DEVS-Graph Web-Application User Manual

Nirmal Patel – 101220783

**Supervisor** – Prof. Cristina Ruiz Martin

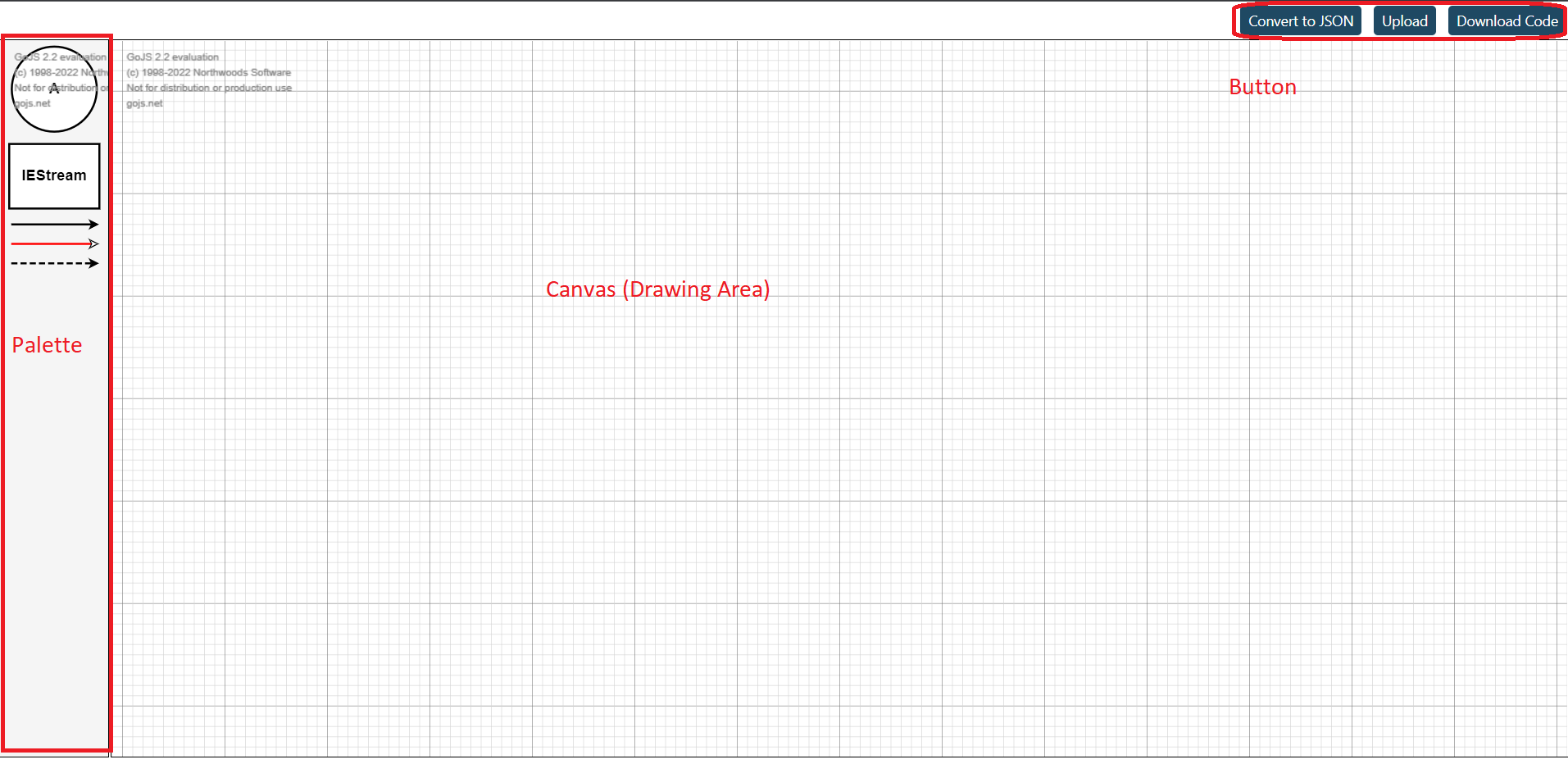
**Introduction**

DEVS-Graph is a graphical formalism to represent DEVS models. With the DEVS-Graph web-application users can draw the DEVS model using graphical user-interface, generate the JSON for the model so that it can be re-used in future. The users can also generate cadmium code for DEVS models from the DEVS-Graph. In this document, a general outline of how to use the web-application is given step by step.

**Step 1:** Open this URL in the browser of your choice:

<https://devs-simulators.sce.carleton.ca/DEVS/>

**Step 2:** Let’s understand three main components of the application,



*Figure 1: Overview of DEVS-Graph web-application*

Palette: On left-hand side of the window, the palette is positioned. There are 5 elements in the palette which can be used to draw the DEVS model.

Diagram, venn diagram

Description automatically generated: The circle is used to define the state. The state name can be edited by clicking on the text inside the circle. By default, the circle has label “A”. This text can be multi-lined label as well.

Shape

Description automatically generated with low confidence: The square is used to indicate external input source to the model. To define the type of the input, click on the text and press enter to type the label.

: The solid line arrow is used for defining external transition in the atomic models. The transition condition can be edited by clicking on the label of the link.

: The red arrow with white arrowhead is used for couplings between atomic and coupled models.

: The dashed line arrow is used for defining internal transition in the atomic models. The transition output can be edited by clicking on the label of the link.

Canvas: The canvas or the drawing area is used for drawing the models. To draw the model, drag and drop different components from the palette window. Some of the short-cuts for the canvas are as follows,

Ctrl + A: Select all the components of the canvas

Ctrl + - : Zoom-out the canvas layout (makes the components look smaller)

Ctrl + +: Zoom-in the canvas layout (makes the components look bigger)

Ctrl + C: Copy the selected components

Ctrl + V: Paste the copied components

Ctrl + G: Group the selected component, puts the selected components in a square box.

Ctrl + Shift + G: Ungroup the selected components.

Buttons: There are three buttons on the right-top part of the webpage. The buttons functions are as follows,

Convert to JSON: Converts the drawn model in canvas to a raw JSON, which will be downloaded in the user’s machine. This JSON can be re-used to load the same model again in future.

Load: Upload JSON from the user’s machine to draw the model defined in the JSON file.

Download Code: To generate cadmium code for the DEVS model click on the download code button which will prompt for a name for the project and upon filling the name and clicking on download will download a zip file containing all the source files needed to run the model.

**Example:**



Step 1: Open any text editor and copy-paste the content of the above file and save it as model.json

Step 2: Open URL <https://devs-simulators.sce.carleton.ca/DEVS/> in the web-browser.

Step 3: Click on load button and upload the file created in step 1 (model.json).

Step 4: Click on load and the drawn model will be loaded in the canvas.

Text

Description automatically generated

*Figure 2: Upload file model.json to the web-application*

*Diagram

Description automatically generated*

*Figure 3: Canvas after loading model.json*

Step 5: Click on “Download Code” button to download the source code, it will open a pop-up to specify the project name. We can put any name of our choice, in this example put name as ‘hotel’.

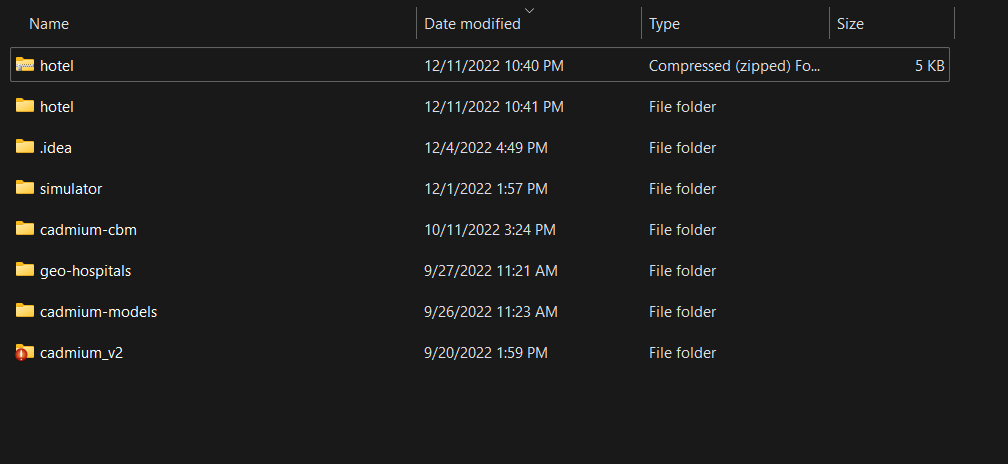
Diagram

Description automatically generated

*Figure 4: Download model named as hotel.*

Step 6: After clicking on download, verify that a zip file named hotel.zip will be downloaded in the downloads folder. Extract this zip file.

Step 7: Place the extracted file in the folder in which the cadmium\_v2 source code is located.

