

# User Manual: SLC Tool

## 1 Download and Setup

The following steps need to be followed diligently for the correct setup of the tool in your machine.

STEP 1: **Pre- requisite:** User must have *Java SDK* and *Eclipse* installed in their machine. Within Eclipse the following plug-in needs to be installed *Acceleo* and *Complete XText SDK*. The “*Eclipse packages required.mp4*” video file in the github repository of the SLC tool shows a simple way to check if the necessary packages are already installed on your machine.

STEP 2: **Download:** User needs to download the following files from the github repository <sup>1</sup>:

- *AcceleoMTLRunner.rar*
- *CompliantFSM.rar*
- *org.xtext.example.mandsl.rar*

STEP 3: **Extract:** The downloaded RAR files need to be extracted. The *CompliantFSM* folder contains supporting files for the tool. It should be saved in the path *C:\CompliantFSM* specifically.

STEP 4: **Import:** User needs to import the *AcceleoMTLRunner* and *org.xtext.example.mandsl* projects into the Eclipse workspace. The screenshots are shown in Figure 1.

- Open an Eclipse workspace.
- Click on *File* → *Import* (Figure 1(a)).
- In the Import Wizard, select *General* → *Projects from Folder or Archive* (Figure 1(b)).
- Select the directory of the extracted folders as the Import Source (Figure 1(c)).
- Select folders *AcceleoMTLRunner* and *org.xtext.example.mandsl* and click *Finish* (Figure 1(c)).

---

<sup>1</sup><https://github.com/cFSM/GRL2FSM>

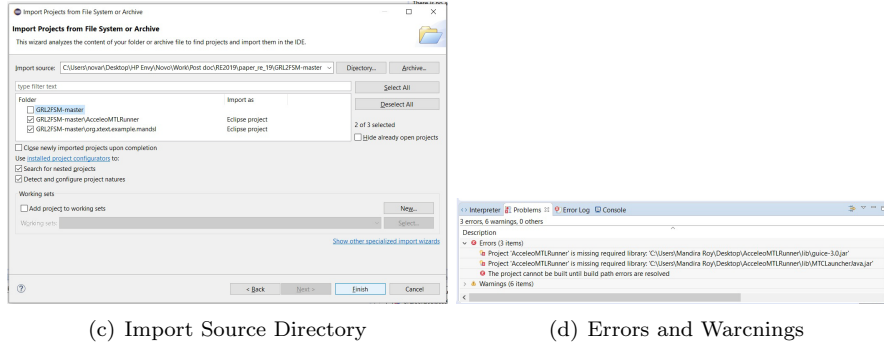
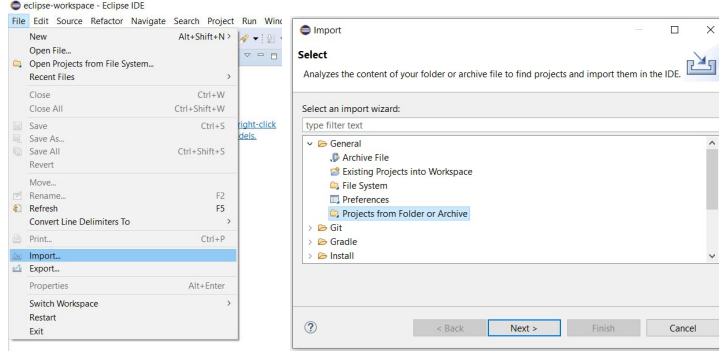
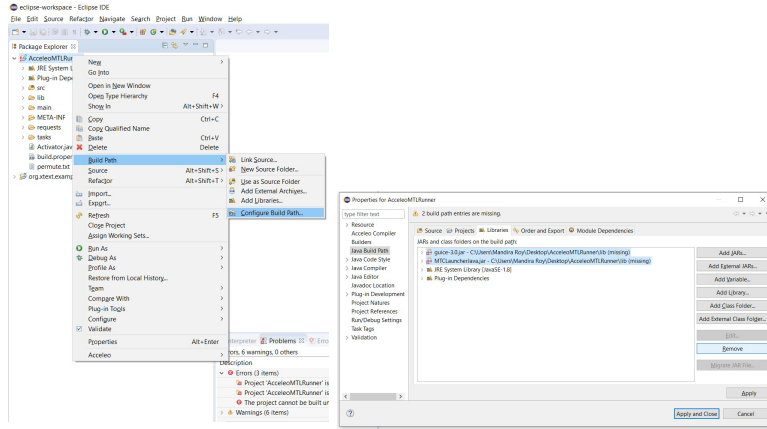


Figure 1: Importing the Eclipse projects for the SLC Tool.

