Logo, company name

Description automatically generated

**Cell-DEVS: Web-Application for CD++**

**User Manual**

*Advanced Real-Time Simulation Laboratory,*

*Department of System and Computer Engineering,*

*Carleton University, Ottawa, Canada*

Date: 11th April 2022

1. **INTRODUCTION:**

This document will allow the users to use the CD++ web-application effectively to simulate and visualize the Cell-DEVS models. The CD++ web-application allows users to simulate their models in the web browser, using two available simulators, the original CD++ simulator and Santi (extended and improved lopez simulator). This web-application has features like, in browser editing of the model, macro, initial value, palette and initial state variable values. After the model is simulated successfully, the visualization will be shown on the right half of the browser window. If there are any errors while running the simulation the descriptive error message will be shown on the right half of the browser window. For more detailed information for the simulation and the rules evaluation, the user can enable debug mode before running the model. The users may share their work by downloading the zip file of the model and the results files. In the next sections, all the features and use cases are explained.

1. **System Requirement:**

**Device:** Windows, MacOS or linux OS laptop or Desktop

**Web browser:** Latest Chrome, Firefox, or Edge (any modern JavaScript enabled browser)

1. **Functionalities:**

* Connect to Carleton VPN (<https://carleton.ca/its/help-centre/remote-access/> )
* Open CD++ web-application in any browser by using the URL <http://devs-simulators.sce.carleton.ca:8080/CD++/Cell-DEVS/> . It should look like the image below.

Graphical user interface

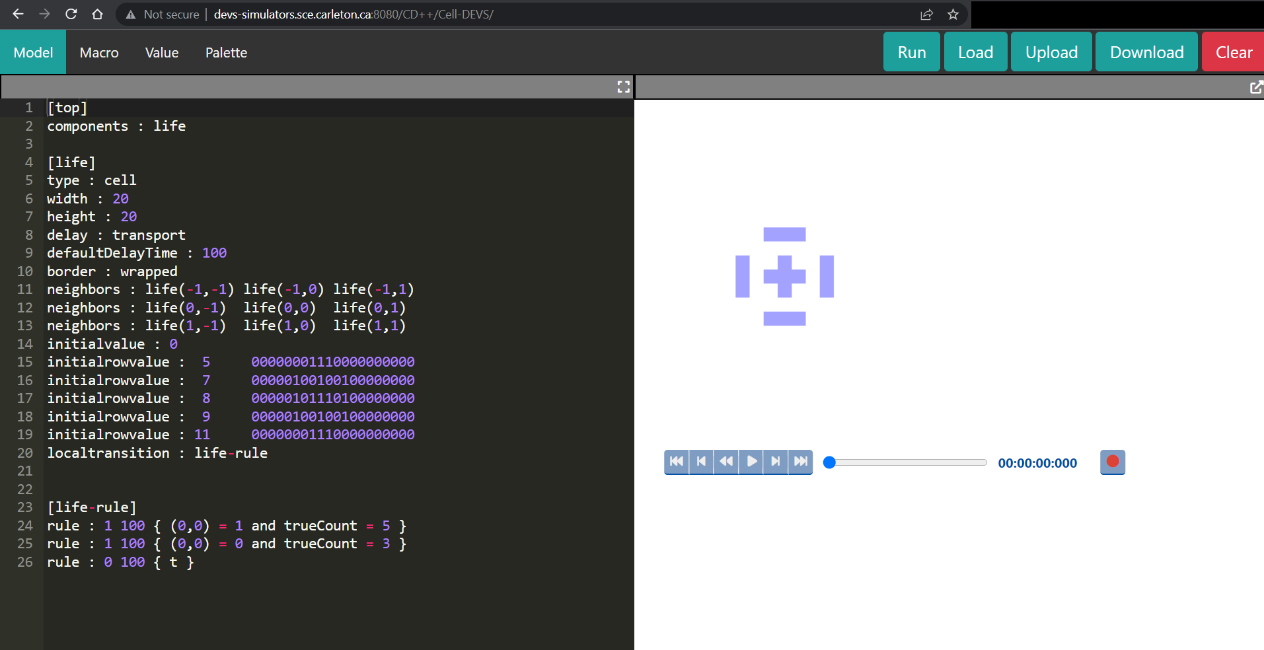
Description automatically generated

* The top-left tool bar options show available code editor windows, model, macro, value, and palette.
* The right-left tool bar buttons show available actions to simulate, edit, load, or create a new model.
* The “Run” button, opens a pop-up for the simulation inputs like project name, macro name (if used) and simulation time.

