SERP Group 9 - Retrospective Report 4

Meeting time: 6pm 11th Oct 2020
Attendees: Basil, Daniel, Josh
The meeting was conducted on the 11th of October, 3 days after the client meeting. The main items of discussion were:

- New UI/UX considerations
- Progress of Python logic model
- Ingenuity presentation

What went well:
Throughout both the mid-semester break and the defined sprint period, a focus was placed on a clear definition of behaviour, both user (client) and server side for the application. Corresponding to this was the design of the user interface to support the desired user experience. The plans made were the product of numerous group discussions, also involving the client. Resulting artefacts of this process included visual interface mockups, behaviour documentation for separate user types and actions, and documentation defining the control schema, i.e. the messages and data sent between the client and servers to perform actions and process updates.

Interface designs were produced for the initialisation wizard of the tool, as well as the instructor view of the main application. A basic version of the main application interface was implemented as a prototype. The designs and the prototype received positive feedback from the client, albeit with a few clarifications and suggestions, most of which were already included in the designs.

The control scheme was implemented in a reduced capacity, acting directly between a test webclient and the Python model, completed to perform the basic actions of initialisation, progress to next step (time iteration), and fetching all the updated cell information. Though not complete in functionality, the process to implement more control messages is simple now that the structure is devised, and proven to be feasible. In addition to this, extra functionality was added to the logic model to support general application configuration, variables and behaviour, such as the number of cells, tracking the current time period, and logging.

What was lacking:

Major progress that was lacking were features in the interface. Although it was at such a state where progress could be shown, it was not in any way close to complete based on the client agreed requirements. There were many features that were lacking, including the ability to iterate through time jumps as well as interact with the simulation and retrieve model data. The form of the data visualisation also had not been improved this sprint, meaning that the moving dots were still the primary visualisation, and failed to effectively communicate what was happening in the model.

The communication between the client interface and the python server was also in an incomplete state, although it is possible to communicate to the server from a local computer.

Steps that need to be implemented to complete the communication chain are setting up the interface to send and receive requests from the python server and reflect those changes in the interface, using NodeJS as the intermediary. This will allow us to create scenarios that multiple users can interact with and track their simulations.

Another feature that was not yet implemented was the ability for a user to set up their own simulation. This was currently only possible through the backend configuration, however dynamic initialisation through the user interface is a vital component of the tool to be completed. The idea is that the user will be guided through a setup process where the user will enter their own configuration of cells, variables and policies, that will then be reflected in the simulation. There will be different pages for different parameters with information describing what the user is entering in each.

What are you planning on doing in the next sprint:

We have a lot planned for the next sprint in our project, this primarily is split into two key sections, the first is further development of the user interface of the system and the second being the changes needed to be made to the server.

As for the user interface development, we are planning on defining the primary charts/visualisations we are going to use and how they are going to be displayed. Along with that we are planning on developing the rest of the general user interfaces that are required to use the training tool. This includes also developing the javascript and communication side of the user interface to be able to communicate back to the rest of the model.

Next, we have set a focus on building out the communication needed to get the model to be able to functionally support multiple clients. This will primarily be handled by the NodeJS server, communicating via a websocket connection to the clients, and via another separate websocket connection to the python model to run simulation functions updating the model. As part of this we also need to develop the backend to differentiate user types between an instructor and students connected to the system, to be able to serve them the appropriate functionality.

Ultimately, at the end of this coming sprint we are aiming to have a functional demo performing a basic run of an “exercise” with core functionality, and working on quickly evolving it into a build that we can demonstrate during Ingenuity.

Work Breakdown Structure: