L Plugin for Eclipse (Draft)

Evgenii Balai

March 6, 2016

Contents

1 Installing the Plug-in

2 Using the Plug-in

2.1 Creating a New Project

Every L program must be a part of a project. To create a project, go to the menu `File` → `New` → `Project` as shown in Figure 1.

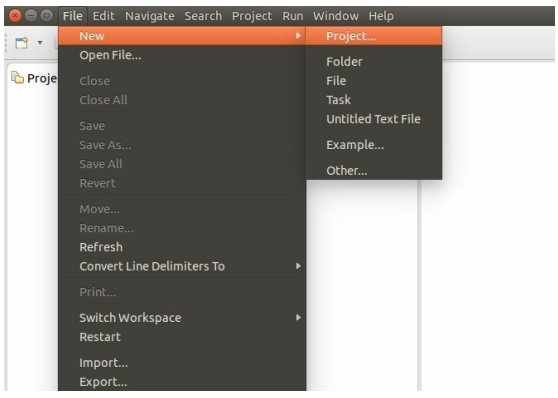

Figure 1: Creating a New Project (Menu)

In the opened window choose `General` → `Project` and click `Next` as shown in Figure 2.

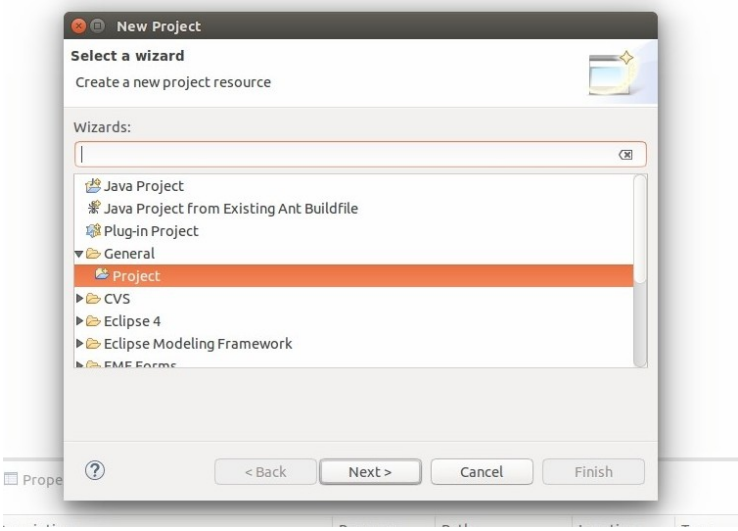

Figure 2: Choose the type of the project

In the next window shown on the screen type the name of the project (in our case the name is `safety_ch5`) and click `Finish`. See Figure 3.

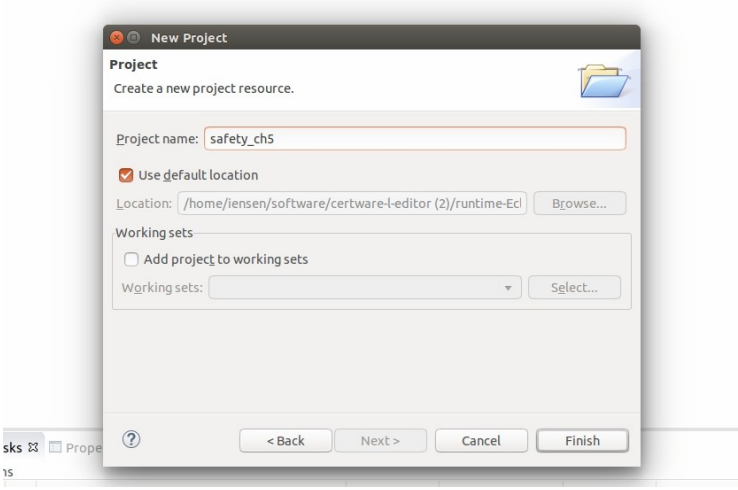

Figure 3: Final step of creating the project

You should see a new project is created and shown in the project explorer (a window in the top left corner), as in Figure 4.

