SIMULATION AND POSTPROCESSING

Lecture 1.2

Modelon

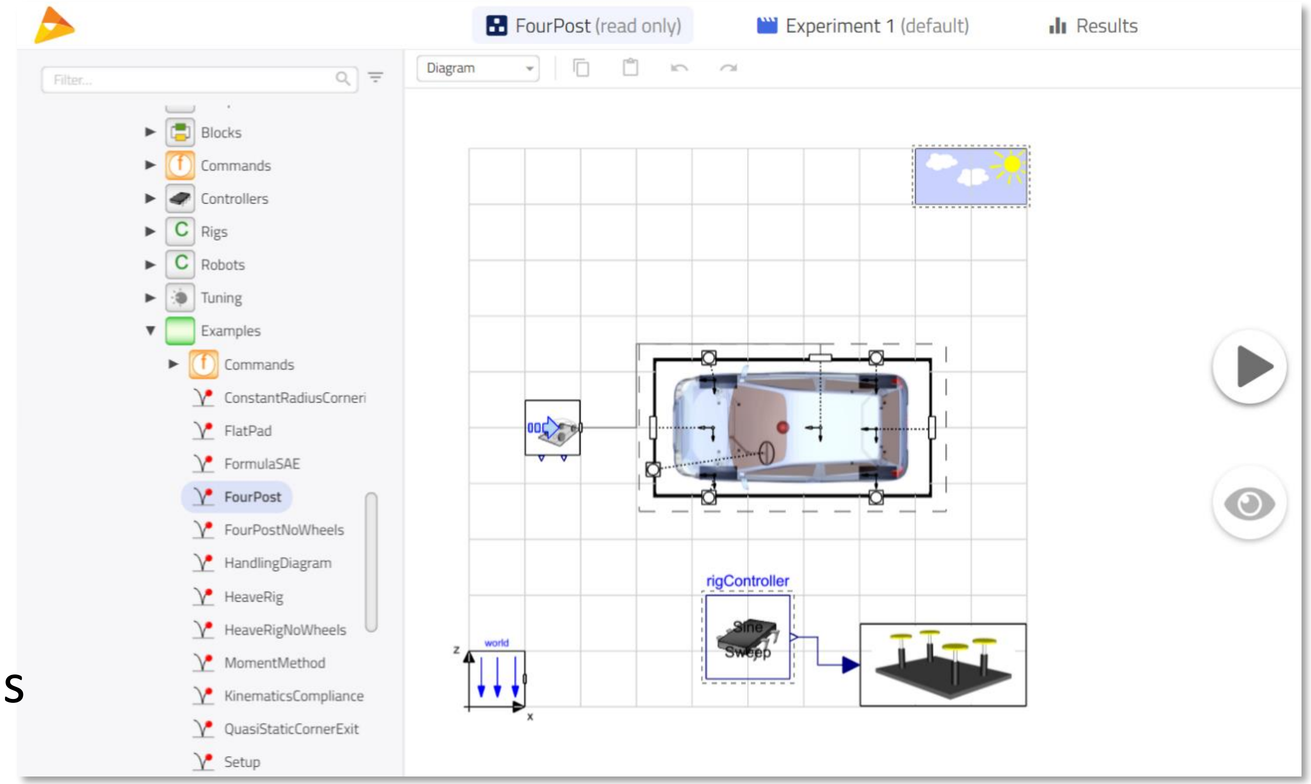
OVERVIEW

- Setting up an experiment model
- Running multiple experiments
- Dynamic Solvers Settings
- Results
- Stickies and views

SETTING UP AN EXPERIMENT MODEL

WHAT IS AN EXPERIMENT MODEL

- A system model with a purpose:
  - Specific data set for parameters
  - Boundary conditions applied
  - Initial conditions applied
  - Can be executed
  - Intention to retrieve results



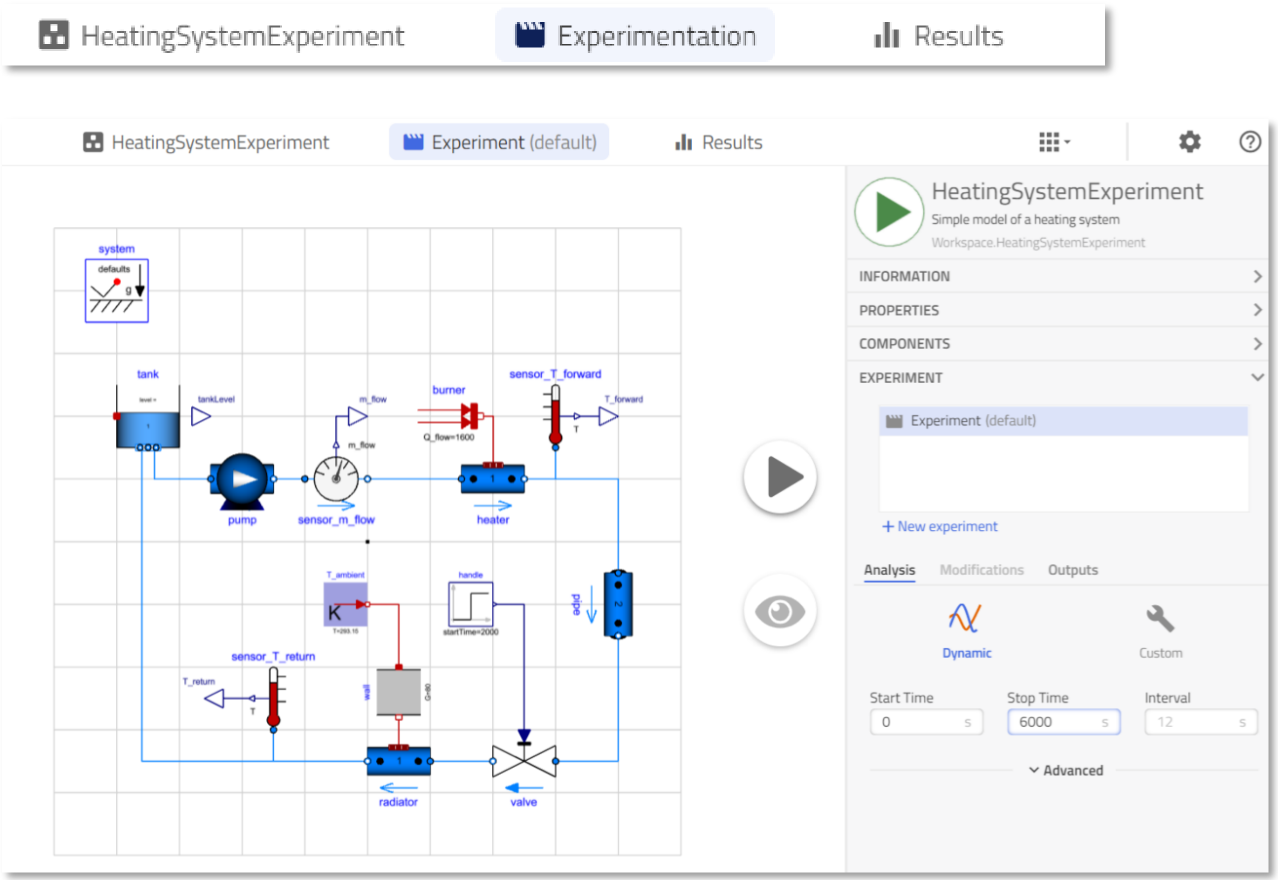
- Good practice to store separatly
- Indicate for users that they are executables

RUNNING MULTIPLE EXPERIMENTS

EXPERIMENTATION MODE

- Impact makes it easy to run several experiments on the same system model
- Each experiment can have separate
  - Data modifications
  - Analysis type
  - Solver settings
  - Compiler settings

