicking the *Print* button. **On UNIX**, a PostScript file will be generated on `/tmp` and the file will be piped to the *Execute* command that you have specified. **In Windows**, a print file will be generated by using the *Execute* command that you have specified to print it, or by using the default printer driver if you have selected MSW Print format.

   If the file is sent to a printer queue, the printer should respond almost immediately. If you are not satisfied with the size of the resulting printout, you may scale it as follows:

   – Click the *Setup* button. The *Print Setup* dialog is opened.

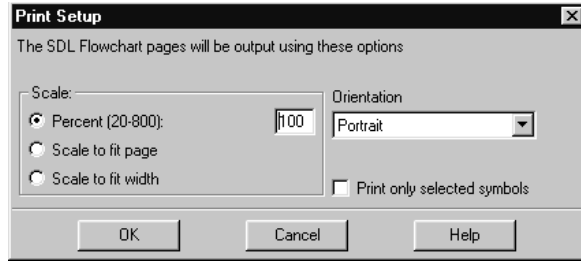


Figure 42: The Print Setup dialog

   – Adjust the scale to the value of your choice and click *OK*.

   – Order the printout once more by clicking *Print*.

Checking the System Diagram

What You Will Learn

- To invoke the Analyzer
- To set analysis options
- To work with the Organizer Log window
- To locate and correct syntax errors

Running the Analyzer

You should now check the syntax of the system diagram you created before proceeding further by creating the remaining diagrams. To do this, you will use the *Analyzer* tool, a back-end tool which is fully integrated with the Organizer.

1. Select the SDL system diagram icon in the Organizer diagram structure. Then, from the Organizer's *Generate* menu, select the *Analyze* command.

2. If you had (perhaps accidentally) modified any diagram, the Organizer first prompts you to save modified diagrams (by issuing the *Save* dialog, see Figure 40 on page 71) — in which case you should click the *Save All* button to make sure everything is OK.

3. Once the Save dialog is closed, the Analyzer dialog is opened.

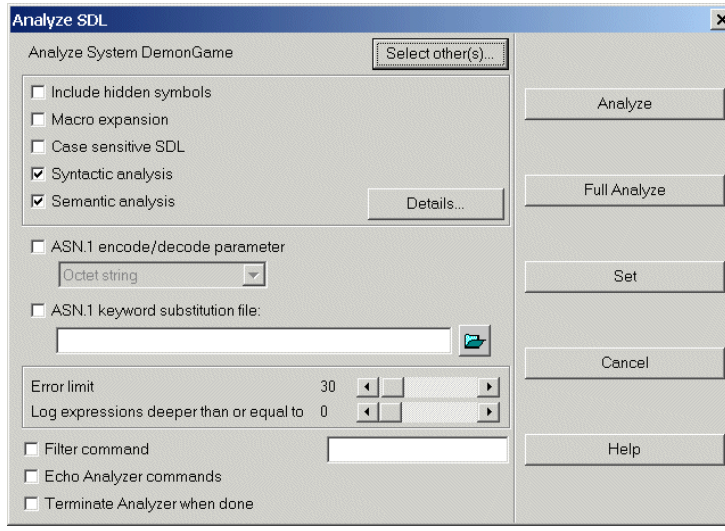


Figure 43: The Analyzer Options dialog

4. Adjust the options in accordance to Figure 43, i.e.

   – *Macro expansion* off
   – *Syntactic analysis* on
   – *Semantic analysis* **off**
   – Adjust, if required, the *error limit* slide bar to a reasonable value. This parameter defines how many errors and warnings the Analyzer will report before aborting the analysis.

5. Click the *Analyze* button. The Analyzer now starts processing the input with the options as specified in the options dialog. When it is finished, the Organizer status bar should read something with the essence “Analyzer done”, possibly appended with extra information.

6. The Organizer is provided with a textual window where important information is logged. By default, the Organizer Log window is raised as soon as information classified as “warning” or “error” is reported. If the window does not appear automatically after the analysis is complete, open the window manually. (Use the *Organizer Log* command from the Organizer’s *Tools* menu to show the window, or the provided quick button).

Looking for Analysis Errors

The diagnostics that are reported by the Analyzer are appended to the Organizer Log (together with other important messages). Look at the tail of the log for the report summary, which should look something like:

```
-------------------------------------------------

Number of warnings: <diagram dependent>
+ Analysis completed
```

(You may need to scroll down the Organizer Log window to bring the tail into view.)

The text “Number of warnings” or “Number of errors” shows how many syntactic warnings or errors that were detected in the diagram (if no warnings or errors were found, then these lines are missing altogether).

1. For the purpose of this exercise, you may need to introduce a syntactic **error** into the diagram. You may for instance remove one of the separating commas in the signal list of channel C1 (but make sure there is a space separating the signals).

   – Such a syntactic error will be detected already in the SDL Editor and marked with a red underline, but we will show how the error is reported by the Analyzer.

2. Save everything and repeat the analysis.

Correcting Analysis Errors

Your Organizer log should now report an error looking something like:

```
#SDTREF(SDL,/opt/home/tmi/demongame/DemonGame.ssy(1)
,131(25,50),3,8) (on UNIX)
#SDTREF(SDL,C:\Telelogic\SDL_TTCN_Suite4.5\work\demo
ngame\DemonGame.ssy(1),131(25,50),3,8) (in Windows)
ERROR 312 Syntax error in rule SIGNALLIST, symbol
Name found but one of the following expected:
,  ;  comment
Result Endgame;
       ?
```

How to Interpret the Error Message

Let us spend a few moments on explaining the contents of this error.

- The first part (`#SDTREF...`) is a reference to the source diagram, page, symbol, line number and finally a position within a line of text where the error was found. All references produced by the Analyzer adhere to this format in its whole or partially; the reference may in some circumstances be less precise than in the example above, depending on the Analyzer's ability to locate the exact source of error.
- The second part (`ERROR 312...`) contains the error number and an explanatory text, telling you, in this case that a comma, a semicolon or a comment was expected.
- The last part (`Result Endgame`) along with the '?' character shows more specifically where the error occurred, in this case the comma should be inserted between the signals Result and Endgame.

To display the diagram and symbol where the error was found, you may use a handy facility:

1. Select, by dragging the mouse, the lines of text containing the error message.

Figure 44: Selecting the error message

2. Select the command *Show Error* form the *Tools* menu.
   - Alternatively, you may click the *Show error* quick button.
3. The symbol where the error was found is immediately selected in the SDL Editor. The more information the reference holds, the more precise the selection. Correct the error (insert the comma).
4. To correct the next error, simply click the *Show error* button again.
5. Save the diagram and repeat the analysis until the Analyzer does not report any errors. If you feel uncertain about how to interpret and correct the errors, look at the printout for the system diagram for a reference (see Figure 27 on page 58).

- You may clear the Organizer Log window at any time, for instance between subsequent passes to make it easier to read the contents of the log. Use the *Clear Log* command from the *Edit* menu for this, or the provided quick button.

- For repeated analysis passes using the same options, you can use the *Analyze* quick button in the Organizer.

You have now designed your first SDL diagram using the SDL suite. You have also verified that the diagram is syntactically correct according to the Z.100 recommendation. Congratulations!

Creating a New Block Diagram

What You Will Learn

- To create and draw a block diagram
- To request signal dictionary support
- To work with multiple diagrams using the SDL Editor
- To open new windows on a page
- To work with multiple SDL Editor windows
- To perform syntax check on a block diagram

Creating a Block Diagram from the Organizer

In this exercise, you will create a block diagram, starting from the Organizer.

1. Locate the Organizer window and double-click the icon named GameBlock. In a similar fashion as when creating the system diagram, you will get the *Edit* dialog (see Figure 25 on page 56). Make sure the *Show in editor* option is on and click the *OK* button.

   Next, the *Add Page* dialog is opened. The dialog is used to specify the type of page (process or block interaction). The page name can also be specified. Decide if you want the pages to be autonumbered. (1, 2,... N). This functionality is enabled by default.

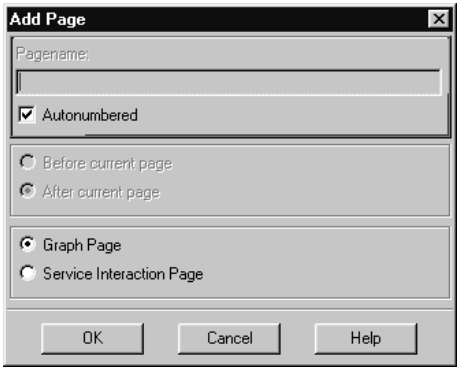


Figure 45: Prompting to add a page

2. Make sure the *Process Interaction Page* button is on and click *OK*.

The SDL Editor opens a window on page 1 of the newly created block diagram. The block diagram editor window is similar to the system diagram window; only the symbol menu differs.

Figure 46 shows the appearance of the finished block GameBlock when printed. As you may see, the diagram contains two process reference symbols, five signal routes, three connection points and a text symbol with a signal declaration.

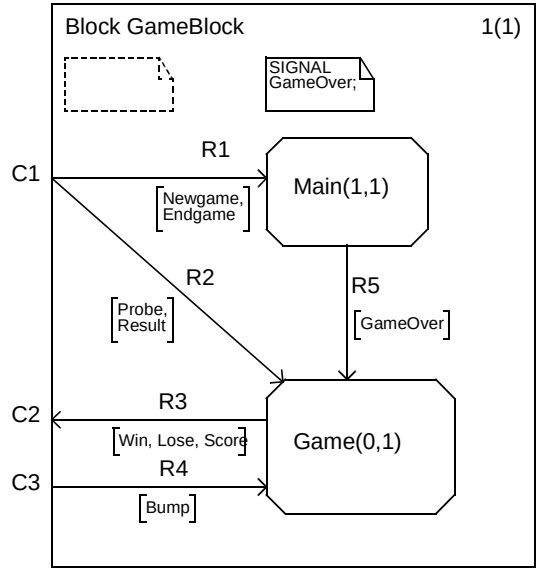


Figure 46: The block GameBlock

You should now draw the block diagram as depicted in Figure 46. You add the symbols and lines in a similar fashion as when editing the system diagram. Please spend a minute reading the three sub-sections below before starting drawing the diagram, in order to familiarize yourself with the new concepts that are introduced and how you manage them.

Process Name and Number of Instances

When you add a process reference symbol, you should specify the number of instances by appending the text directly after the name of the process reference symbol. The number of instances is the text between parentheses '( )'.

Signal Routes

Signal routes are drawn in a similar way as channels. When you select a process reference symbol, **two** "handles" are displayed.

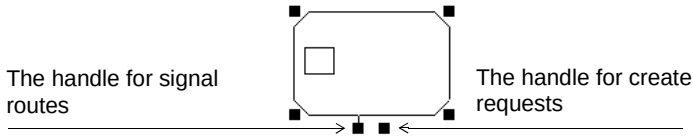


Figure 47: The two handles of a process reference symbol

- The left handle is used for drawing signal routes, in a similar fashion as channels.
- The right handle is used for drawing create requests. It is not used in this tutorial.

Connection Points

When you draw signal routes to / from the frame symbol, you should not only fill in the name and signal list, but also take advantage of graphical *connection points* to establish connections between the signal routes and the parent system diagram, i.e. connecting the signal routes to the channels. When you draw a signal route to the frame, an additional text object is created close to the frame symbol. In this text object, the name of the corresponding channel is entered. See Figure 48.

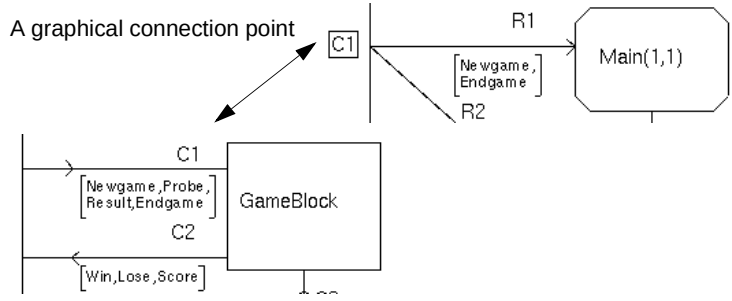


Figure 48: Graphical connection point

The figure depicts a signal route (R1) in the block diagram and the referencing system diagram with the connected channel (C1).

To edit a connection point:

- Simply select it and enter its textual contents.

Editing the Block Diagram

Now, draw the diagram as described in the following steps:

1. Start by adding the process reference symbol “Main(1,1)”.
   - As when drawing the system diagram, all text you enter is subject to an immediate syntax analysis. Errors are indicated by a red underlining, which disappear as soon as you have entered the complete text according to the SDL syntax.
   - Remember that before text editing has started, the text cursor is not flashing. Pressing `<Delete>` at this stage deletes the whole selected symbol, instead of just a typed character.
2. Draw a signal route **from** the environment to the process (use the *Redirect* command). Enter the name of the signal route: R1.

Using the Signal Dictionary for Individual Signals

You are now to specify the name of the signals to be conveyed on the signal route R1. The SDL Editor has the ability to assist you in reusing the signals that are already defined in the SDL structure (i.e. defined in the system diagram, since you are working in a top-down fashion!), with a facility known as the signal dictionary.

1. Select the signal list text field.
2. Select the command *Signal Dictionary* from the *Window* menu. The Signal Dictionary window is displayed. If necessary, move it so that you can see the signal list in the Editor window.

> **Note:**
>
> To function properly, the Signal Dictionary utility requires that the input SDL diagrams are syntactically correct. If not, you need to go back to the previous exercise (see “Correcting Analysis Errors” on page 77) and run the Analyzer in order to correct any errors.

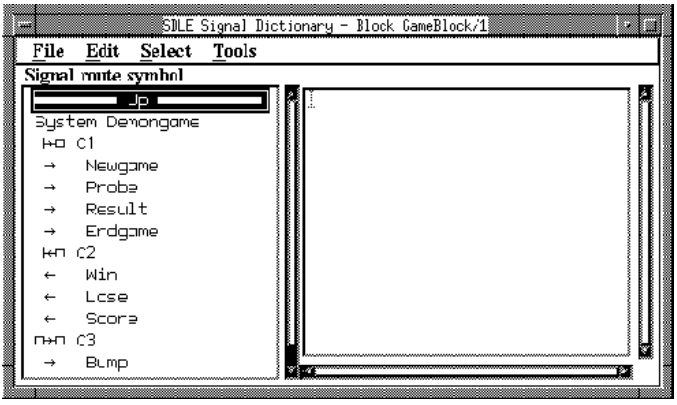


Figure 49: The Signal Dictionary window **(on UNIX)**

The exact appearance of the list to the left depends on the graphical capabilities available on your terminal.

3. Look at the left list in the Signal Dictionary window. The first section starts with the separator *Up*. This section includes the signals that are available by looking one level up in the diagram structure (i.e. in the parent diagram, system DemonGame).

Figure 50: The Up separator **(on UNIX)**

Figure 51: The Up separator **(in Windows)**

4. In this section, the first item identifies the *System DemonGame*, as expected:

Figure 52: The item symbolizing the system DemonGame **(on UNIX)**

Figure 53: The item symbolizing the system DemonGame **(in Windows)**

5. Since you are editing a signal route from the system diagram to the block diagram, you should look for all icons/lines symbolizing channels from the parent diagram, i.e. IN channels.

Figure 54: The icon symbolizing a channel from the parent diagram to the current diagram **(on UNIX)**

Figure 55: The line specifying a channel from the parent diagram to the current diagram **(in Windows)**

   – You should find exactly one channel that matches the criteria: C1. The remaining channels are either internal (C3) or are not directed into the current diagram (C2).

6. Beneath the channel all the signals that it conveys are listed. Click on the signal named Newgame:

Figure 56: The Newgame signal **(on UNIX)**

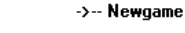


Figure 57: The Newgame signal **(in Windows)**

7. From the *Edit* menu, select the command *Insert*. The signal list in the SDL Editor is immediately updated.

   – Alternatively, you may double-click the signal in the list.

Note: Undoing the operation

If you happen to insert the signal into the wrong text field (such as the signal route name), you may select the *Undo* command from the Signal Dictionary's *Edit* menu. (The SDL Editor's Text Window has no *Undo* facility).

8. Insert the required comma and a newline in the Editor window (move the signal list if needed).
9. Double-click the signal Endgame in the Signal Dictionary.
10. The channel in the Signal Dictionary also gives a suggestion about how to fill in the connection point: C1. Select the connection point text field in the diagram and double-click the channel in the Signal Dictionary.

Using the Signal Dictionary for Multiple Signals

1. In the SDL Editor window, add the process reference symbol: Game(0,1).
2. Draw the signal route R3 from Game to the frame symbol.

   When you are to enter the signal list for R3, you do not need to enter the signals one by one as for R1, since the channel C2 is not split up into multiple signal routes, in the way that C1 becomes R1 and R2 (see your printout or Figure 27 on page 58). Instead, you may insert all signals with one single operation:

3. Select the channel C2 in the Signal Dictionary. The right list is updated to list all signals conveyed on the channel. Insert them with the *Insert* command (or double-click the channel C2).

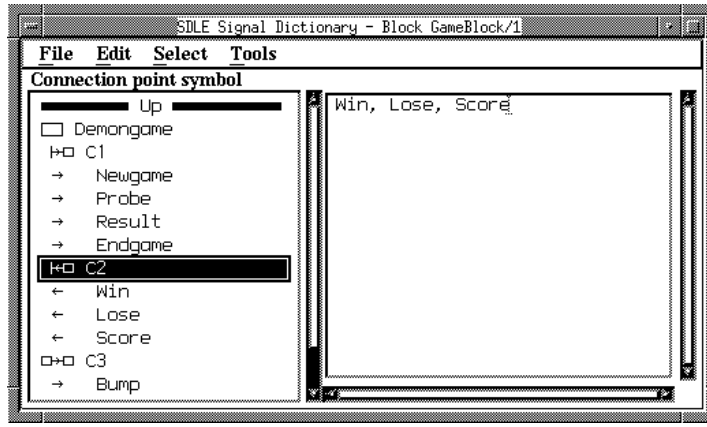


Figure 58: Selecting a channel in the Signal Dictionary window lists all signals **(on UNIX)**

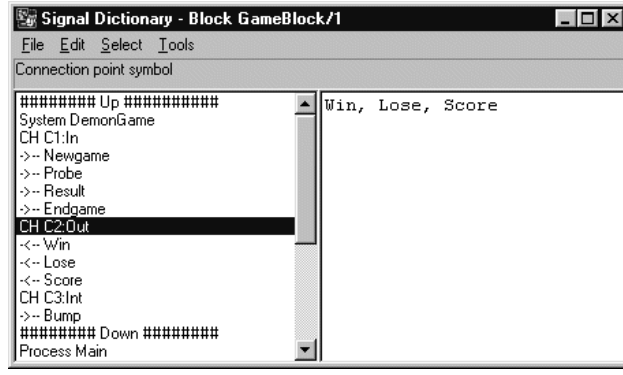


Figure 59: Selecting a channel in the Signal Dictionary window lists all signals **(in Windows)**

You are now somewhat familiar with the Signal Dictionary utility.

Completing the Diagram

1. Fill in the remaining parts of the diagram using your preferred method. When done, you may close the Signal Dictionary window (use the *Close* command from the *File* menu).

2. Once you are finished with the block GameBlock, save everything, for instance by clicking on the Organizer's *Save* quick button. Accept the default file name suggested by the Save dialog.

Working with Multiple Diagrams

You have now created two SDL diagrams. Both diagrams are currently opened by the SDL Editor; however only the diagram currently being edited is visible in a window.

The SDL Editor provides a menu named *Diagrams* where all diagrams and pages currently opened by the editor are listed.

1. Click on the *Diagrams* menu. It should now list **two** diagrams:

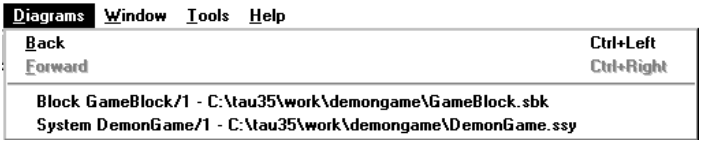
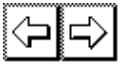


Figure 60: The Diagrams menu

Each of these menu choices correspond to a diagram and page currently opened by the SDL Editor. The file the diagram is stored on is also displayed to the right of the diagram name.

Now, bring the diagram for the system DemonGame into view:

2. Select the menu choice
System DemonGame/1 ..... DemonGame.ssy[1]
The system diagram is instantly displayed. (The block diagram is now hidden.)



– The menu choices *Back* and *Forward* can also be used to switch between diagrams and pages. The SDL Editor keeps track of

1. The exact appearance of the menu choice depends on the directory structure you are working on.

which pages you have edited and you can go back and forward in this list, in much the same way as for visited Web pages in a Web browser. There are also two quick buttons for this.

Working with Multiple Windows

So far, you have only worked with one single window on a page. The SDL Editor allows you to open new windows on the same diagram, which makes it possible to work on multiple views on a page. This is also called *instantiating* a window.

