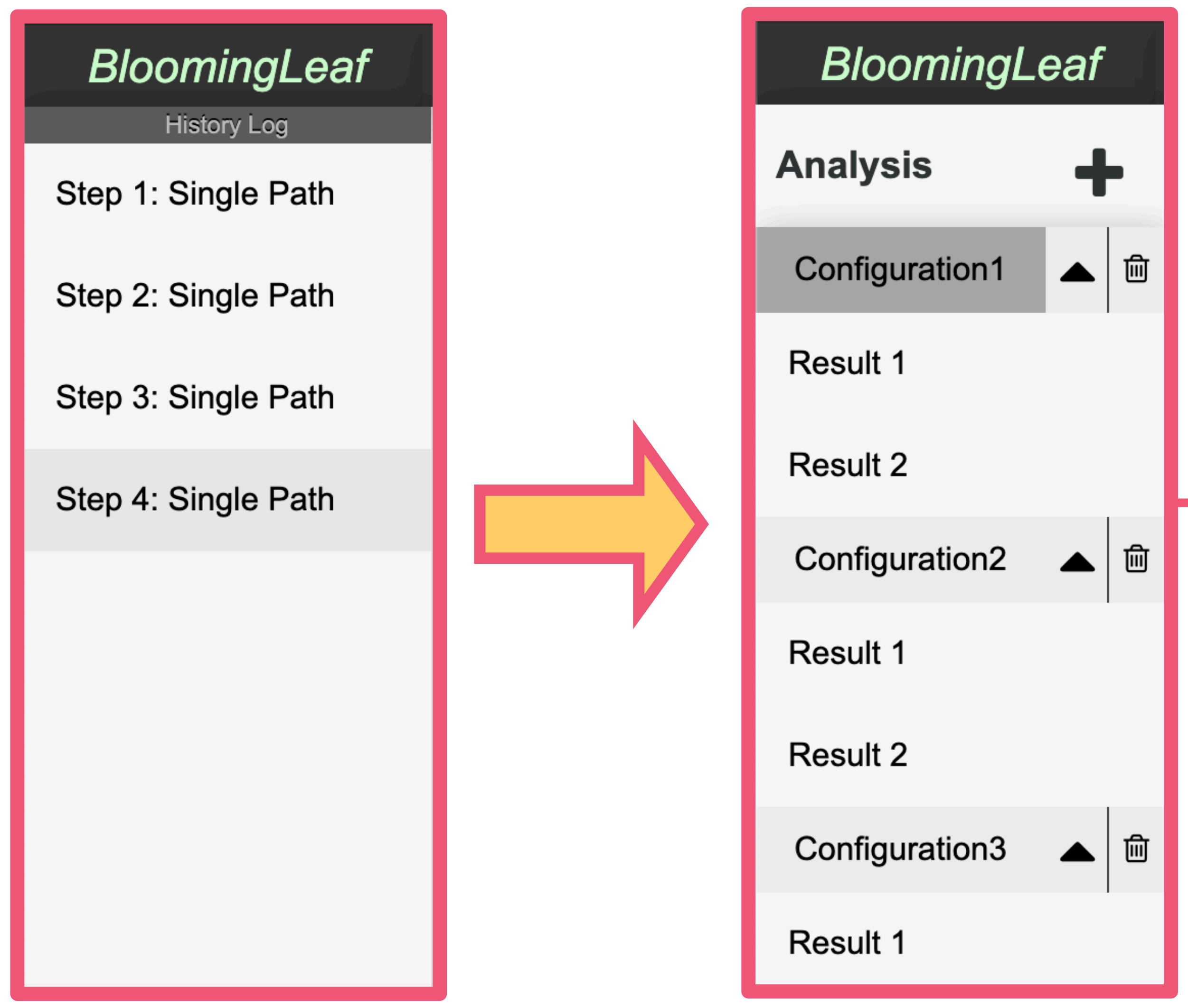
Restructuring Model Analysis Features
to Support Multiple Analysis Configurations in BloomingLeaf

Kathleen Hablutzel '23, Ali Eshghi '21, Ester Zhao '21, Isabella Brody-Calixto '21, Yanning Tan '23, Yesugen Baatartogtokh '23
Supervised by Dr. Alicia M. Grubb, Department of Computer Science

Overview

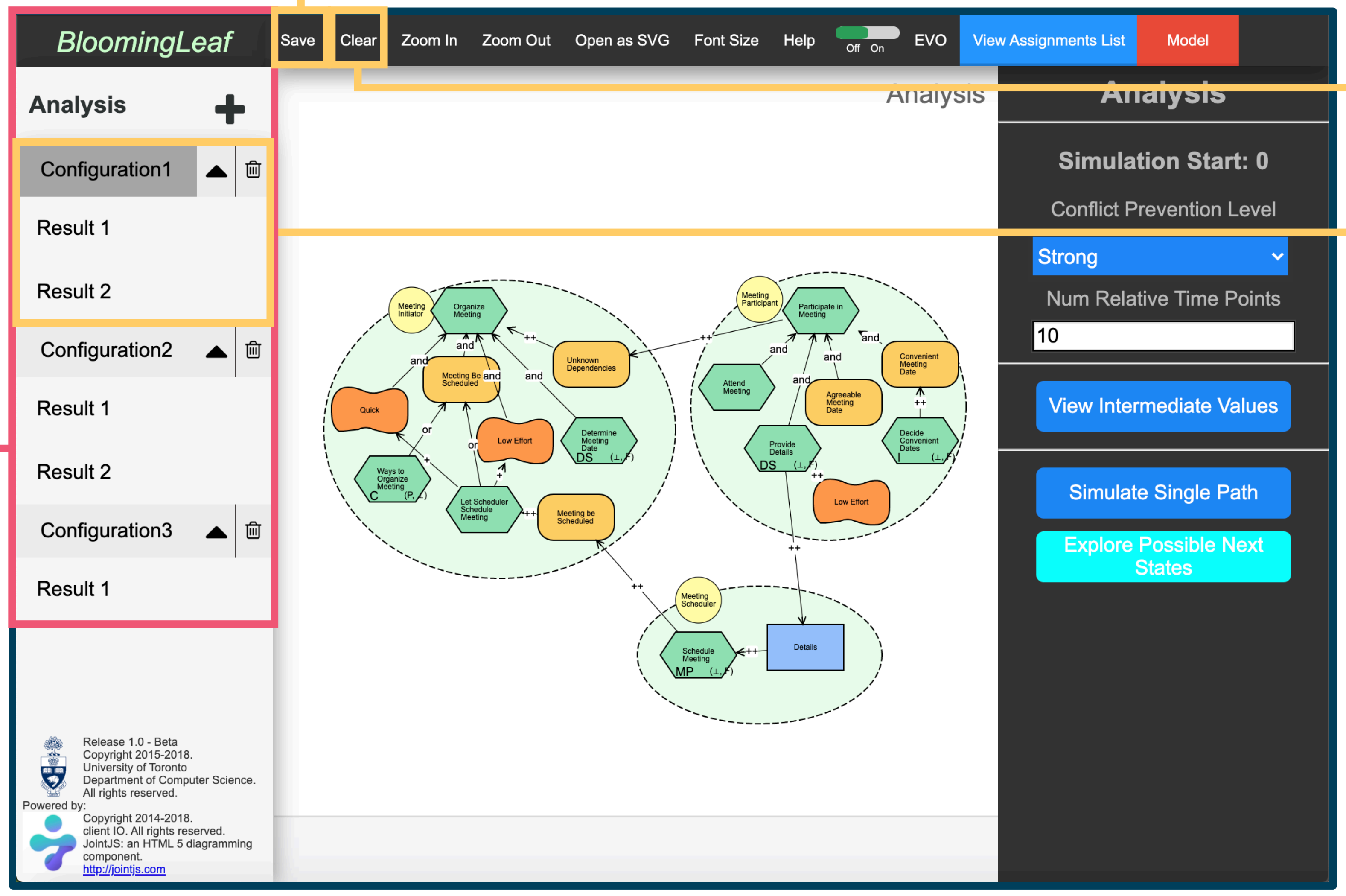
- BloomingLeaf is a web-based tool for modeling and analyzing goal models, which assist stakeholders in decision-making.
- We led a major refactoring effort of the existing code base to allow users to create and store multiple analysis configurations and associated results.
- We employed an agile software development methodology, including weekly sprint meetings. As part of our software engineering process, we documented the existing code base with class diagrams and addressed existing bugs in the software.
- We divide this project into three parts:
  (1) implementing analysis configurations,
  (2) redesigning the user interface, and
  (3) building supporting functionality
- These features improve the ability of users to compare different analysis results and configurations, as well as collaborate on model analysis with other users.

New Configurations Sidebar



Adding Configurations

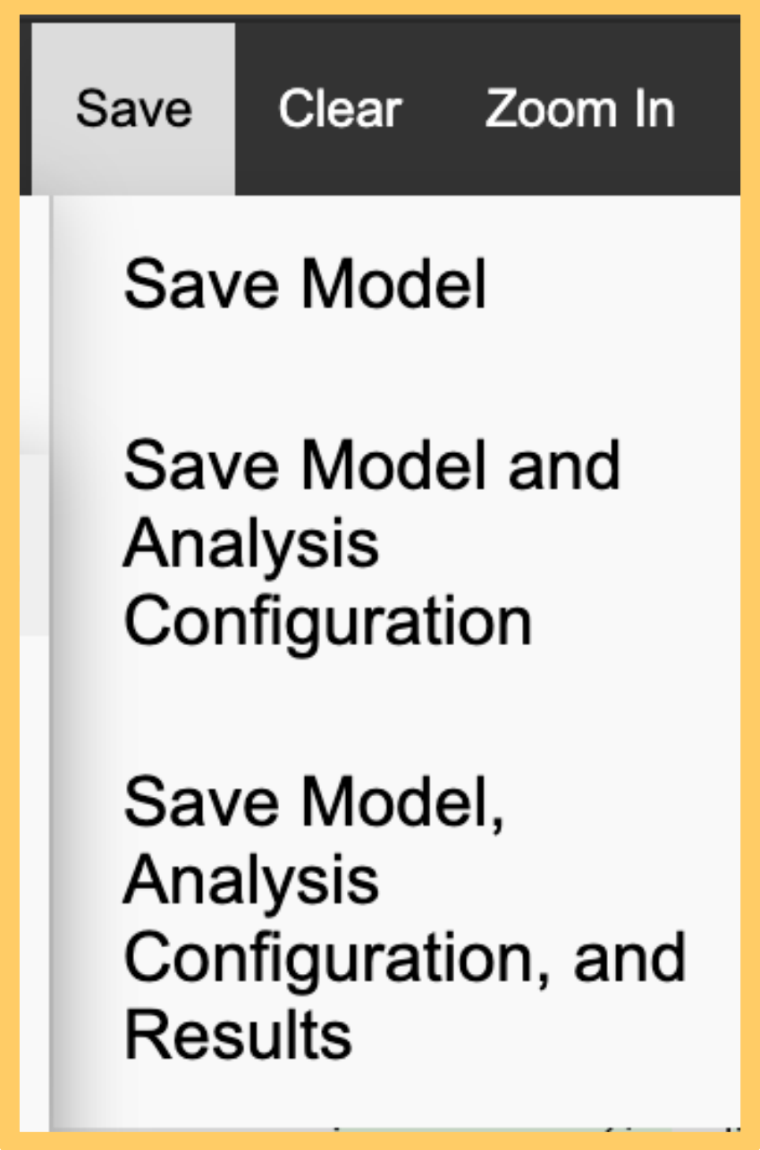
- Previously, users could simulate a single path analysis with a set of conditions on a model only once. Running the same analysis again would clear the previous simulated results, and updating the analysis conditions would result in the previous analysis being lost.
- To allow users to run and store multiple analyses with multiple results, we implemented a grouping functionality called "Analysis Configurations" to help save multiple results under their specific analysis conditions.
- To complete this refactor we created multiple new classes, a data structure to hold the AnalysisConfiguration, and restructured the frontend UI of the tool.



UI Redesign

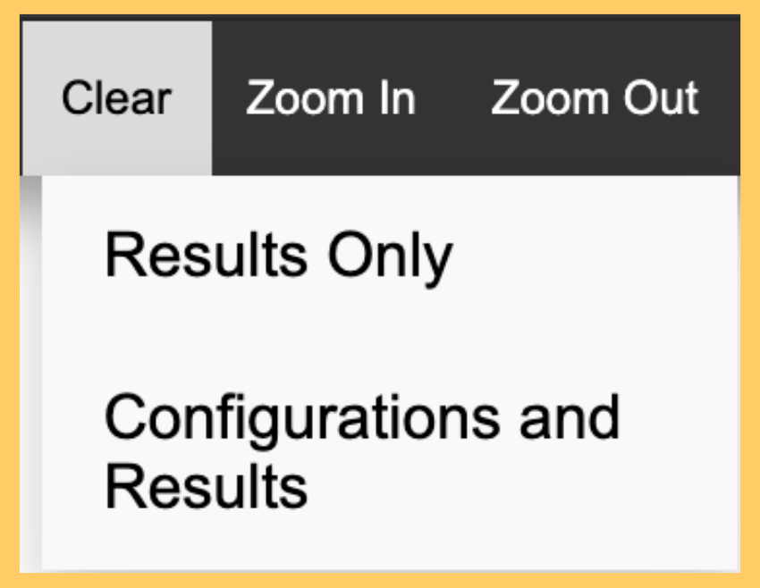
- To implement the configurations sidebar in the frontend, we cleared out code from the defunct history log sidebar and rebuilt it with new functionality.
- We added the ability to create and show multiple configurations at a time, each with unique parameters.
- To compare different modelling results across configurations, we had results populate in a dropdown menu under their respective configurations in real time.
- Now users can switch between multiple configurations and their respective results, and the frontend will update accordingly.

Features


Save
Clear
Zoom In
Save Model
Save Model and Analysis Configuration
Save Model, Analysis Configuration, and Results

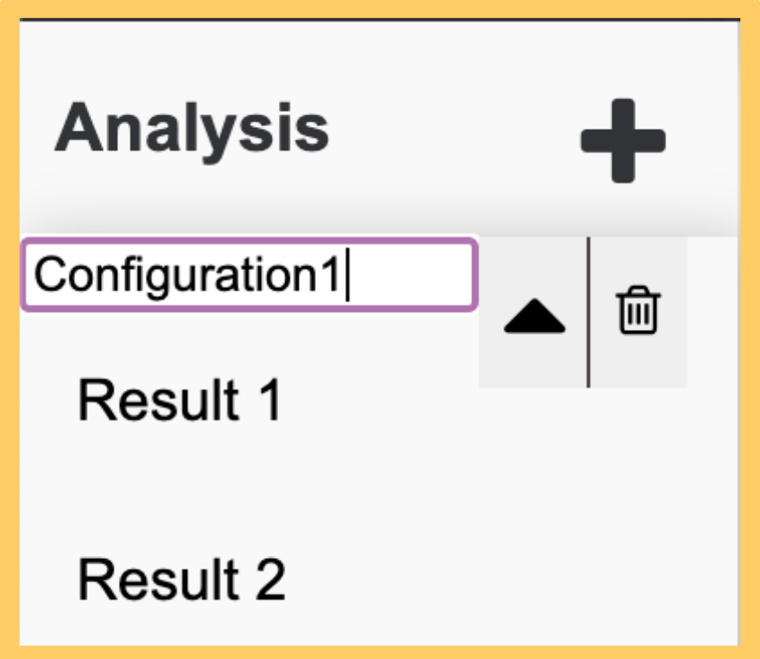
Save/Load

Allows the user to save a model and associated analysis configurations and results as a JSON file, which can be later loaded back into BloomingLeaf for continued use.



Clear

Gives the user the option to clear results with or without clearing analysis configurations as well.



Rename/Delete

Allows the user to better label and differentiate between configurations through renaming. For example: "Strong Conflict Prevention," "Max Absolute Time 100," etc. Allows the user to delete Analysis Configurations from the garbage can icon on the sidebar.

Updated Model

When model updates invalidate past analysis results, we clear results and update configurations with new model information (not shown).

Future Work

- We plan to validate that these additional model management features improve the user experience with analyzing models.
- Additionally, we plan to extend our refactoring effort into a general reorganization of BloomingLeaf's frontend codebase, including developing standards for jQuery event handling syntax and for file organization.

View BloomingLeaf on GitHub | View the Grubb Lab Website
https://github.com/amgrubb/BloomingLeaf | https://amgrubb.github.io/grubb-lab/