Warsaw University of Technology

Faculty of Mathematics and Information Science

User Guide for Deterministic Actions with Cost

Submitted by:

Haran Dev Murugan
Rahul Tomer
Rishabh Jain
Kuldeep Shankar
Alaa Abboushi
Bui Tuan Anh.

Approver: Dr Anna Radzikowska

Date: 26/06/20

Introduction:

A dynamic system is designed for deterministic actions with cost. This document will help a user to understand the design and functionality of the system.
Following workflow is performed based on our Example 1 (Travel -Fuel – Reserve) from our drafted functional document. As system is dynamic hence other examples can be verified exactly same way as shown below.
Numerical Indices shown in images have been explained respectively followed by image.

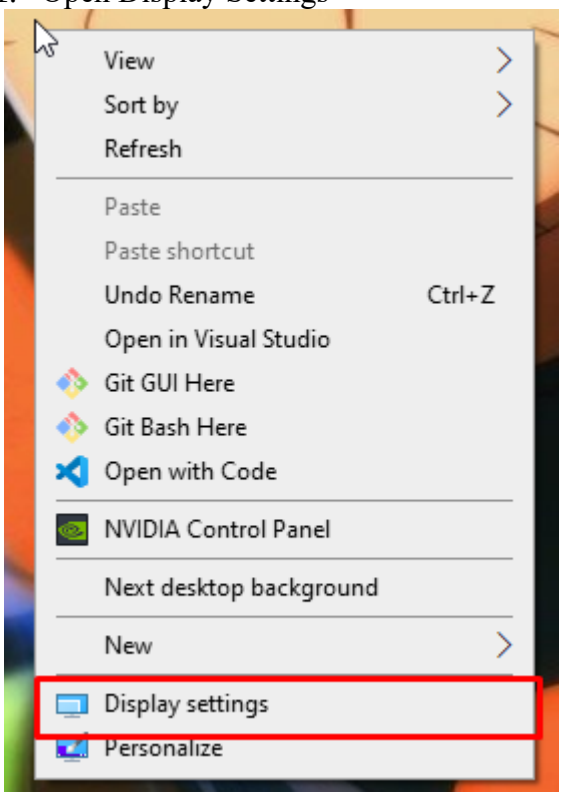
System has been designed in C# language.

System Requirements:

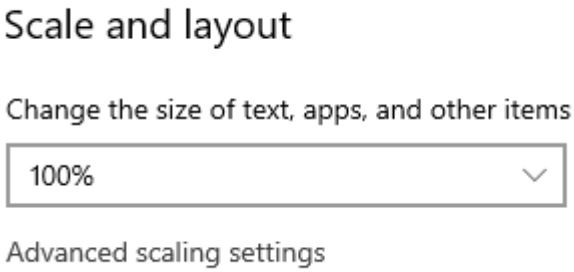
To run this project you need to have display settings to 100%.

Steps:

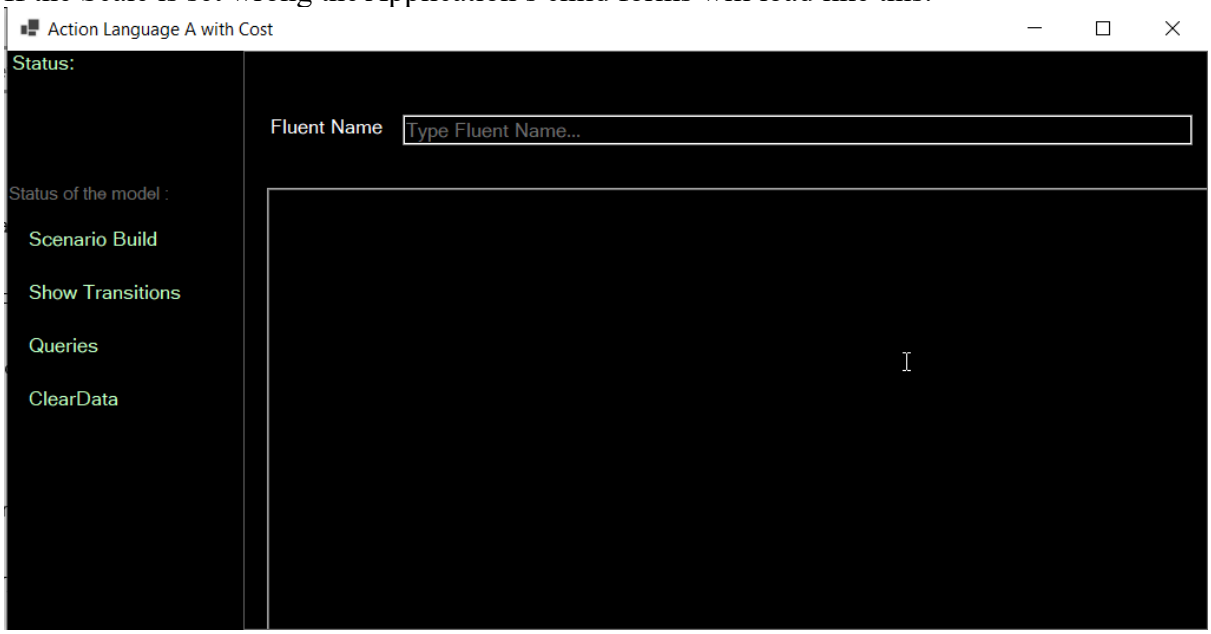
1. Open Display Settings



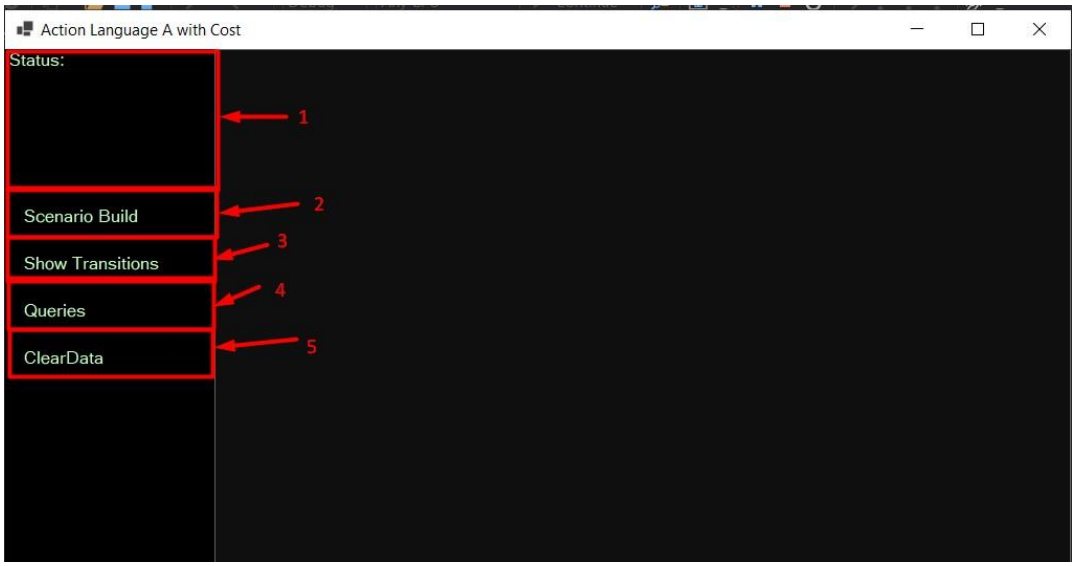
2. Set Scale to 100%



If the Scale is set wrong the Application's child forms will load like this.



Home Screen:



Legend:

1. Status
   Displays the status of the project (If the fluents are submitted, if the actions are added, If the states are added etc.)

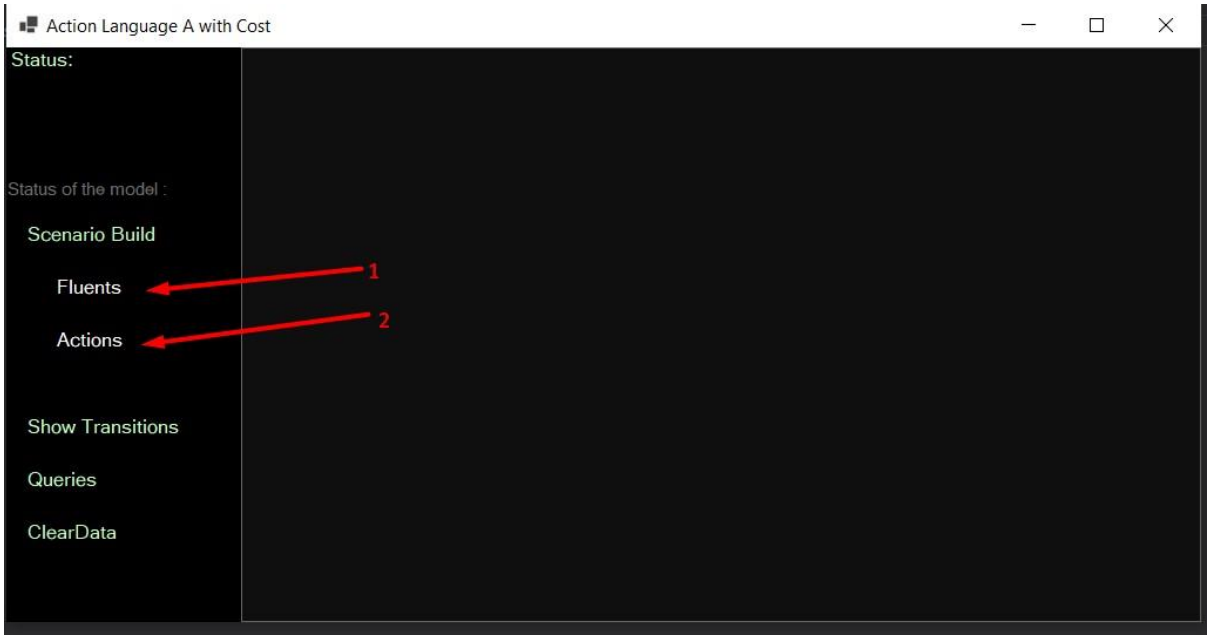
2. Scenario Build
   This button is a drop-down menu button used for adding fluents and actions needed for scenario building. (Refer the Next image)

3. Show Transitions
   This Button opens the show states form that lets us Set the initial state and then shows the generated states and transitions of the system

4. Queries
   This Button opens the queries form window that lets us check a query against the scenario that we have built.

5. Clear Data
   This Button clears all the data inputted by the user and reset the system.

Scenario Build:

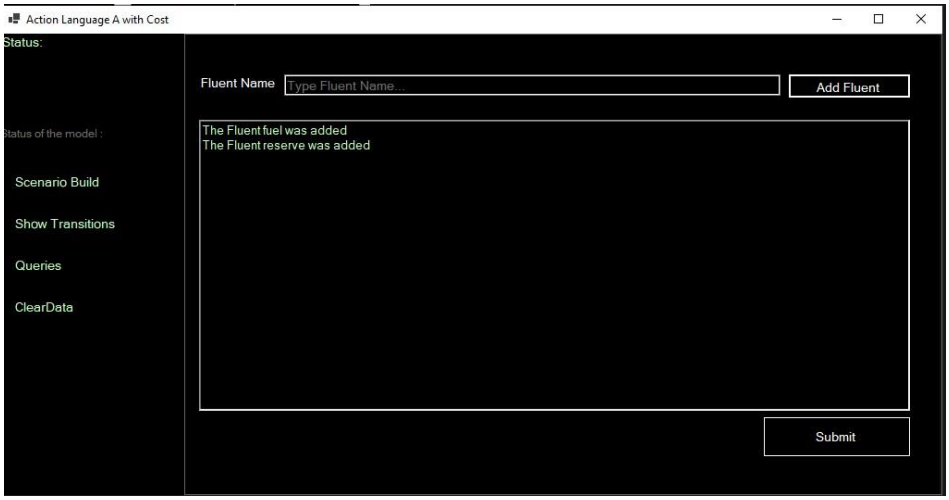


Legend:

1. Fluents
   This Displays the Add Fluents form window that lets us add fluents of the scenario

2. Actions
   This button displays the Add Actions form window that lets us add actions and their respective conditions to the scenario.

Fluents:



How to Add Fluents

1. Type the name of the fluent name in the First Text box.
2. Press add fluent
3. Status of the addition will be displayed on the big text box
4. Press submit button if all fluents are added. (Very important as it will change the status of fluents in the system)

Actions:

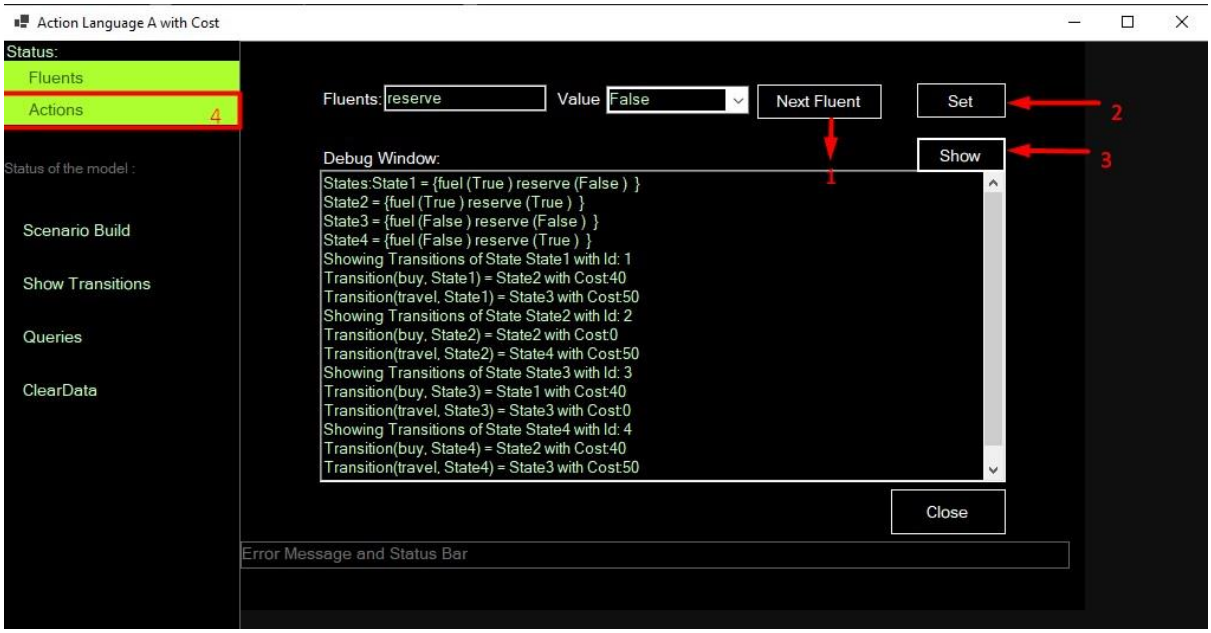
Legend:

1. The Status of adding fluents will turn green after you submit the fluents in the fluents form. If its red this Form will not be displayed because it is important for us to have fluents before adding actions.
2. Add Effect Fluent Button.
3. Add Condition Fluent Button.
4. Add the Action.