To open a new window on a page:

1. Make sure the page whose window is to be instantiated is the page currently in view in the SDL Editor. If not, use the *Diagrams* menu.
2. From the *Window* menu, select the command *New Window*. A new window showing the current page is instantly displayed (see Figure 61).
3. You may now use any window to work on the page. Any change causes both windows to be simultaneously updated. Try this, for instance by moving a symbol!
4. You probably need one window only for this tutorial, since the diagrams you are working on in this tutorial are small, on purpose. Close any of the two windows with the *Close Window* command from the *Window* menu.

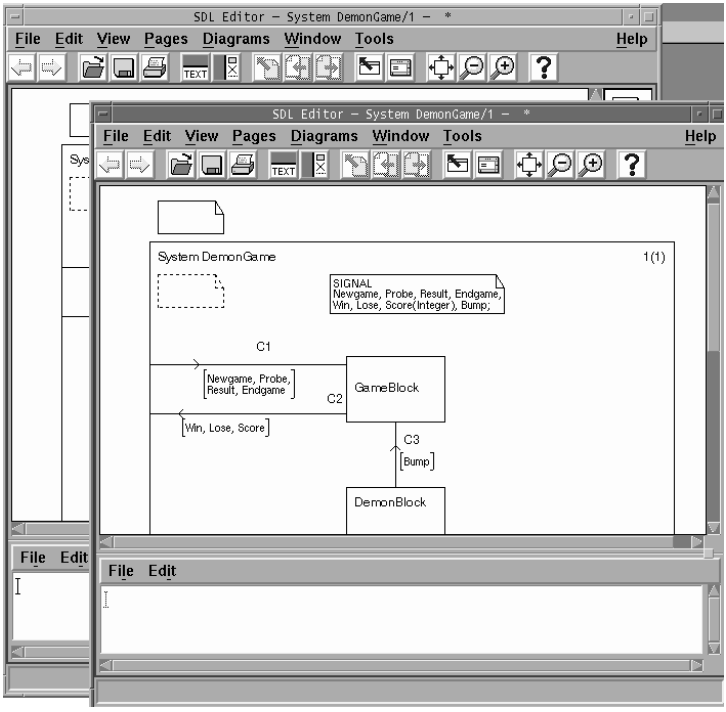


Figure 61: Two windows of the same page

Resulting Organizer View

Save everything. The resulting Organizer View should now look like:

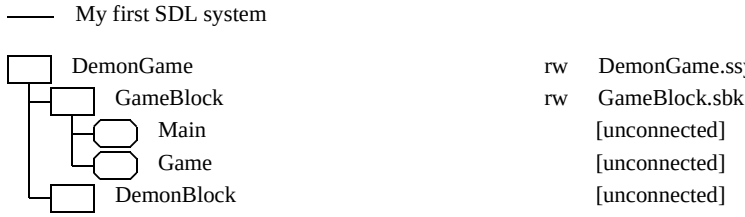


Figure 62: The resulting Organizer view

Checking the Syntax of the Block Diagram

You may now want to use the Analyzer to check the syntax of the block diagram you just created.

To analyze the block GameBlock, do as follows:

1. Select the icon for the block GameBlock to specify the block as input to the Analyzer.
2. Select the command *Analyze* and analyze the block diagram.

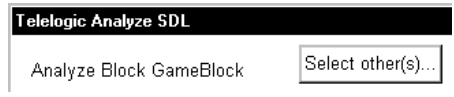


Figure 63: Specifying the input to the Analyzer

   – Note that the Analyze dialog lets you select which part(s) of the system you wish to analyze. Make sure the top text in the dialog says *Analyze Block GameBlock* before clicking *Analyze*.

3. Proceed as for the system diagram, i.e.
   – Look for any syntax errors in the Organizer Log.
   – Correct these errors.
   – Repeat the procedure if required (see "Correcting Analysis Errors" on page 77 if you do not remember how to do this).

Creating a Block Diagram From a Copy

What You Will Learn

- To create a diagram from an existing copy
- To save a diagram on a new file

Creating the Block DemonBlock

You created the block GameBlock from the Organizer by double-clicking the symbol. You may also do this from within the SDL Editor, by double-clicking on diagram reference symbols.

To create the block DemonBlock:

1. Locate the block reference symbol DemonBlock in the SDL Editor. Double-click on the reference symbol. A dialog is opened.

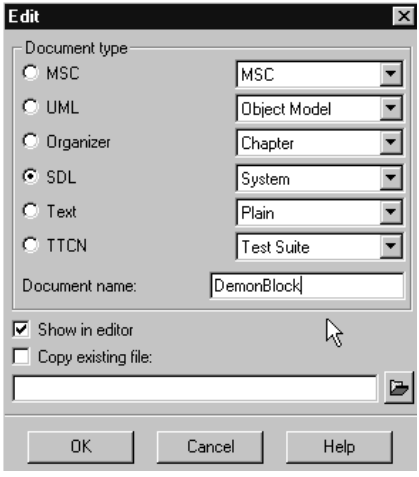


Figure 64: Prompting to create the block DemonBlock

You will now create the diagram by using a copy of an existing file. The Telelogic Tau installation contains a number of SDL examples, among which the diagrams that build up the DemonGame example may be found. These diagrams are by default stored in a subdirectory to the installation directory. The name of the directory should be `$telelogic45/sdt/examples/demongame` **(on UNIX)**, or

`C:\Telelogic\SDL_TTCN_Suite4.5\sdt\examples\demon-game` **(in Windows)**.

Your next task is to specify the location of the file that contains the block DemonBlock. This file is named `DemonBlock.sbk`

Note:

You may need to contact your system manager to find out the exact location of the directory mentioned above. If you fail in finding the directory with the DemonGame example, do not give up! You may always create the remaining diagrams with the New option, and design them with the SDL Editor in a hand-drawn fashion, as you learned in the previous exercises.

2. Make sure the *Copy existing file* button is turned on.
3. Then, either:
   - Type in the file name, **including the directory path** (according to above),

     or
   - Click on the *folder button* to the right of the text field. A file selection dialog (with the title *Select File to Copy for Block DemonBlock*) is displayed.
   - In the dialog, navigate in the directory structure until you have located the directory where the diagrams are stored (see the directory path above). **On UNIX**, you double-click the directory names in the left list and find any block diagram files in the right list. **In Windows,** you select directories from the *Look in* box or double-click directories in the list.
   - Select the file `DemonBlock.sbk` that appears in the list and click *OK*.

Note: Accessing files on other disks

In Windows, you may use so-called UNC paths to access network disks by using the syntax `\\disk\directory\` when typing the path to file names.

On UNIX, the possibility to change to the `[root]` directory using the Telelogic Tau file selection dialog may or may not work properly, depending on your computer system and network file system. You may need to type in a leading slash ('/'), followed by a name, then click *OK* in order to access files that are stored on another disk than the one you are currently working on (but try first to double-click the `[root]` directory).

4. Close the *Edit* dialog by clicking on the *OK* button. The SDL Editor shows the diagram in a window. The diagram when printed should look like Figure 65.

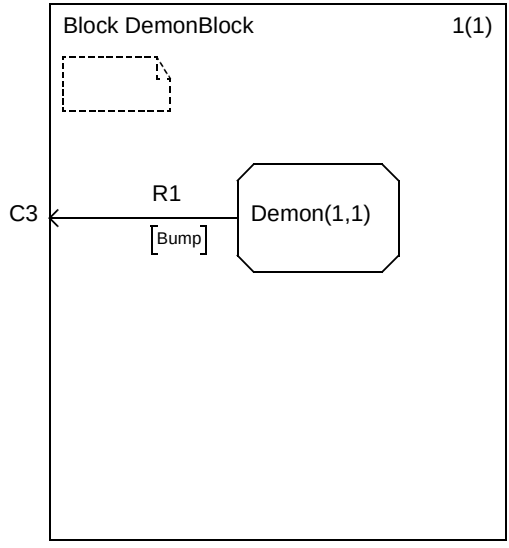


Figure 65: The block DemonBlock

5. Now, save the diagram from the SDL Editor (use the *Save* quick button). A file selection dialog is displayed, with the suggested file name `DemonBlock.sbk`.

6. Accept the suggestion by clicking the *OK* button. The resulting Organizer's diagram structure should be as follows:

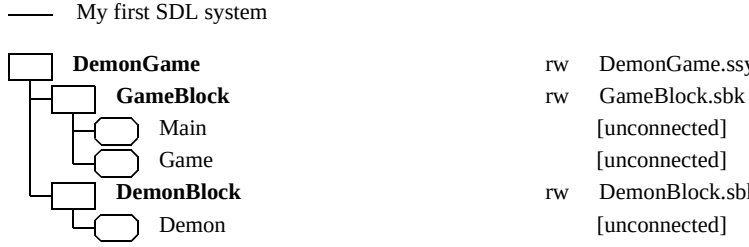


Figure 66: The resulting Organizer list

You may use the method described above to take a copy of an existing diagram for any diagram in the remainder of this tutorial. However, we recommend that you draw the diagrams from scratch to get yourself acquainted with all editing features of the SDL Editor. How to draw a process diagram (described next) is somewhat different from drawing a block diagram.

Creating a Process Diagram

You have now created the structural elements of your SDL system. This structure needs now to be completed with the implementation, i.e. the process flow charts that describe the behavior of the system.

In the previous exercises, you have learned how to create new diagrams, either from the Organizer or from the SDL Editor, so we will not focus on these details any more. Feel free to double-click icons in the Organizer or in the SDL Editor, or to use the Organizer *Edit* command, depending on your preference.

In the next exercise, you will instead learn how to use the SDL Editor for drawing process diagrams. Let us start with the process Demon, which is depicted in Figure 67.

Editing the Process Demon

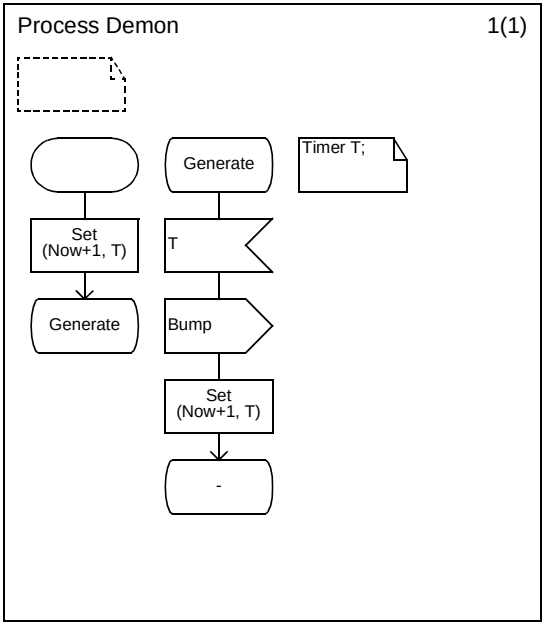


Figure 67: The process Demon

On the next pages, you will find suggestions about how to use the SDL Editor to draw the diagram.