When clicking the run button you implicitly run an experiment

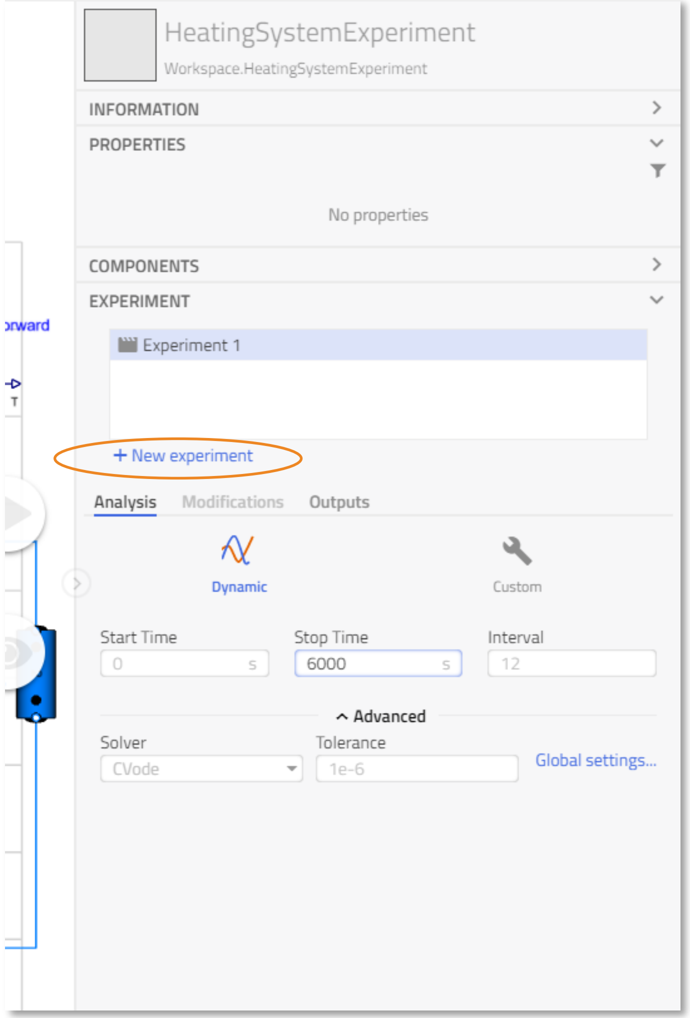


CREATE NEW EXPERIMENT

- Create several new experiment with 

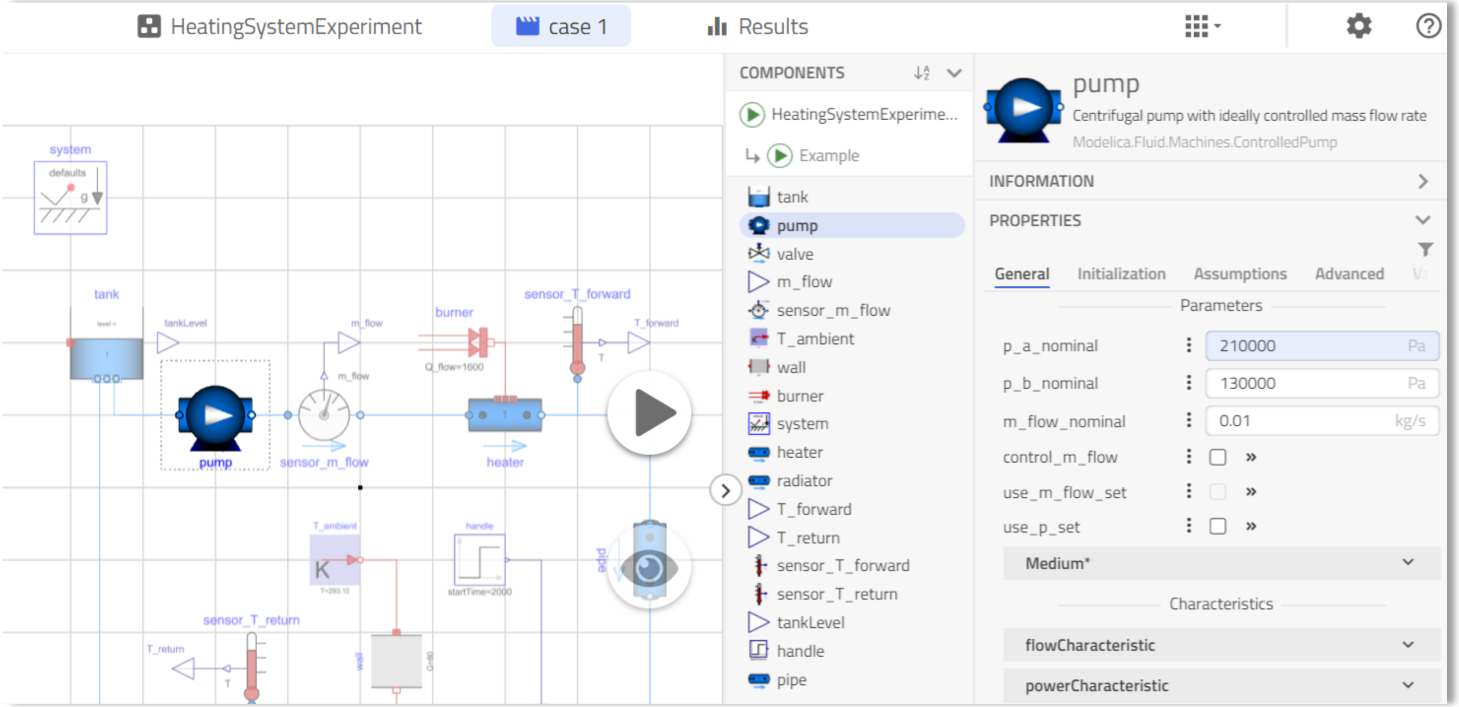
- Experiments can be renamed

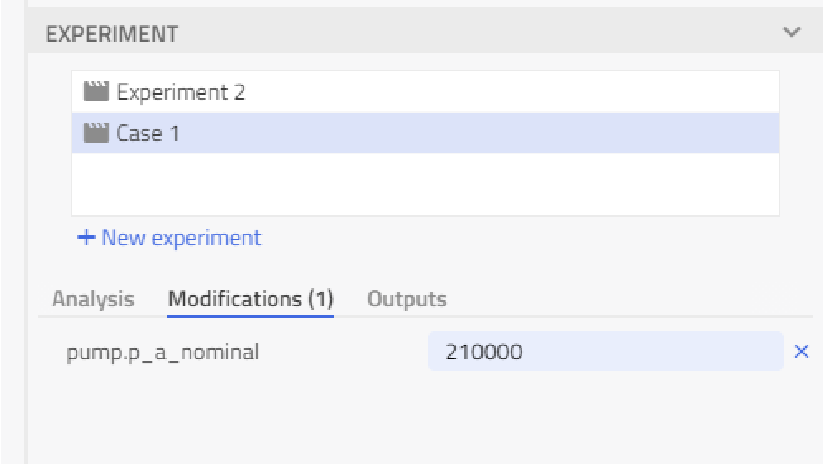
- When a specific experiment is chosen, all changes to the model in that mode, will be saved as modification to the model and only applied to that experiment.



CHANGES IN EXPERIMENTS

- In the example, Case 1 is active, and variable p_a_nominal has changed its value
- All parameter changes in a specific experiment can be viewed under ”Modifications” tab





WORKFLOW

From one system model, you can define several experiments to generate results

System model -> Experiment -> Results

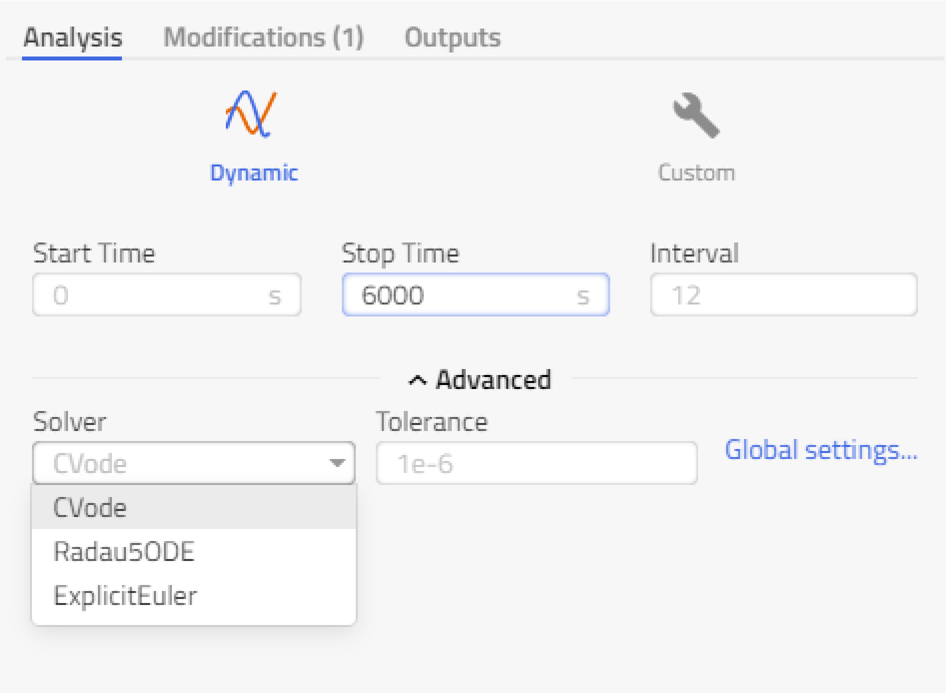
DYNAMIC SOLVER SETTINGS

SOLVING A DYNAMIC PROBLEM

- The resulting mathematical problem formulation is an ODE (or DAE).

$$\dot{x} = f(x, u)$$

- Modelon Impact has a set of ODE solvers to chose from.



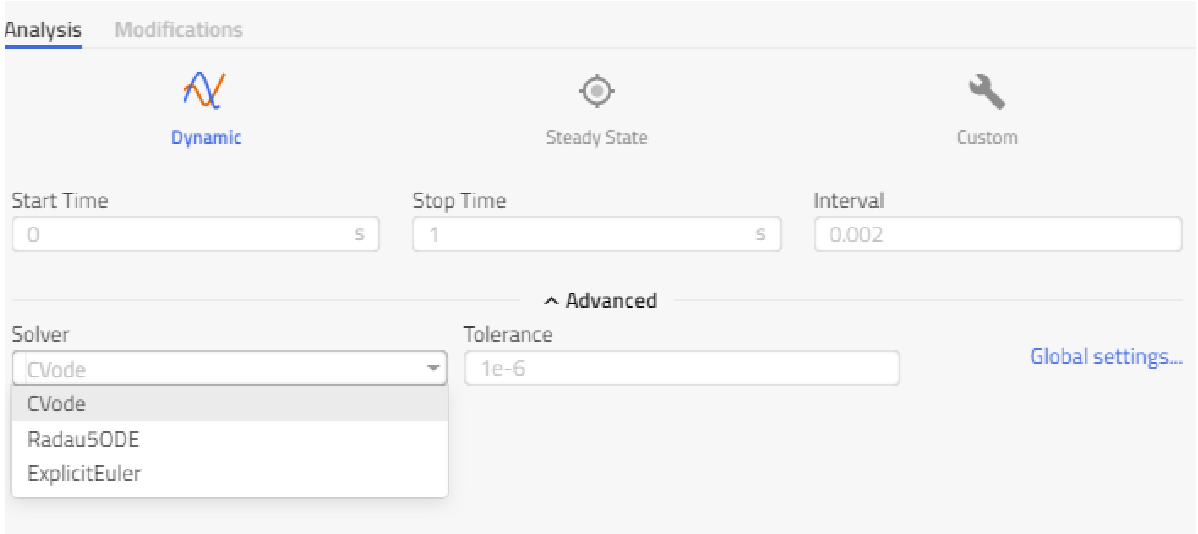
SELECTING SOLVERS

- Variable step size
  - Most integration algorithms available have a *variable step size algorithm.*
  - The integration step size is chosen in such a way, that the error is smaller than the desired maximum error, defined via the set tolerances.
  - This implies, that usually smaller step sizes are used, if smaller tolerances are defined.

- One-step algorithms versus multi-step algorithms
  - One-step algorithms (like Radau) are basically designed such that they start fresh on every step
    - The cost of restarting them after an event is substantially reduced
  - Multi-step algorithms such as CVODE base the next step upon previous steps
    - Expensive to restart the simulation after an event

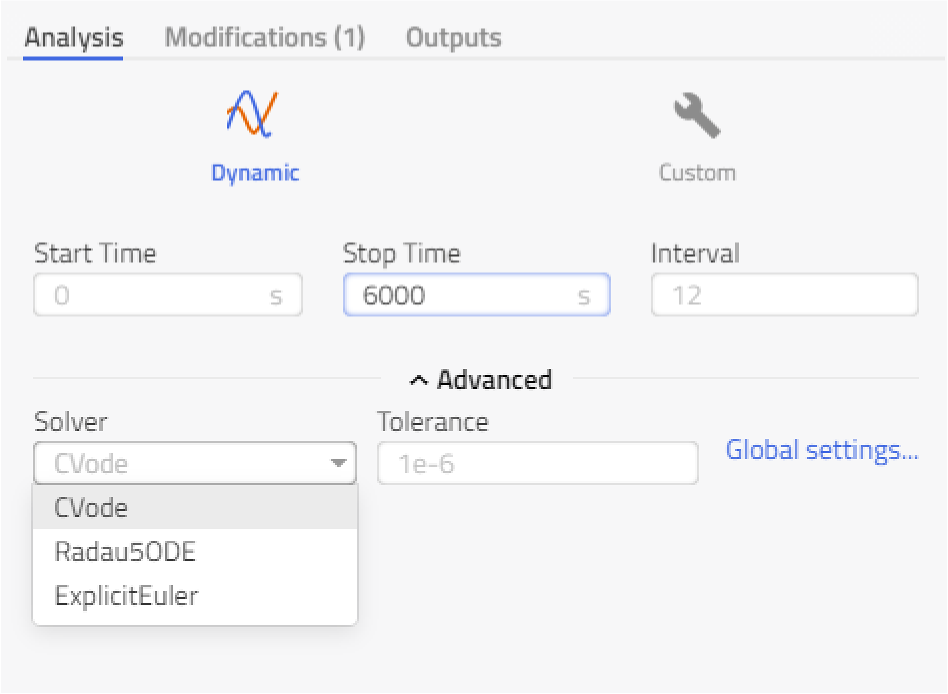
SELECTING SOLVERS

- **CVODE** is a solver for stiff and non-stiff ordinary differential equation (ODE) systems (initial value problem) given in explicit form y' = f(t,y). The methods used in CVODE are **variable-order, variable-step, multistep** methods. For nonstiff problems, CVODE includes the Adams-Moulton formulas, with the order varying between 1 and 12. For stiff problems, CVODE includes the Backward Differentiation Formulas (BDFs) in so-called fixed-leading coefficient form, with order varying between 1 and 5.
- **Radau5ODE**: Radau IIA fifth-order three-stages with step-size control and continuous output. Based on the FORTRAN code RADAU5 by E.Hairer and G.Wanner.
- **ExplicitEuler:** This solver solves an explicit ordinary differential equation using the explicit Euler method.



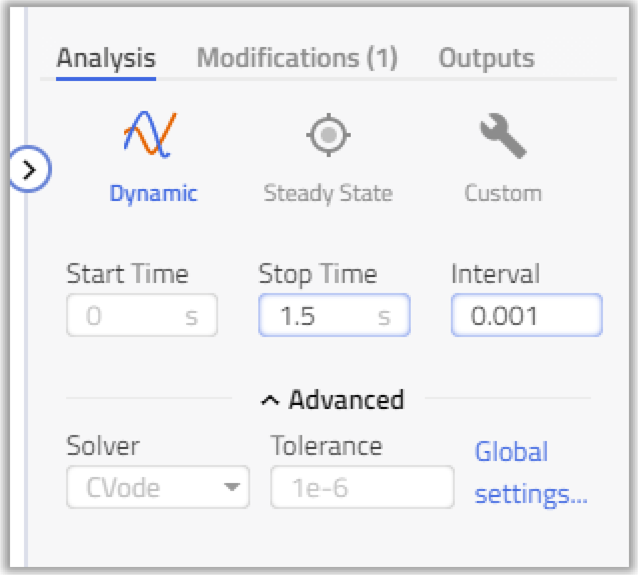
SELECTING SOLVERS

- Tolerance: affects the accuracy of the solution
- Interval: affects the resolution of the stored solution

