Universal System Simulation via Hypergraphs

John Morris

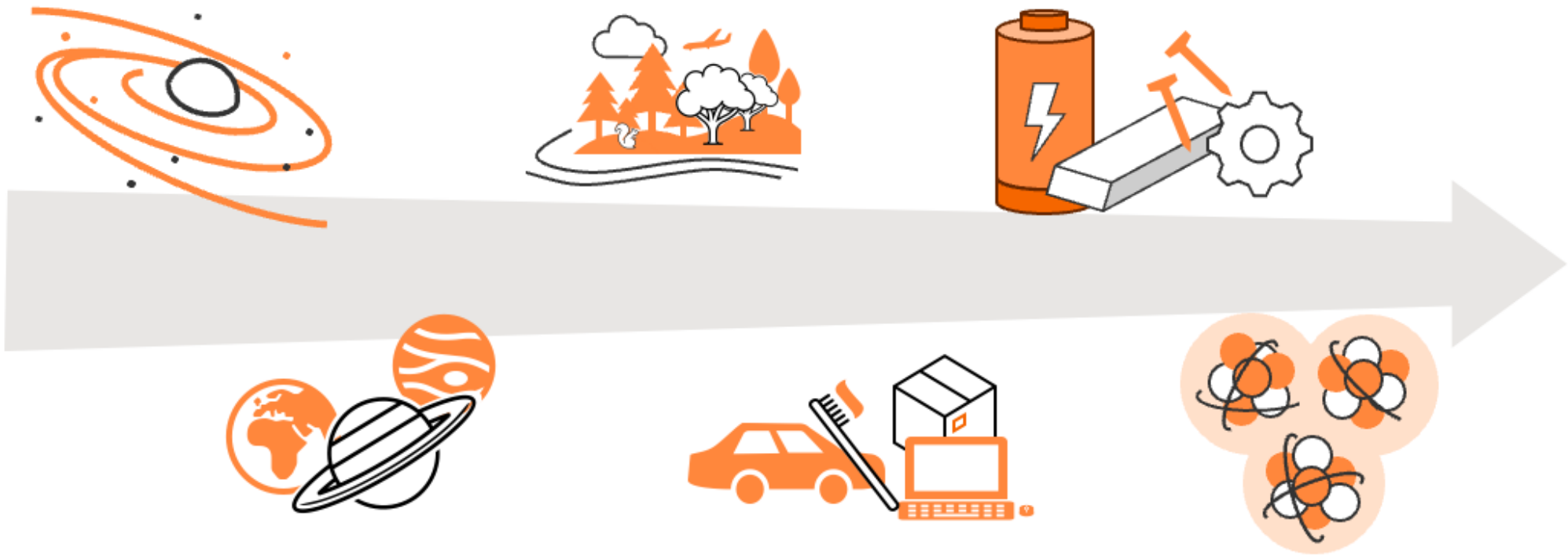
Dept. of Mechanical Engineering
Clemson University

PLMC

CEDAR
Clemson Engineering Design Applications and Research

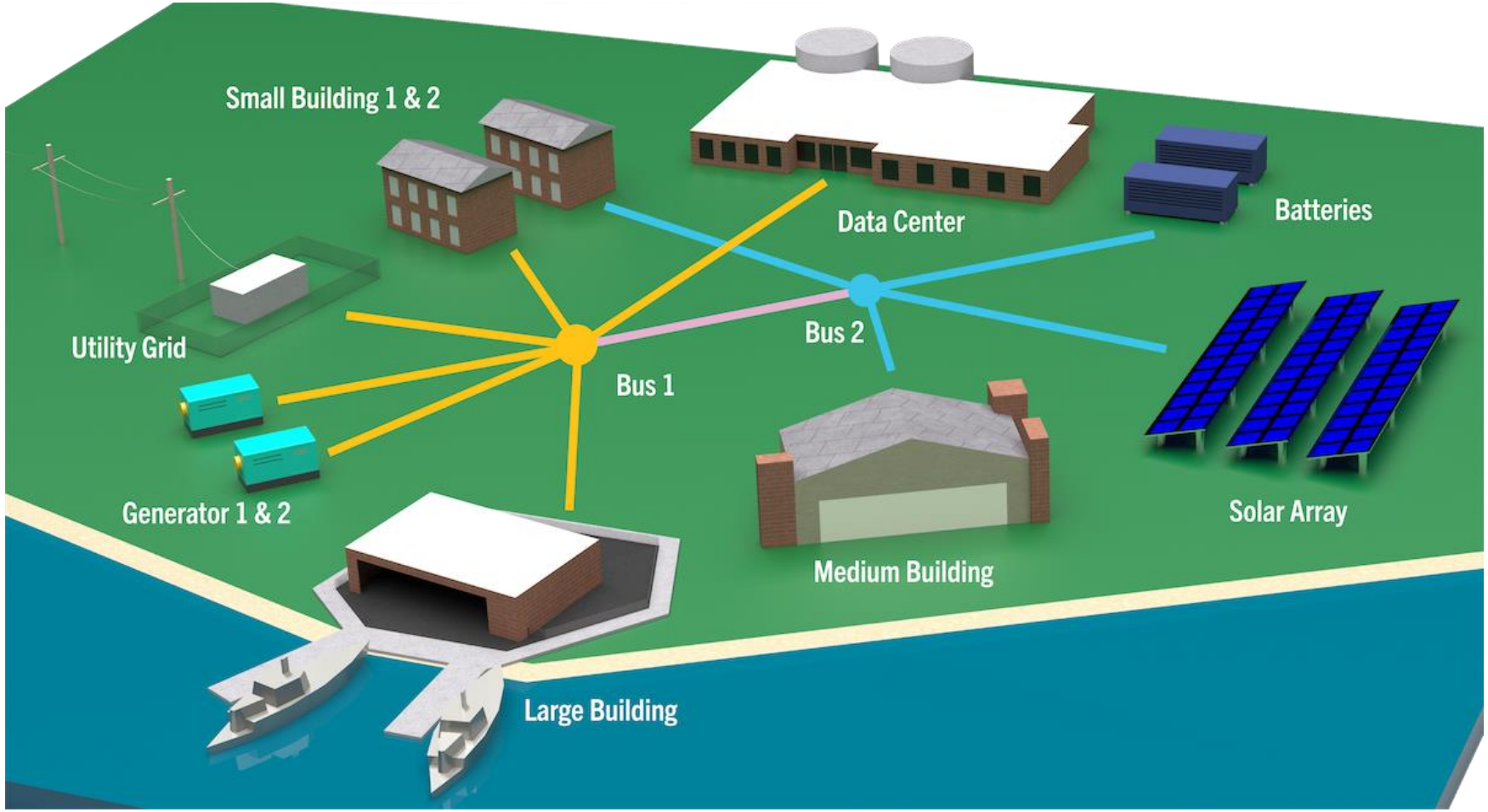
CLEMSON

Science is the work of characterizing the systems around us

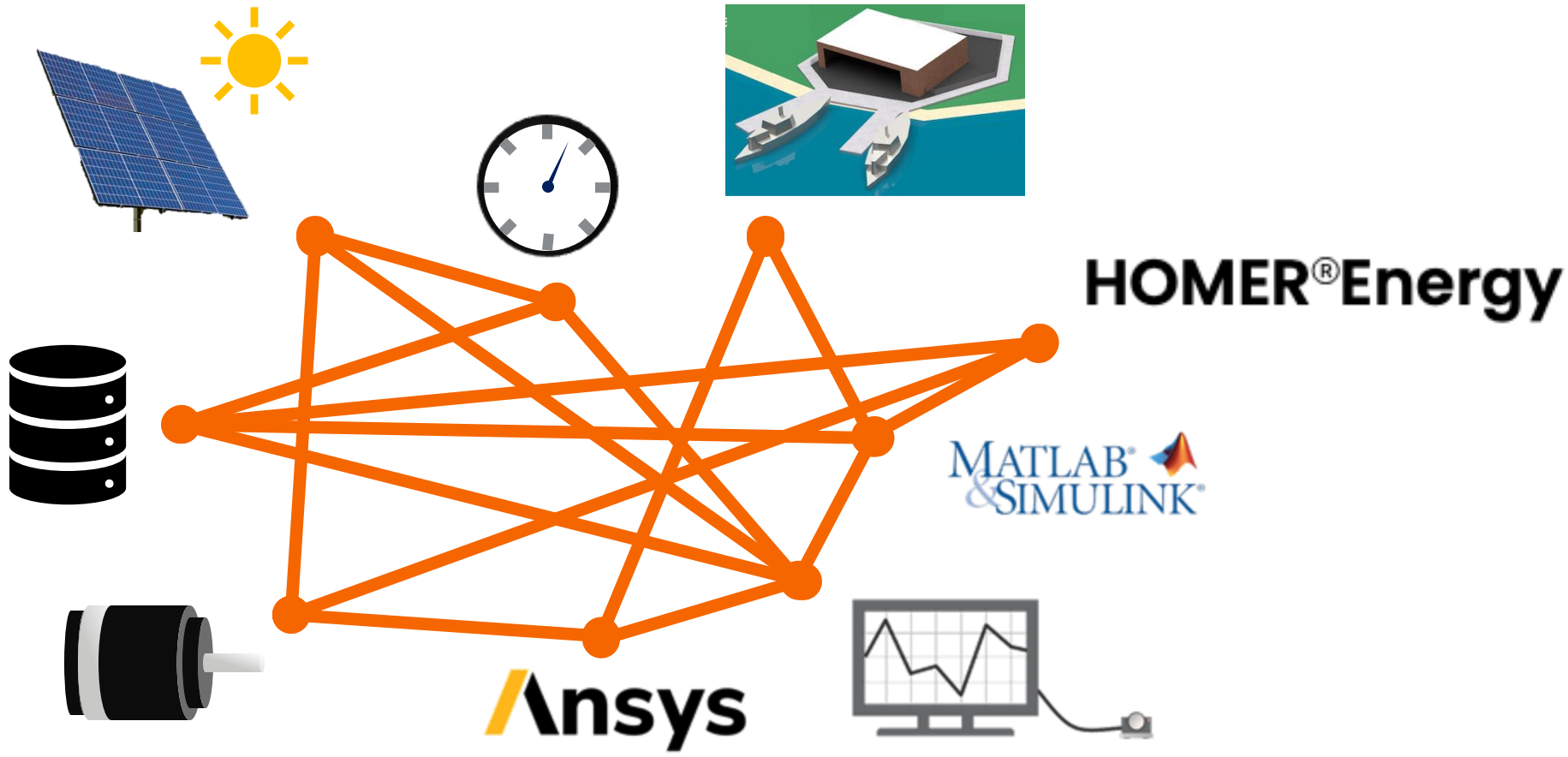


Goal of an agent is to observe facts about a system (its state)

