MusicBox Interactive

An Intuitive User Interface for the MusicBox Chemistry Model

Simon Thomas
MusicBox Model

- CAMP solver
- MICM solver
- Command line interface
Design Considerations

• Scientific use cases
• Ease of use
• Docker distribution
• Web-based interface
• Application as a tool for education
Software Structure

- Web Browser
- Django Server
- MusicBox Docker Container

- HTTP request/response
- Command-line interaction
Development Process

- Addition of Bootstrap
- NCAR color scheme
- Addition of features
- Continuous integration testing
- Docker Hub deployment
Mechanism Configuration

- Reaction/species menus
- Editing panel
- Rate law equations
- Rate constant units
Model Configuration

- Dynamic input forms
- Unit conversion fields
- Conversion tool
- File uploads and interpretation
Chemical Network Visualization

- Nodes and edges generated for each species and chemical reaction
- Generated with PyVis tool on backend
Integrated Reaction Rate Visualization

Integrated reaction rate data from MusicBox model
Configuration I/O

- Easy exporting and importing of complete model configurations.
- Benefits for development and sharing of information.
Evaluation and Testing

• Supervised alpha testing
• Beta testing
• Evaluation as a tool for education
Next Steps

• MusicBox as a web service
• New reaction types
• Additional tools for analysis
• Validation on user inputs
Thank you to Andrew Conley and Matt Dawson for your mentorship during this project.