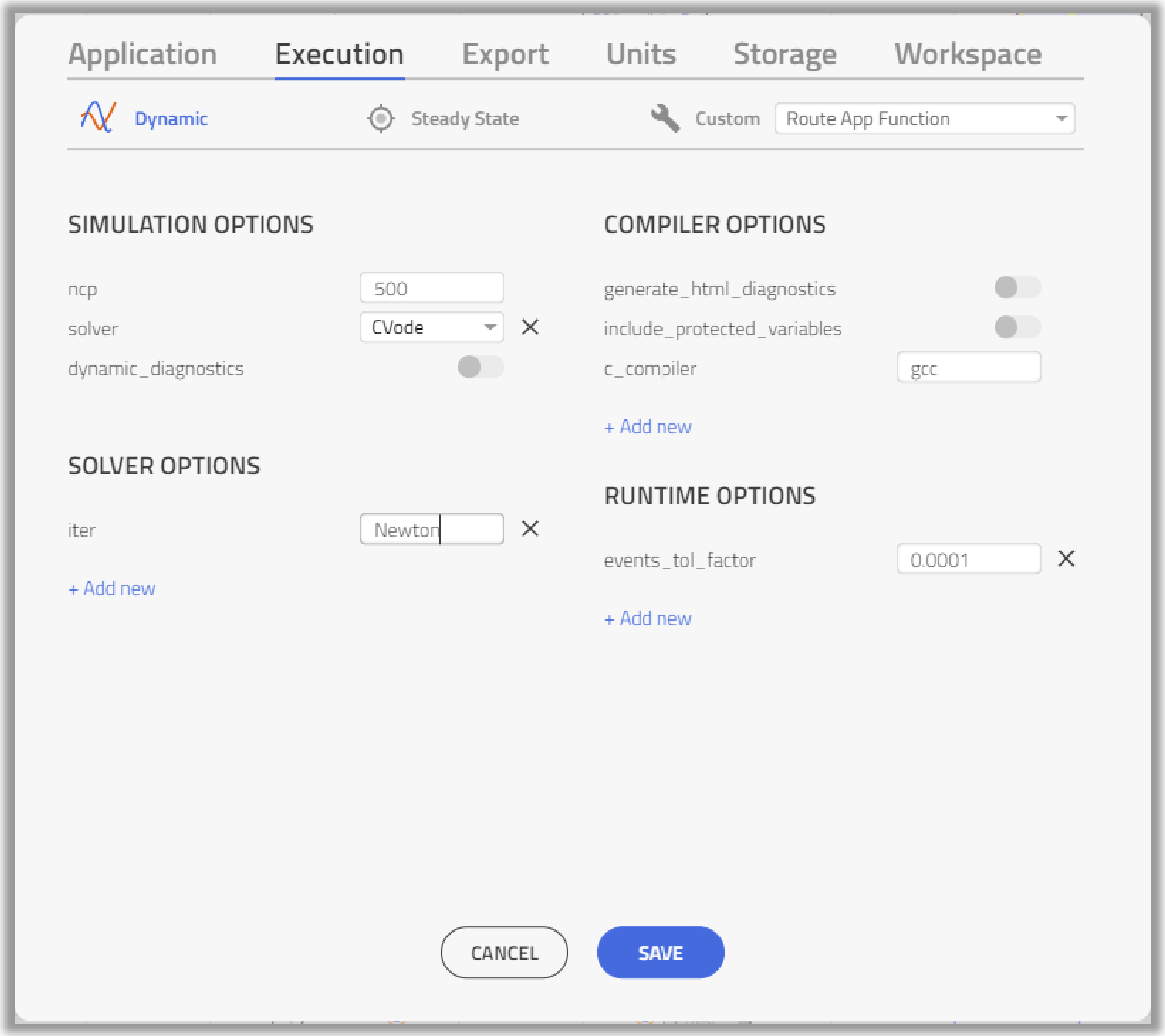


SETTINGS

- Found under ⚙ or "Global settings…"



- Detailed information found in Help Center:
  - OPTIMICA Compiler Toolkit User's Guide



Modelon

SELECTING SOLVERS

- Should not rely on just one integration algorithm for simulation experiments.
  - Instead, some selected results should be checked using different integration algorithms to find out which works best.
    - CPU time
    - Stability
    - Accuracy

Modelon

SIMULATION BUTTON VISUAL FEEDBACK

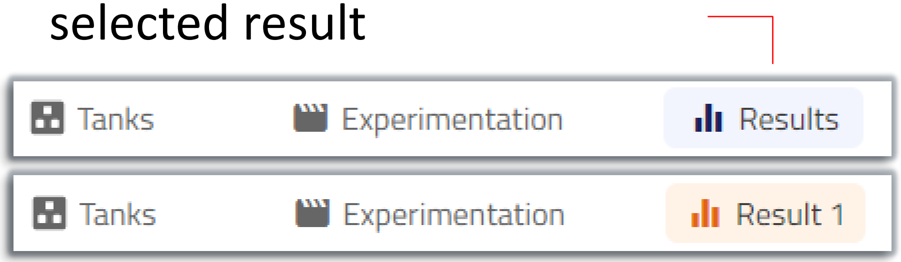
- Progression:

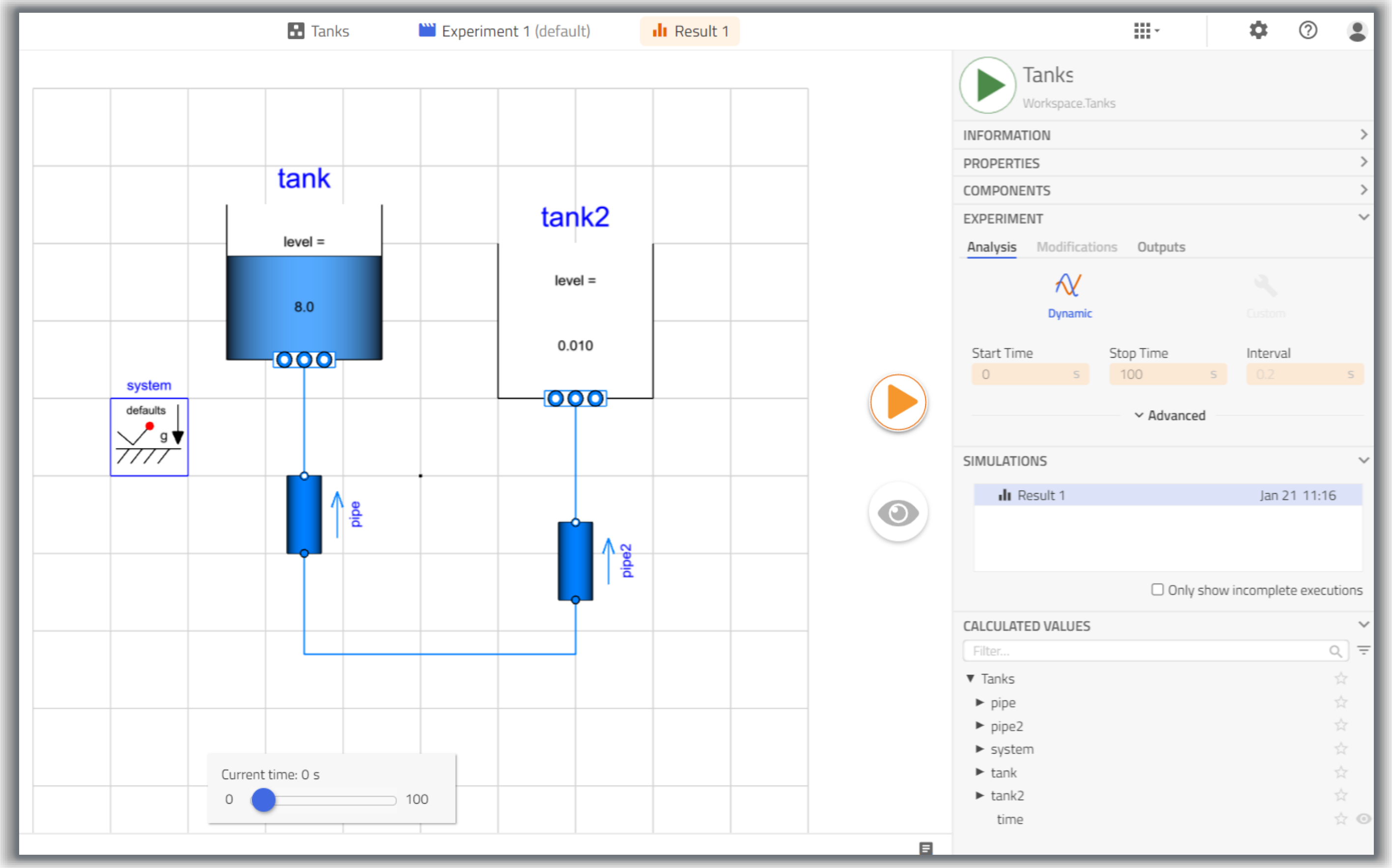
- Compiled model available:

- Result available for viewing:

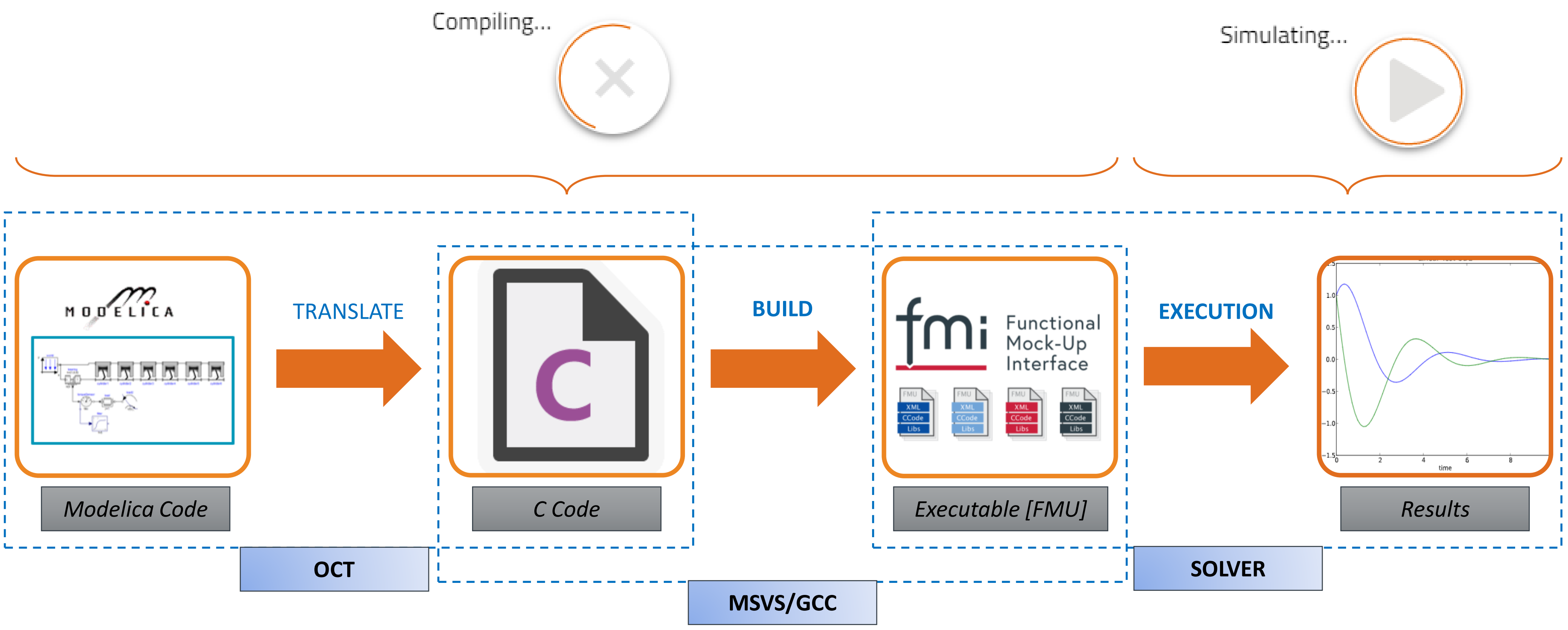
SIMULATION

- When simulation is finished, a result becomes available for analysis in the **Simulations** list.
- By default, the latest result is selected
- Variables organized according to components are listed in **Calculated Values**
- Once simulation is completed, the mode is automatically switched to Results and the name is updated based on the selected result