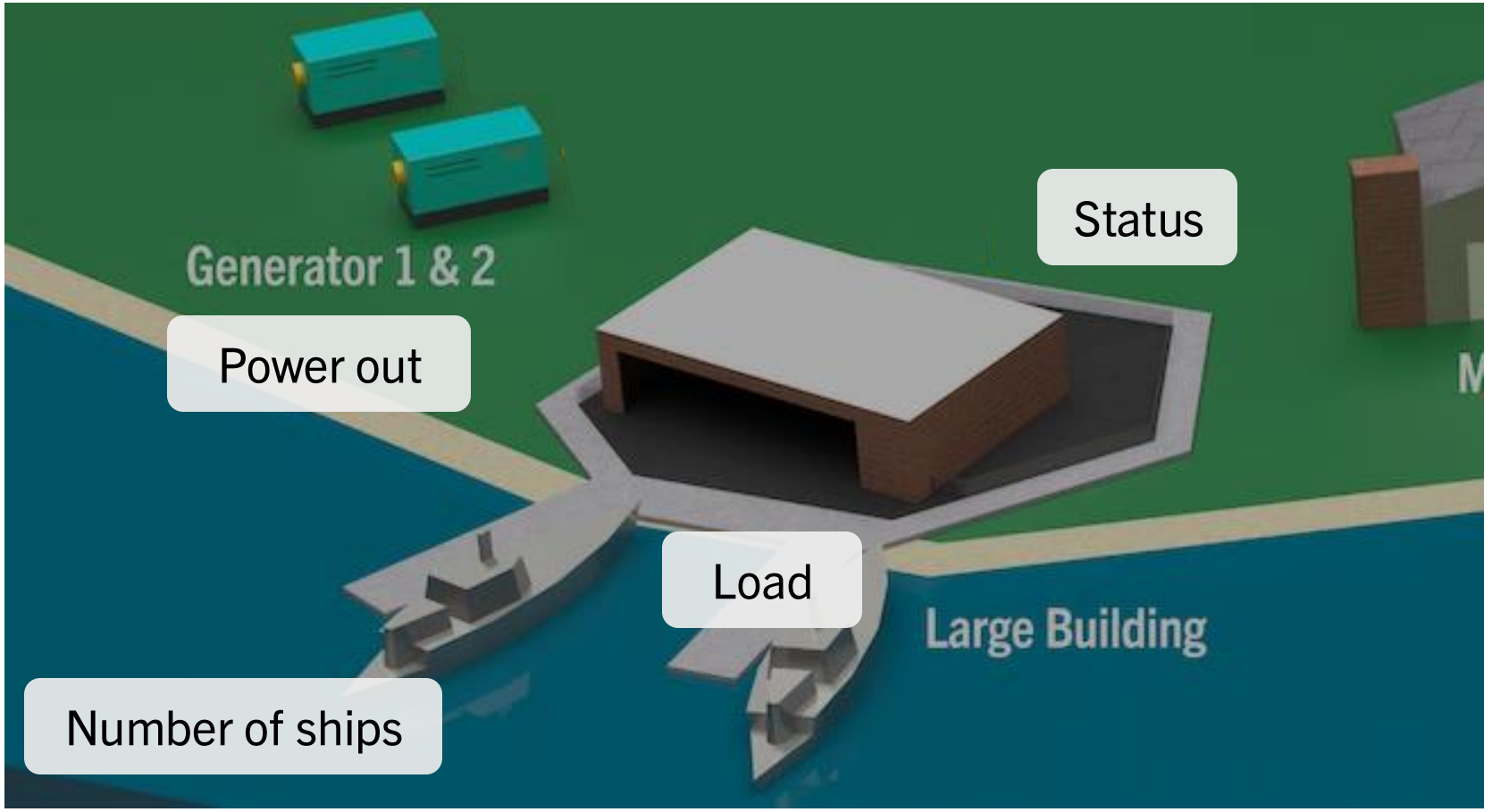
Example: Naval Microgrid



Complex systems requires models to be represented in different formalisms and software

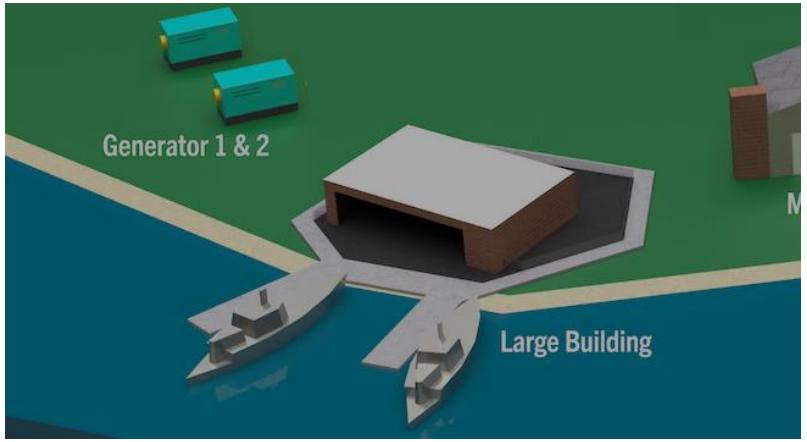


“A system is a collection of variables” (Ross Ashby)



Ashby, W. R. (1956). An Introduction to Cybernetics. Chapman & Hall.

States are sets of values representing specific properties

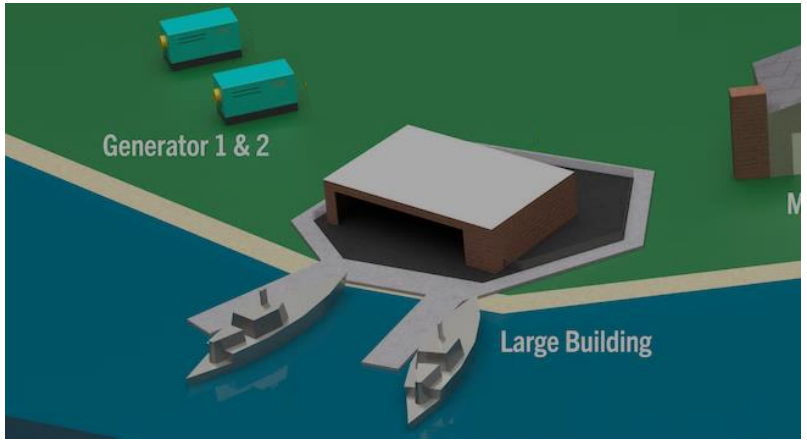


Number of ships	Load	Power out	Status

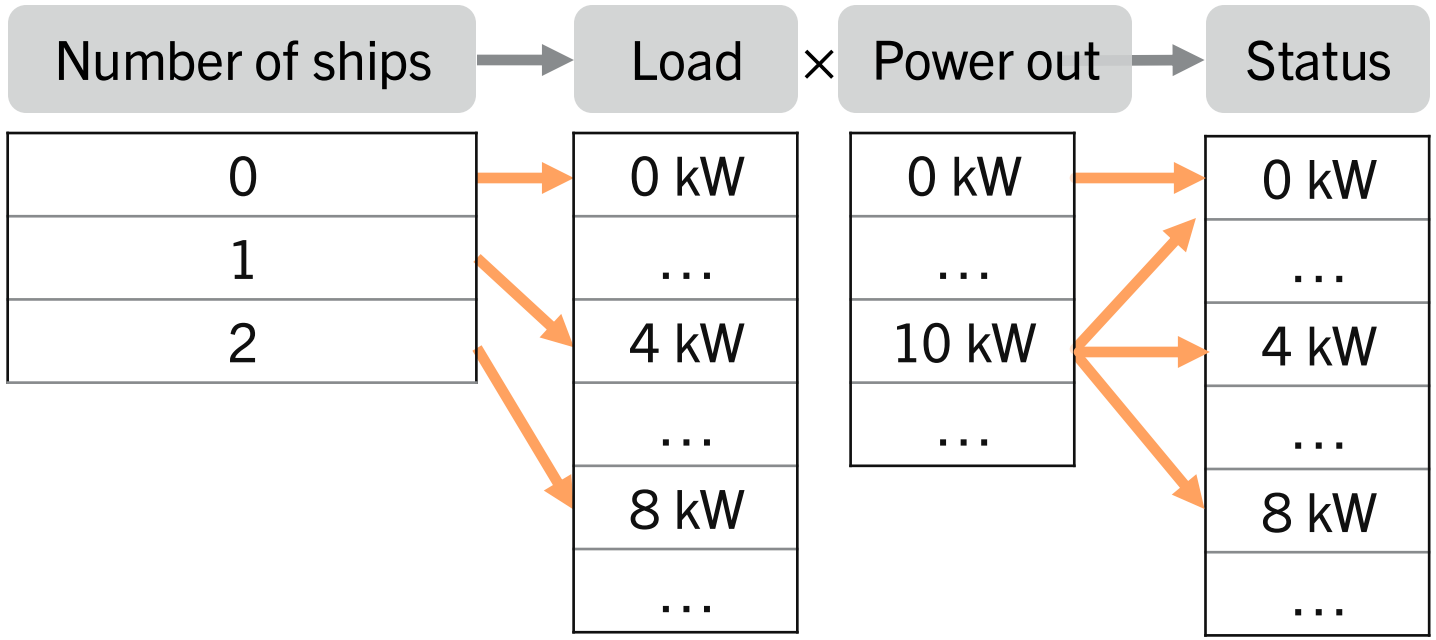
$$\mathbb{N}_2 \times \mathbb{R} \times \mathbb{R}_{\geq 0} \times \mathbb{R}$$

Polderman, J. W., & Willems, J. C. (1998). Dynamical Systems. In Introduction to Mathematical Systems Theory: A Behavioral Approach. Springer.

