

Wyatt Evans, Kyle Krueger, Melody Pressley, Evan Russell

Faculty Mentors: Austin Sanders, Eck Doerry

College of Engineering, Informatics, and Applied Sciences - Northern Arizona University

Clients: Dr. Hélène Coullon & Dr. Frédéric Loulergue

Problem

Background: Our clients, Dr. Coullon and Dr. Loulergue, are working on a new model for software deployment called Madeus/MAD. This new model increases the speed of the deployment process by leveraging parallelism. However, the model has a few core issues that exist during the development phase that are holding it back.

Core Issues:

- **Time Consuming** - MAD deployments (“assemblies”) must be coded by hand, which is tedious for developers who do this frequently.
- **Difficult to Edit** - Any change to the code risks breaking the assembly’s structure.
- **No Easy Testing** - Aside from complicated testing of the code directly, there is no viable way to test these assemblies without actually executing the deployment on a system.

Development Methodology

Tools: Our team used several tools to facilitate the development of MAB:

- **Electron** - Facilitates desktop GUI applications with a Node.js backend, and a Chromium frontend.
- **Konva** - Provides interactable 2D canvas objects with event listeners, using HTML5 and JavaScript.
- **Greensock Animation Platform** - Allows the creation of powerful timeline based animations with precise control of more involved animation sequences.

Modules: Our system has been developed modularly to allow for easy extensibility and maintenance. There are three core modules:

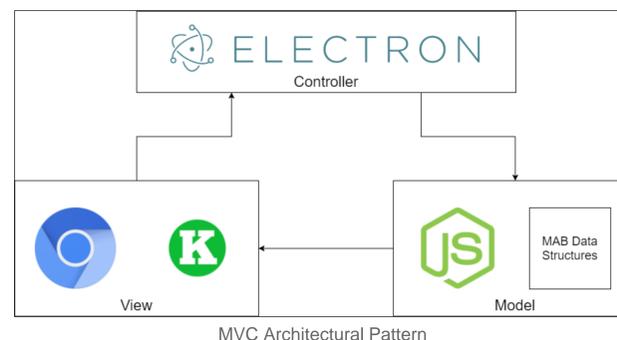
- **User Interface** - The interface through which the user will interact with MAB. Here they will be able to create assemblies in the workspace and execute validated plugins.
- **Data Structure** - A tree-like structure that matches the assembly as it is created and modified in the workspace.
- **Plugins** - Provide additional functionality with minimal dependence on one another. Documentation for valid plugin development exists on our website.

Solution

Overview: Our goal is to provide a solution to our Clients’ three core issues. To do this, we are developing the MAD Assembly Builder (MAB). This software will work as an interface to expedite the process of developing MAD assemblies by alleviating the core issues in their current workflow.

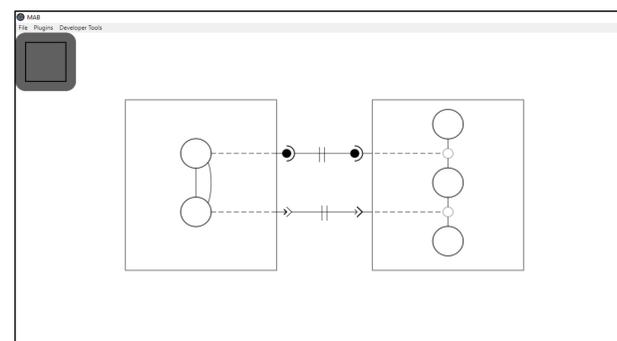
Architecture: Model-View-Controller architectural pattern.

- **Controller** - Electron is a GUI Framework developed by GitHub that utilizes Chromium and Node.js.
- **View** - The HTML-based viewport enabled by Electron; utilizes Konva for shapes and figure creation.
- **Model** - Data structure designed by our team based on elements of the Madeus model (components, places, transitions, etc). The model is updated dynamically as the user interacts with the GUI.



Key Features:

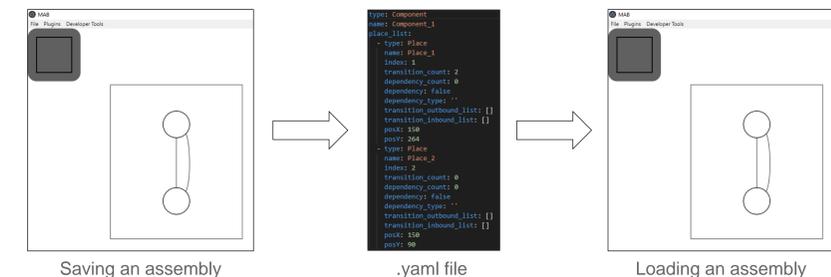
- **MAD Development** - MAB’s interface allows for fast and easy Madeus assembly creation.
- **Mutable Assembly Elements** - MAB provides quick and easy modification and deletion of all Madeus elements.
- **Generation of MAD Code** - MAB’s Code Generation plugin translates Madeus assemblies into executable MAD code.
- **Testing via Simulation** - MAB’s simulation plugin lets users simulate the deployment process of their assembly while they are building it.



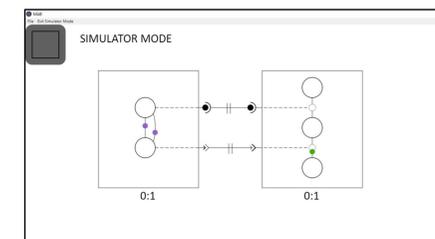
MAB User Interface w/ Example Assembly

Challenges

Saving and Loading of User-Created Assemblies: Our solution was to amend our data structure to store the konva objects (shapes and figures) in order to save them into .yaml files. Saving will serialize all objects and their attributes, including names, positions, scale, etc. Loading builds an assembly from a serialized data-structure.



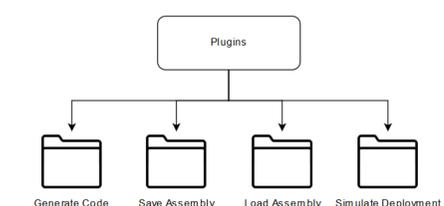
Animating a Deployment Simulation: Our solution was a “simulation mode” which creates a layer on top of the workspace. The simulation mode pauses all event listeners and ensures that the user will not be able to modify the assembly while the simulation is being animated. We chose to use the GSAP animation library which was perfect for animating the directed acyclic graph design of the Madeus assemblies.



MAB Simulated Deployment Plugin w/ Example Assembly

Future Work

New functionality for MAB can be added through plugins, linked via the designated “Plugins” folder. MAB will automatically detect and implement properly structured plugins at launch. MAB’s plugin folder forms the foundation for future extensions - for example, converting pre-existing MAD code to a MAB diagram. Currently MAB has four pre-packaged plugins: Code Generation, Saving of an Assembly (.yaml format), Loading of an Assembly (.yaml format), and Simulation of Assembly Deployment.



MAB Plugins Folder Structure