How to Add Actions

1. Type the action name and cost of the action.
2. Select the effect fluent and its corresponding value in the Drop-down list and press button 2.
    a. Buy causes not fuel. For this we select fuel as effect and value as false and press button 2.
    b. Error message will be shown to user in case button 2 is pressed after adding one effect fluent already.
3. If the Action statement is “Buy Causes Fuel”, we can keep the condition state as No Condition (Condition state drop down list). But if the action is “ Buy causes fuel if.” , then the condition state should be has condition which upon selecting will display the dropdown for condition fluents, their respective values and the add condition fluent button (Button 3).
4. Adding the Condition Fluents is as same as adding effect fluent only change is instead of button 2 its button 3.
5. Add Action Button (Button 4) is pressed to add the action to the action list and display the log in the status textbox below.
6. After adding all the actions press Submit Button (Very important!).

States:

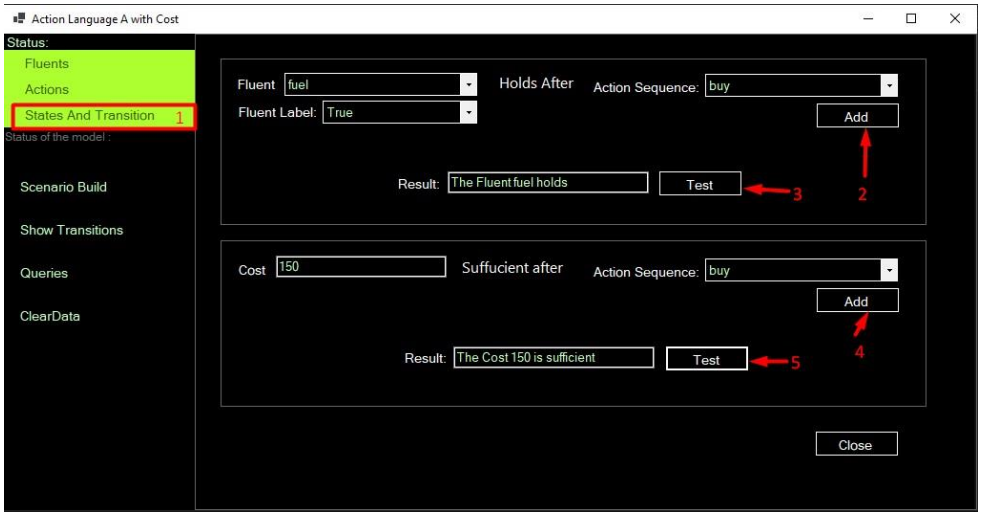


Legend:

- (4) The Status of adding Actions will turn green after you submit the Actions in the Actions form. If its red this Form will not be displayed because it is important for us to have Actions before setting initial state and generating transitions.

1. This button is used to go to the next fluent after you have set the value of a respective fluent specified. if all fluents are set with respective values the status bar will prompt to set the initial state.
2. Sets the initial state for the scenario with the values inputted by the user.
3. Shows the generated states and transitions for the respective initial state, fluents and actions of the scenario in the Debug Window.
4. The Button Close should be pressed after the transitions are shown and the results are verified because it will turn the states and transitions to be in green status, a prerequisite for the Queries Form.

Queries:



Legend:

1. The Status of states and transitions will turn green after you close the show states form after setting the initial state and generating the various states and transitions. If its red this Form will not be displayed because it is important for us to have states and transitions generated before setting queries.
2. The Add button adds the respective Action selected to the Action sequence.
3. The button test is pressed after adding the Actions to the Action sequence and it will display if the fluent and value selected by the user holds or not in the Result Text Box.
4. The Add button adds the respective Action selected to the Action sequence.
5. The button test is pressed after adding the Actions to the Action sequence and it will display if the cost value submitted by the user is sufficient or not in the Result Text Box.

Clear Data:

This Button clears all the data inputted by the user and reset the system.