Behaviors are reductions of the set of possible configurations, forming a model

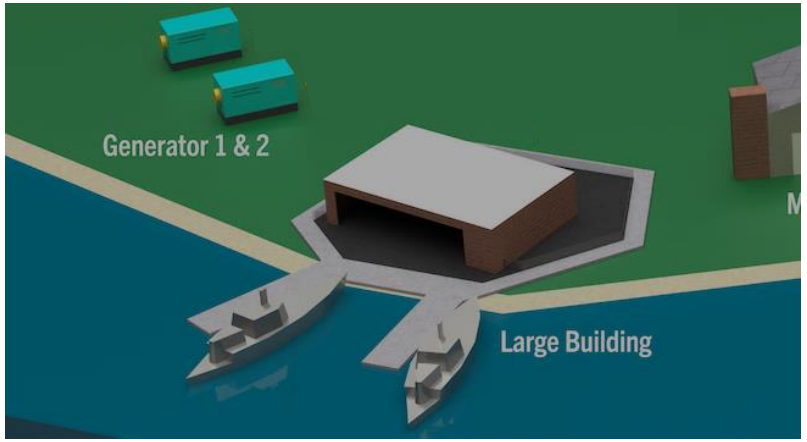


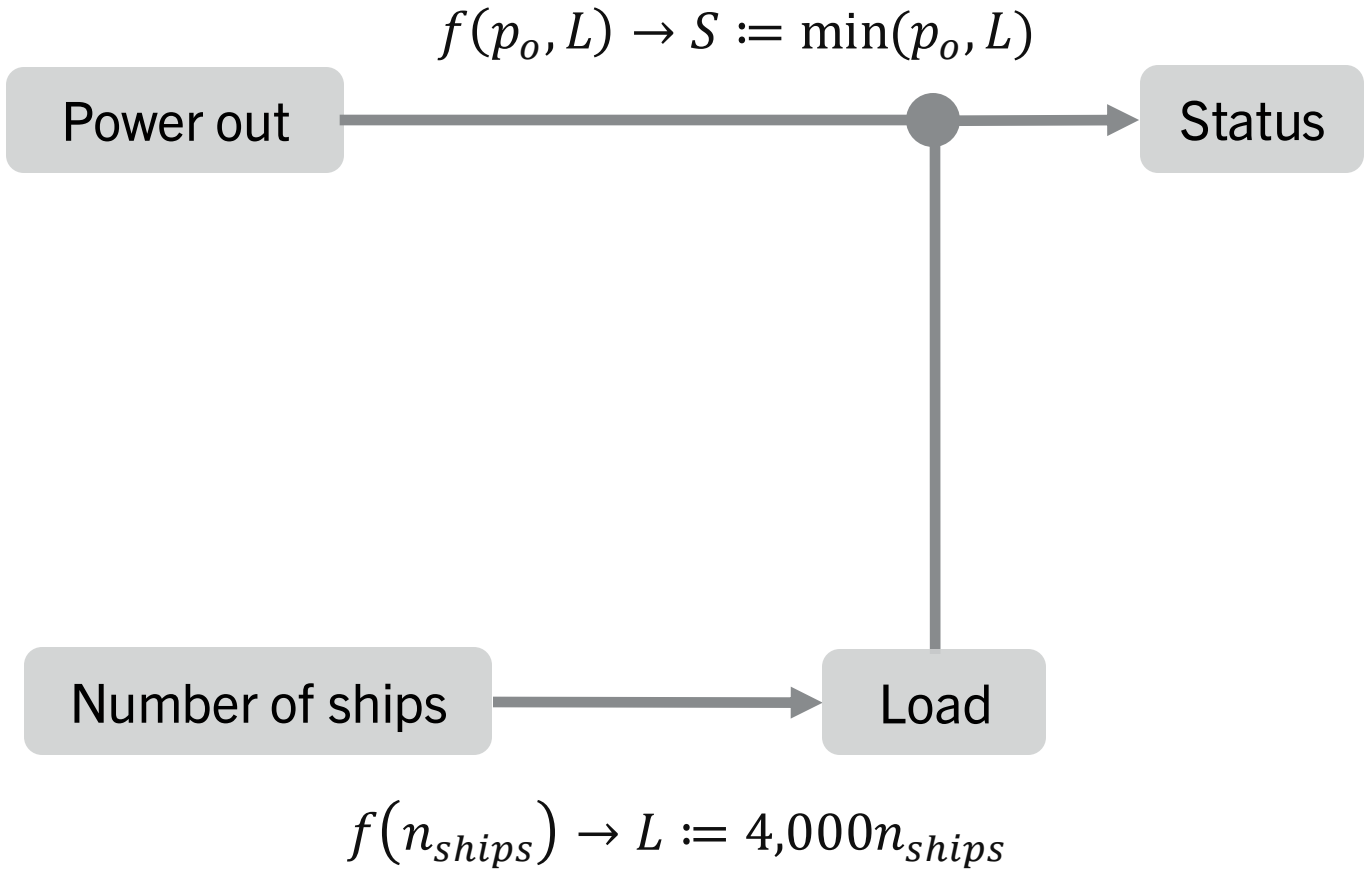
$f(n_{ships}) \rightarrow L \coloneqq 4{,}000 n_{ships}$ $\quad$ $f(p_o, L) \rightarrow S \coloneqq \min(p_o, L)$



Polderman, J. W., & Willems, J. C. (1998). Dynamical Systems. In Introduction to Mathematical Systems Theory: A Behavioral Approach. Springer.

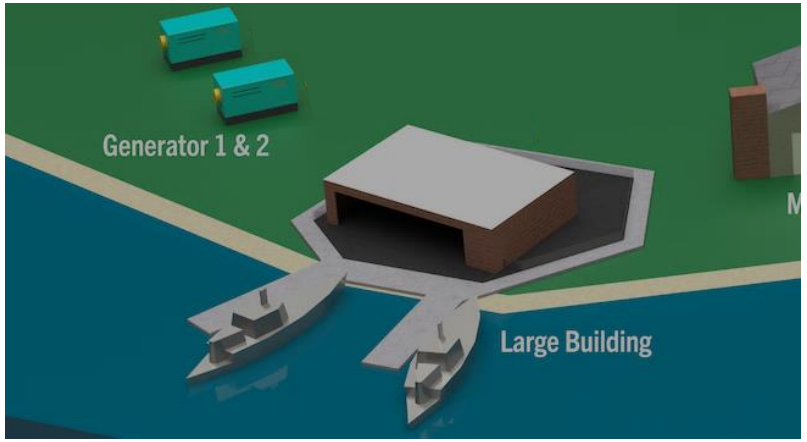
Simple (pragmatic) model of a system is a hypergraph





Schultz, P., Spivak, D. I., & Vasilakopoulou, C. (2019). Dynamical Systems and Sheaves (No. arXiv:1609.08086). arXiv. https://doi.org/10.48550/arXiv.1609.08086

There are 2 mechanisms for observing system states: coupling and simulation



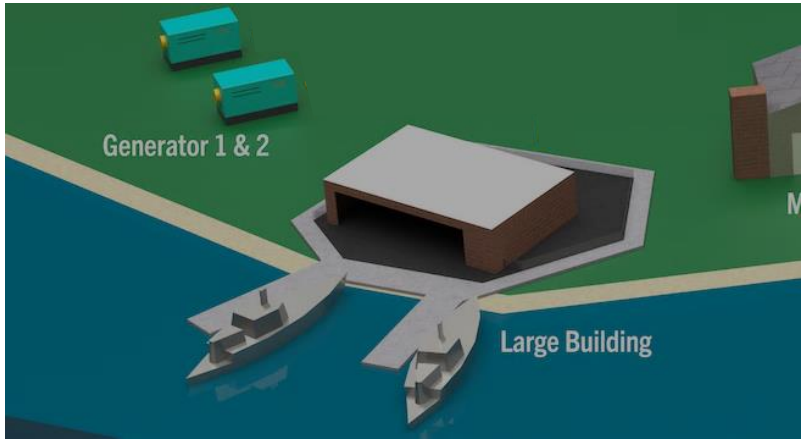
$$f(p_o, L) \rightarrow S \coloneqq \min(p_o, L)$$

Power out → Status

Number of ships → Load

$$f(n_{ships}) \rightarrow L \coloneqq 4{,}000 n_{ships}$$

Coupling: synchronizing the states of the system are with the observer



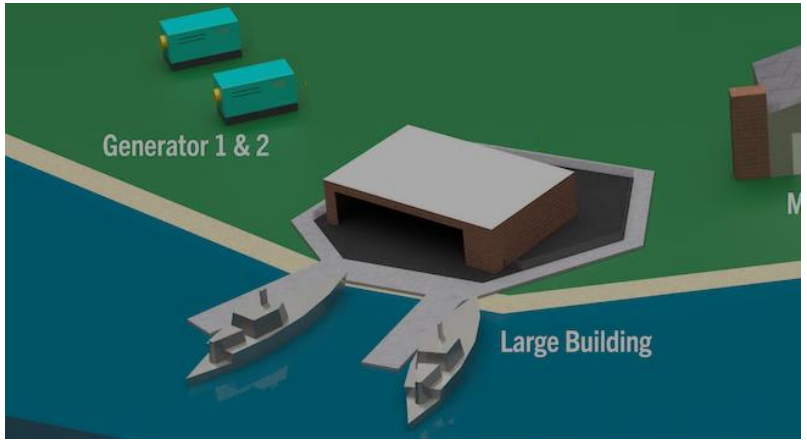
$$f(p_o, L) \rightarrow S \coloneqq \min(p_o, L)$$

Power out → Status

Number of ships → Load

$$f(n_{ships}) \rightarrow L \coloneqq 4{,}000 n_{ships}$$

Simulation: approximation by relating inputs to outputs



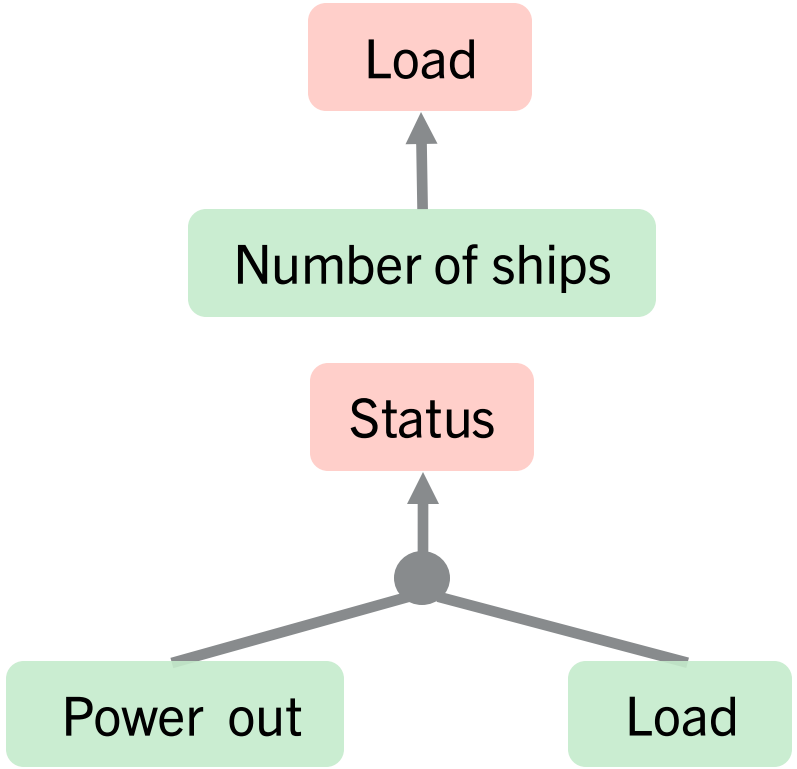
$$f(p_o, L) \rightarrow S \coloneqq \min(p_o, L)$$

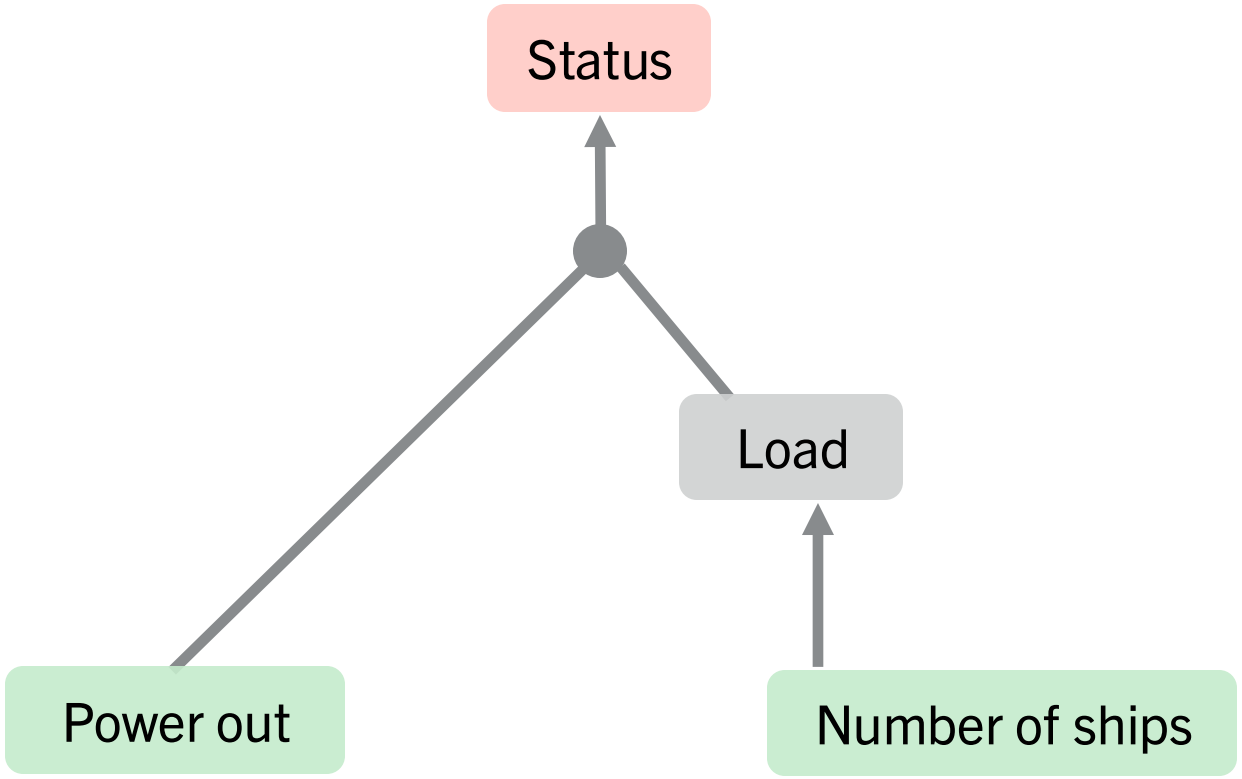
Power out → Status

Number of ships → Load

$$f(n_{ships}) \rightarrow L \coloneqq 4{,}000 n_{ships}$$

All possible simulations pairing inputs to outputs are given by the set of paths through the hypergraph

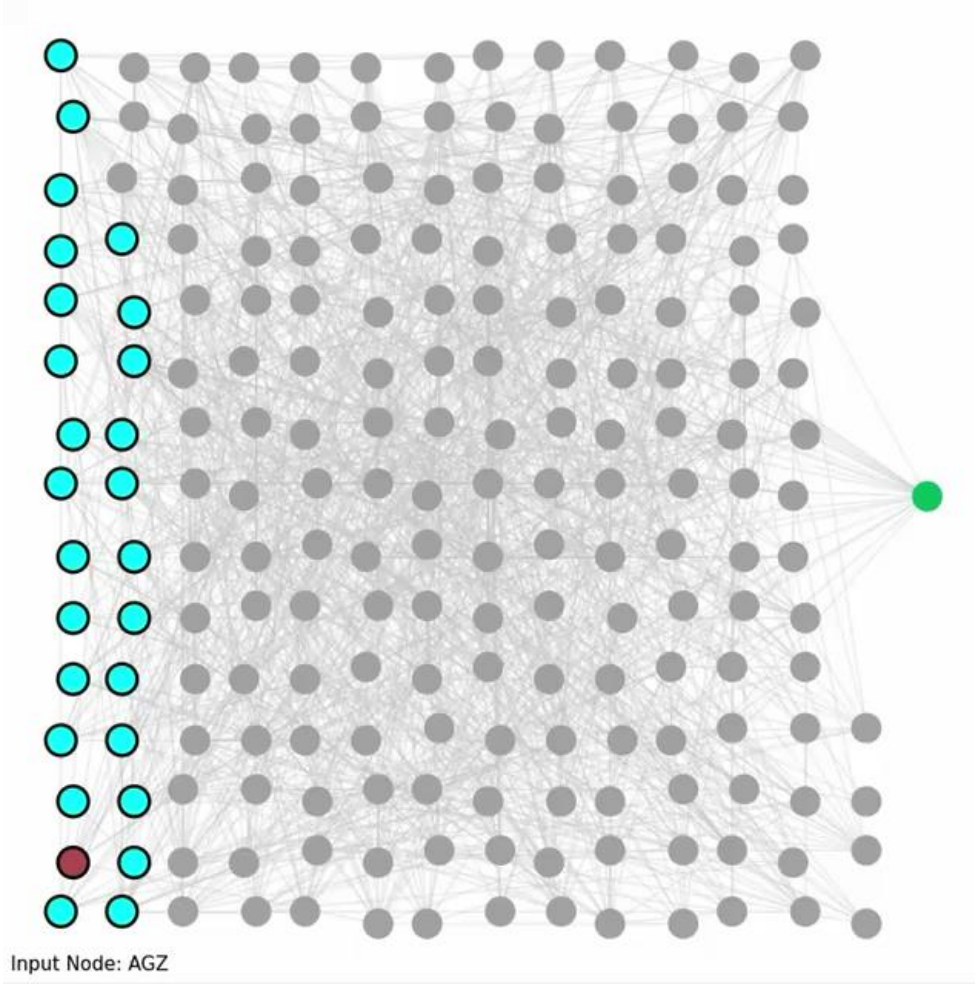




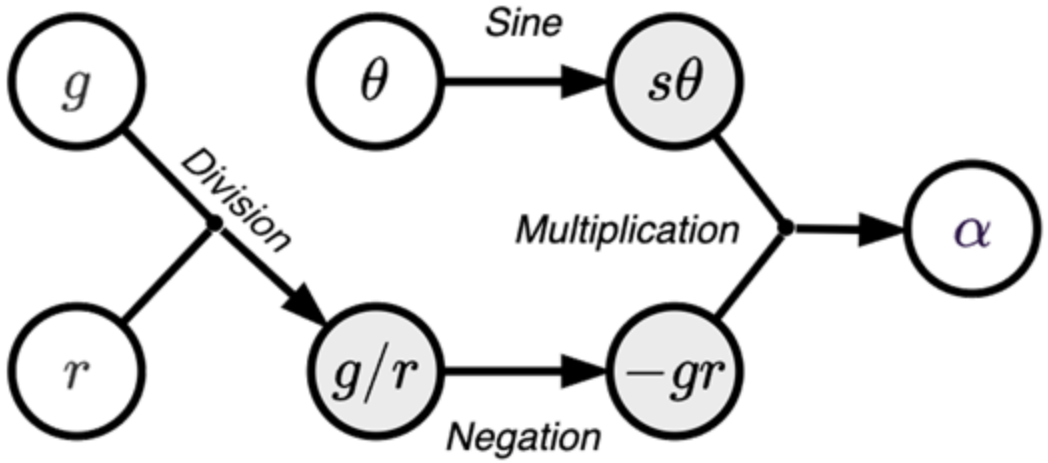
Robinson, M. (2017). Sheaf and Duality Methods for Analyzing Multi-Model Systems. In Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science. Springer International Publishing.

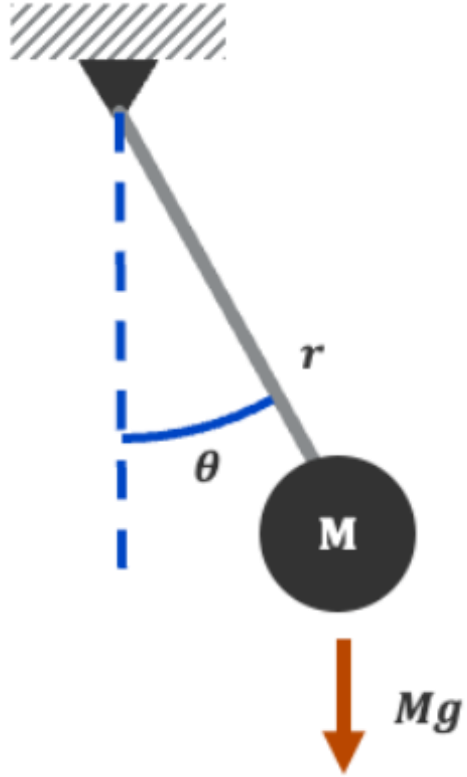
Solving engine available on the Python Package Index

Implements a Breadth-First Search to simulate a system



Software Demonstration

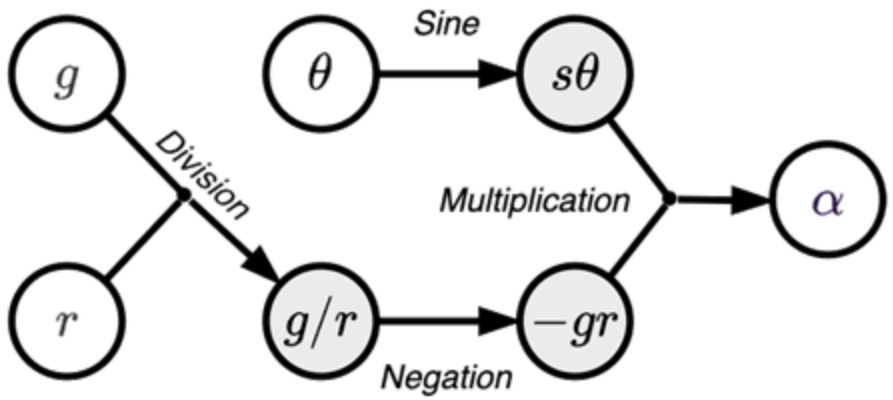




$$\alpha = -\frac{g}{r}\sin\theta$$

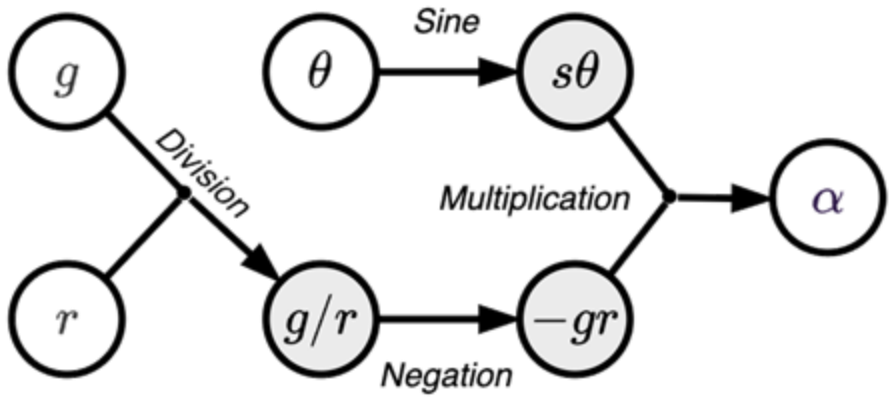
Add Nodes

```
g = Node('gravity', -10)
r = Node('radius', 0.5)
theta = Node('theta')
F = Node('gravitational force')
omega = Node('omega')
alpha = Node('alpha')
time_step = Node('time_step', .03)
time = Node('time', 0)
```



