Chapter

9

Tutorial: Using SDL-2000 features

This tutorial describes how to use the SDL-2000 features in Telelogic Tau.

In order for you to fully take advantage of this tutorial, you should be familiar with the SDL suite and the SDL Editor functionality.

This tutorial is independent from the Demon game example presented in the tutorial on the editors and the Analyzer.

You can find additional information on how to use the SDL-2000 features in *"Working with Classes" on page 1876 in chapter 44, Using the SDL Editor, in the User's Manual*.

Purpose of this Tutorial

The purpose of this tutorial is to make you familiar with the SDL-2000 support in Telelogic Tau SDL Suite.

Introduction

Support in the SDL Suite

The Telelogic Tau SDL Suite supports the following SDL-2000 related features:

- Graphical design of data types (using class symbols)
- Textual algorithms
- Case sensitivity
- Operators without parameters
- Operators without return results.

Why SDL-2000?

The useful features in UML are included in SDL to facilitate the transport from UML to SDL.

Graphical Design of Data Types

Telelogic Tau SDL Suite offers the following graphical data type design features:

- *Class Symbols*
- *Association Lines*
- *Aggregation Lines*
- The *Browse & Edit Class* dialog, see *Editing a Diagram*.
- The *Class Information* dialog, see *Viewing a Class*.

Class Symbols

With class symbols you can graphically design data types. The SDL editor handles class symbols much the same way as other SDL symbols.

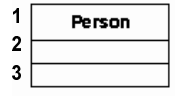

Figure 202 Class symbol

A class symbol has three separate text fields:

- name field (1)
- attribute field (2)
- operator field (3)

The properties of a class symbol can be set in the preference manager in the same way as for other SDL symbols, except for case sensitivity, see "Case Sensitivity" on page 336. The size of the class symbol is adjusted to the size of the text, and you cannot change them manually. Class symbols cannot be collapsed or expanded.

Association and Aggregation Lines

There are two types of lines:

- Association lines
- Aggregation lines

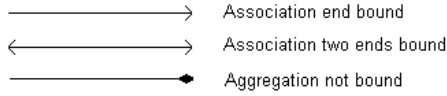

Figure 203

Association Lines

An association links two types using UML notation. The types are:

- Block types
- Process types
- Data types
- Interfaces

Association lines can be both redirected and bidirected. Single association lines (unidirectional associations) have a name and a role name. Double association lines (bidirectional associations) have a name and two role names.

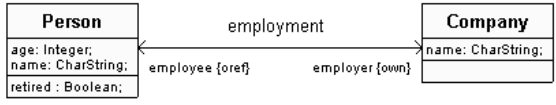

Figure 204: Example of an association line

Aggregation Lines

Aggregations use the same format as associations.

Aggregations are used to indicate that a class is a subset of another class or a part-of relationship, e.g. a steering wheel is a subset of or a part of a car. Aggregations can only be single directed.

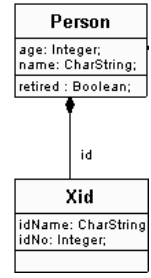

Figure 205: Example of an aggregation line.

Creating an SDL Structure

In this section you will create a small example and perform a number of actions to learn what possibilities and limitations there are in the SDL-2000 support in the SDL Suite.