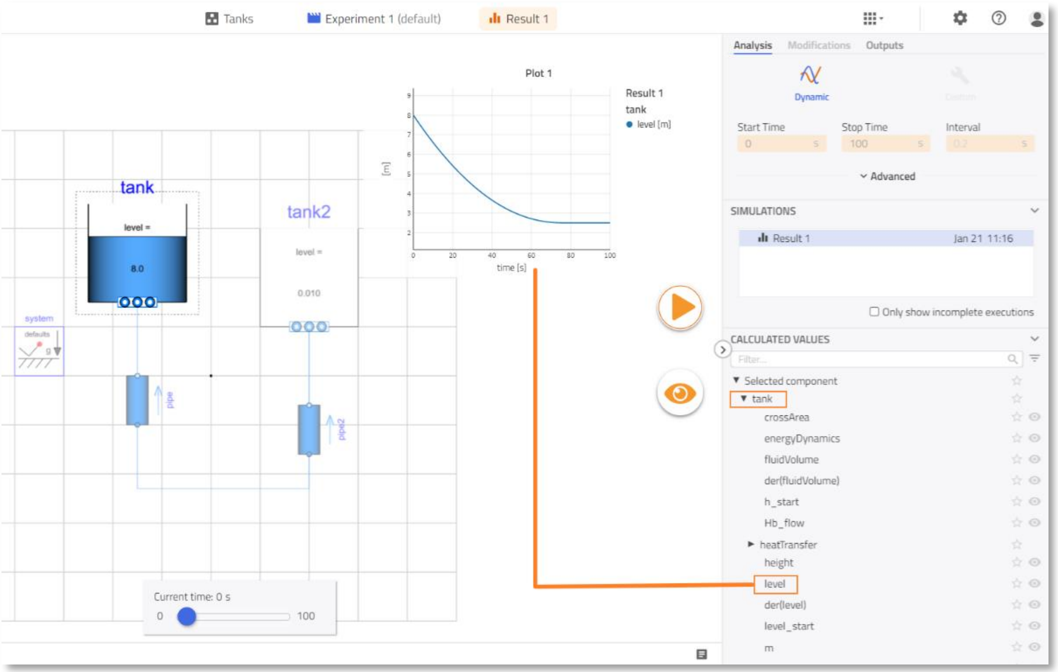
WHAT HAPPENS BEHIND THE SCENES



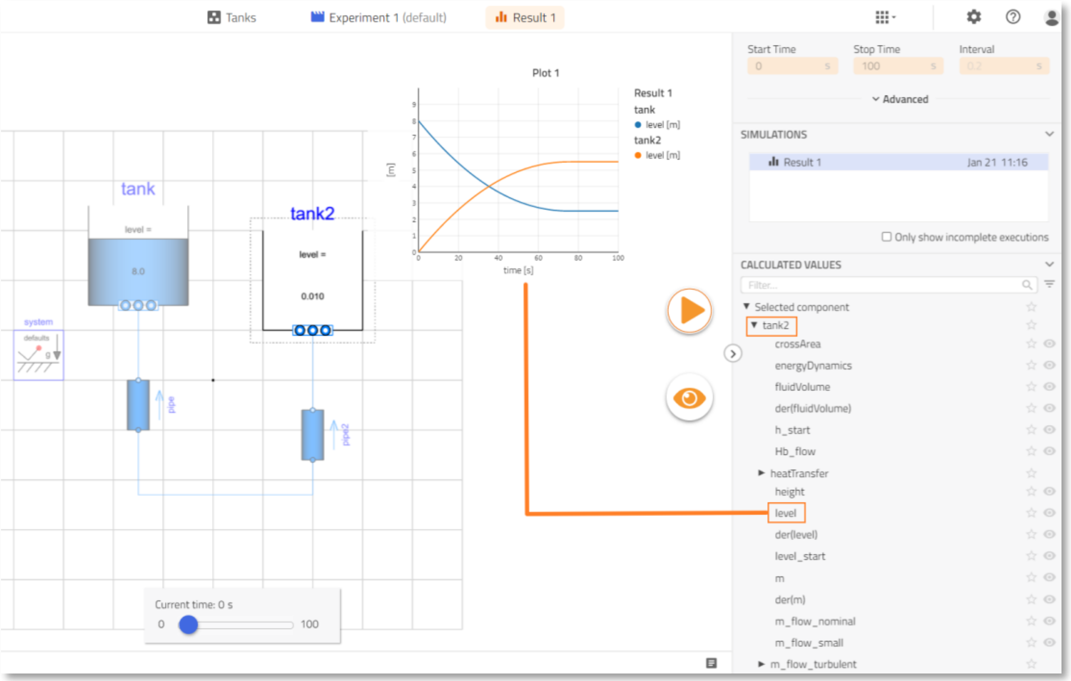
RESULTS

INSPECTING RESULTS: PLOTS

- To plot a variable, simply drag a variable on to the canvas from the variable list

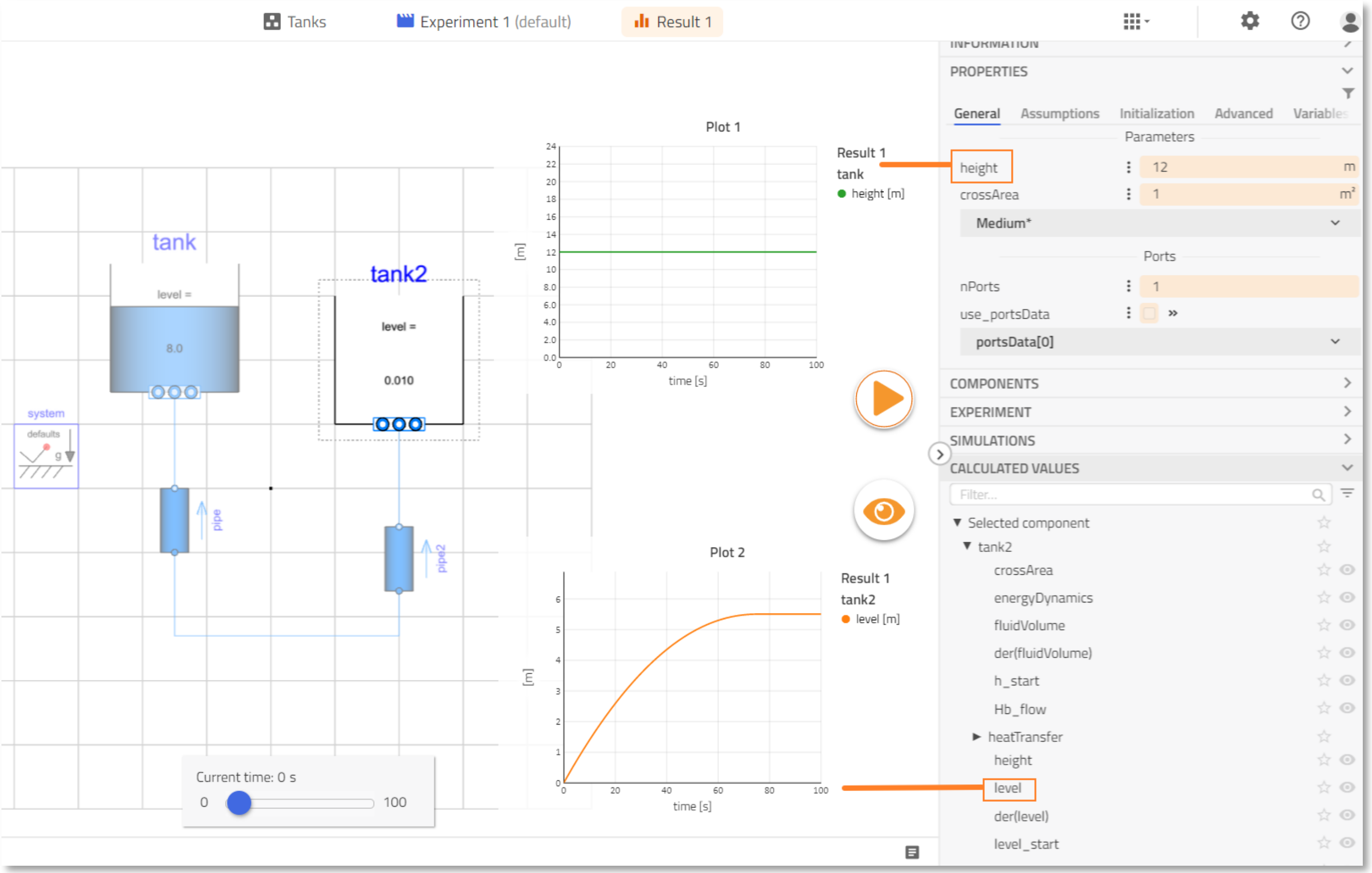


- Multiple variables can be dropped on the same plot for comparison



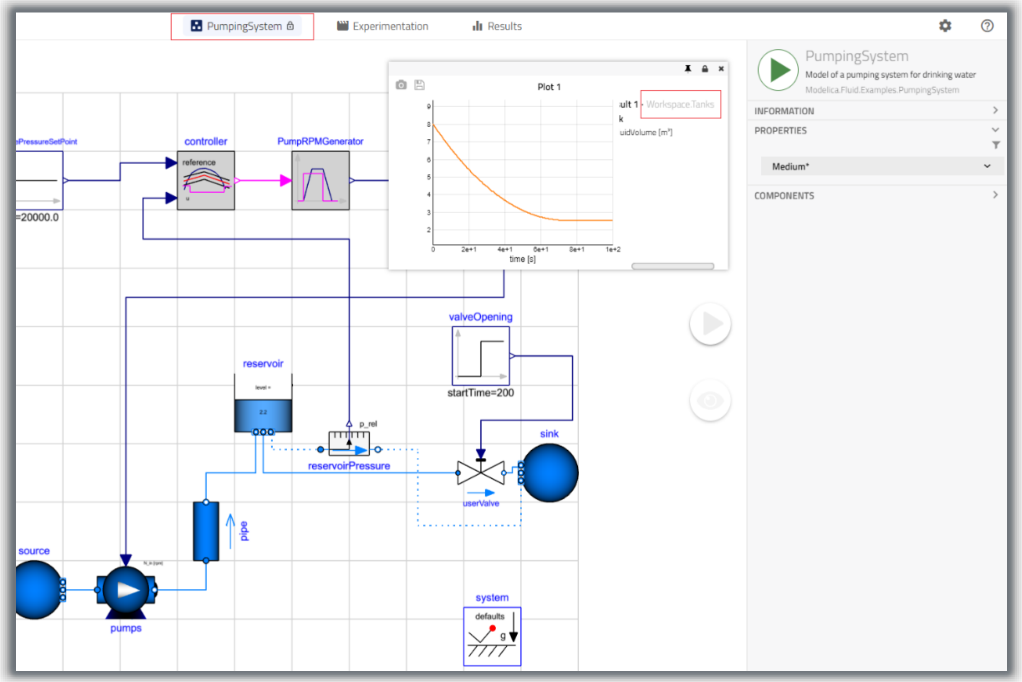
INSPECTING RESULTS: PLOTS

- Parameters and variables can also be dragged to be plotted on the canvas

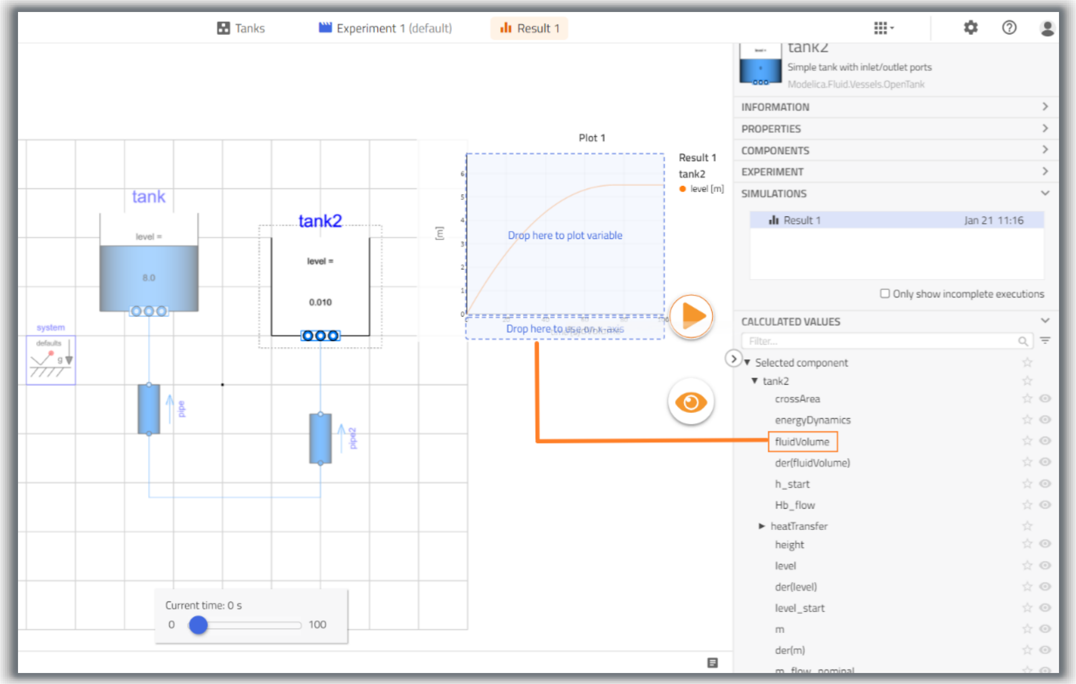


INSPECTING RESULTS: PINNED AND X-Y PLOTS

- Plots can be pinned such that they remain on the canvas even if the model is switched
  - Useful for comparing results between models



- Independent variable can be changed in order to generate X vs. Y or parametric plots