Graphical user interface, text, application

Description automatically generated

* The run pop-up will look like the above screen, here the simulator version CD++ is selected, project name is ant, macro file name is hor.inc, simulation time is 1 minute 20 seconds, and the debug mode is enabled.
* The simulation time needs to be in defined format of MM: SS, where MM is 2-digit minutes (between 00 to 59) and SS is two digit seconds (between 00 to 59).
* Click on run button to simulate the model and wait for the simulation loading icon to complete the simulation.
* If the simulation runs successfully the visualization will be shown on the right of the window.



* If there is any error while running the simulation, it will be shown on the right half of the window.

Text

Description automatically generated

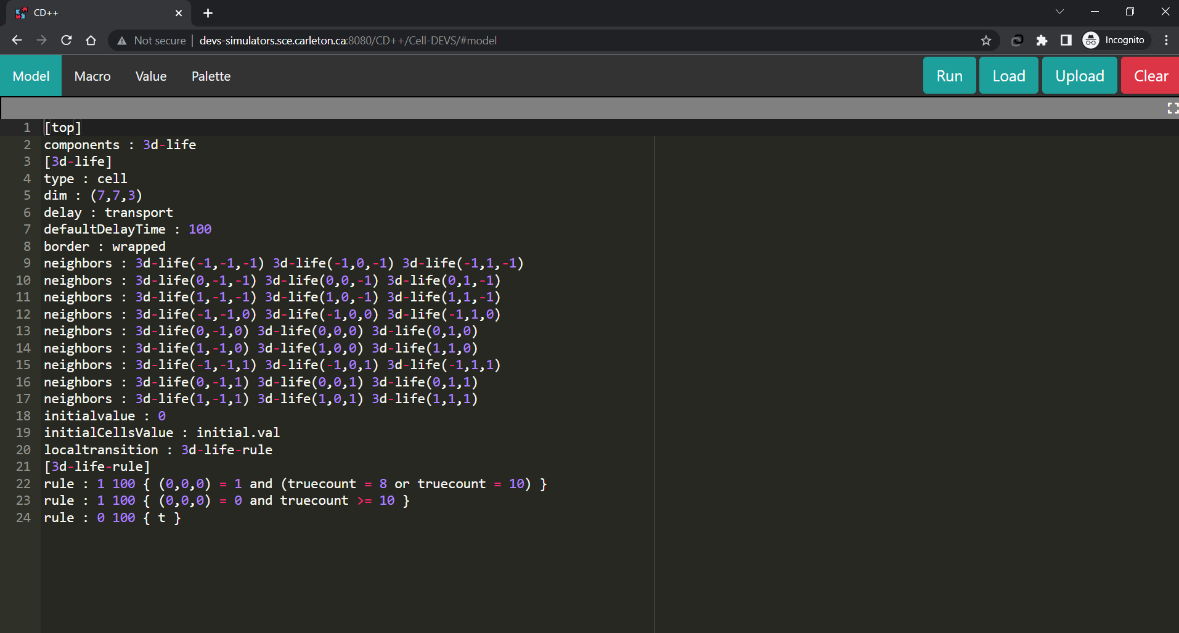
1. **Using a sample model**

* To use any of the sample models, click on the “load” button from the right-top of toolbar.
* It will open a pop-up with list of available sample models. If the model requires CD++ simulator, while running the simulation user may select any of the simulator (CD++ or Santi), but if the model requires Santi simulator, it can only run using the Santi option.

Graphical user interface

Description automatically generated

* Above screen will be presented on pressing “Load” button, the Smog, Reaction and Virus model run with Santi simulator only, while all the other sample models can be run using any of the simulators.
* For example, we will select Life-3d model. The model, value, and palette window will auto populate. There is no macro file for this model. The screen will look like below,



* If user want to make any changes in model, value, and palette file, they may do so, otherwise click on “Run” button to configure simulation parameters.
* Select any of the simulator (CD++ or Santi), as this model does not have state variables, we will leave them empty. Fill in the project name, leave macro file name empty as there are none, and fill out simulation time as per the format (for example 01:00). Lastly, click on the run button from the pop-up and wait for the simulation to complete.
* After successful execution, user should be able to see the screen as below,