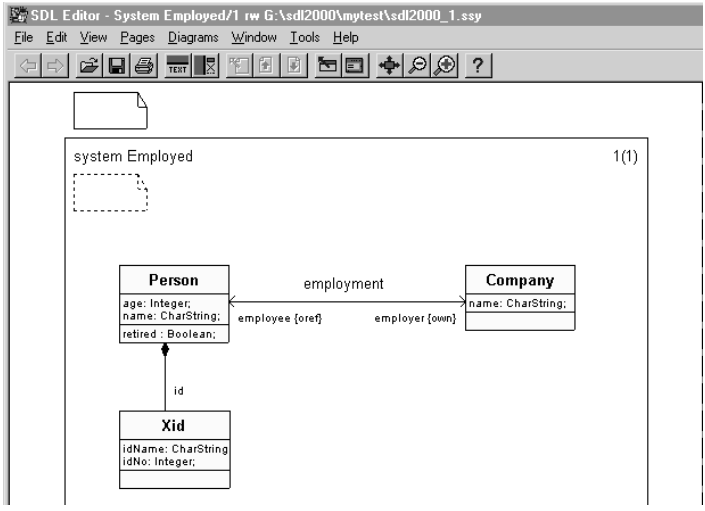

Figure 206:SDL diagram with class symbols

Working with Class Symbols

1. Start the SDL Editor.
2. Place a class symbol in the diagram.

   You place symbols the same way as other SDL symbols, see "Placing Block Reference Symbols" on page 59 in chapter 3, *Tutorial: The Editors and the Analyzer*. You can move symbols but not resize them.
3. Name the class symbol **`Person`**.

   All class symbols with the same class name are treated as a single, merged class. See "Limitations" on page 336 for more information on naming issues.
4. Fill out the attribute and operator fields.

5. Place two more class symbols in the diagram and name them **`Company`** and **`Xid`**.

6. Fill out the attribute and operator fields.

Working with Lines

1. Click the class symbol `Person`. Two handles appears.

2. Select the square association handle and drag it to the class symbol `Company`. The editor draws a line while you are dragging. You drag lines the same way as lines in other SDL diagrams, see "Drawing Channels between Blocks" on page 61 in chapter 3, *Tutorial: The Editors and the Analyzer*.

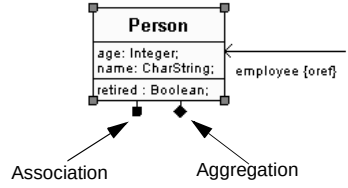

Figure 207

3. Click the mouse button when you have reached the class symbol `Company`. The line is connected at both ends.

4. Name the association line and the role.

The name of an association is often a verb phrase and the role often has a noun as a name.

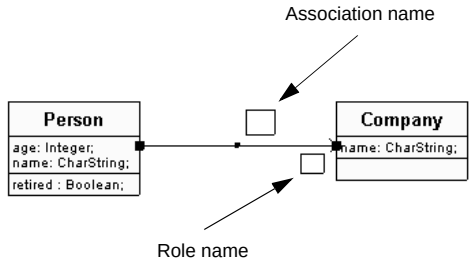

Figure 208

5. Bidirect the line by selecting *Bidirect* from the *Edit* menu.
6. Give the second role a name.
7. Select the diamond shaped aggregation handle and drag it to the class symbol `Xid`.
8. Name the aggregation line and the role.

Moving Symbols and Lines

You move symbols and lines the same way as in other SDL diagrams, see "Moving and Resizing Symbols" on page 61 in chapter 3, *Tutorial: The Editors and the Analyzer*.

Saving the Diagram

1. Save the diagram by the menu choice *Save*, by clicking the *Save* button or by **`<Ctrl> s`**.

   The file is saved with the extension `.ssy`.

Editing a Diagram

The *Browse & Edit Class* dialog lets you view and edit the complete definition of a class to ensure consistency between the attributes and operators.

The dialog consists of two parts. The above part is where you browse classes and all occurrences of each class. In the below part you can edit the name, attributes and operators of a class.

All symbols that belong to the class you select in the *Browse & Edit Class* dialog will be affected by the changes you make in the dialog. If you want to edit the content in only one class symbol, you have to select the single symbol and edit the content in that symbol. For detailed information on how to edit class information, see"Browse & Edit Class Dialog" on page 1877 in chapter 44, *Using the SDL Editor*.