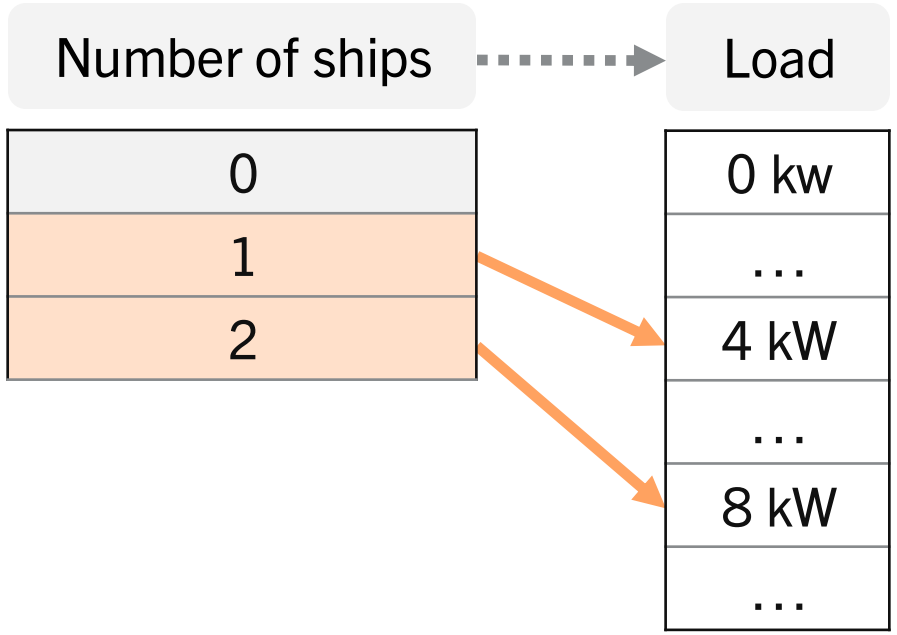
Add Edges

```
hg.add_edge(
   {'s1': g, 's2': r}, target='g/r',
   rel=R.Rdivide,
)
hg.add_edge(
   theta, target=s_theta,
   rel=R.Rsin,
)
hg.add_edge(
   {s_theta, 'g/r'}, target=F,
   rel=R.Rmultiply,
)
hg.add_edge(
   source=F, target=alpha,
   rel=R.Rmean,
   index_offset=1,
)
```



Handling model validity frames

Models with given validity frames can be described by functions that map from a *subset* of a node's values

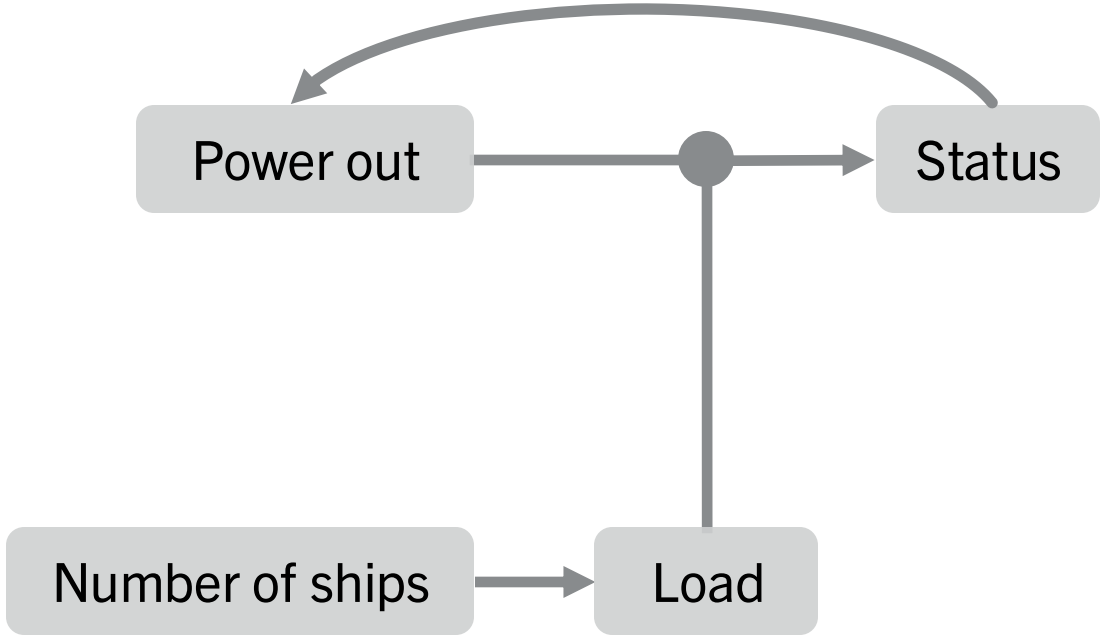


Morris, J., Mocko, G., & Wagner, J. (2025). Unified System Modeling and Simulation via Constraint Hypergraphs. Journal of Computing and Information Science in Engineering, [In Press]

Cycles violate transitivity, indicating conflicting models

Can be dealt with in two ways:

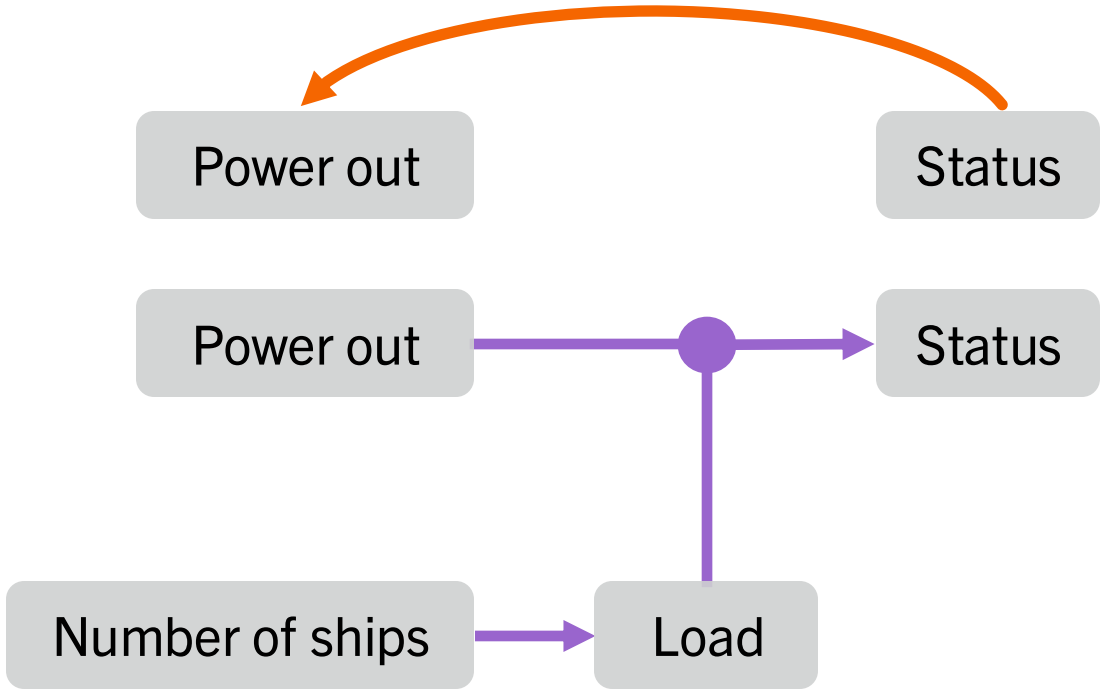
1. Separation
2. ~~Cloning~~ ***Unraveling (Goren-Roig, 2025)***



Morris, J., Mocko, G., & Wagner, J. (2025). Effects of Functional and Declarative Modeling Frameworks on System Simulation. [Manuscript Submitted for Review to the ASME Journal of Dynamic Systems, Measurement and Control].

Separating cycles into distinct graphs

Good for isolating *competing* models

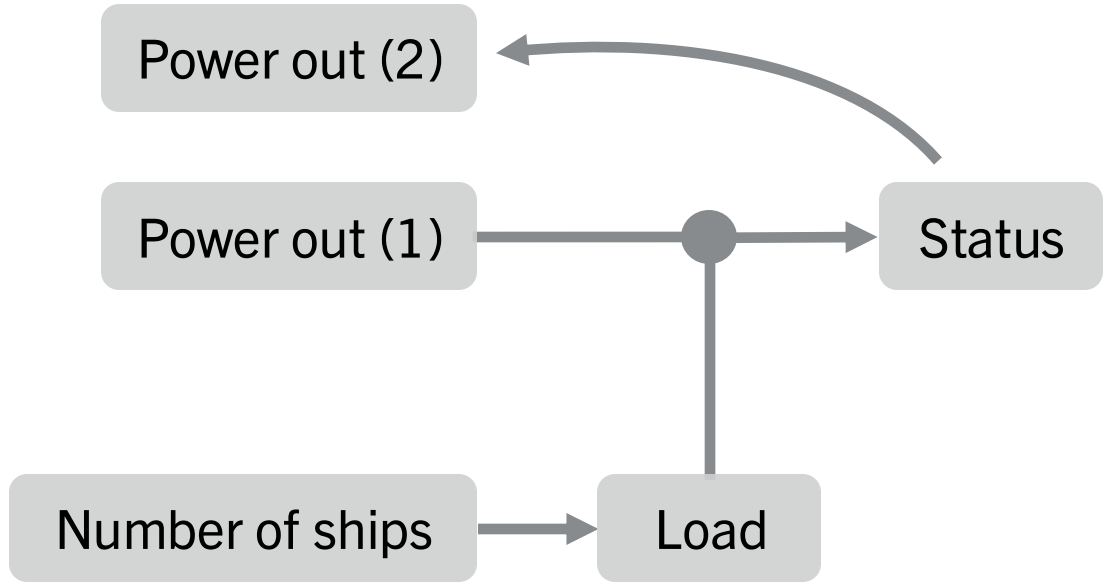


Robinson, M. (2017). Sheaf and Duality Methods for Analyzing Multi-Model Systems. In Recent Applications of Harmonic Analysis to Function Spaces, Differential Equations, and Data Science. Springer International Publishing.

~~Cloning~~ *Unraveling* nodes allows cycles to extend the hypergraph

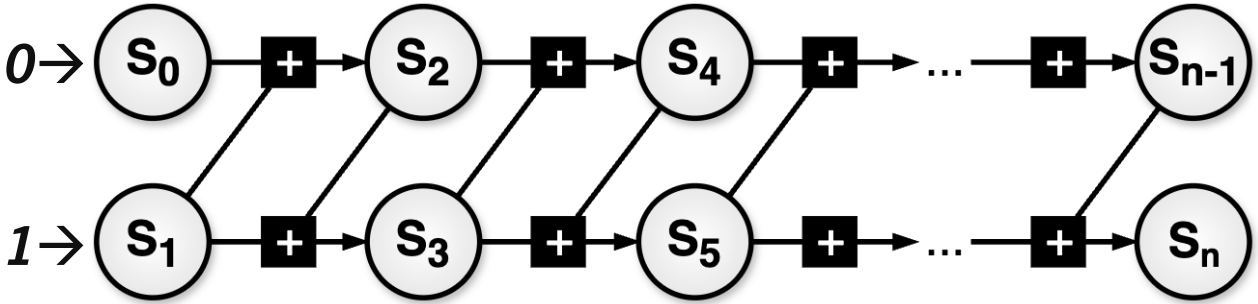
Good for *pattern* adaption

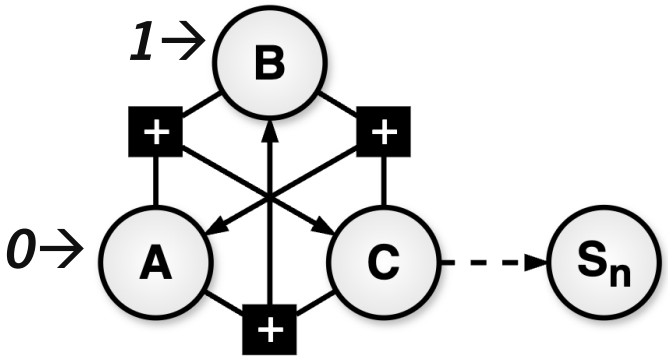
~~Cloning~~ *Unraveling* occurs while searching for simulation paths and requires an exit edge

