

User Manual: SLC Tool

Novarun Deb, Mandira Roy, Agostino Cortesi, Nabendu Chaki

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 ¹:

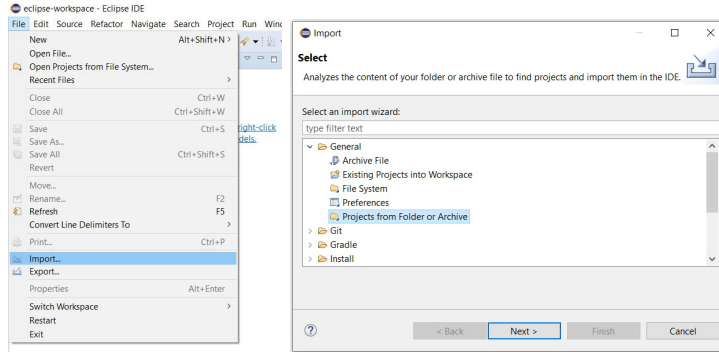
- *AcceleoMTLRunner.rar*
- *CompliantFSM.rar*
- *org.xtext.example.mandsl.rar*

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.

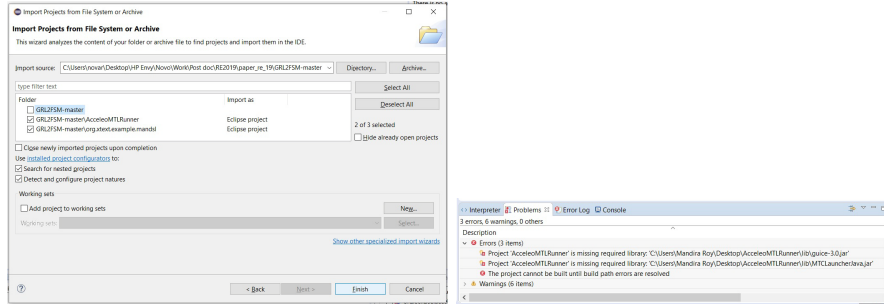
- Open an Eclipse workspace.
- Click on *File* → *Import* (Figure 1(a)).
- In the Import Wizard, select *General* → *Projects from Folder or Archive* (Figure 1(b)).
- Select the directory of the extracted folders as the Import Source (Figure 1(c)).
- Select folders *AcceleoMTLRunner* and *org.xtext.example.mandsl*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

¹<https://github.com/cFSM/GRL2FSM>



(a) To Import Projects

(b) Import Wizard



(c) Import Source Directory

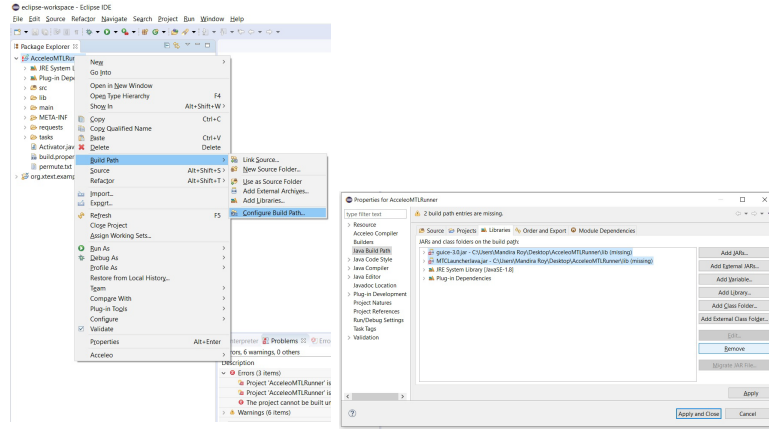
(d) Errors and Warnings

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 *AcceleoMTLRunner* project. The two main errors are due to incorrect referencing of two external JAR files. The warnings can be ignored.

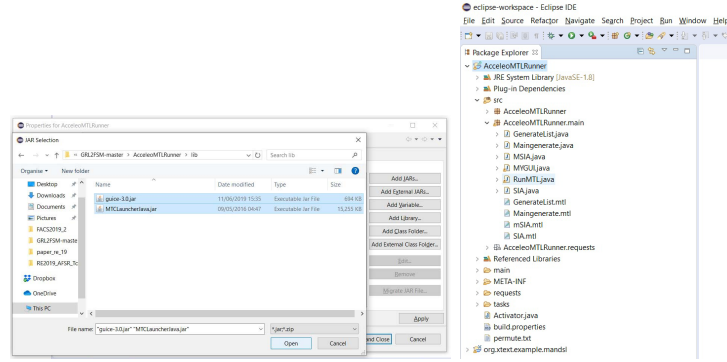
STEP 5: Resolve reference errors: User has to resolve the two external JAR reference errors in the *AcceleoMTLRunner* project. Warnings should be ignored. The following steps remove these errors. The screenshots are shown in Figure 2.

- Right-click on the project root *AcceleoMTLRunner* 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 *AcceleoMTLRunner* 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.

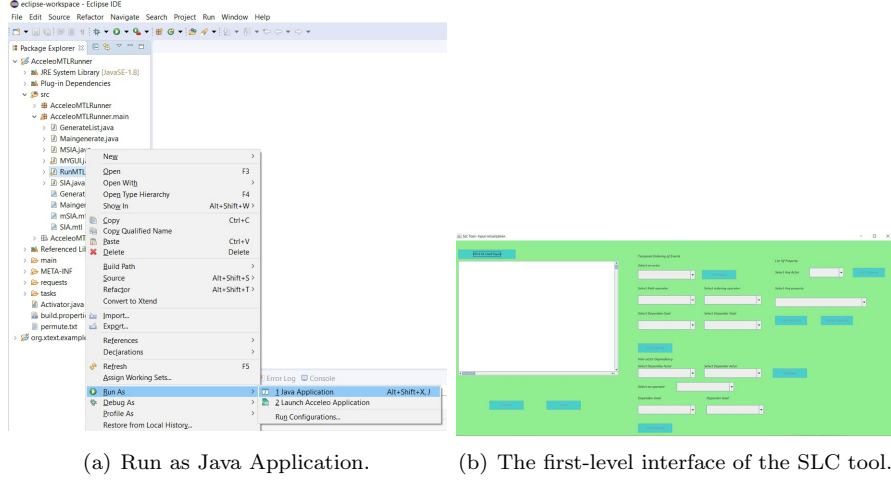
- Click on *Apply and Close* button (Figure 2(d)).

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



(a) Run as Java Application.

(b) The first-level interface of the SLC tool.

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**).
- 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(c)).

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 4(d).

SLC Tool- Input initialization

Click to Load Input

```

gsn model{
  actor Patient{
    goal SeekHealthcare{decompositionType='and'; decomposedBy send_sym
    goal send_symptoms{decompositionType='none';}
    goal send_reports{decompositionType='none';}
    goal get_medicine{decompositionType='none';}
  }
}

```

Temporal Ordering of Events

Select an actor

Patient

Get Goals

Select Path operator

AG

Select ordering operator

AF

Select Dependeo Goal

send_symptoms-FU

Select Depender Goal

send_reports-FU

Set Property

Store Temporal Property

(a) Load goal model.

(b) Specify properties.

SLC Tool- Input initialization

Click to Load Input

```

gsn model{
  actor Patient{
    goal SeekHealthcare{decompositionType='and'; decomposedBy send_sym
    goal send_symptoms{decompositionType='none';}
    goal send_reports{decompositionType='none';}
    goal get_medicine{decompositionType='none';}
  }
  ctl 'AG((send_symptoms-FU) -> AF(send_reports-FU)) AND ((send_reports-FU) -> AF(get
  )
}

```

List Of Property

Select Any Actor

Patient

Get Property

Click to see the stored property

Select Any property

AG send_symptoms-FU AF send_reports-FU
AG send_reports-FU AF get_medicine-FU

Edit Property

Delete Property

(c) Save Properties with goal model.

(d) Select property to Edit or Delete.

Temporal Ordering of Events

Select an actor

Patient

Get Goals

Select Path operator

AG

Select ordering operator

AF

Select Dependeo Goal

send_reports-FU

Select Depender Goal

get_medicine-FU

Set Property

List Of Property

Select Any Actor

Patient

Get Property

Select Any property

AG send_reports-FU AF get_medicine-FU

Edit Property

Delete Property

(e) Edit and re-specify property.

Figure 4: The first level SLC tool interface.

- In List of Property 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 Figure 11).
- On clicking **edit** button the selected property gets loaded into the property specifying boxes on the immediate left panel (refer Figure 4(e)).
- Modify the property and again click on **Set Property** button.

STEP 4: **Setting Inter-Actor property.** Refer to Figure 5.

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

Figure 5: Setting inter-actor properties.

STEP 5: **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 6(a) shows the second-level interface when it is loaded. All buttons in the frame remain de-activated except the **Generate FSM** button.

STEP 6: **Generate FSMs.** Refer to Figure 6(b).

- 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*”.