What You Will Learn

- To add symbols with the double-click facility
- To work with the clipboard functions
- To insert symbols in a flow
- To request grammar help

Creating the Diagram

1. Edit the Demon diagram. When you are prompted to add a page, make sure that you specify a page with the type set to *Graph Page*.

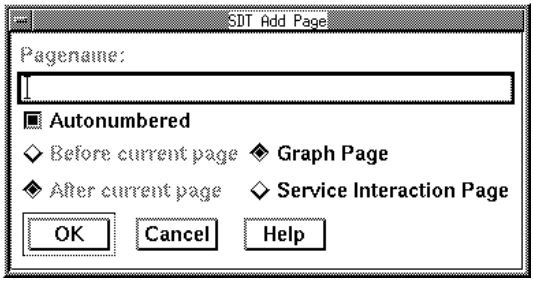


Figure 68: Specifying page type to graph page

2. When the SDL Editor responds by displaying the (empty) diagram, you notice that the appearance of the symbol menu is different; it now contains the symbols that are allowed on a flow diagram (such as state and input symbols).

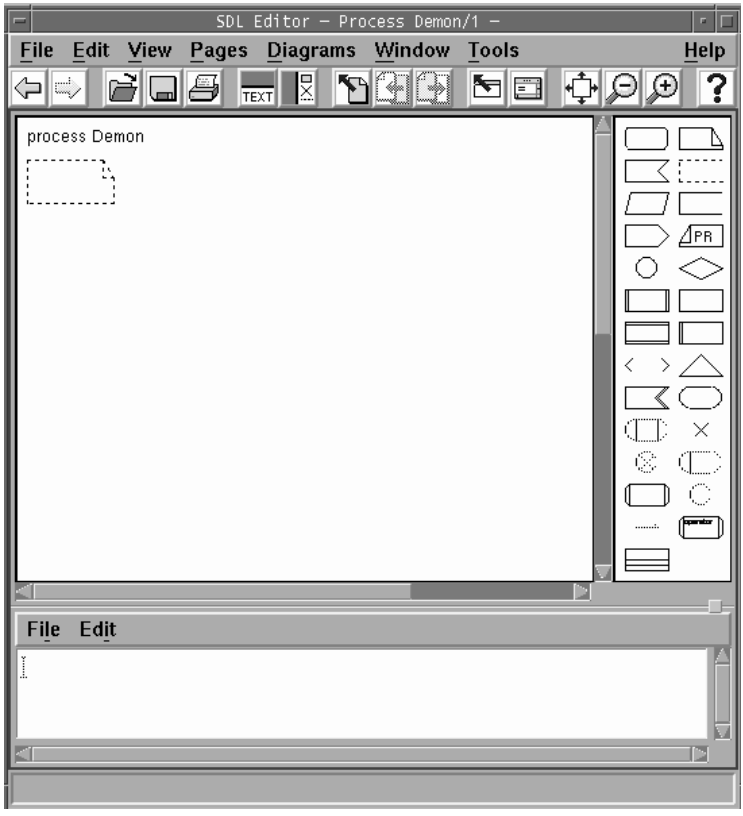


Figure 69: The SDL Editor window for flow diagrams **(on UNIX)**

The diagram consists of two branches of symbols (see Figure 67 on page 96). When you append symbols to a branch, the editor may automatically interconnect the symbols with flow lines.

You may select to enter the text into each symbol once the symbol has been inserted, or insert all symbols and then edit the text, or a mix of both methods.

Creating the Left Branch with Grammar Help

To create the left branch:

1. Select the start symbol in the symbol menu and place it in the drawing area at a suitable location.

   – Remember that when you point to or select a symbol in the symbol menu, its type is displayed in the Status Bar at the bottom of the SDL Editor window. Use this if you are not sure what symbol to pick.

2. Double-click the task symbol in the symbol menu. An empty task symbol should now be appended to the start symbol.

   When you are to edit the task symbol containing the statement that sets the timer, let us assume, for the purpose of this exercise, that you do not have the grammar for the Set statement in mind.

   The SDL Editor provides a context-sensitive facility, the *Grammar Help window*, that assists you in entering correct SDL expressions. You will now use it in order to fill in a correct set expression.

3. Select the command *Grammar Help* from the *Windows* menu. The SDL Editor responds by displaying the *Grammar Help window*.

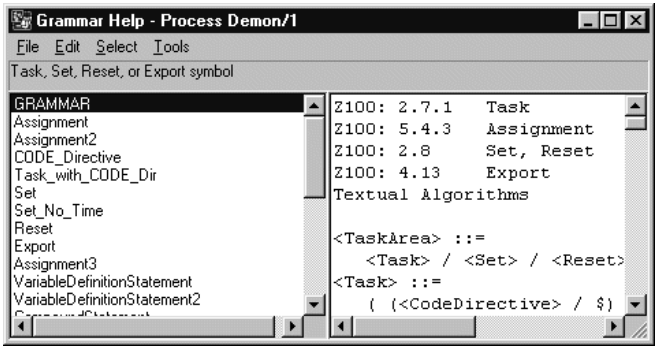


Figure 70: The Grammar Help window

– The left list shows a number of “use cases”, each of them identified with their name. The first one is the GRAMMAR for the selected object.

– The top of the right list shows a number of references to the Z.100 definition.

– Beneath the Z.100 references are listed the formal textual (SDL/PR) expressions that are legal to add to the symbol. (The formal expressions need of course to be replaced by the actual values that are used in your context).

4. The use case you are to use is the set of a timer, so locate the item titled *Set* in the left list and select it.

   – The right list is updated to reflect the formal grammar for the expression: “SET(Now+Expr, TimerName)”.

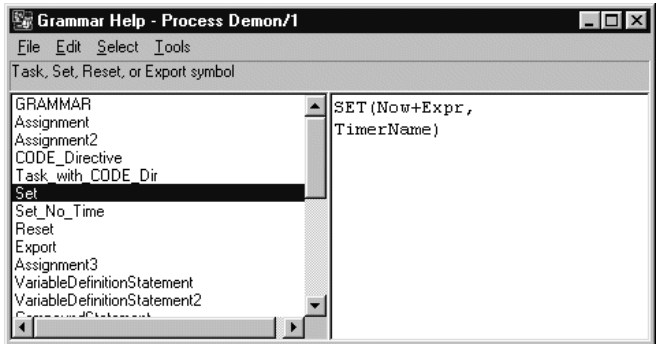


Figure 71: The grammar for a Timer Set

5. Insert the formal text into the task symbol by selecting the *Insert* command from the *Edit* menu. The task symbol is immediately updated.

   – You may also double-click the *Set* item in the list.

6. Now, change the generic names *Expr* and *TimerName* to their actual values (*1* and *T*, respectively).

   – You use the SDL Editor's text window for this. Drag for instance the mouse over the text to be changed and type in the new text to substitute it with.

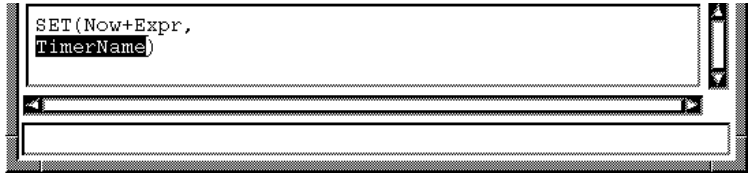


Figure 72: Edit the text in the text window **(on UNIX)**

You have now learned the basics about how to work with the Grammar Help.

7. To finish the left branch, double-click a state symbol and enter the text: `Generate`

Creating the Right Branch

To create the right branch:

1. Copy the newly added state symbol to the clipboard. You find the clipboard commands, e.g. *Copy*, on the *Edit* menu or on the pop up menu that is activated with the right mouse button.
2. Paste the state symbol. Following *Paste*, you should specify the location of the new symbol; move the mouse until you point to a suitable location and terminate with a click with the left mouse button.
3. Append an input symbol with a double-click. Enter the text: `T`
4. Append the output of the signal Bump with a double-click and enter the text `Bump`.
5. Copy the task symbol with the text “SET (Now+1, T)” to the clipboard. But, **do not** paste right now.
6. Point to the output symbol Bump. Press the right mouse button and select the *Insert Paste* command. This pastes and connects the task symbol.
7. Terminate the branch by double-clicking a state symbol and typing a hyphen (-).
8. Finally, add a text symbol and type in the declaration of the timer T.
9. If desired, resize the frame symbol.
10. *Save* the diagram with the file name `Demon.spr`.

This concludes the editing of the process Demon.

Editing the Process Game

First, create the process diagram Game in the usual way. In this exercise, you will learn some other editing functions:

What You Will Learn

- To edit parallel flow branches
- To interconnect symbols

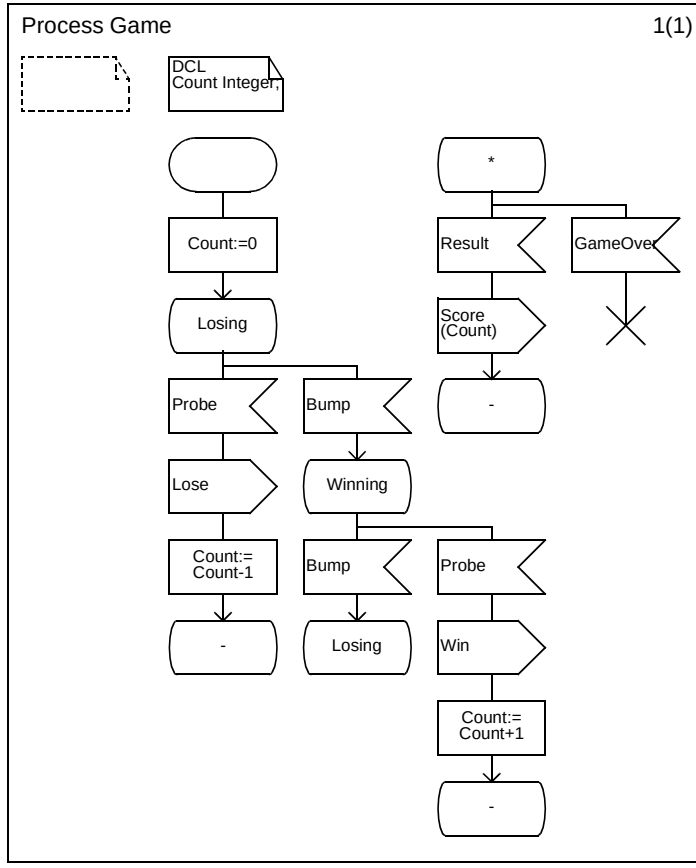


Figure 73: The process Game

You may proceed editing the process diagram in Figure 73, as will be described below:

Editing the Start Transition

1. Insert the start symbol, the following task symbol and the state symbol Losing.
2. You will now insert two input symbols in parallel. To do this, first make sure the state symbol is selected. Then, press `<Shift>` and double-click two input symbols (`<Shift>` must be kept pressed while you do this).
3. Release `<Shift>` and select the left input symbol.
4. Fill in the name of the input symbol (**`Probe`**), and complete the left branch.
5. Select the right input symbol, fill in the name (**`Bump`**) and complete the branch without bothering about the subbranch that starts with the input of the signal Probe in the state Winning.
6. Select the Probe input symbol in the left branch. On the *Edit* menu, use the *Select Tail* command to extend the selection to the end of the branch.
7. *Copy* the selection and *Paste* it. Move the selection (which appears as a set of symbols) to a suitable place and paste it with a click with the left mouse button. If *Paste* fails (because of insufficient space or overlap), an alert sound is issued — please try again.
8. Change the text in the input symbol from Lose to **`Win`**.
9. Change the text in the task symbol to **`Count:=Count+1`**.
10. To interconnect the state symbol Winning with the input symbol Probe: select the state symbol – a handle appears –

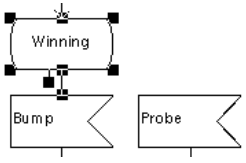


Figure 74: The selected state and its handle

drag the handle while pressing the mouse and release the mouse when it points to the input Probe symbol. A line is drawn between the symbols:

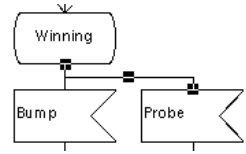


Figure 75: The two branches are connected

11. Conclude the diagram by drawing the remaining parts and saving the diagram.

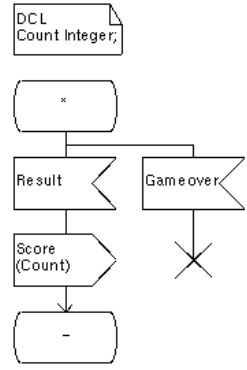


Figure 76: Remaining parts to edit

Editing the Process Main

The process Main is the last diagram to create and edit. If you find this tedious, you may skip this exercise and create the diagram as a copy from the files that are enclosed in the distribution (how to do this is described in section "Creating a Block Diagram From a Copy" on page 92). Figure 77 shows the appearance of the diagram to create.

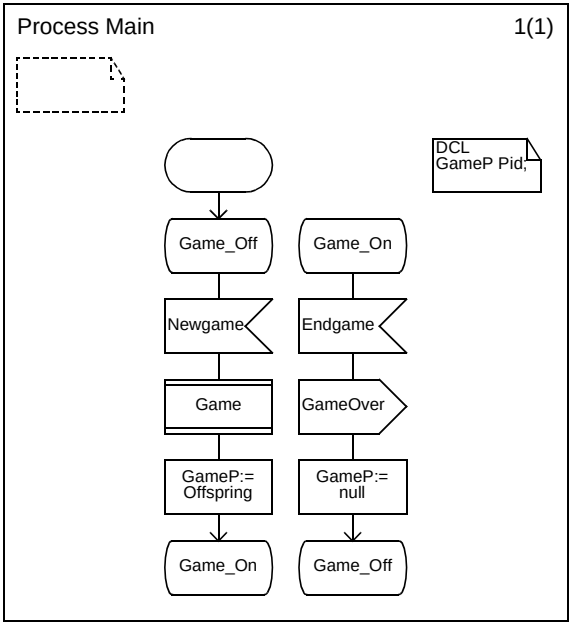


Figure 77: The process Main

More About the Organizer

When you are ready with the diagram, save everything. The diagram structure in the Organizer Window should now look like this:

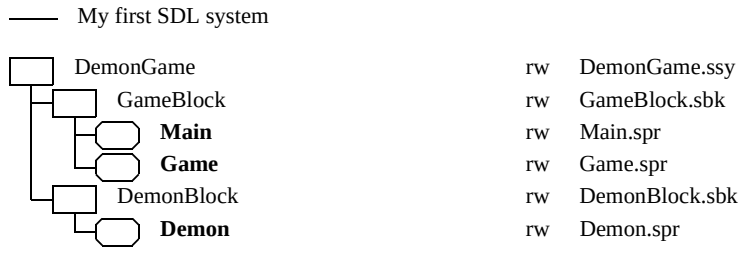


Figure 78: The resulting diagram structure

In this tutorial, you have only browsed through a minor part of the available functionality. You may for instance customize the Organizer to display the information using different view options.

What You Will Learn

- To work with vertical trees
- To expand and collapse the diagram structure
- To rearrange diagrams in an Organizer structure
- To display directories and pages
- To print the entire system

Tree View

1. Bring up the *View Options* dialog (see Figure 33 on page 66), click on the *Vertical tree* radio button and click on *Apply*. The Organizer window changes its presentation mode:

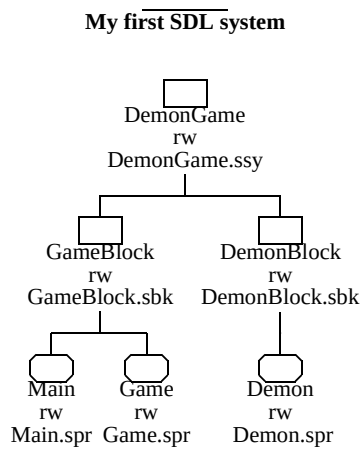


Figure 79: A vertical tree structure

2. Apply the *Indented list* mode again.

Expand / Collapse

You may make parts of the diagram structure invisible (and visible again) with the *Expand* and *Collapse* commands from the *View* menu. (*Expand* is available on collapsed nodes only (indicated by the small triangle), while *Collapse* is available on expanded nodes that have a substructure).

1. *Collapse* the block GameBlock. The subtree for the block GameBlock is reduced to the node, with a small triangle added.

Figure 80: A collapsed node

2. Expand the subtree again (use *Expand Substructure*).

Rearranging Diagrams

The Organizer lets you rearrange the order of appearance of symbols.

You can either do that with arrow keys (`<Up>`, `<Down>`, `<Left>` and `<Right>`) or with the quick buttons *Move up* and *Move down*.

Say that you want to rearrange the order of appearance of the blocks DemonBlock and GameBlock:

1. Select the block GameBlock.

2. Click once on the quick button *Move down*, alternatively press `<Shift>` and type the `<Down>` arrow key. The result becomes:

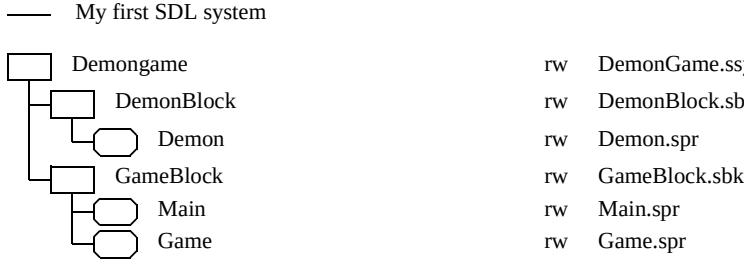


Figure 81: Rearranged GameBlock and DemonBlock

3. Change back to the original order of the diagrams.

Diagram Pages

1. In the *View Options* dialog, turn the *Page symbols* item on.
2. Apply the options – the result becomes a list where the SDL pages are made visible.

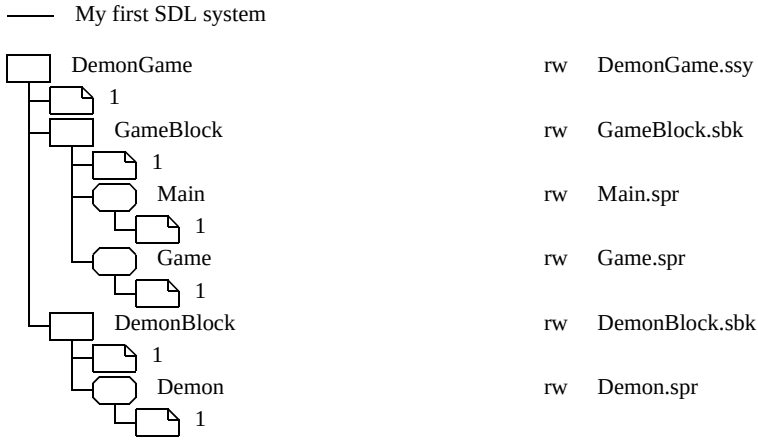


Figure 82: Diagram pages are displayed

Printing the System

The Organizer lets you print all diagrams that are included in the system with a single command. You may also include a table of contents:

1. De-select all diagram symbols, or select the system diagram.
2. Click the quick button for *Print*. This opens the Organizer's Print dialog.
3. Turn the *Table of contents* toggle button on and click on *Print* to order a global printout of all SDL diagrams, including a table of contents (see Figure 83).

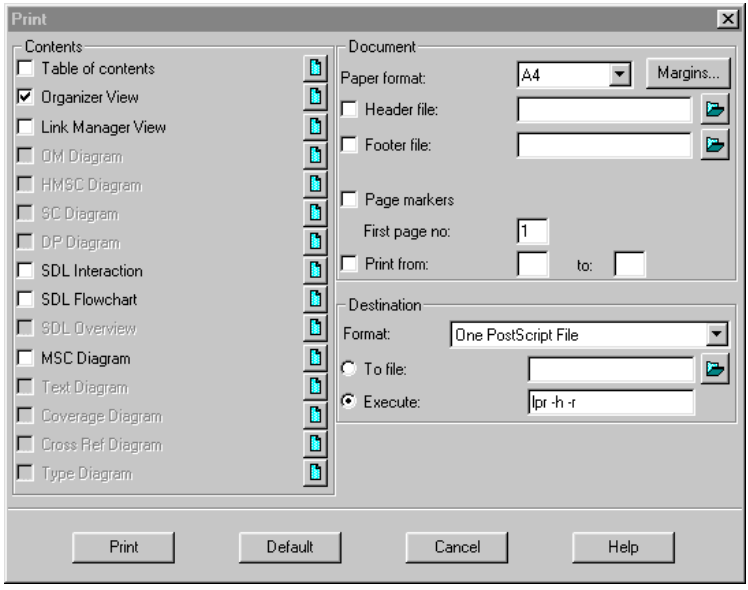


Figure 83: Including a table of contents

You have now created and printed your first complete SDL system using the SDL suite. Your next task is to check the complete system with respect to SDL syntax and semantics.