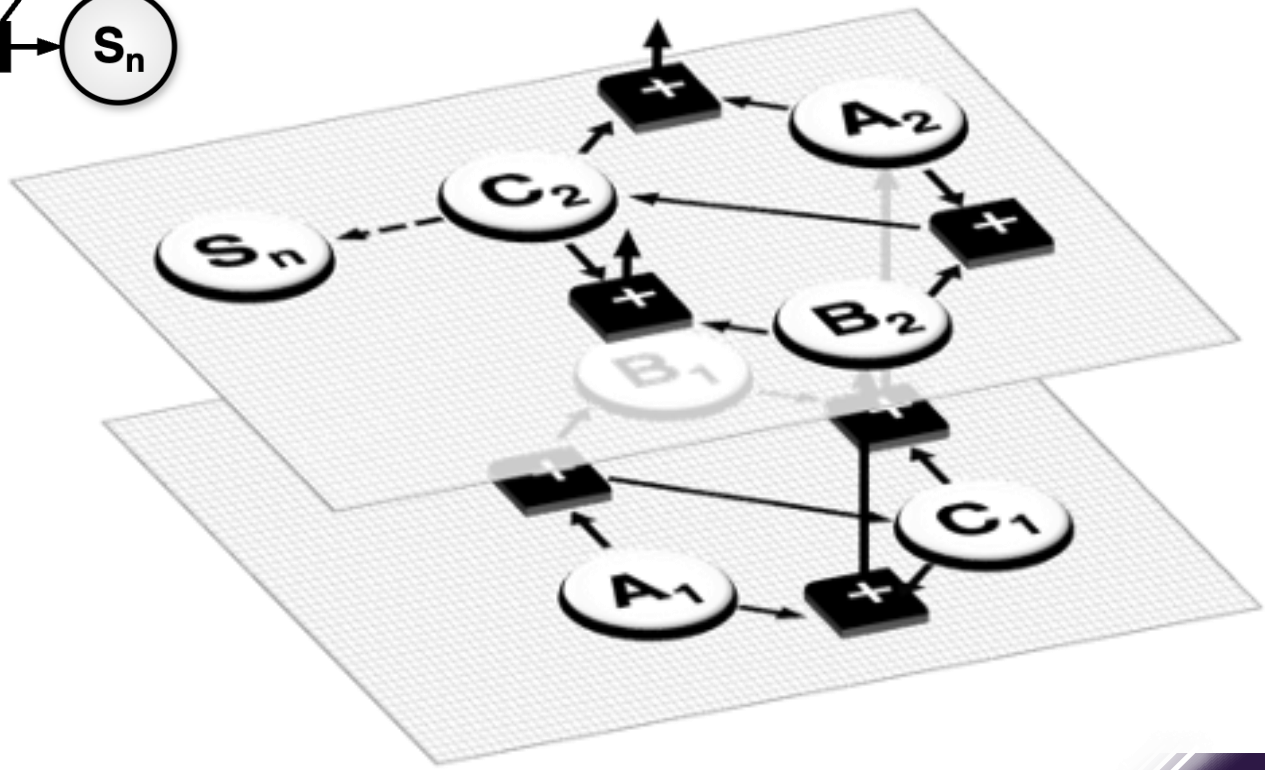


Morris, J., Mocko, G., & Wagner, J. (2025). Effects of Functional and Declarative Modeling Frameworks on System Simulation. [Manuscript Submitted for Review to the ASME Journal of Dynamic Systems, Measurement and Control].

Example computing Fibonacci sequence using cloning

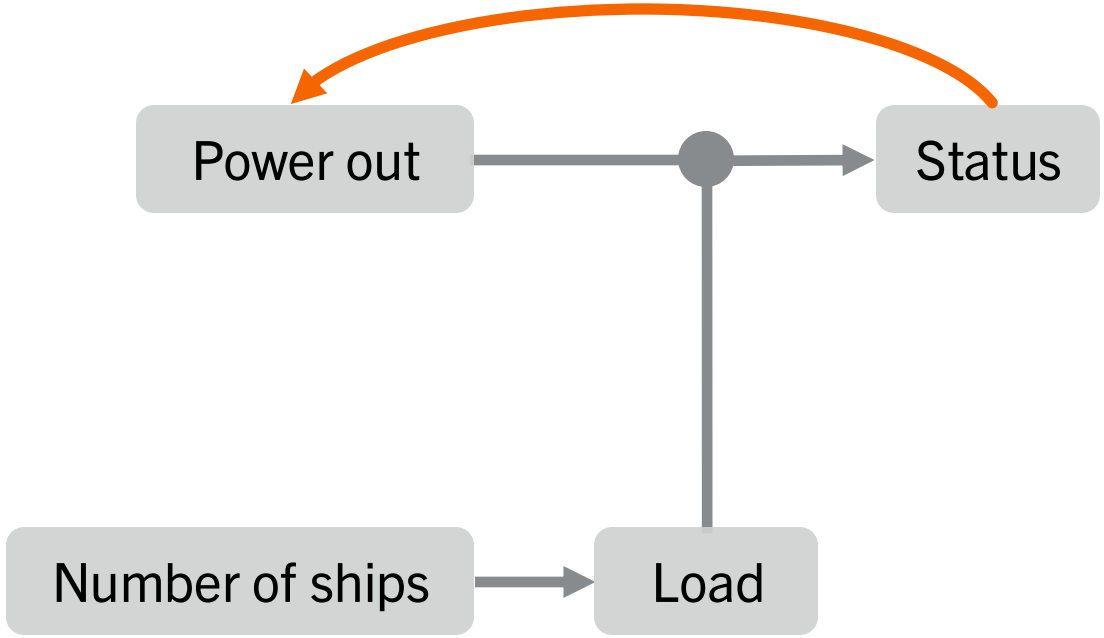






Independent behaviors make the hypergraphs immediately modifiable

Functions can be rewritten, removed, or added without affecting the consistency of the graph *(no side effects)*

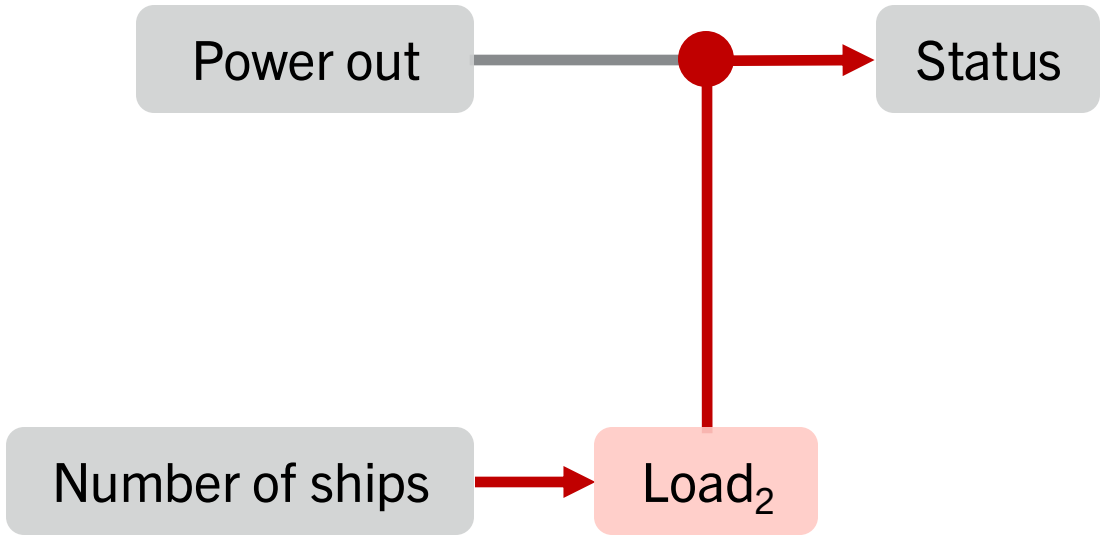


Morris, J., Mocko, G., & Wagner, J. (2025). Effects of Functional and Declarative Modeling Frameworks on System Simulation. [Manuscript Submitted for Review to the ASME Journal of Dynamic Systems, Measurement and Control].

Caveat: scope changes are not guaranteed to be consistent

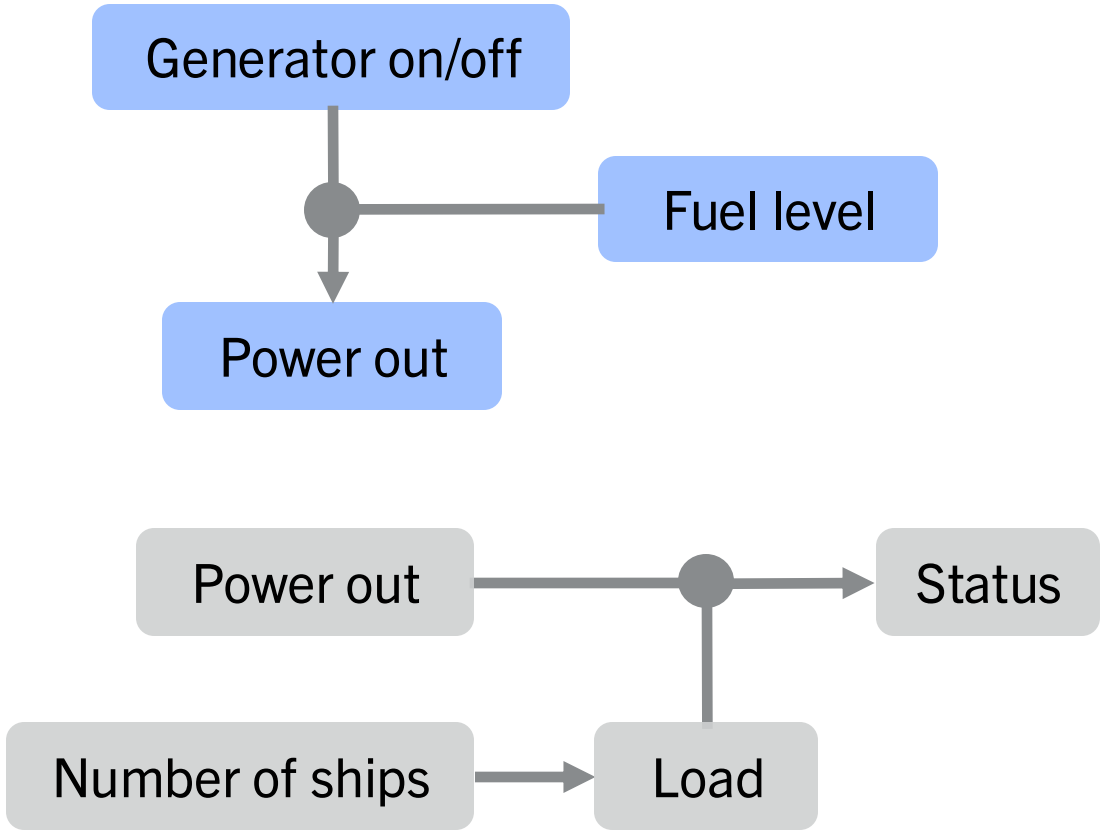
Nodes can be modified only if all connecting edges are also updated to maintain consistency *(only local side effects)*