The two projects get imported with *Warnings* and *Errors* as shown in Figure 1(d). Errors are associated only with the *AcceleoMTRLRunner* project. The two main errors are due to incorrect referencing of two external JAR files. The warnings can be ignored.

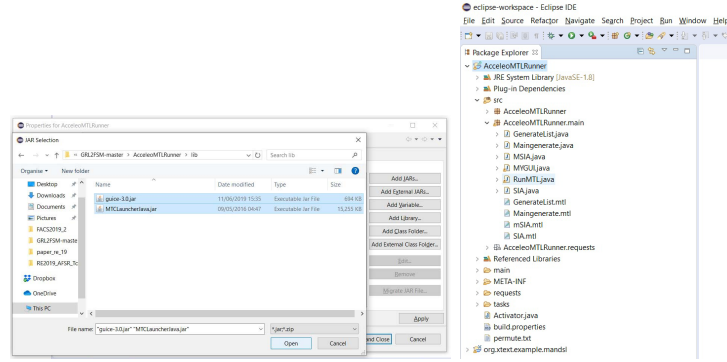
**STEP 5: Resolve reference errors:** User has to resolve the two external JAR reference errors in the *AcceleoMTRLRunner* project. Warnings should be ignored. The following steps remove these errors. The screenshots are shown in Figure 2.

- Right-click on the project root *AcceleoMTRLRunner* and click *Build Path* → *ConfigureBuildPath..* (Figure 2(a)).
- In the *Libraries* tab, select the *guice-3.0.jar* and *MTCLauncher-Java.jar* files and click *Remove* (Figure 2(b)).
- Click *Add External JARs*. In the JAR Selection Window, go to the *lib* directory under the *AcceleoMTRLRunner* project. Select the 2 JAR files - *guice-3.0.jar* and *MTCLauncherJava.jar* and click *Open* (Figure 2(c)).



(a) Configure Build Path

(b) Remove old JAR references



(c) Select JARs from lib

(d) Apply New References

Figure 2: Resolving errors in the *AcceleoMTLRunner* project.

- Click on *Apply and Close* button (Figure 2(d)).

These steps remove the errors but there are a lot of *Warnings* that still exist in the project. Users of the SLC Tool can simply ignore them.

## 2 Opening the SLC Tool Interface

Once Setup is complete (as shown in the previous section) , users can start using the SLC Tool. To open the tool, users must do the following:

- Go to the *RunMTL.java* file in the path *AcceleoMTLRunner* → *src* → *AcceleoMTLRunner.main*.
- Right click on *RunMTL.java* file, click on *Run As* and execute as a *Java application* (refer Figure 3(a)). SLC Tool first-level interface appears on

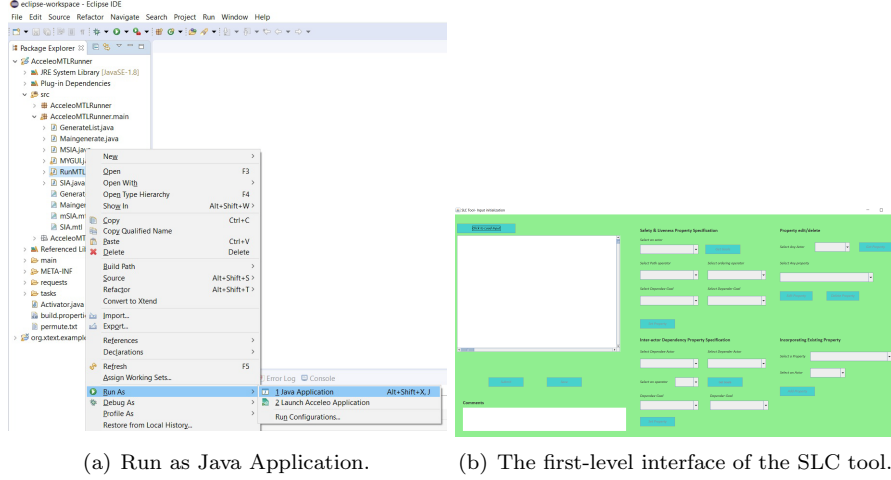


Figure 3: Running the SLC Tool.

the screen as shown in Figure 3(b).