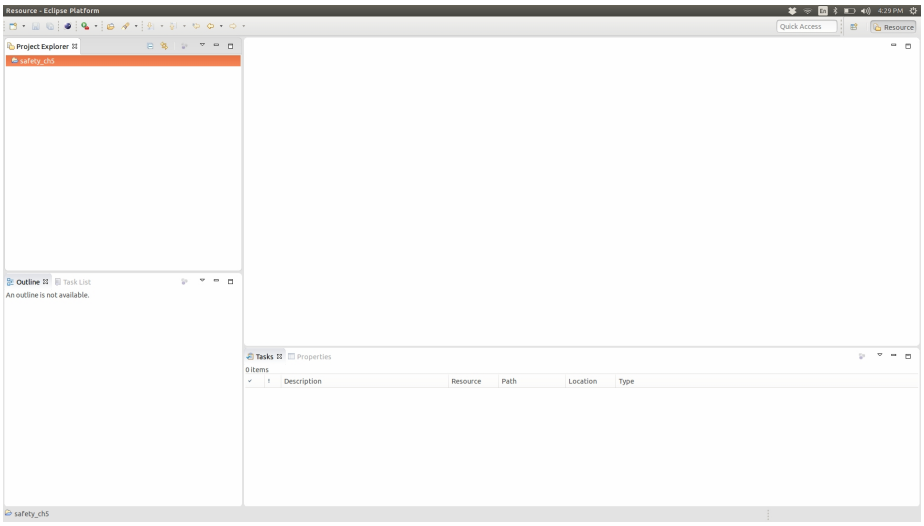

Figure 4: The newly created project is shown in the project explorer

2.2 Adding a New L Source File

To create a file where an L program can be stored, right click the project name in the project explorer and choose the menu `New` → `File` as shown in Figure 5.

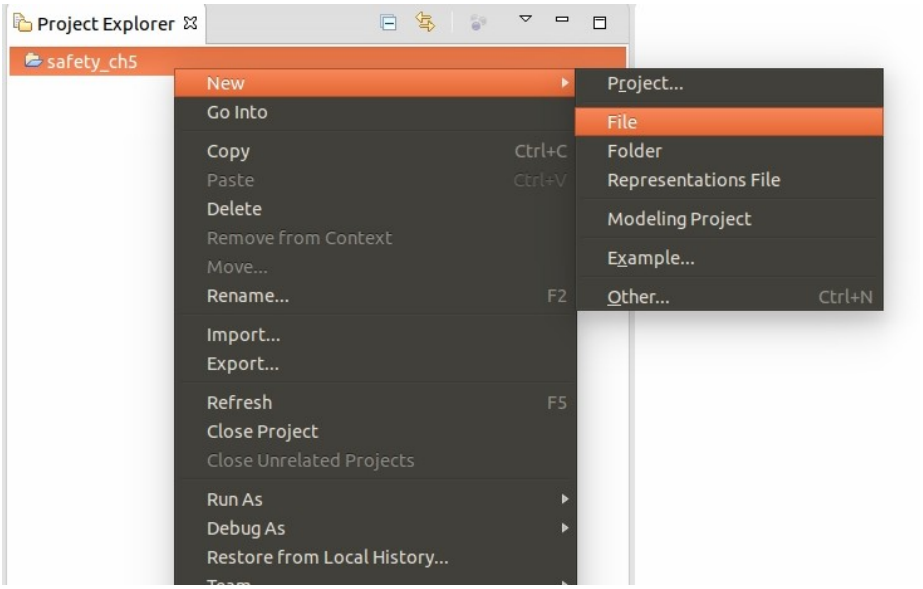

Figure 5: Creating a New L source file (Menu)

In the newly opened window type the name of the new file. **The name must have `.L` or `.l` suffix**. After the name is chosen, click `Finish`. See Figure 6.
If this is the first L source file added to the project, you should see a dialog