Graphical user interface

Description automatically generated

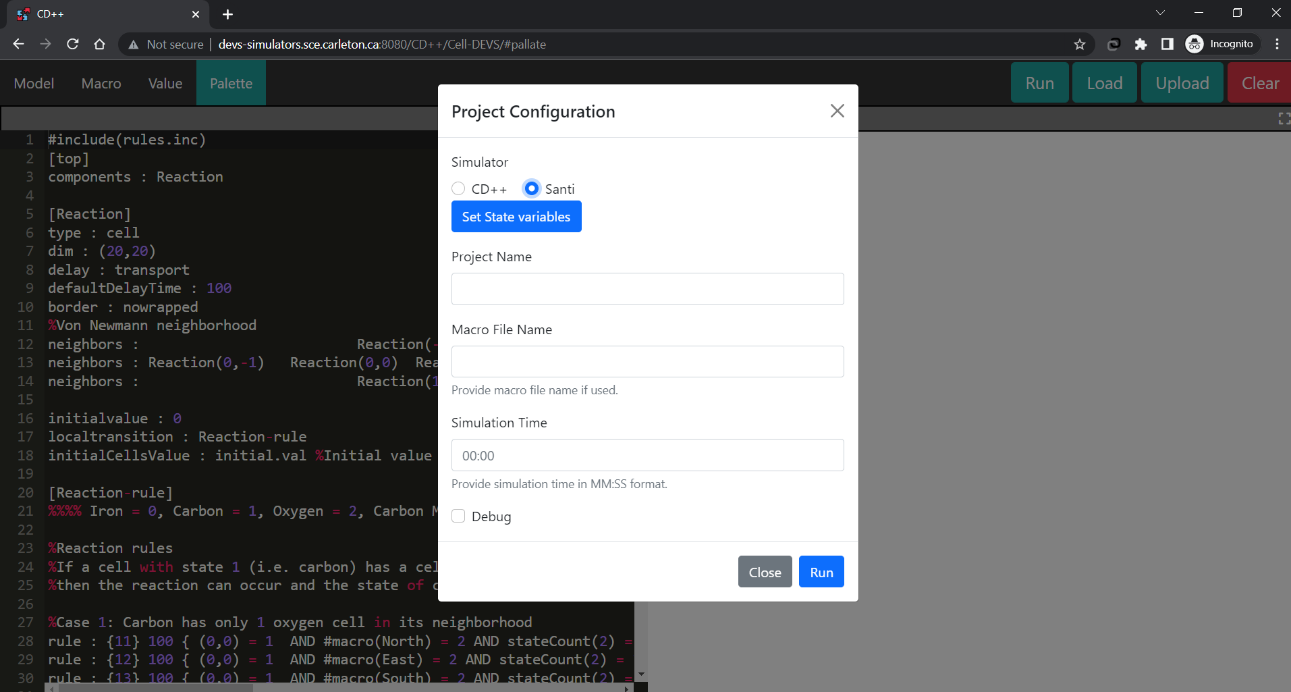
* As we can see, since the visualization has three dimensions (three layers), the visualization window is overflowing. To see the visualization tool in full window, use the square button at the top-right of the visualization tool. It will make the visualization window cover the entire screen,

Qr code

Description automatically generated with medium confidence

1. **Running a model with state variables**

* As per previous section, to use an existing model with state variables, click on “load” button.
* Select “Smog (santi)” model to load the model and auto populate the model, state values, and palette files.
* Click on “Run” button to set the state variables. To set the state variables, click on the “Santi” radio button first, it will present a button to set the state variables.

