

CARLETON UNIVERSITY
Department of Systems and Computer Engineering
SYSC 5104. METHODOLOGIES FOR DISCRETE EVENT MODELLING AND SIMULATION

Assignment 2

The goal of this assignment is to show understanding of modeling and simulation using the Cell-DEVS formalism, and application of the techniques using the CD++ tool. You must identify a real system that can be represented using Cell-DEVS, build a cellular model of the system and run simulations of the system under analysis. The real system under analysis can be any natural or artificial system.

Part I:

The first stage in the assignment will be to identify one of such systems in the Cell Spaces modeling literature. You can use any of the available resources (libraries, the Internet, digital libraries, etc.), and look for examples of different cellular models. You should provide a copy of the chosen material, and provide a one-page conceptual model description, including a brief sketch of the model structure if a coupled Cell-DEVS is chosen.

The model should be able to be represented as a Cell-DEVS model, to be later implemented in CD++. Students working in teams must define more complex models. In this case, the Cell-DEVS model provided could be integrated with other DEVS models, or be represented as three or higher dimension specifications (maximum: three members per team). You should include one extra dimension or two base models (atomic or coupled) for each extra member.

NONE OF THE MODELS EXPLAINED IN THE COURSE OR AVAILABLE IN OUR WEBPAGES CAN BE USED.

Part II

After this stage is finished, the conceptual model will be used as a basis to develop the final work. You should:

- . Write the formal specifications for the Cell-DEVS models (and for the DEVS models, if any).
- . Using the specification as a basis, build the models using CD++.
- . Once the models have been tested, run simulation examples, showing the reaction of the model to different inputs than those defined in the specifications.
- . Provide a graphical output using CD++ Web Viewer (<https://simulationeverywhere.github.io/CD-WebViewer-2.0/index.html>) (you need to provide a .pal file storing the palette used to visualize the outputs).
- . Generate a video of interesting results using CD++ Web Viewer.

Write a report showing these execution results and analyzing the behavior of the model according to the specifications.

Important dates:

November 4. Assignment 2 requirement document delivery (earlier submissions are encouraged).

November 15. Assignment 2 Report delivery.

The reports should be submitted via email before 10:30 pm.

Deliverables and grades

Part I

This first activity represents 15% of the final mark. Those students not able to provide a model by themselves, will be given one with the proposed characteristics (but they lose the opportunity to work on a model of their choice). If the real system you want to analyze is too complex, select a subset of the real system, and work with it. The deliverable is the one-page report associated to this stage of the assignment, including the corresponding reference (Word or PDF format).

Part II

This specification document is worth 25% of the final mark. It will include the model used in the third stage.

The simulation results and the implementation models are worth 60% of the final mark. You should include a final report explaining the execution results of your model, and any variations you have done to the original specification, showing the new results obtained. It should include a brief description of each of the models (as a conceptual model definition), besides the formal specification.

If you have included a new DEVS model, the source code should run cleanly in the simulation tool. The source will be installed in a 'clean' directory, including only the original toolkit. Do not include the final executable file. Zip all the source code (header files, C++ code, .ma files, .ev files, .pal files, .log). Include shell scripts to run the model, explaining how to run them, which input files are needed for each case. **DELIVER ONLY THE SAME FILE TYPES THAT YOU INCLUDED IN ASSIGNMENT 1. You SHOULD NOT include:**

- .o files
- .exe files
- .out files

- Use meaningful and self-descriptive names for every file included. Consider using names that represent the models you built (DO NOT use “Assign1.doc”, or “assign1.h”; neither “MyNameAssign1.cpp”, etc... Use a name that represents the content of the model).

DEVSmModelsForm

You should fill out this form, and submit it with your report.

All documents for the 2 parts should be combined in a unique delivered assignment document.

The reports will be analyzed as follows:

- . The zip file will be imported to CD++Builder
- . The video files submitted will be viewed
- . The results will be visualized using CD++ Web Viewer. You should include a .pal file to define different colors associated to each of the state values.
- . If you are working in teams, and you provided a DEVS model, it will be recompiled using the makefile provided by you.
- . If needed, the model should compile cleanly, generating the proper simulator in the end
- . The examples will be run using the scripts you provided.
- . After these basic steps are carried out, different changes will be studied, in order to find possible errors.

In order to ensure proper execution, try to execute these same steps by yourselves, before delivering the models.

Include detailed information on each one of the scripts provided.

Check the “Life.ma” model in the Examples webpage to see what is expected.

CHECKLIST

Read carefully the requirements specified before, and to everything there explained; if something is missing, the assignment will be returned unread.

Tick each one of these items, and verify you comply with them.

- You have combined the 2 parts of the report in ONE report. Each part has a meaningful name (not Part I, II, but a descriptive title explaining the content of the subchapter).
- You have included the report document in your zip file
- You have **only** included the needed files:
 - Source code (*.h/*.cpp) for each of the atomic models (if you wrote a DEVS model as part of yours)
 - Coupled model definitions (.ma files)
 - Scripts to execute each of the models using different experimental frameworks
 - Explanations for the detailed behavior for each one of the scripts provided.
 - .ev files containing input values, if needed
 - .log files showing the execution results
 - Your final report document (.doc files, or .pdf versions)
 - .pal files to automatically load with each of the models in the CD++Modeler
 - Video files of visualization results
- You have put meaningful and self-descriptive names to each of the files (including the zip file and folders used)
- You have imported your zip in a “clean” directory, and tried to recompile and run each of the tests, and you were successful in each of them.
- You have checked the “Life” zip file to see an example of what is required.
- You have filled out and submitted the DevsModelForm file.