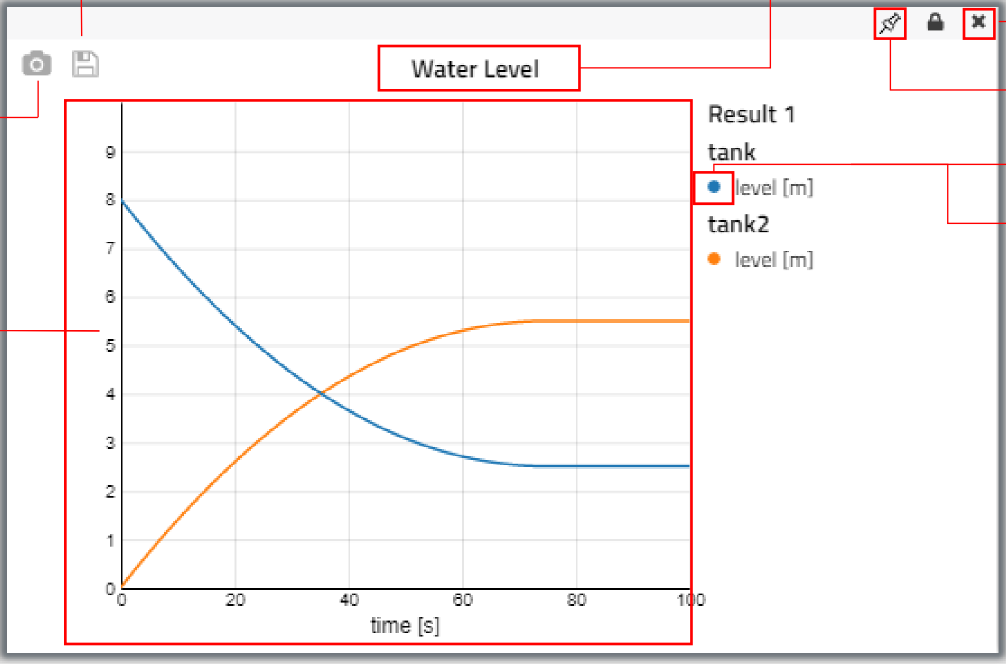


INSPECTING RESULTS: PLOTS - WINDOW FUNCTIONALITY

Download plot

- CSV
- PNG

Zooming capabilities



Rename plot

Remove plot

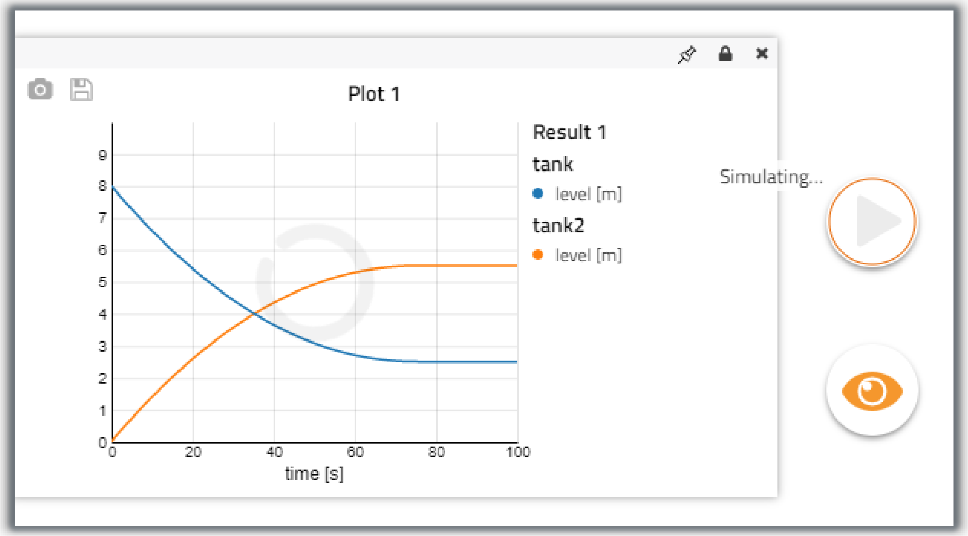
Pin plot

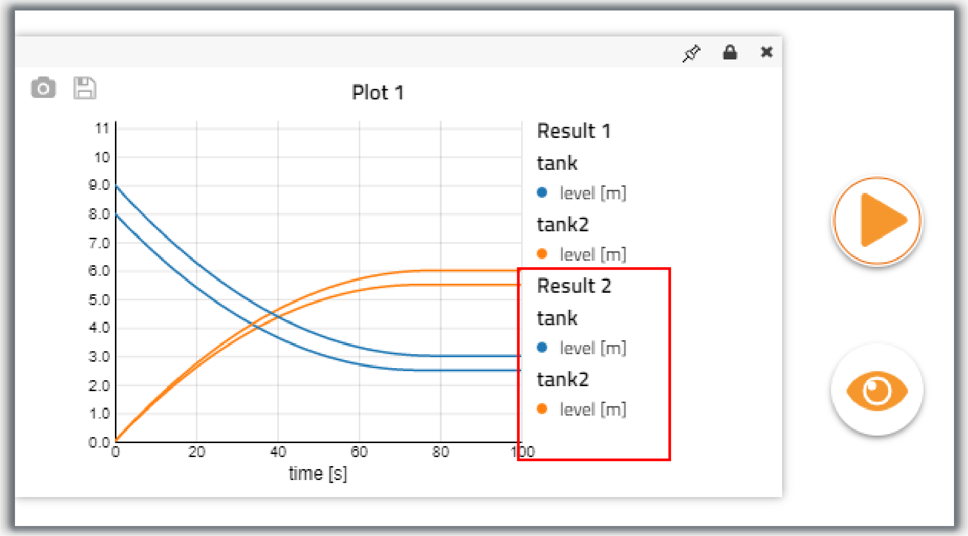
Edit color

Remove trajectory

INSPECTING RESULTS: PLOTS – COMPARING RESULTS

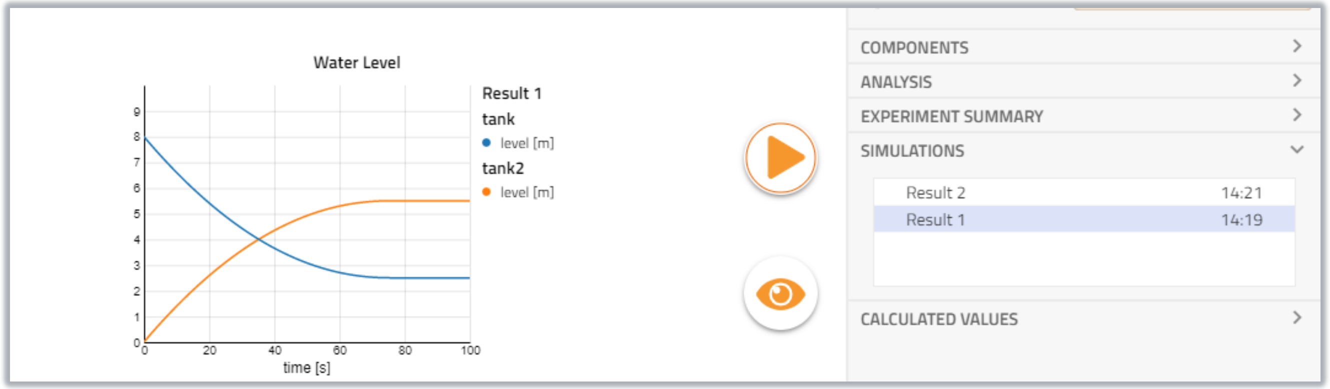
- When running a new simulation with the plot window open, new results will automatically be displayed in the plot

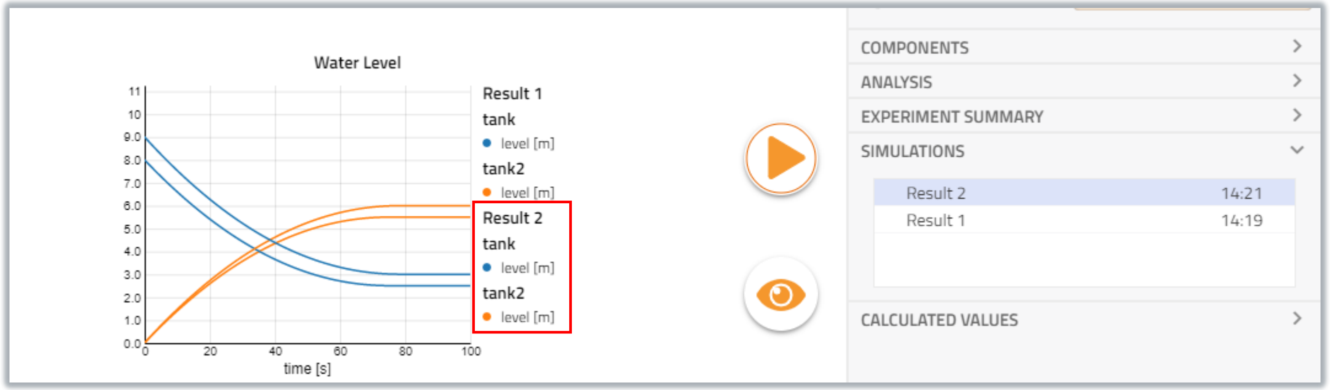




INSPECTING RESULTS: PLOTS – COMPARING RESULTS

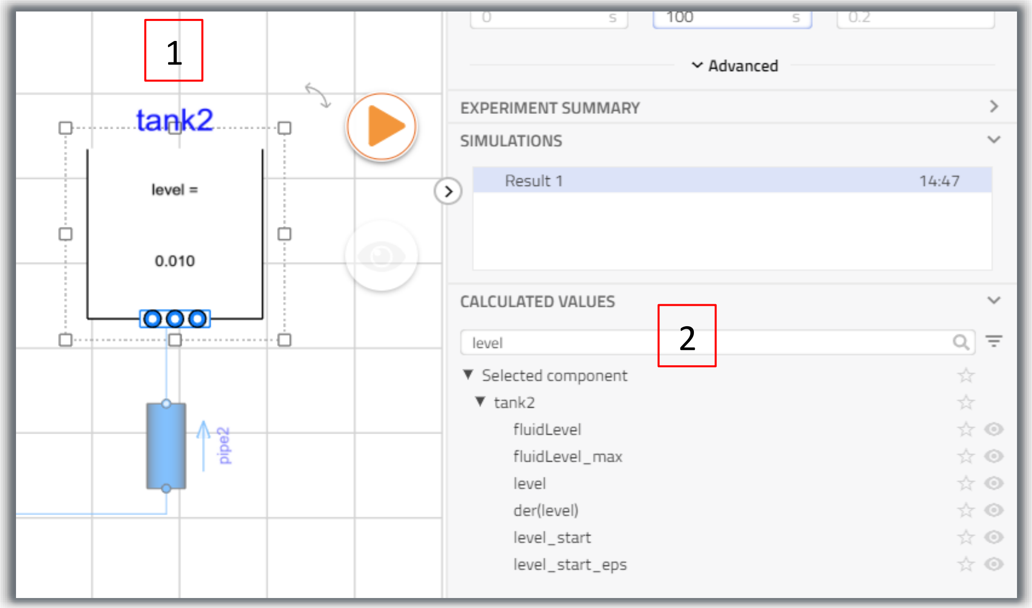
- Compare results by selecting another result while having a plot window open