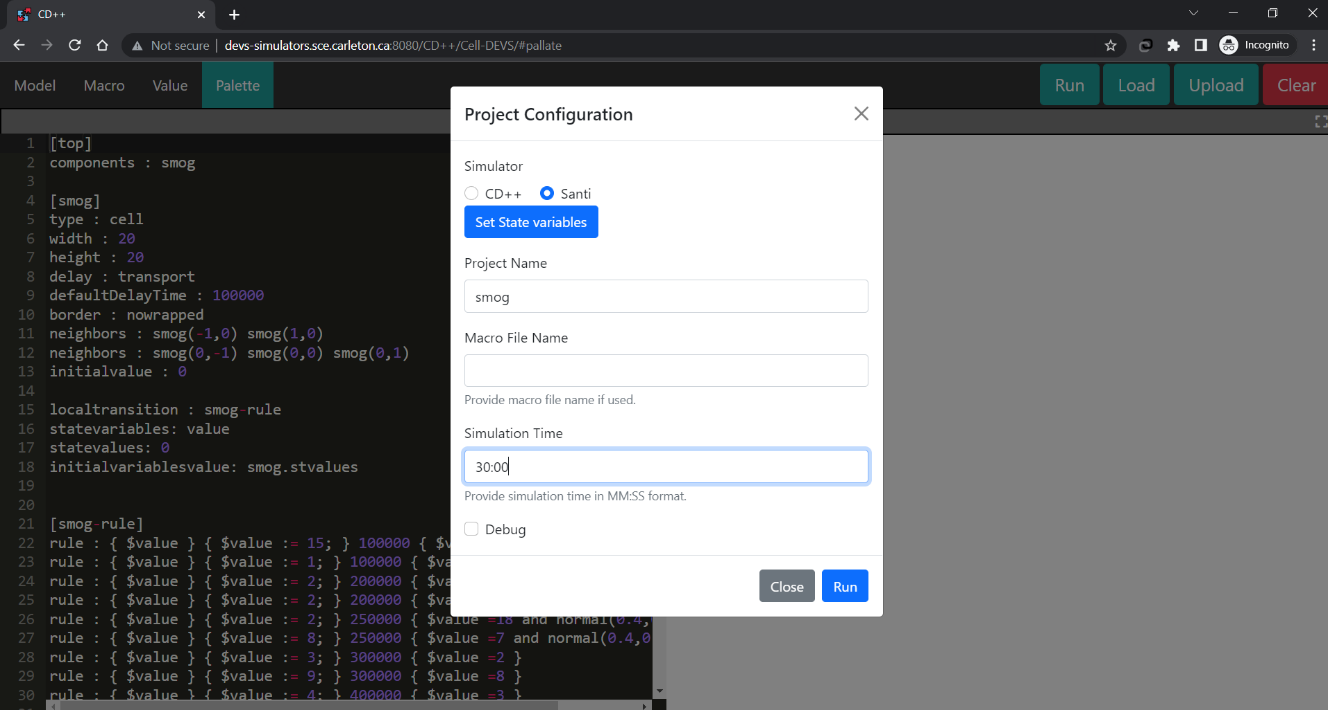


* Click on “Set state variables” button to set the values using a “.stvalues” file.

Graphical user interface, text, application

Description automatically generated

* Enter the state values file name with extension, and in the text box fill out the state variables. For this smog model, the file name and the state variable values will be auto-populated. Edit the values if needed and click on “Done” button to return to the run configuration.
* While setting the state variables using a “.stvalues” file, make sure that the file name matches the name used in the model file.



* Fill out the project name and the simulation time (for example 30:00). Click on the “Run” button and wait for the simulation to compelete.
* After the simulation complete, the visualization can be obsereved on the right half of the screen. Also the “Download” button will appear on the right of the tool-bar (beside Upload button).

Graphical user interface

Description automatically generated

* Click on the “Download” button to download the zip file containing the model file, macro file (if used), value file (if used), palette file, state values file (if used), output file (under the name messages.log), debug files under the name parse.log and debug.log (if debug option is selected).

1. **Error Handling**

* If there are any error, it will be shown on the right half of the screen. The error message will be displayed under the “Description” tag. If the users want to get more information about the simulation and the debug logs, run the simulation again with the “debug” option checked.

Graphical user interface, text, application

Description automatically generated

* Let’s look at some of the error messages with example. Load the “Life (CD++)” model using the Load button. As before, the model and palette windows will be populated.
* Now edit the model file, on line number 20, change the “localtransition: life-rule” to “localtransition: lifez-rule”. Since there is no rule defined as lifez-rule, it should throw error. Run the simulation by clicking on Run button, and filling out project name and simulation time. You should get the error as the below screen, stating that the lifez-rule is missing.

Text

Description automatically generated

* Change the transition rule to life-rule and run again, it should work fine. Now, remove line number 26 completely, making some of the transition to not maching any rule. In this case we should get an error as below,

Text

Description automatically generated

1. **Important Notes**:

* Since this is the web-application and the simulations are being run on a server, there are some limitations in terms of the size and the time of the response. The response size of the simulation should not be more than 25 MB and the response time should not be more than 55 seconds. (Please not that the response time does not correlate to the “simulation time” provided in the project configuration directly.)
* Do not use the debug mode in all the cases, only use it if necessary otherwise it will cost more time and memory on the server while runnig the simulation.
* Use clear button before loading new projects, directly loading new project might not clear all the windows and will lead to false run. It is a good practice to clear all the windows before starting new simulations.
* The editor supports almost all the short-cuts available in the IDEs, such as Ctrl+Z (undo), Ctrl+C (copy), Ctrl+V (paste), Ctrl+F (search) and Ctrl+H (find and replace).
* While using the “Upload” functionality, the respective file will be loaded in the editing windows based on the extention. For model file “.ma”, for macro file “.inc”, for value file “.val”, for palette file “.pal” and for state variables “.stvalues” extensions are used.
* Make sure that the intial value file (.val file) is refered only as “initial.val” file in the model file.