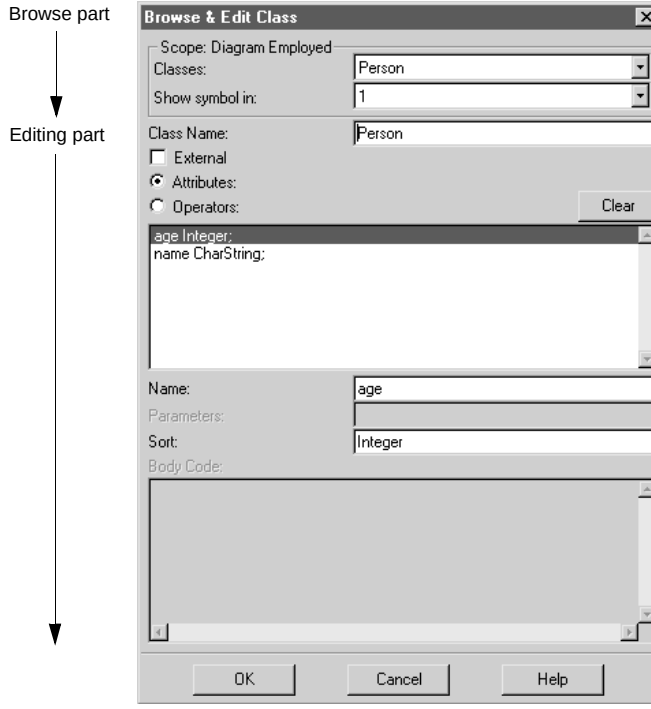

Figure 209: The Browse & Edit Class dialog

Case Sensitivity

Case sensitivity applies to class symbols and it cannot be turned off in the Preference Manager, though case sensitivity for other symbols in the SDL editor can be switched off.

Class sensitivity applies to both the name field and the attribute field as well as the operator field.

If two classes have the same name but different cases (for example `Person` and `person`), they will be treated as two separate classes in the *Browse & Edit Class* dialog and will **not** be merged into a single class when generating code.

For a detailed description, see "Set-Case-Sensitive" on page 2416 in chapter 55, *The SDL Analyzer*.

Limitations

You can edit a class in this dialog when you have selected a single class symbol, but the dialog is not available if you select more than one class symbol.

If the class name contains incorrect syntax the dialog will not be displayed.

You cannot add syntax errors to your classes as the values you enter are syntactically checked.

Incorrect text in a symbol is not displayed in the dialog.

Edit the Diagram

1. Double-click the class `Xid` or select the class `Xid` and then select *Class.*. on the *Edit* menu.

   The *Browse & Edit Class* dialog is displayed with the class name and its attributes or operators filled in.

   If there are more than one class with the same name, attributes or operators for all classes with the same name will be shown in the dialog.

2. Select class `Person` from the *Classes* drop down menu. The content of the attributes and operators fields changes to that of the class `Person`.

3. Change the name of the class `Person` to **`Employed`** in the *Class Name* field and click *OK*.

   The class name has changed to `Employed` in the diagram.

4. Open the *Browse & Edit Class* dialog again and select either of the radio buttons *Attributes* or *Operators* to edit or view the attributes or operators of the class `Employed`.

   The list of the attributes or operators is displayed in the field below the radio buttons.

5. You can now edit the information in the *Name, Parameters, Sort and Body Code* fields.

   The *Parameters and Body Code* fields are for operators only.

6. Click *OK*.

   The dialog is closed and all appropriate class symbols in the current SDL diagram are updated.

Viewing a Class

The Class Information dialog lets you view the complete textual (PR) definition of a class in read-only format. If you want to edit the diagram, see "Editing a Diagram" on page 335.

For detailed information on how to view class information, see "Class Information" on page 1876 in chapter 44, *Using the SDL Editor*.

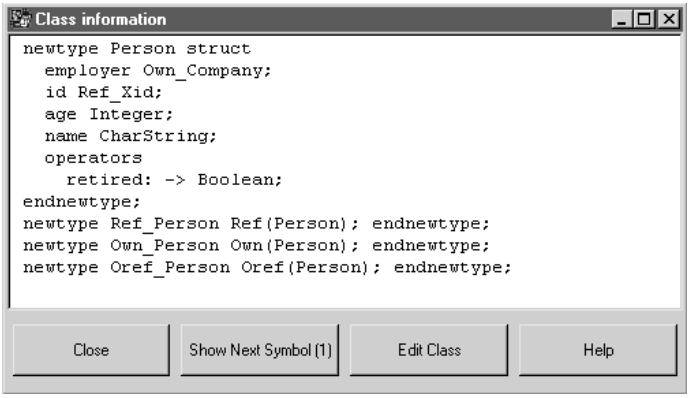

Figure 210: The Class Information dialog

Limitations

The dialog is not available if you select more than one class symbol.