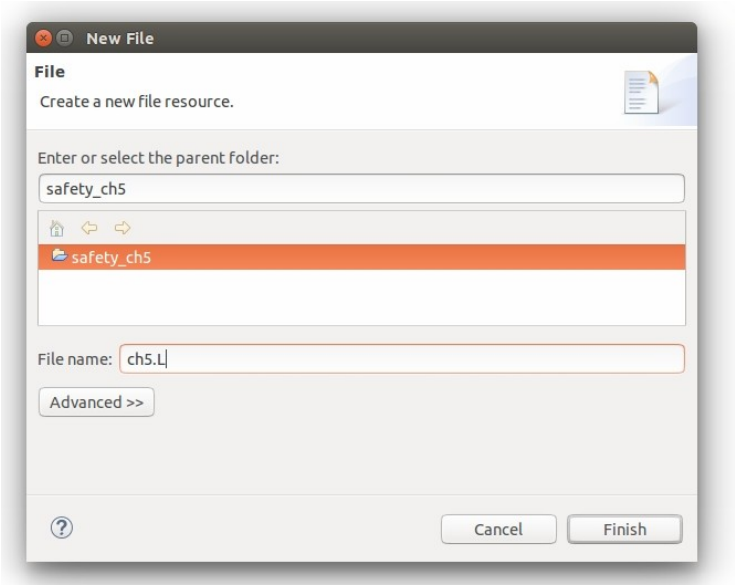

Figure 6: Giving the new file a name

shown in Figure 7.

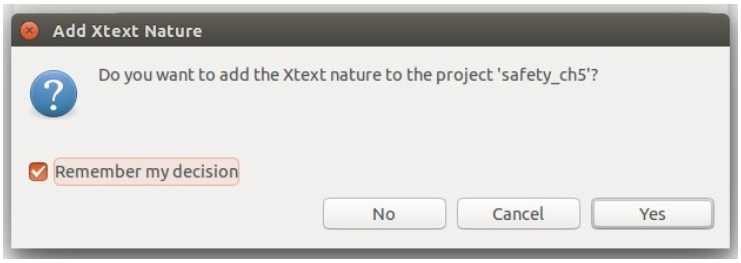

Figure 7: Adding Xtext nature to the project

Choose the option `Remember my decision` and click `Yes`. After this step you should see the new file added to the project (and shown in the project explorer) and the L editor opened, as shown in Figure 8.

2.3 Editing the File

To edit the newly created file, switch the cursor to the largest text area of the workbench and start typing. See Figure 8. The outline window in the bottom left corner shows the structure of the program. If the program in the editor window has syntax errors, they will be shown in the bottom horizontal text area called `Problems`, as shown in Figure 9.

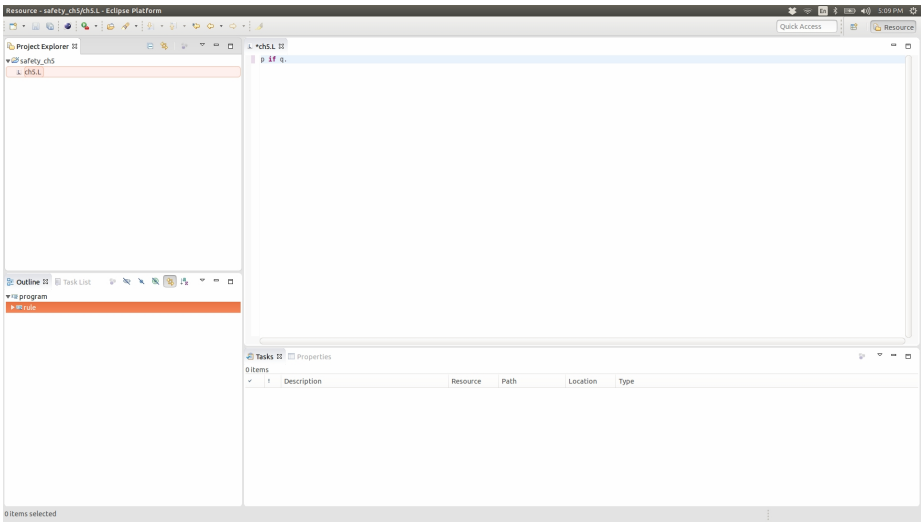

Figure 8: Editing the program

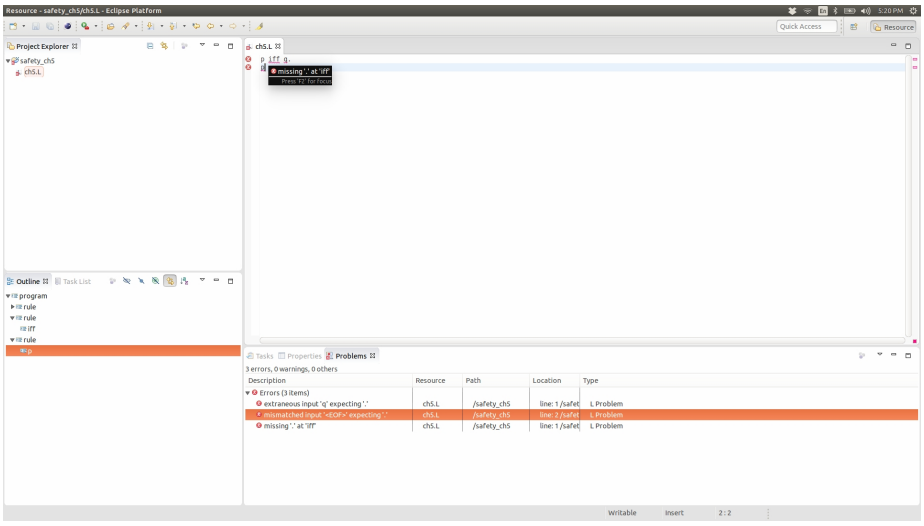

Figure 9: Erroneous program

The corresponding parts of the program will be underlined with red color in the editor. (If the window `Problems` is not shown, go to the menu `Window` → `Show View` and choose `Problems`.)

2.4 Computing the Models of a Program

To compute the models of the program stored in the L source file opened by the editor, save the program (use `File` → `Save` menu or `Ctrl + S` shortcut) and right click the editor window and click the menu item `Compute Models`, as shown in Figure 10.

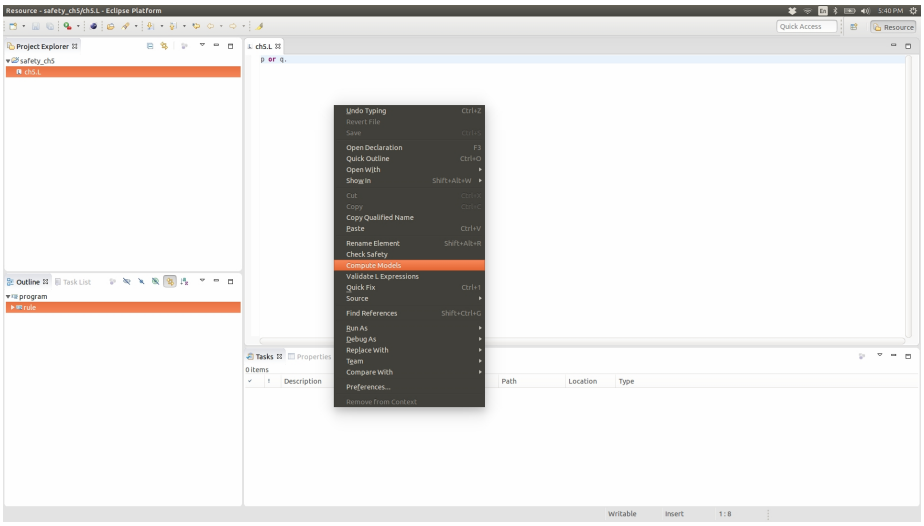

Figure 10: Computing the models of a program (Menu)

The program shown in the editor in Figure 10 has two models, p and q. After `Compute Models` is clicked, the models will be shown in a separate editor, one per line, as shown in Figure 11.

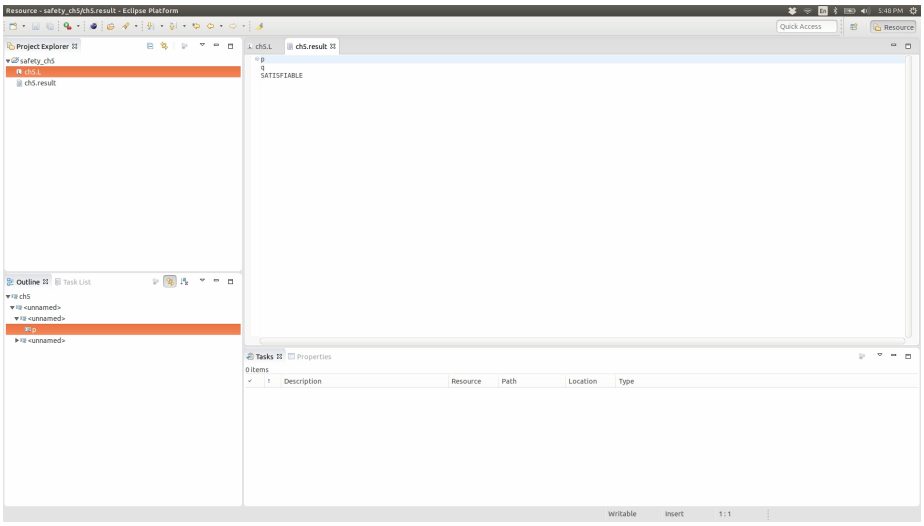

Figure 11: Computing the models of a program (Models)