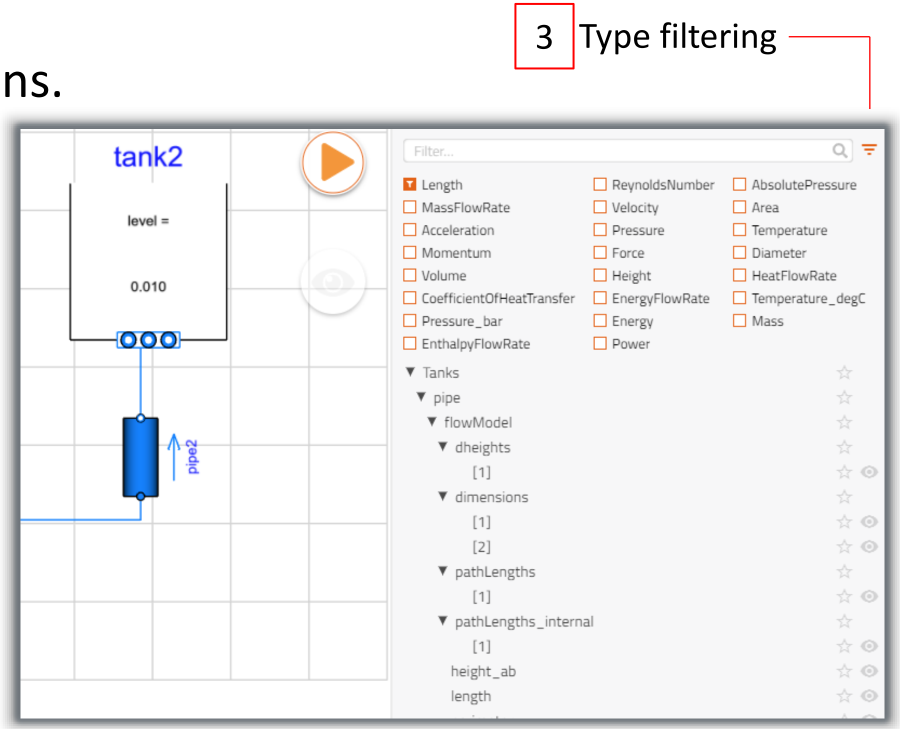




INSPECTING RESULTS: FILTERING

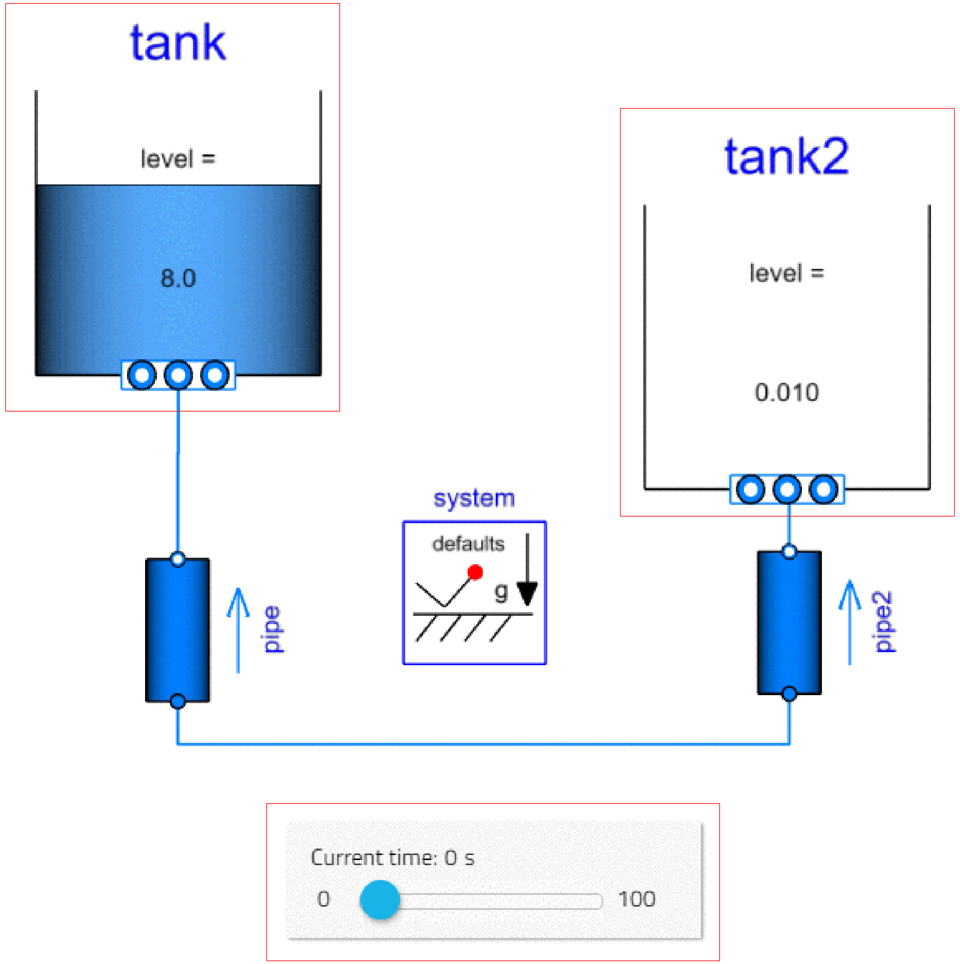
- Find variables by filtering. Three options exist:
  1. Selecting component in canvas
  2. Text filtering
  3. Type filtering
- Filters are additive, only giving results matching all conditions.



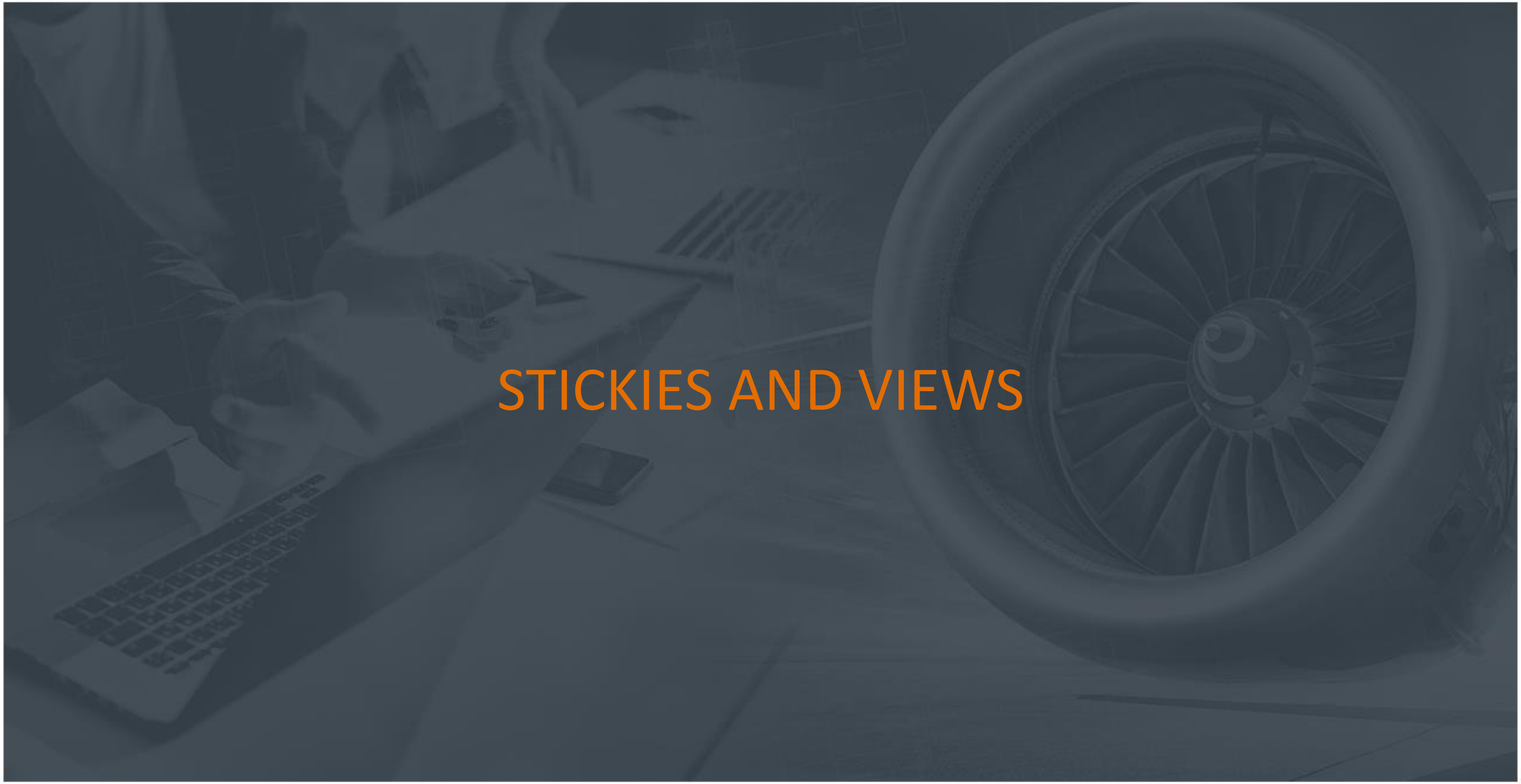


INSPECTING RESULTS: MODEL ANIMATION

- Certain models support visualization (ex. *OpenTank* from MSL)
- Model changes appearance when sliding the time-slider