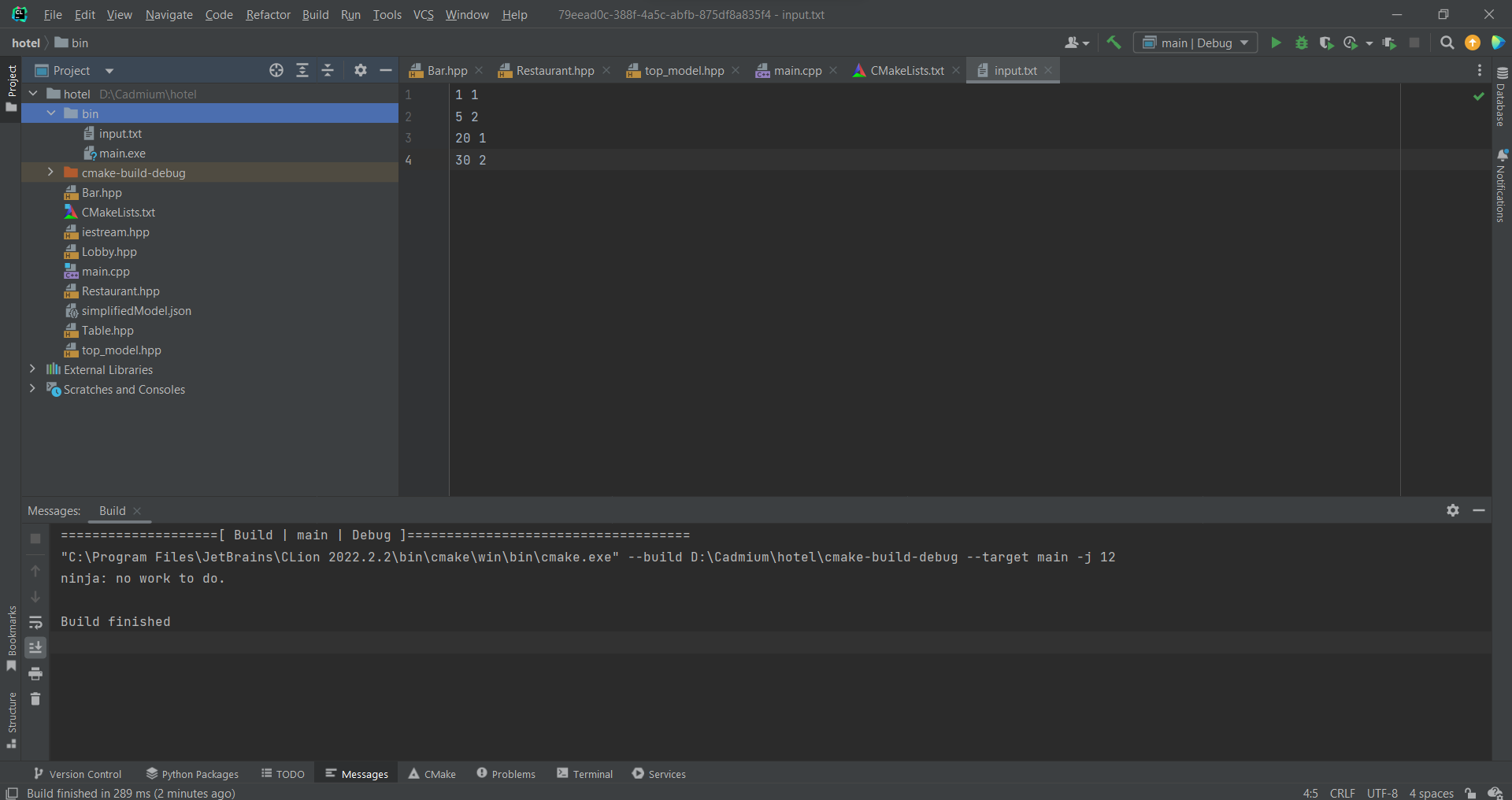


*Figure 5: Place the hotel folder in the cadmium\_v2’s root directory.*

Step 8: Open any IDE of your choice for c++ source code and build the project hotel using the CMakeLists.txt file provided. For this example, IDE Clion is used.

Step 9: Build the project, after successful build, in the hotel/bin folder executable file will be created. In case of windows the executable will be “main.exe”.

Step 10: Create a sample input.txt file to run this executable for a simple input message sequence.



*Figure 6: Sample input.txt file created in the bin folder of the project*

Step 10: Edit the run configuration to add run-time arguments for the simulator or optionally run this code using command line command “main.exe ./input.txt 50”.

Graphical user interface, text, application

Description automatically generated

*Figure 7: Add Program arguments ./input.txt 50*

Step 11: Run the current configuration of the program and after successful run an output file named messages.csv will be created in the hotel/bin folder.

A screenshot of a computer

Description automatically generated

*Figure 8: Sample output file for this example with the messages in csv format.*