View the Definition

1. Rename two or more classes with identical names (for testing of the Class information dialog).
2. Select one of the classes and then select *Class Information* on the *Window* menu.

   The *Class Information* dialog, with the PR code for all classes with the same name, is displayed.
3. Select the *Show Next Symbol* button.

   The next symbol, which is described by the line on which the cursor is placed, is displayed.

The number in brackets shows the number of symbols that are described by this line in the current diagram. You can browse through all symbols, one by one, with the *Show Next Symbol* button.

4. If you decide that you want to edit the class, select the *Edit Class* button and the *Browse & Edit Class* dialog for that class is displayed.

Textual Algorithms

Textual Algorithms

Telelogic Tau SDL Suite supports textual descriptions of algorithms. Algorithms can be expressed within a Task symbol, e.g. if-then-else, loops and decisions. For a detailed description of this feature, see "Algorithms in SDL" on page 137 in chapter 3, *Using SDL Extensions*.

Operators without Parameters and Operators without Return Results

Telelogic Tau SDL Suite supports operators with no parameters or no return value. For example, an alternative to an assignment statement could be that an operator application statement invokes a non value returning operator.

For a detailed description of operators and parameters, see "Operators" on page 82 in chapter 2, *Data Types*.

Limitations

The following limitations are worth noticing:

- Inheritance lines are not supported.
- Comments added to a class symbol will not appear in the generated PR code.
- CIF code generation for class symbols is not supported.
- If the same association has been drawn in several places, all instances have to be edited to change the association.
- Navigating from a class symbol is possible only by using *Tools>Navigate*, since double-clicking a class symbol will open the *Browse & Edit Class* dialog.

An Example of Using Class Symbols

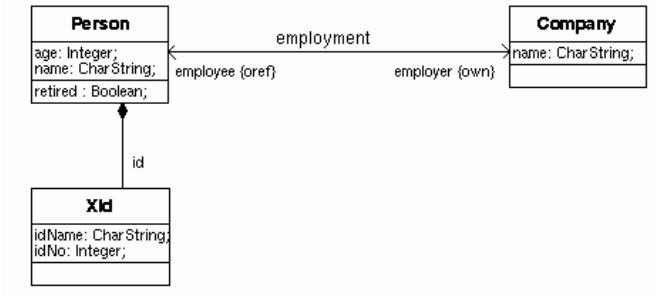

Figure 211 Graphical representation of a data structure

The operator definition must be added in the *Body Code* part of the *Browse & Edit Class* dialog. It might look like this:

```
operator retired returns Boolean { /*start of user defined
body code*/
return age > 64;
/* End of user defined body code */}
```

The above graph will then generate the following PR code.

```
newtype Person struct
  employer Own_Company;
  a_Company Ref_Company;
  id Ref_XId;
```

```
  age Integer;
  name CharString;
  operators
    retired: -> Boolean;
    operator retired returns Boolean {/*start of user defined
body code*/
return age > 64
/* End of user defined body code */}
endnewtype;
newtype Ref_Person Ref(Person); endnewtype;
newtype Own_Person Own(Person); endnewtype;
newtype Oref_Person Oref(Person); endnewtype;
newtype XId struct
  idName CharString;
  idNo Integer;
endnewtype;
newtype Ref_XId ref(XId); endnewtype;
newtype Own_XId own(XId); endnewtype;
newtype Oref_XId oref(XId); endnewtype;
newtype Company struct
  employee Oref_Person;
  a_Person Ref_Person;
  name CharString;
endnewtype;
newtype Ref_Company Ref(Company); endnewtype;
newtype Own_Company Own(Company); endnewtype;
newtype Oref_Company Oref(Company); endnewtype;
```

Chapter 9 Tutorial: Using SDL-2000 features