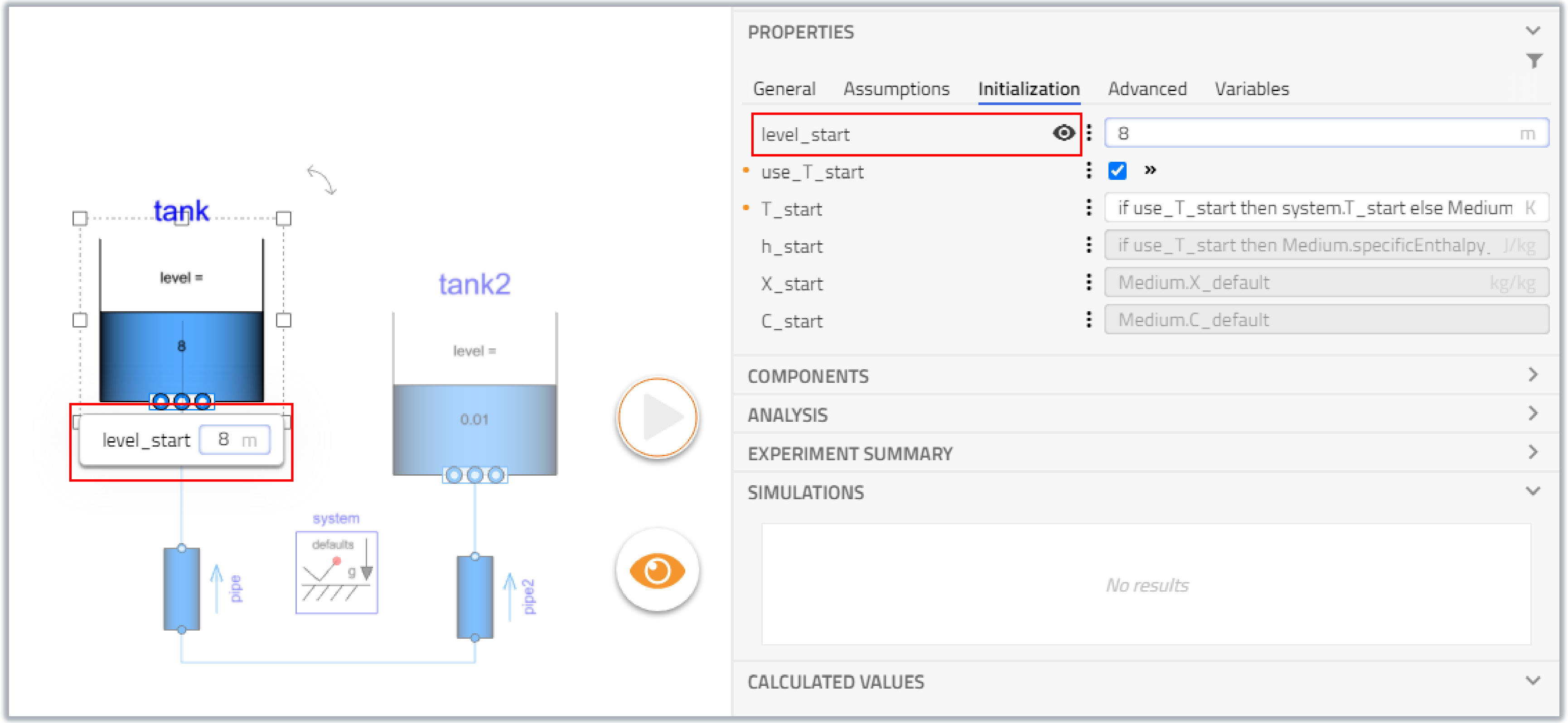
Modelon



STICKIES AND VIEWS

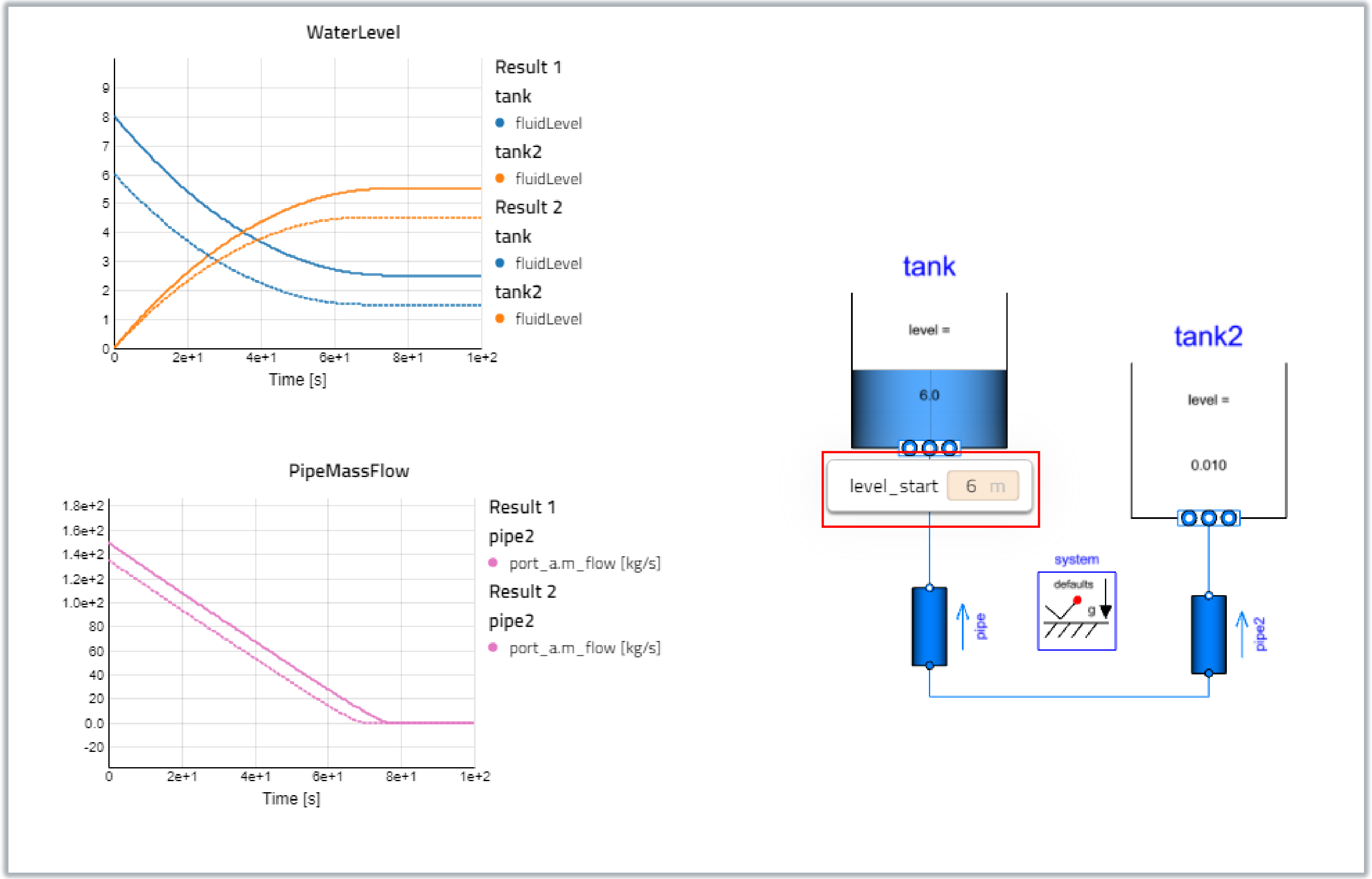
STICKIES

- Stickies are small widgets that can be added to the model canvas for interaction with the model
- Stickies are added with the -button in the control panel
- Modelon Impact-unique feature



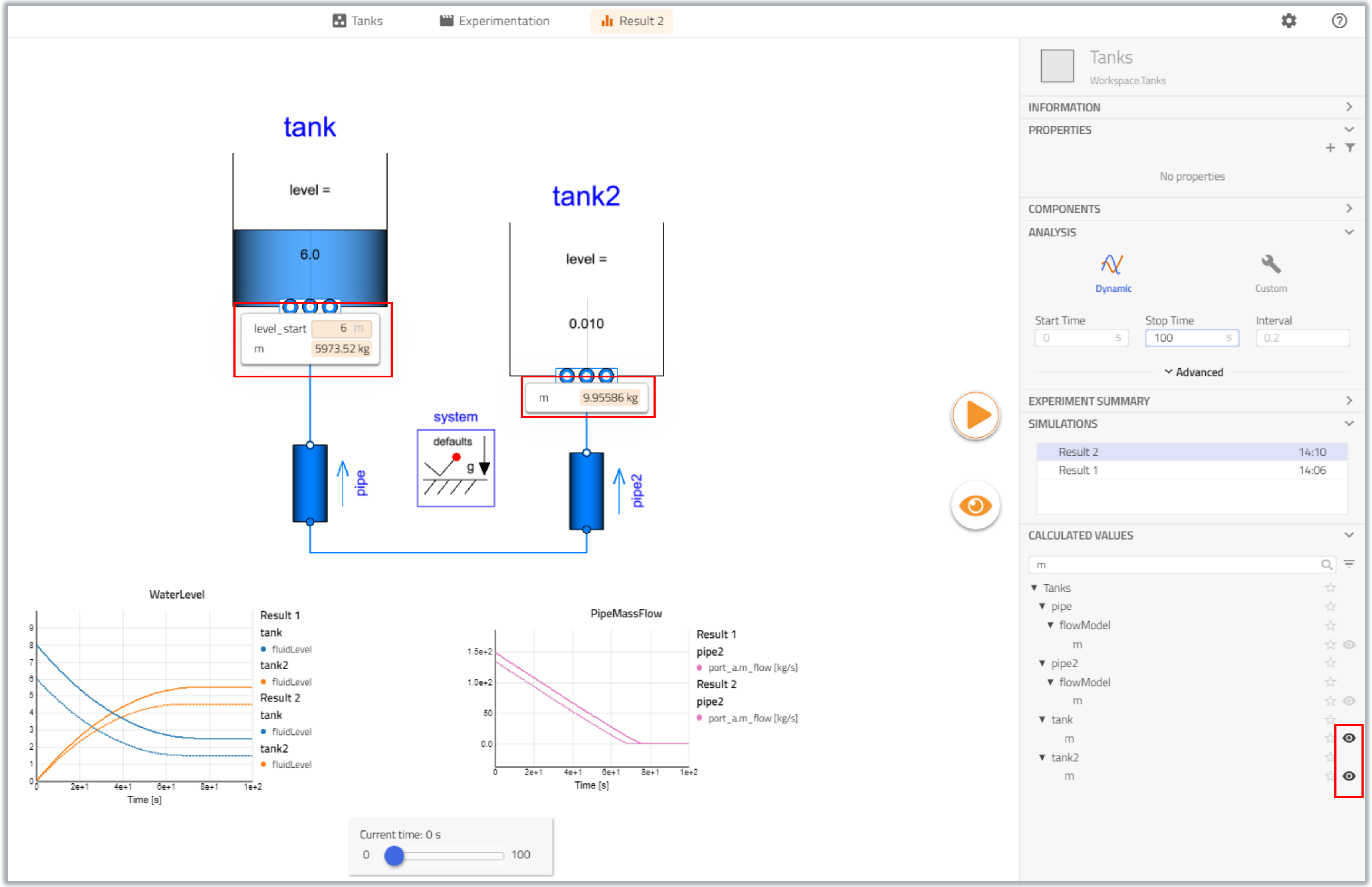
STICKIES

- Can be used to provide input to the model
- Equivalent to parameter input in control panel



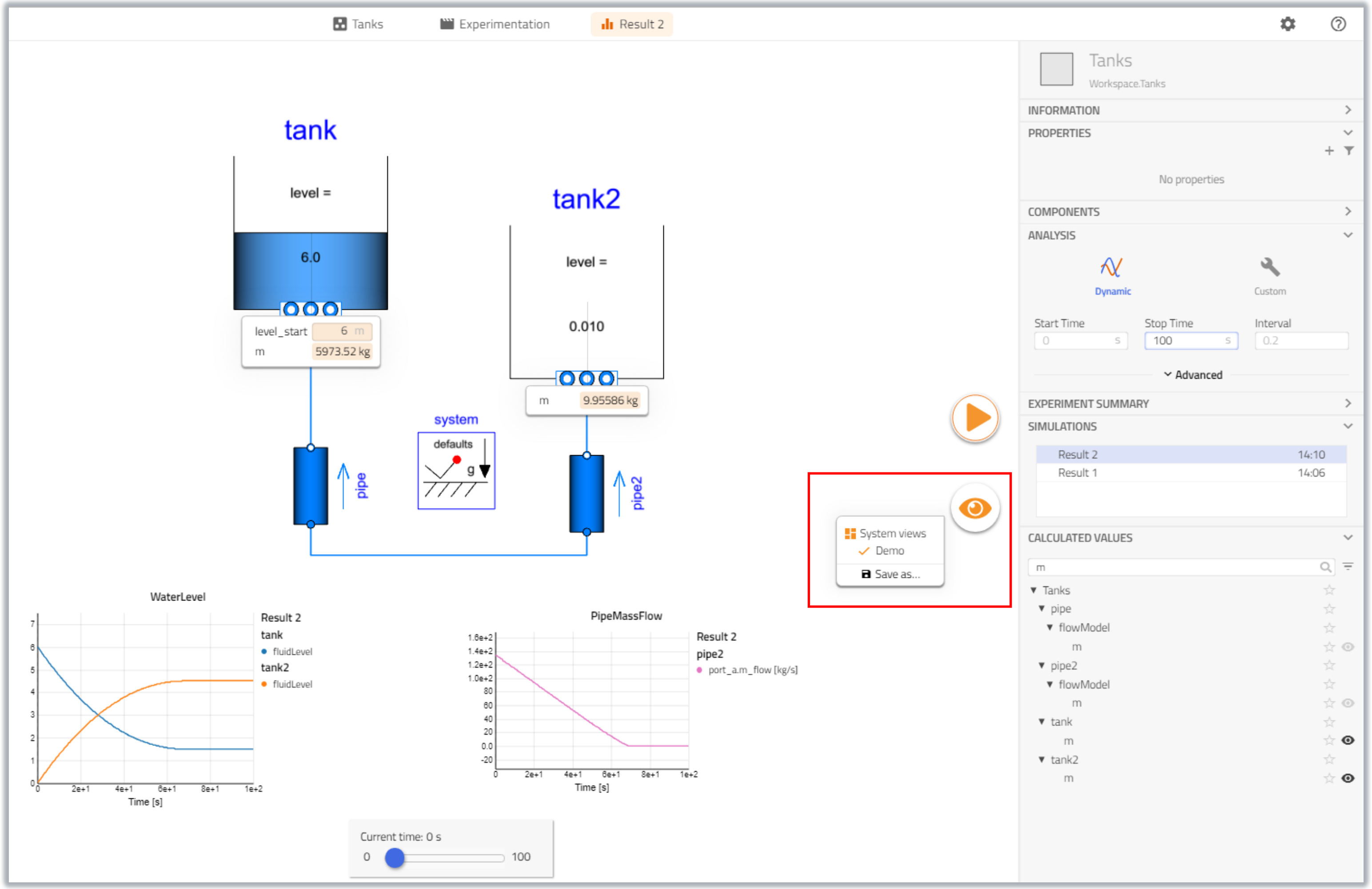
STICKIES

- Can be used to display simulation results
- Moving the slider will update the stickies
- Both parameters and variables can be displayed in the stickies



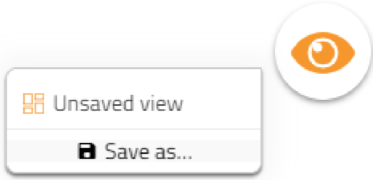
VIEWS

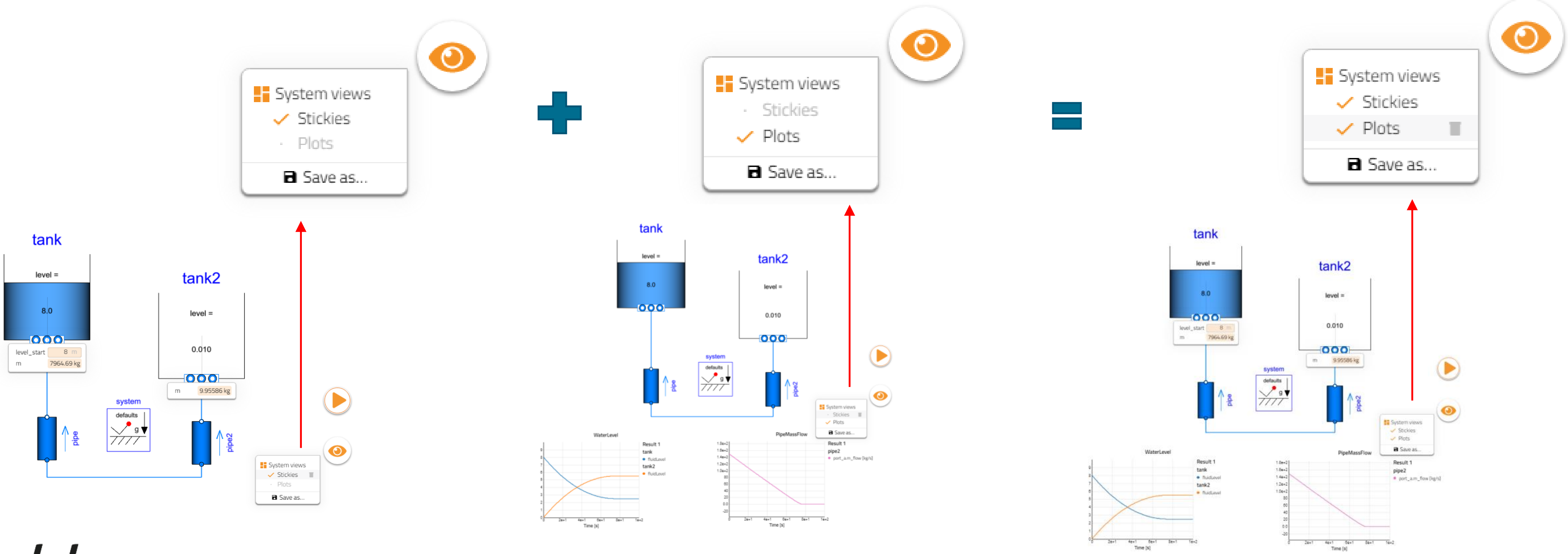
- The views are used to save a set of plots and stickies for a model
- This includes what variables are plotted/shown in stickies as well as the positioning of plots and stickies
- The views are saved in the context of the active workspace for the workspace package and global libraries.