A complete video tutorial (*DownloadAndSetup.mp4*) of how to download, setup and run the tool on a user machine is available in the github repository.

### 3 Using the SLC Tool

**STEP 1: Loading Input.** Select an input file by clicking on **Click to Load Input** button (refer Figure 4(a)). After loading input goal model all other buttons get activated.

**STEP 2: Setting CTL Property.** Refer Figure 4(b).

- Select an actor from **Select an actor** drop down menu and click on **Get Goals** button. The goals of the selected actor appear in the drop down menus – **Select Depende** goal and **Select Depender goal**.
- Select a path operator (**Select path operator**), a temporal ordering operator (**Select ordering operator**) and dependee (**Select Depende** goal) and depender goals (**Select Depender goal**).
- On specifying all the required parts of a CTL property, the comment section displays its textual meaning (refer Figure 4(c)).
- Click on **Set Property** button to save it in a temporary file.
- Click on **Save** button to see the updated goal model appended with the CTL properties specified (refer Figure 4(d)).

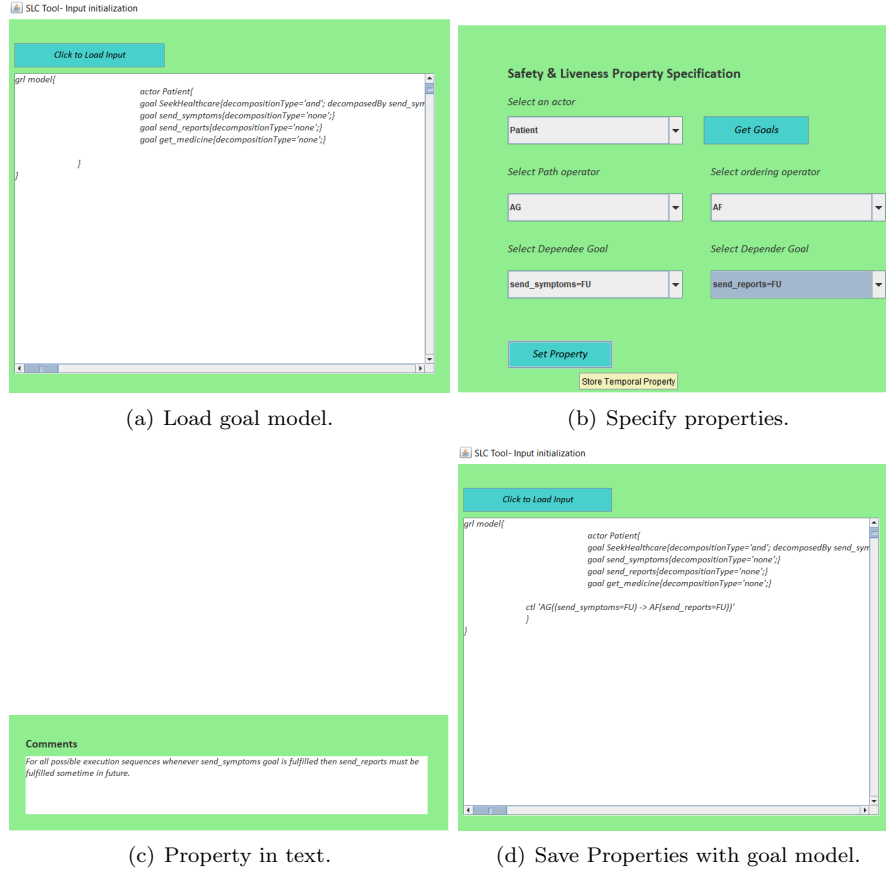
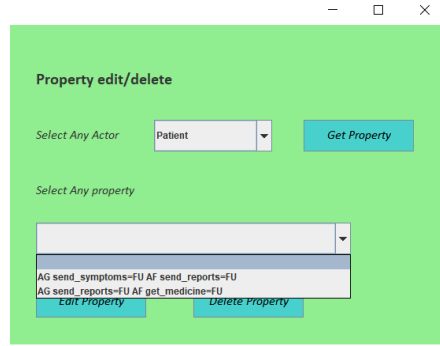