Behavioral assumptions may be affected



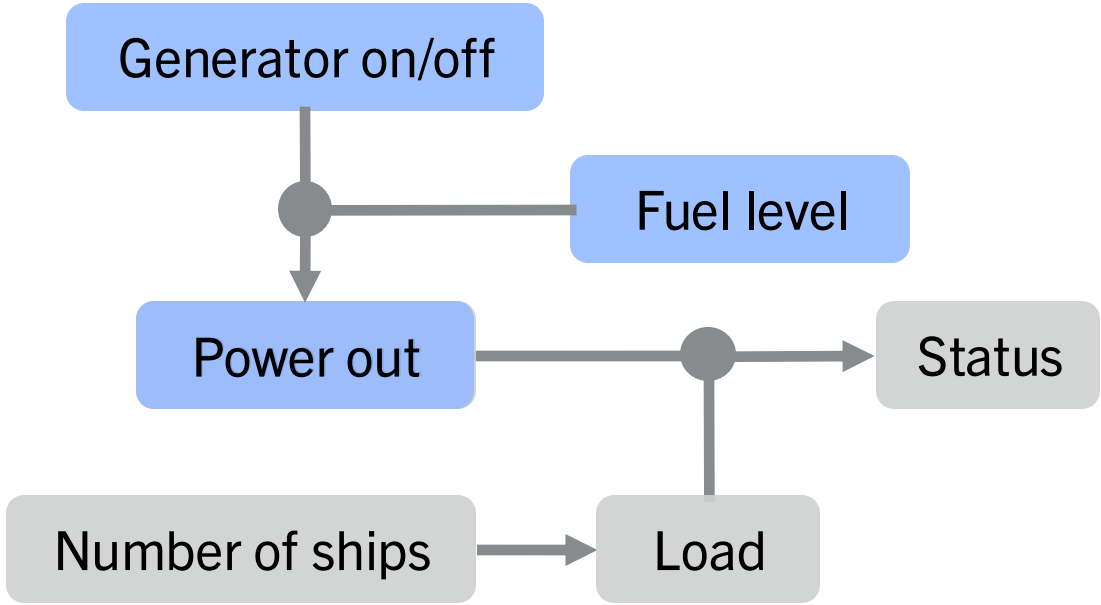
Composable with other graphs along shared nodes (union)

All simulation paths remain consistent *if* new scope does not violate behavioral assumptions



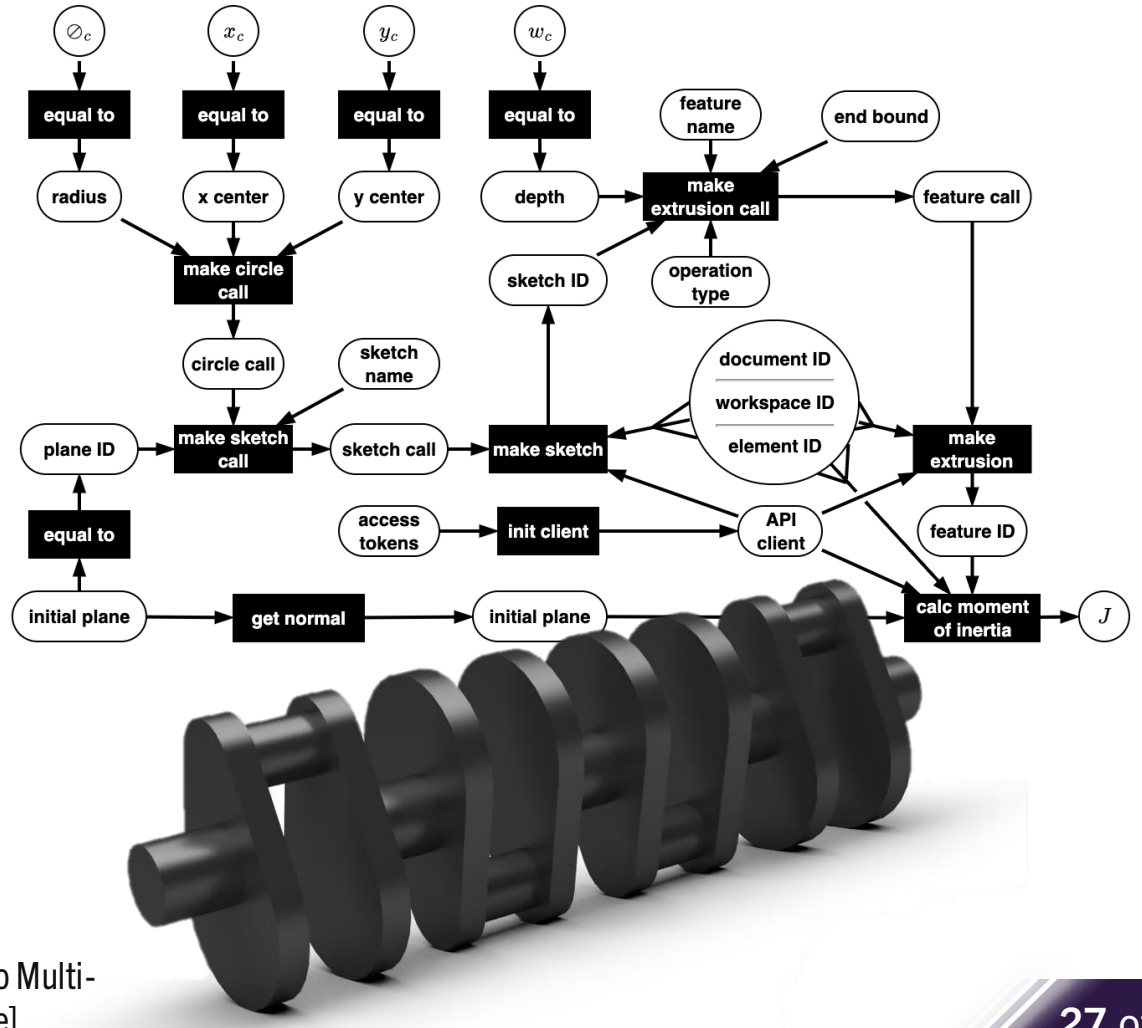
Emergent behavior is provided entirely from composition

All simulation paths remain consistent *if* new scope does not violate behavioral assumptions



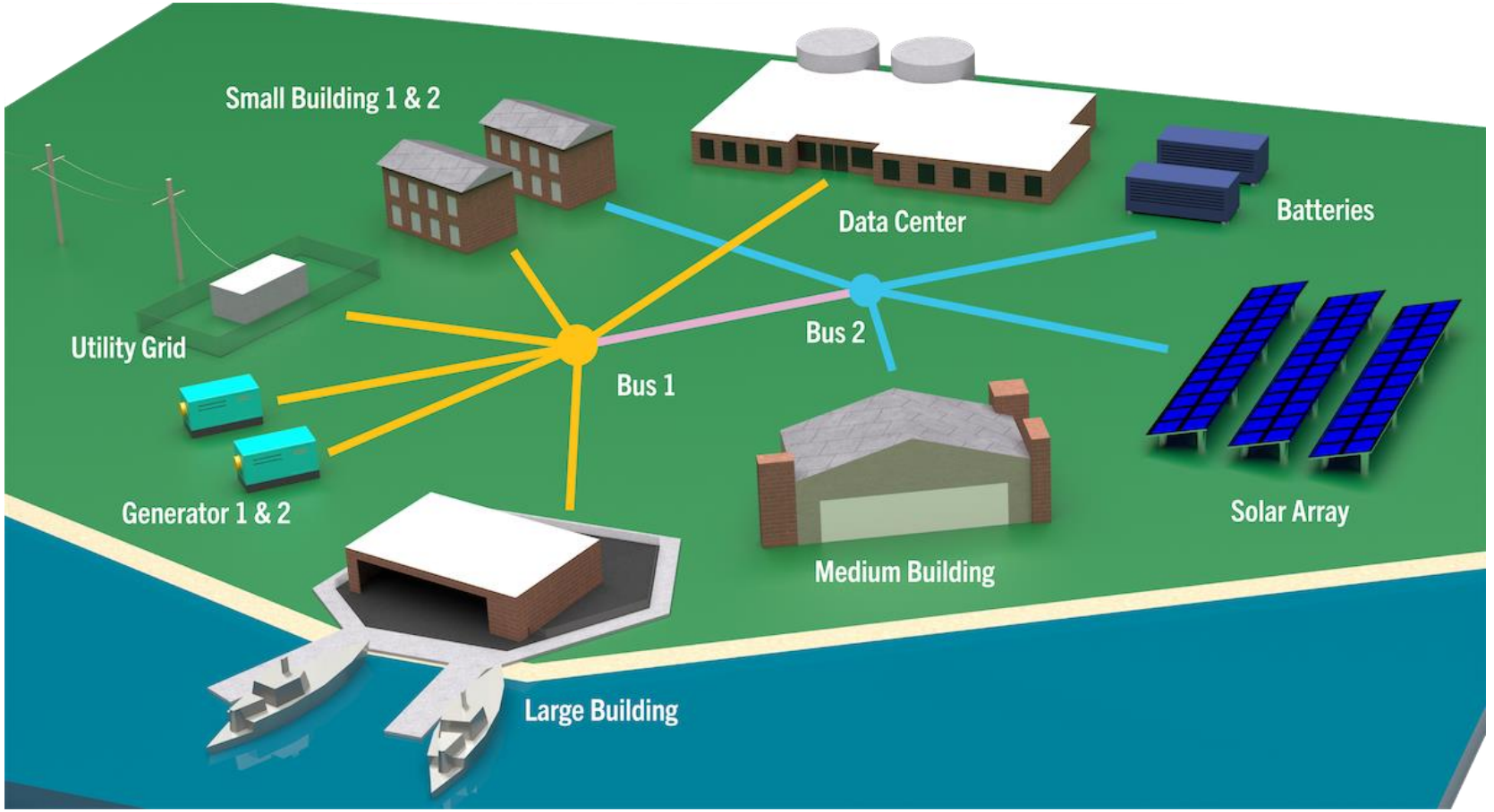
Constraint hypergraphs tie siloed software into a single representation