Analyzing the Complete System

What You Will Learn

- To perform syntactic and semantic analysis on the whole system
- To generate files containing definitions and cross references

Enabling Semantic Analysis

To analyze the system, you should also enable the semantic checker. To do this:

1. Select the system diagram icon.
2. Use the *Analyze* command from the *Generate* menu.
3. Adjust the analyzer options according to the picture below:

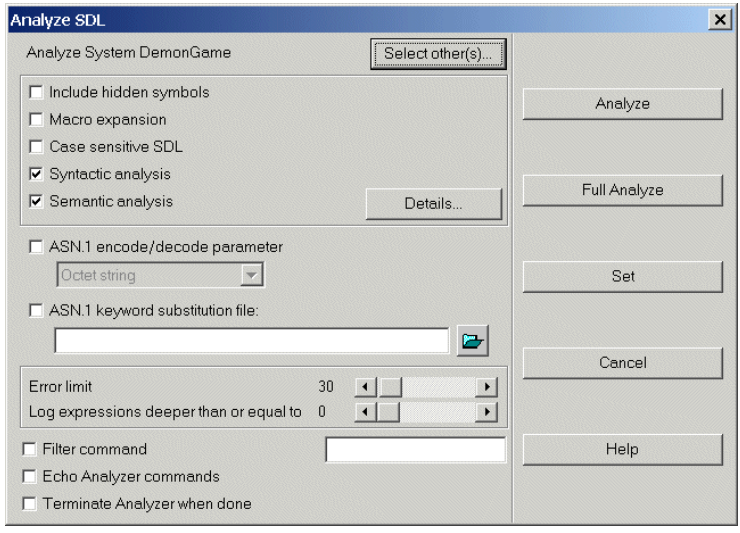


Figure 84: Including semantic analysis

- Make sure the Analyzer generates a file with *cross references*, by turning the toggle button *Generate a cross reference file* on. You will need this file in a later exercise in this tutorial.
- The semantic Analyzer has some other options, each one of these individually activated with a toggle button. They have no impact on this tutorial.

4. Click the *Analyze* button.
5. When the Organizer status bar reads “Analyzer done”, look at the *Organizer Log* for any errors reported by the Analyzer.
6. If required, correct the errors and repeat the procedure. How to locate errors in the source SDL diagrams was described in a previous exercise, see “Looking for Analysis Errors” on page 76 and “Correcting Analysis Errors” on page 77.
7. The tail of the Organizer log should contain the following output when the system is syntactically and semantically correct:

```
+ Analysis started
Conversion of SDL to PR started
Conversion to PR completed
Syntactic analysis started
Syntactic analysis completed
Semantic analysis started
Semantic analysis completed
+ Analysis completed
```

Terminate this exercise by saving everything. You may also want to print the diagrams again (see “Printing the System” on page 110 for how to do this).

Managing Message Sequence Charts

Besides the SDL tools, Telelogic Tau also support the Z.120 recommendation, also known as Message Sequence Charts (MSC). You should have a basic understanding of MSC symbols to fully understand this exercise.

In this tutorial we will demonstrate some application areas of MSCs.

- First, an MSC may be used for describing the requirements on the dynamic behavior of a system, viewed as a “black box” which receives external stimuli (corresponding to SDL signals issued from the environment) and respond by sending SDL signals to the environment.
- MSCs may also help you to understand a problem, by offering a way of presenting, in graphical form, some use cases which have been identified, before proceeding with the design in SDL.
- Generating MSCs as the result of a simulation of a system also provides a mean to understand the dynamic behavior and verify it against the expected behavior.
- Finally, MSCs can be input to a *Validator* where you can verify that the scenario that the MSC is describing may actually occur and under what circumstances.

What You Will Learn

- To add MSCs to the diagram structure
- To associate SDL diagrams and MSCs
- To create MSCs
- To edit MSCs

Inserting an MSC into the Organizer

To create an MSC, you use the Organizer, where the MSC will be managed as an *Other Document*. In this exercise, we will create an MSC where you will describe the dynamic behavior of the system DemonGame. You will also use this MSC as a reference when simulating and validating the system (this is done in later exercises).

To create an MSC:

1. Select the Organizer chapter *Other Documents*.

2. From the *Edit* menu, select the command *Add New*, or click the quick button for this.
3. The *Add New* dialog is opened, prompting you to specify a diagram name and type.

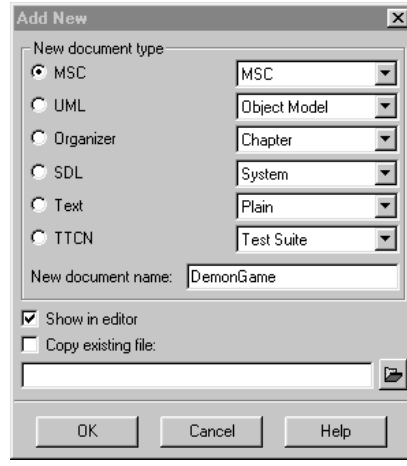


Figure 85: Specifying the name and type of the diagram to add

Adjust the dialog options as in Figure 85 above:

- Set *New document type* to *MSC*
- Change the name to **`DemonGame`**.
- *Show in editor* should be turned on

4. Click the *OK* button. An *MSC icon* appears in the Organizer's *Other Documents* chapter; the lower part of your Organizer window should look like Figure 86, once the MSC Editor is started (you may have to raise the Organizer window if the MSC Editor covers it).

Figure 86: The Organizer structure with an MSC added

The MSC you have inserted into the Organizer is intended to describe the behavior of the system and you will associate it with the system diagram. The association will be visible in the Organizer.

5. Make sure the MSC icon is selected and select the *Associate* command from the *Edit* menu. A dialog is displayed.

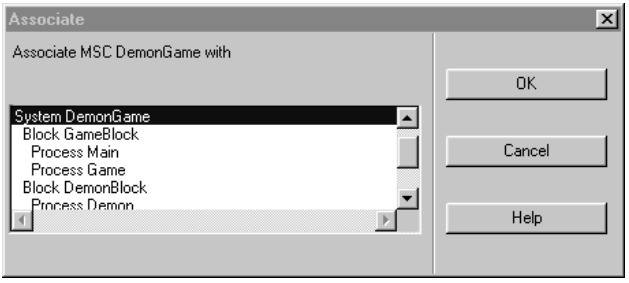


Figure 87: Associating an MSC with an SDL diagram

6. Select the system DemonGame item in the list and click *OK*.
7. Look at the resulting Organizer structure. In addition to the MSC icon in the *Other Documents* chapter, an MSC Link icon appears, connected to the system diagram icon. If you select it, the Organizer's status bar informs you about the link to the actual MSC.
   - If you do not see any MSC Link icon, check the Organizer's *View Options*, turn the option *Association Symbols* on and click *Apply*.

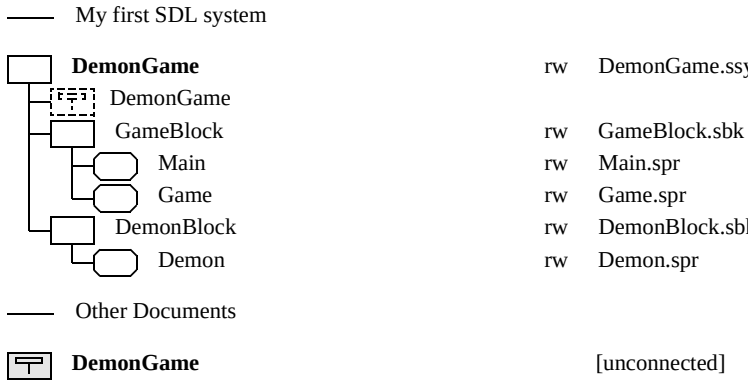


Figure 88: Association between the System diagram and the associated MSC

Editing an MSC

1. Raise the MSC Editor window for the newly added MSC symbol. The window of the MSC Editor looks similar to the SDL Editor window, but provides of course a different symbol menu and different set of commands and quick-buttons.

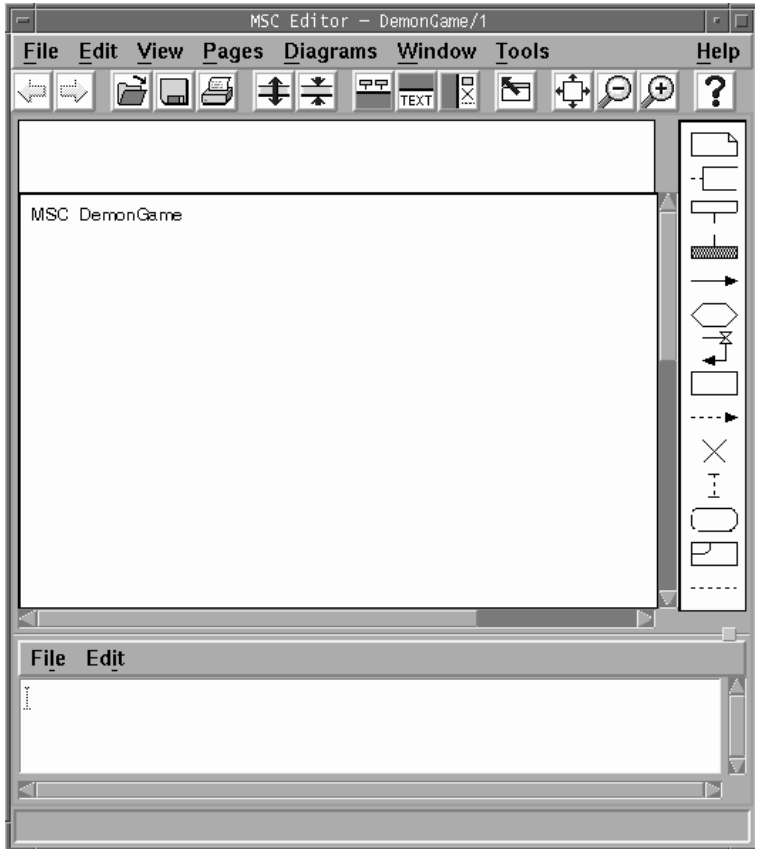


Figure 89: The MSC Editor window **(on UNIX)**

Your next task is to use the MSC Editor to create the following diagram:

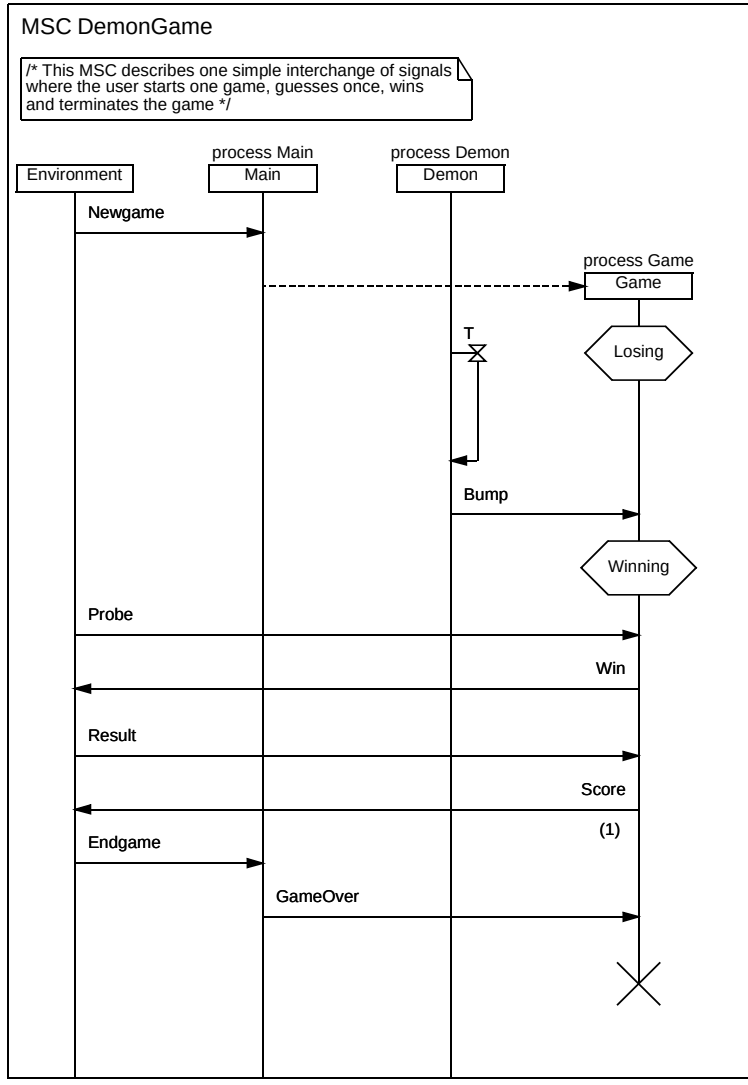


Figure 90: The MSC for the system DemonGame

The MSC basically consists of four instances (the vertical lines starting with a rectangle), a number of messages (the horizontal lines ending with an arrow), a create process (the dashed horizontal line), a timer (the symbol starting with an hourglass and ending with an arrow) and two condition symbols (with the shape of a hexagon). You also find a text symbol, containing a textual comment in it.

How to Draw the MSC

We suggest that you draw the MSC as described below. If you are unsure what symbol in the symbol menu to use, select or point to a symbol and look at the description in the Status Bar.

1. Start by entering the text symbol and fill in its contents. (This is done in the same way as with the SDL Editor).
2. Then, insert the three instances with the instance name Environment, Main and Demon:
   - To insert an instance, locate the instance head symbol in the symbol menu, select it and place it into the drawing area as shown in Figure 90; as soon as you insert an instance head, the MSC Editor automatically appends an instance axis (with an infinite length).
   - Type in the text to assign the instance name (**`Environment`**, **`Main`**, **`Demon`**)
   - To assign the instance kind (**`process Main`**, **`process Demon`**), select the small rectangle located immediately above the instance head symbol and type in the text.

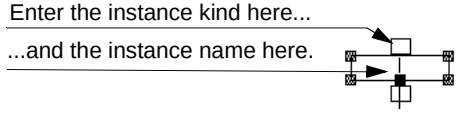


Figure 91: The text attributes associated to an instance head

   - If you are not satisfied with the placement of an instance head, you may drag the symbol to a new location.

3. Once the three instances are added, insert the message Newgame:
   - Select the message symbol in the symbol menu
   - Move the pointer into the diagram. You will notice that a circle indicates a start position outside an instance axis.
   - Click once on the instance axis Environment to define the start of the message.
   - Move the pointer towards the instance axis Main. The message arrow follows the pointer, and a filled circle now indicates an end position outside an instance axis.
   - Click a second time on the instance axis Main to specify the end of the message.
   - Type in the name of the message (`Newgame`).
   - If you are not satisfied with the placement of a message, you may move it up or down by dragging the mouse. You may also move only the start or end position of the message along the instance axis.
4. The instance Game is dynamically created. To add Game, you use the create process symbol. You insert it in a similar fashion as a message:
   - Select the create process symbol in the symbol menu.
   - Click once on the instance axis Main to specify the source of the create process symbol.
   - Click a second time to specify the location of the instance head. A process create and an instance head with its axis are inserted.
   - Fill in the instance kind and instance name fields (after you have selected the instance head).
   - If desired, you may move the instance head symbol.

5. Continue by adding the first condition symbol to the instance axis Game:
   - Select the condition symbol in the symbol menu, and move the pointer to the instance axis. Click to insert the symbol and fill in the name of the condition: `Losing`.
   - The condition symbol may now be moved vertically along the instance axis.
6. Add a timer to the instance axis Demon:
   - Select the timer symbol in the symbol menu.
   - Click once on the instance axis to specify the base of the timer symbol.
   - Move the pointer downwards and click a second time on the same instance axis to locate the end of the timer (the end must reside below the source).
   - Enter the name of the timer: `T`
   - You may drag the start or endpoint to resize the timer symbol, if required. You may also drag the symbol to move it up or down.
7. Insert the message Bump.
8. Add the second condition symbol, `Winning,` to the instance axis Game.
9. Add the remaining messages. The message Score also contains a parameter with the value 1. To enter the parameter value, select the lower of the two selection rectangles and type in the text `1`.

Figure 92: The text attributes associated to a message

10. Conclude the editing of the MSC by adding a process stop symbol.
    - Select the symbol in the symbol menu.
    - Place it by a click on the instance axis Game, below the last message.

11. Before leaving the MSC Editor, you should save the MSC. When saving the newly created diagram, the editor suggests the file name `DemonGame.msc`. Accept the suggested file name by clicking the *OK* button.

Using the Index Viewer

In this exercise, you will practice on the Index Viewer. The Index Viewer is a dedicated tool that presents a graphical view of the definitions and references to SDL entities available in an SDL system. It manages virtually all SDL information related to a system and has a number of facilities for navigating back to the source SDL diagrams.

A prerequisite to this exercise is an up-to-date cross reference file containing the definitions and references, which was generated when you last performed a semantic analysis of the system.

If you have changed any of the SDL diagrams since the last analysis of the system, you should regenerate the file. Perform a new semantic analysis and make sure the option *Generate a cross reference file* is on (see Figure 84 on page 112).

What You Will Learn

- To start the Index Viewer
- To look for definitions
- To look for references

Starting the Index Viewer

1. You start the tool with the *Index Viewer* command from the Organizer's *Tools* menu and its sub-menu *SDL*. Select the sub-menu *SDL* and the menu choice *Index Viewer*.

   – As an alternative, you may click the quick button for the *Index Viewer*. You will then be prompted to save unsaved diagrams. The SDL system is then analyzed and a new cross reference file is generated automatically.

2. The Index Viewer window is displayed. Start by opening the newly created file `DemonGame.xrf` (unless you used the quick button, in which case the file is automatically opened).

3. The Index Viewer reads the file, interprets the content and displays it in graphical form.

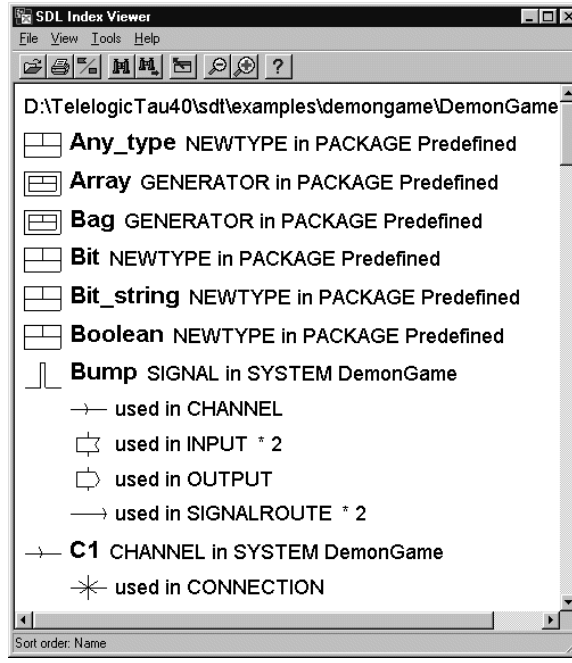


Figure 93: The Index Viewer window

– Some definitions are predefined in the SDL suite environment (the ones containing *PACKAGE Predefined*). They are not of interest for the purpose of this tutorial.

In the next exercise, you will use the Index Viewer to identify all possible situations where a certain signal may be sent or received. We will also look for the definition of the signal.

Finding a Definition

Let us look for the definition of the signal Probe. By default, the definitions in the window are sorted alphabetically, but you do not need to scroll the window manually to find a definition.

There is a *Search* quick button that can be used to find **any** text in the window. However, if you want to search for the name of a definition, there is an even faster way.

1. Simply start keying in the name “Probe”. When you start typing, a search is started on the names that are displayed in bold face. As you type each letter on the keyboard, the status bar at the bottom of the window indicates what text is being searched for, and the window scrolls to show the first matched name. After a few keystrokes, the Index Viewer window shows the signal Probe selected:

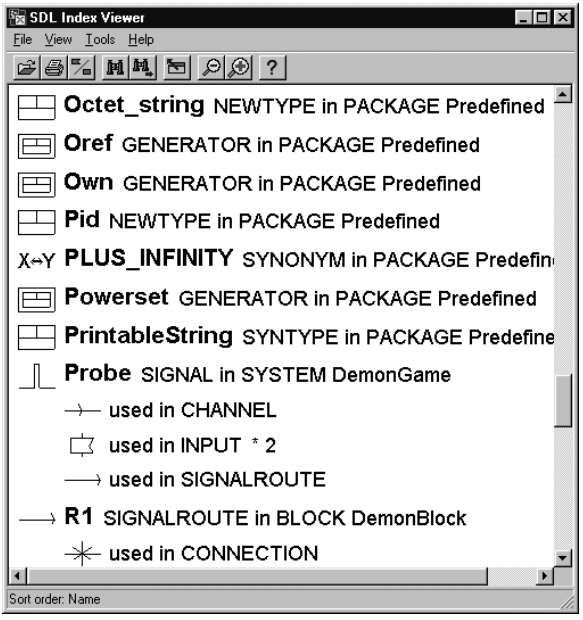


Figure 94: Finding the signal Probe

The selected row shows the icon for a signal, the name and type of the definition (“Probe SIGNAL”) and which diagram the signal is defined in (“SYSTEM DemonGame”).