VIEWS

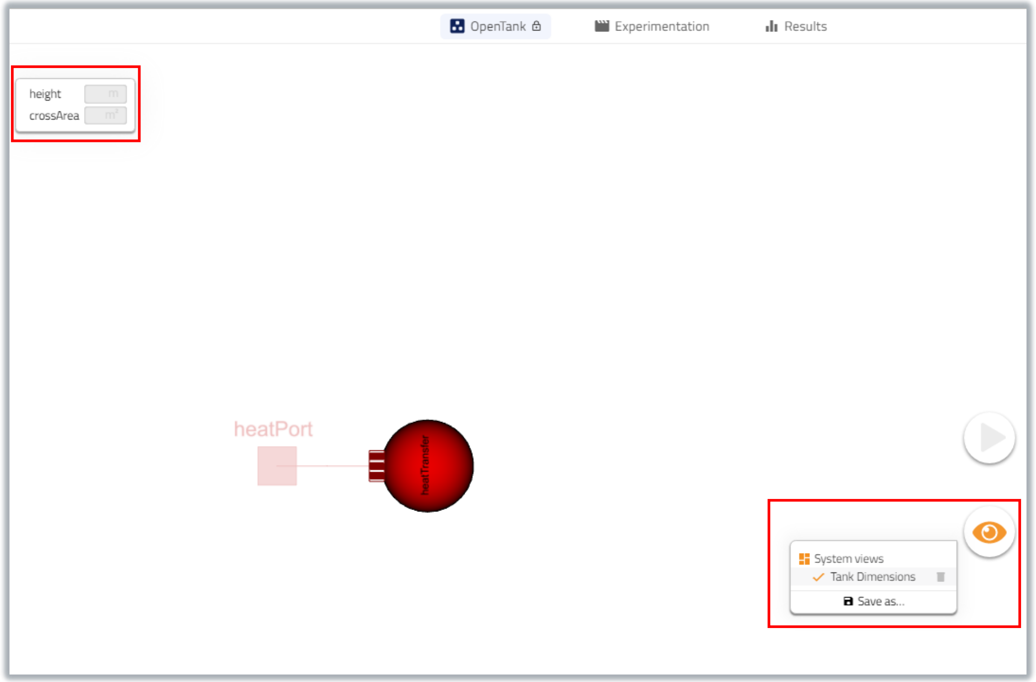
- Save a view to a model via the -button (press enter)
- At least one sticky or plot needs to be present
- Can combine several saved views

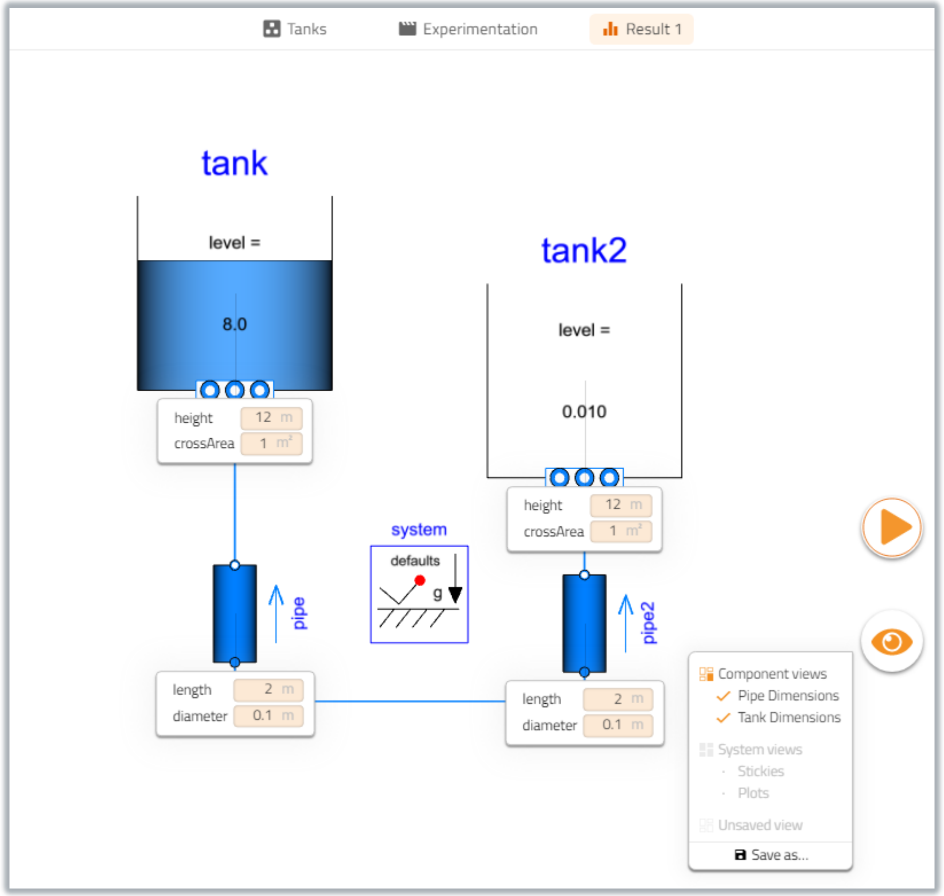




VIEWS

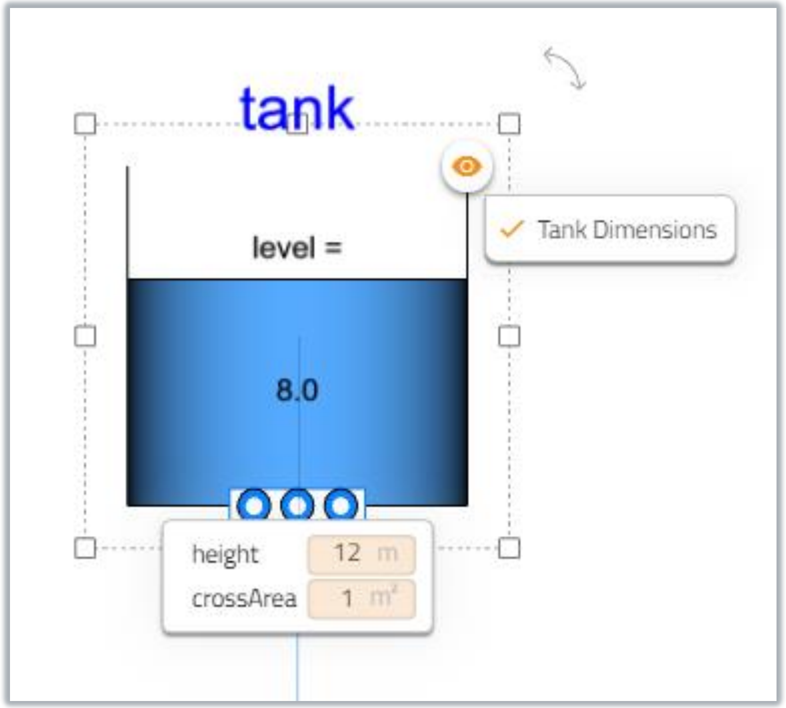
- Views are saved with the model and can be reused when it is used as a component in a system model
- Referred to as “Component views“ in the system model
- Only within workspace
- Works both for global libraries and workspace packages





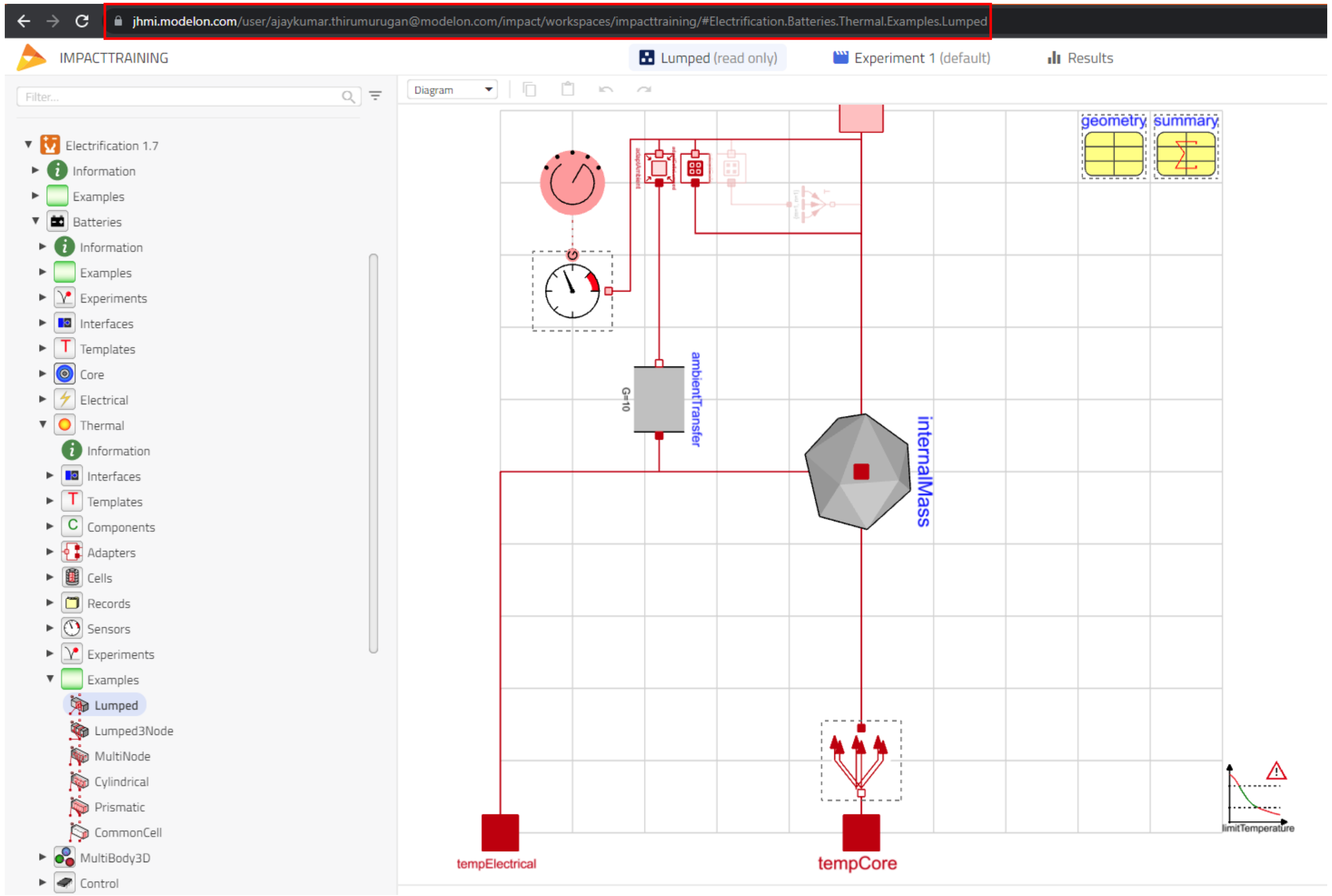
VIEWS

- Component views can be activated/deactivated directly from the canvas if selected



VIEWS

- Note that every level in a hierarchical model has its own URL
  - Example shows a link to a sub-component
- Can be shared with colleagues to point them directly towards a specific part of a model
- Combine with views to point users to specific places of interest in a model



WORKSHOP 1.2

In this workshop you will:

- Build a heating system
- Set boundary conditions
- Simulate
- Create stickies and views