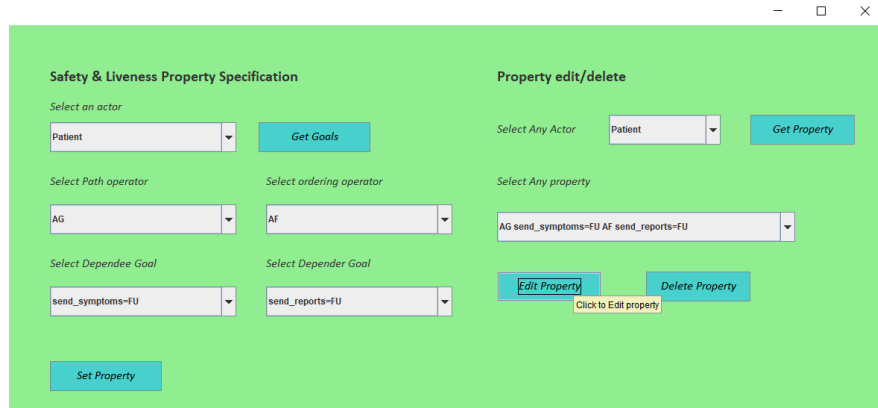
Figure 4: The first level SLC tool interface.

**STEP 3: Modifying Property.** (*Optional*). This panel may be used by the user to edit an incorrect CTL property specification or to delete it altogether. Refer to Figure 5(a).

- In **Property edit/delete** panel select any actor using **Select Any Actor** drop down menu and click on **Get Property** button.
- The saved CTL properties of the selected actor appear in the drop-down menu (**Select Any property**).
- Select any property and click on **edit** or **delete** button (Refer to Figure 5(a)).
- On clicking **edit** button the selected property gets loaded into the property specifying boxes on the immediate left panel (refer Figure 5(b)).
- Modify the property and again click on **Set Property** button.



(a) Select property to Edit or Delete.



(b) Edit and re-specify property.

Figure 5: Modifying Properties.

**STEP 4: Using Existing Property.** (*Optional*). This panel may be used by the user to select a CTL property from the existing list of properties that were already defined (for any other goal model). Refer to Figure 6.

- In **Incorporating Existing Property** panel select any property using **Select a Property** drop down menu.
- Select any actor using **Select an Actor** drop down menu.
- Click on **Add Property** button to save the property for the selected actor.
- Click on **Save** button to see the updated goal model.

**STEP 5: Setting Inter-Actor property.** Refer to Figure 7.

- Select **dependee** (**Select Depende Actor**) and **depender** (**Select Depender Actor**) actors.

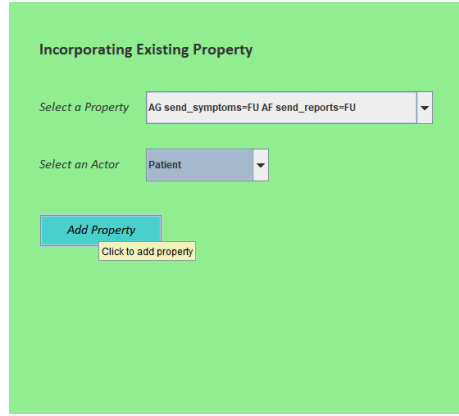


Figure 6: Selecting Existing Property.

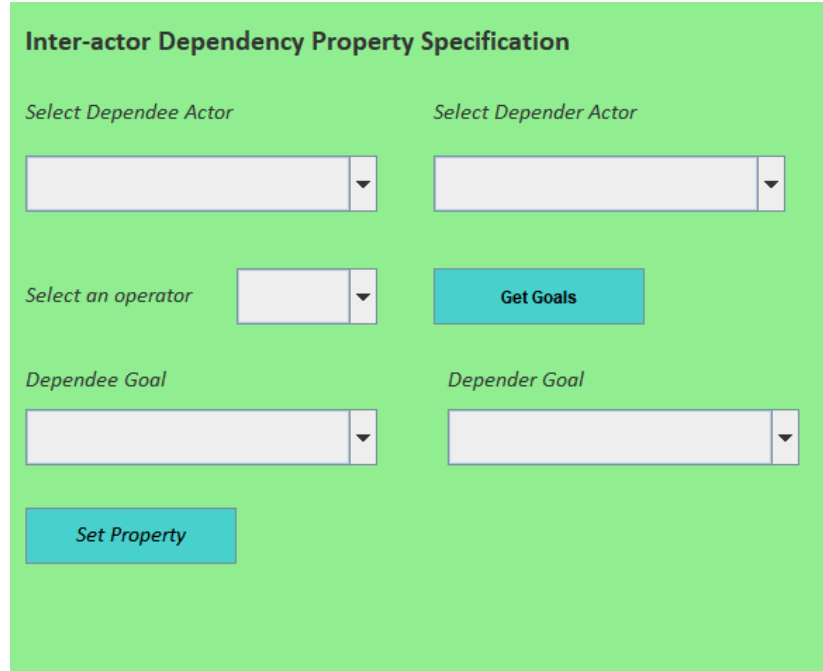
- Click on **Get Goals** button.
- Select a temporal operator from the **Select an operator** drop-down menu.
- Select the goals (**Dependee goal** and **Depender Goal**) from the respective menus.
- Click on **Set Property** button below.

**STEP 6: Loading the level-2 interface.** Click on the **Submit** button to load the second level interface of the SLC tool. The SLC Tool- FSM Generation interface gets loaded. This interface is yellow in color. The **INPUT PANEL** displays the input goal model appended with the CTL properties specified in the previous interface. Figure 8(a) shows the second-level interface when it is loaded. All buttons in the frame remain de-activated except the **Generate FSM** button.

**STEP 7: Generate FSMs.** Refer to Figure 8(b).

- Click on **Generate FSM** button to initiate FSM generation using both SIA and M-SIA algorithms. After FSM generation is complete all other buttons in the button panel gets activated.
- The different metrics that provide a comparison between the two algorithms appear in the left-bottom panel.
- Click on **Save Metrics** button to save the metrics in *Output\_metrics.txt* file in the Metrics folder created in *CompliantFSM* folder in C:/ drive.
- Click on **View File** button to open the *Output\_metrics.txt* file.

**STEP 8: Using Button Panel.** Refer to Figure 8(b).



The form is titled "Inter-actor Dependency Property Specification" and is set against a light green background. It contains several input fields and buttons:

- Select Dependeo Actor:** A dropdown menu.
- Select Dependee Actor:** A dropdown menu.
- Select an operator:** A dropdown menu.
- Dependeo Goal:** A dropdown menu.
- Dependee Goal:** A dropdown menu.
- Get Goals:** A blue button.
- Set Property:** A blue button.

Figure 7: Setting inter-actor properties.

- Click on **Check FSM(SIA)** or **Check NuSMV Input(SIA)** button to check the finite state machine and NuSMV input generated using SIA.
- Click on **Check FSM(M-SIA)** or **Check NuSMV Input(M-SIA)** button to check the finite state machine and NuSMV input generated using M-SIA.
- Click on **Validate State Transition Model(SIA)** to perform model checking of the FSM generated using SIA with the given CTL properties.
- Click on **Validate State Transition Model(M-SIA)** to verify the compliance of the FSM generated using M-SIA with the given CTL properties.

A detailed video tutorial of how to use the tool has been provided in the github repository with the name “*Using SLC-Tool.mp4*”.



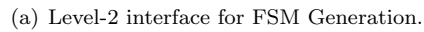


Figure 8: The second level FSM-Generation interface of the SLC Tool.