We now wish to see where the signal is defined.

2. Make sure the Probe icon still is selected.

3. From the *Tools* menu, select the command *Show Definition*.

   – You can also double-click the icon.

   An SDL Editor window is displayed on the diagram for the system DemonGame. The text symbol containing the declaration (i.e. the definition) of the signal is selected.

Finding References

Below the Probe icon in the Index Viewer, all uses (references) of the Probe signal are listed, including the icons for the SDL entities in which the signal has been referred. The information displayed in Figure 94 should be interpreted as:

- The signal is conveyed on one channel,
- The signal may be input in two states,
- The signal is conveyed on one signal route.

To conclude this exercise, you will now locate the places where the signal is input.

1. Select the input icon.
2. The *Tools* menu should now contain the two menu choices *Show Use 1* and *Show Use 2*.
3. Select the menu choice *Show Use 1*– an input symbol is selected in an SDL Editor window, showing the diagram for the process Game.
4. Select the menu choice *Show Use 2* – the second input symbol is selected, also in process Game. These are the two situations where the signal may be input.
5. Double-click the signal route icon. The signal route containing the Probe signal is selected in an SDL Editor window.

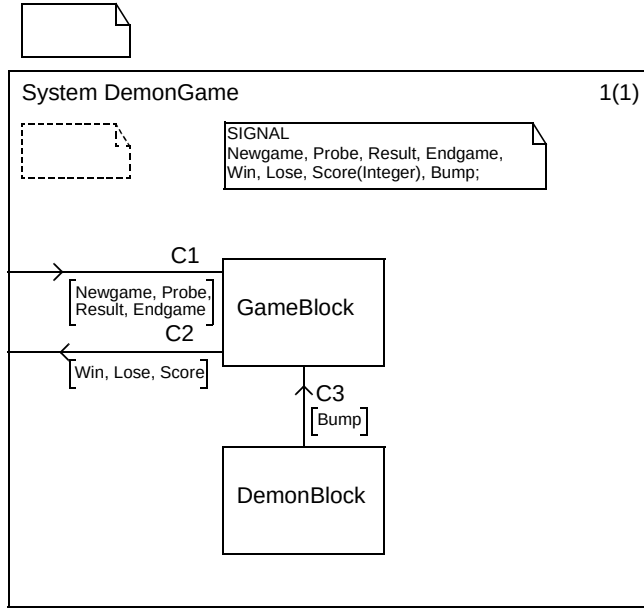
   – If you double-click an icon with more than one reference, the selection in the SDL Editor is automatically changed to the next occurrence. You may try this with the input icon.

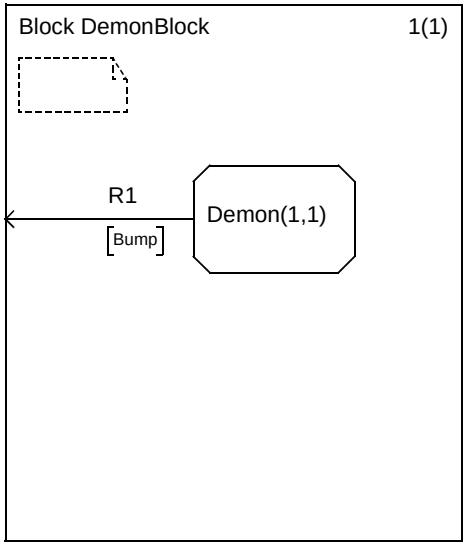
So Far...

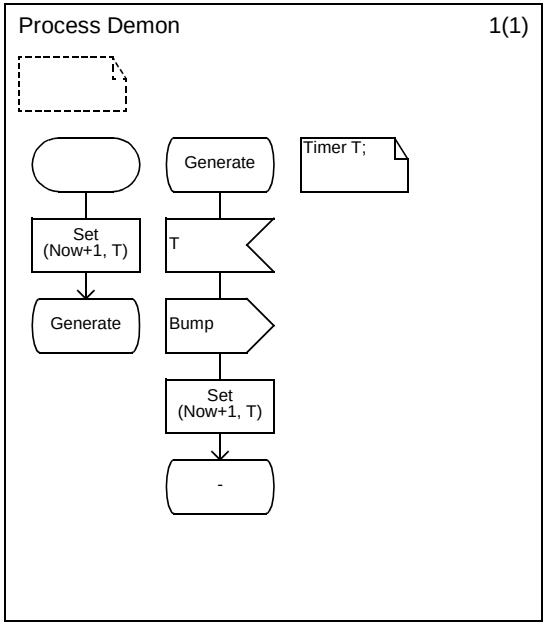
You should now have learned how to use the basic functions in the SDL suite – creating, managing, editing and printing SDL diagrams – and how to create Message Sequence Charts in the MSC Editor. You have also practiced on syntactic and semantic checks on your SDL diagrams. Finally, you have acquainted yourself with the Index Viewer.

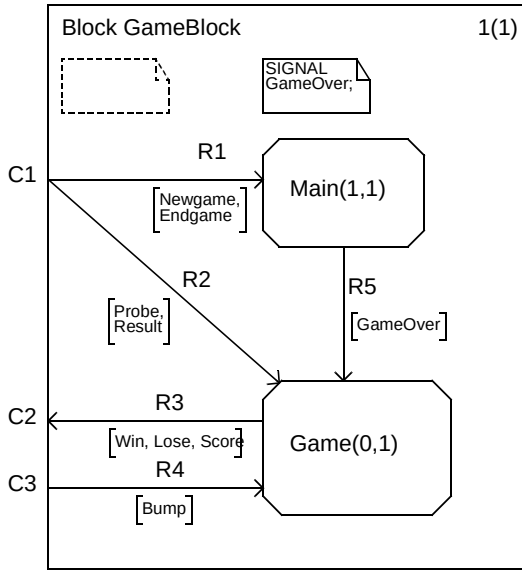
Your next task will be to "animate" your first SDL system by simulating it. A number of exercises are prepared in the next tutorial, starting with "Purpose of This Tutorial" on page 136.

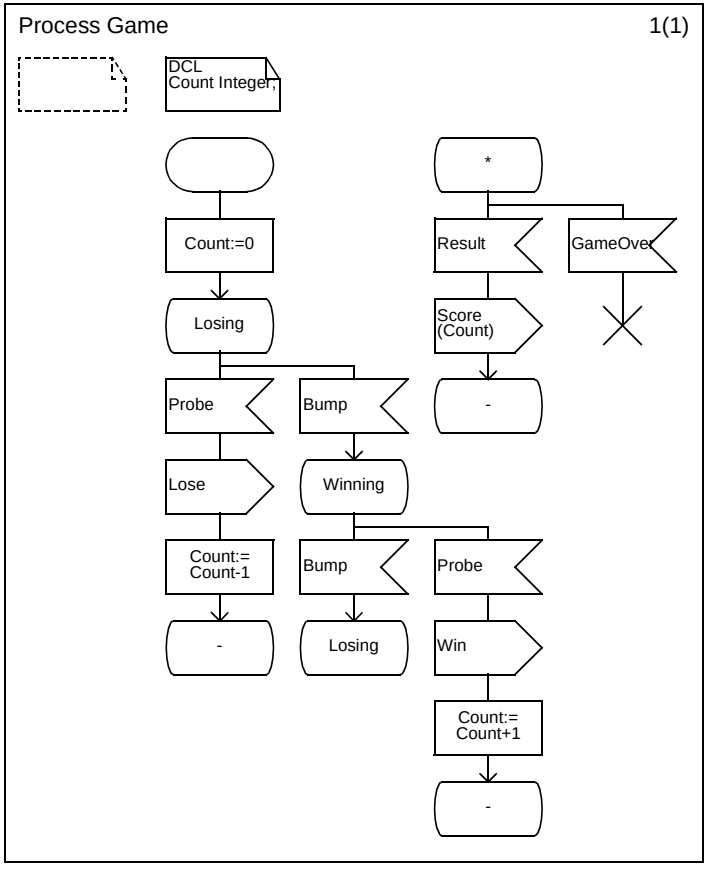
Appendix A: The Definition of the SDL-88 DemonGame

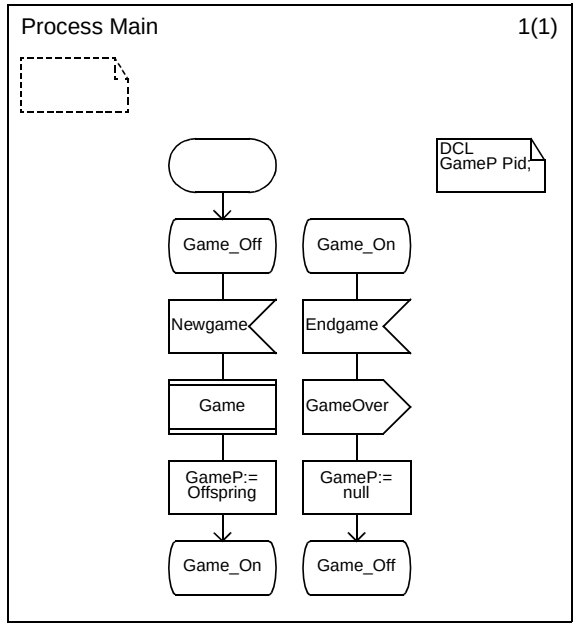



Block DemonBlock
1(1)
C3
R1
Bump
Demon(1,1)


Process Demon
1(1)
Generate
Timer T;
Set
(Now+1, T)
T
Generate
Bump
Set
(Now+1, T)
-


Block GameBlock
1(1)
SIGNAL
GameOver;
R1
C1
Main(1,1)
Newgame,
Endgame
R2
R5
Probe,
Result
GameOver
C2
R3
Win, Lose, Score
Game(0,1)
C3
R4
Bump


Process Game
1(1)
DCL
Count Integer;
Count:=0
Losing
Probe
Bump
Lose
Winning
Count:=
Count-1
Bump
Probe
-
Losing
Win
Count:=
Count+1
-
*
Result
GameOver
Score
(Count)
-


Process Main
1(1)
DCL
GameP Pid;
Game_Off
Game_On
Newgame
Endgame
Game
GameOver
GameP:=
Offspring
GameP:=
null
Game_On
Game_Off

Appendix B: The MSC for the DemonGame