2.5 Checking if Safety Requirements Defined by the Program are Met

To check if the safety requirements defined the by the program are met, load the program into the editor, switch the cursor to it and right click to show the context menu from Figure 10. Click the menu item called `Check Safety`.

- if every model of the program includes the atom $safety\_reqts\_fully\_realised$, we say that *safety requirements defined by the program are met*;
- otherwise, *safety requirements defined by the program are not met.*

For instance, the safety requirements defined by the program from https://github.com/iensen/LtoASPtranslator/blob/master/src/examples/ch5.l which is based on chapter 5 of the EUR RVSM Pre-Implementation Safety Case [1] are met. Therefore, if the program is entered into the editor and the menu item `Check Safety` is called, the message shown in Figure 12 indicating that the safety requirements are met can be observed.

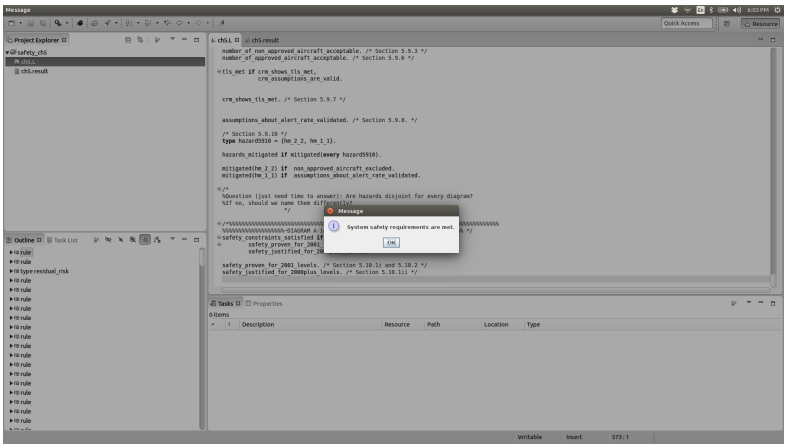

Figure 12: Checking safety - 1

However, if, for example, the fact `mitigated(level_busts)` (indicating that level busts threat was mitigated) is removed, the safety requirements defined by the program are no longer met. The corresponding message is shown in Figure 13.

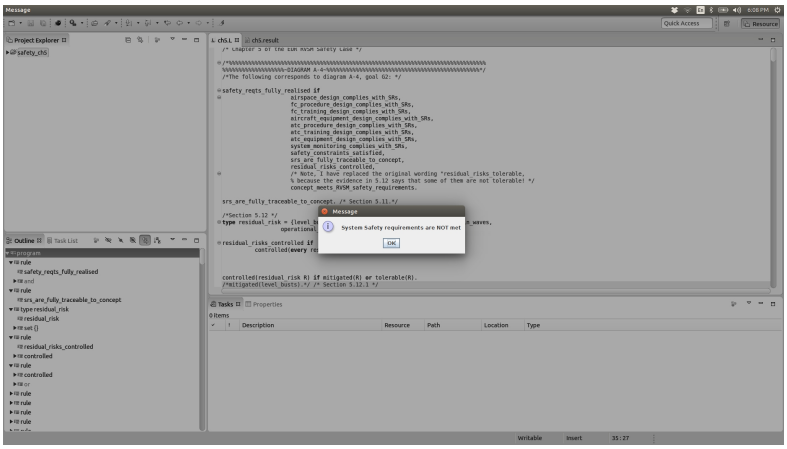

Figure 13: Checking safety - 2

References

[1] European Air Traffic Management Programme, "The EUR RVSM Pre-Implementation Safety Case," RVSM 691, Version 2.0, 14 August 2001.