Software can be integrated by wrapping API calls as edges



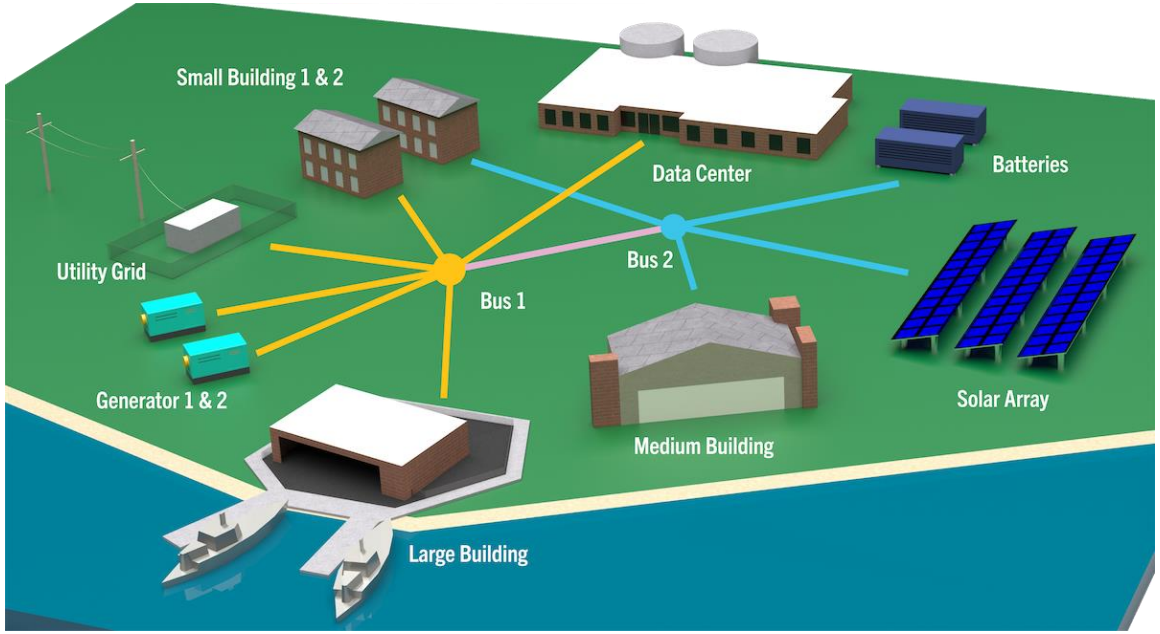
Morris, J., Mocko, G., Wagner, J., & Ramnath, S. (2025). Declarative Integration of CAD Software into Multi-Physics Simulation via Constraint Hypergraphs. [Submitted to the ASME IDETC-CIE 2025 conference]

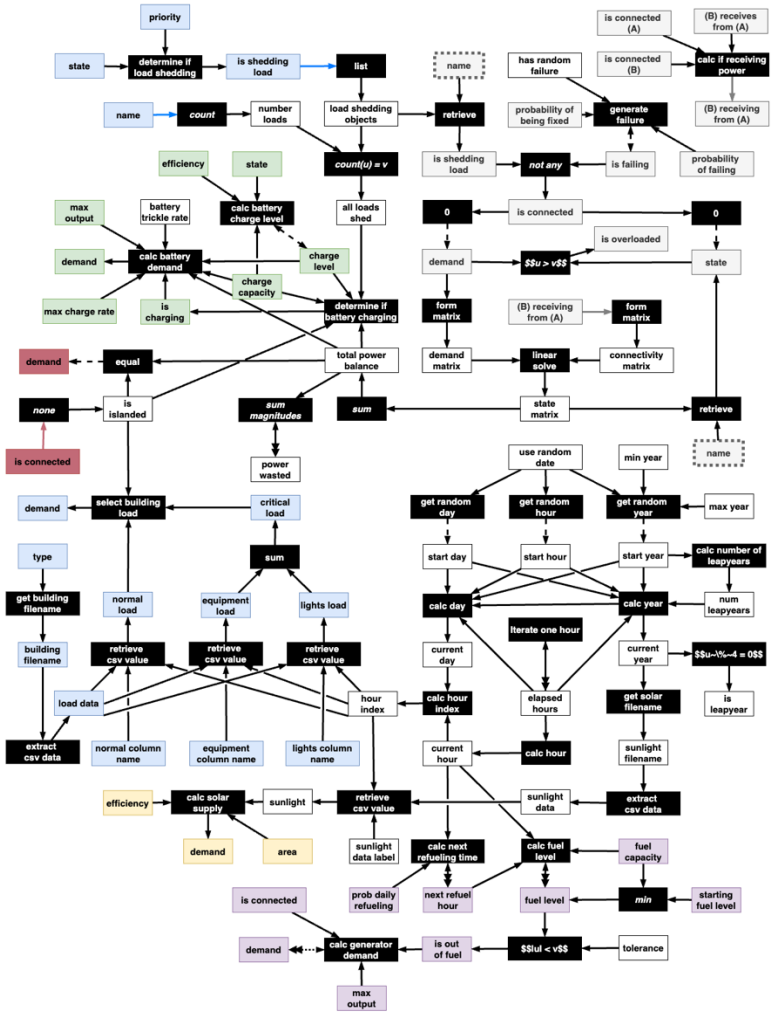
Demo: Integrating more complex systems



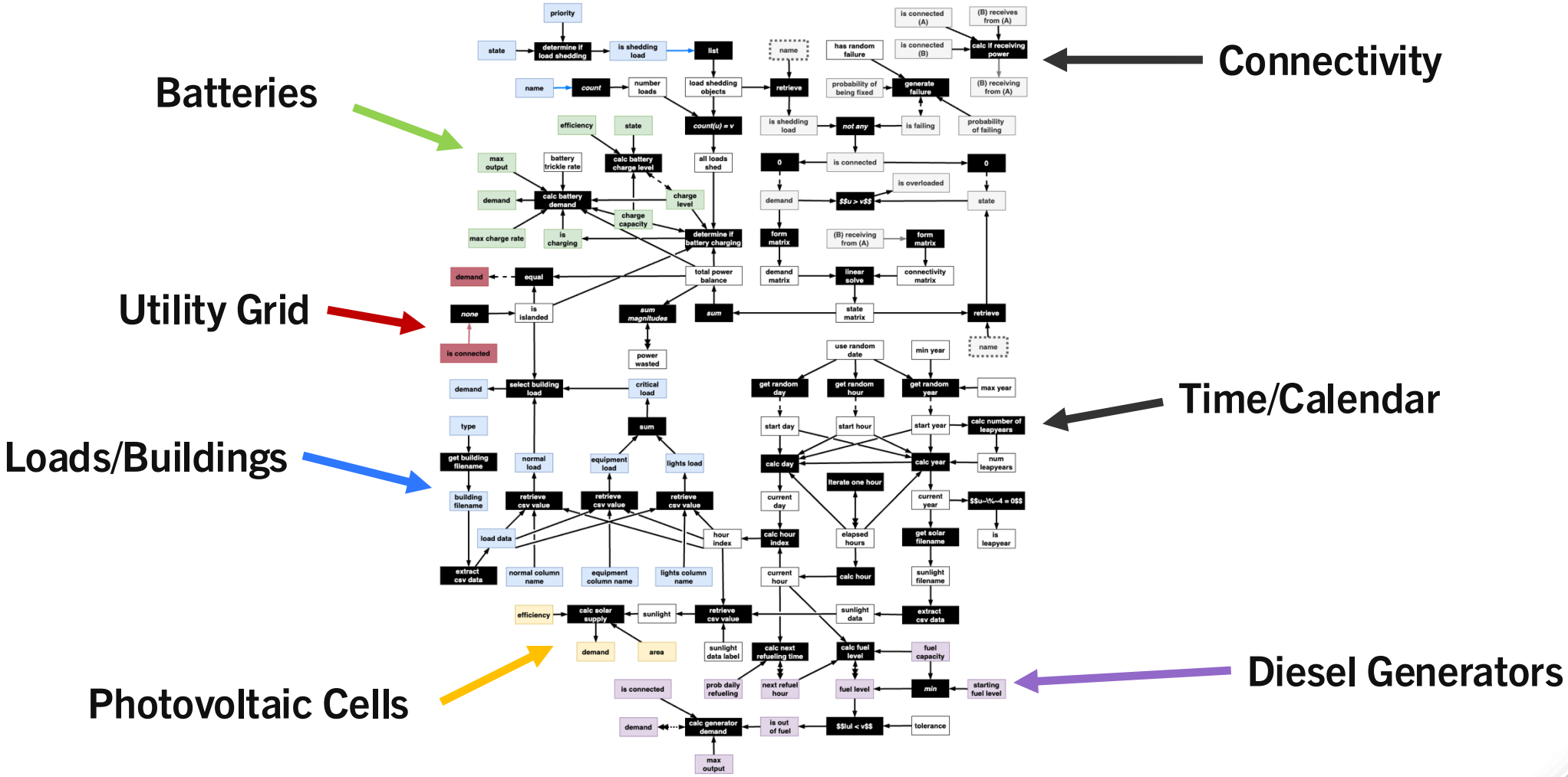
Example available on GitHub by following QR code in corner

Hypergraph contains 594 nodes and 334 edges

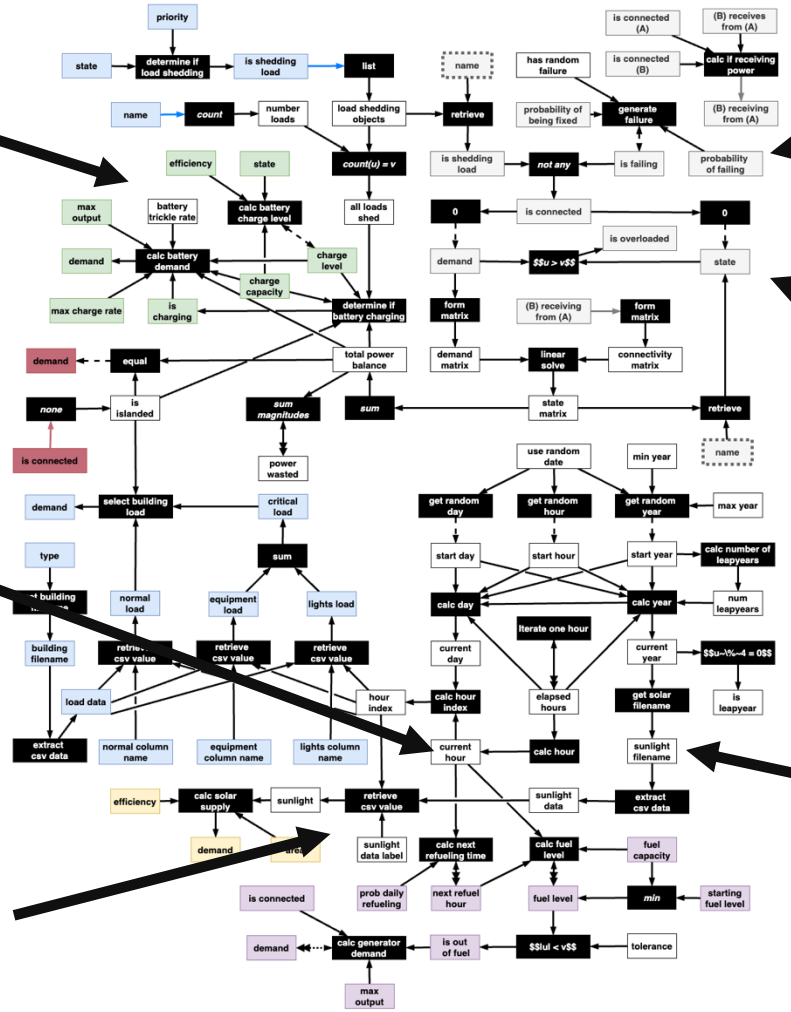




Regions of the hypergraph represent different *things* in the system



Includes all kinds of simulation and modeling



Algebraic rules

Stochastic models

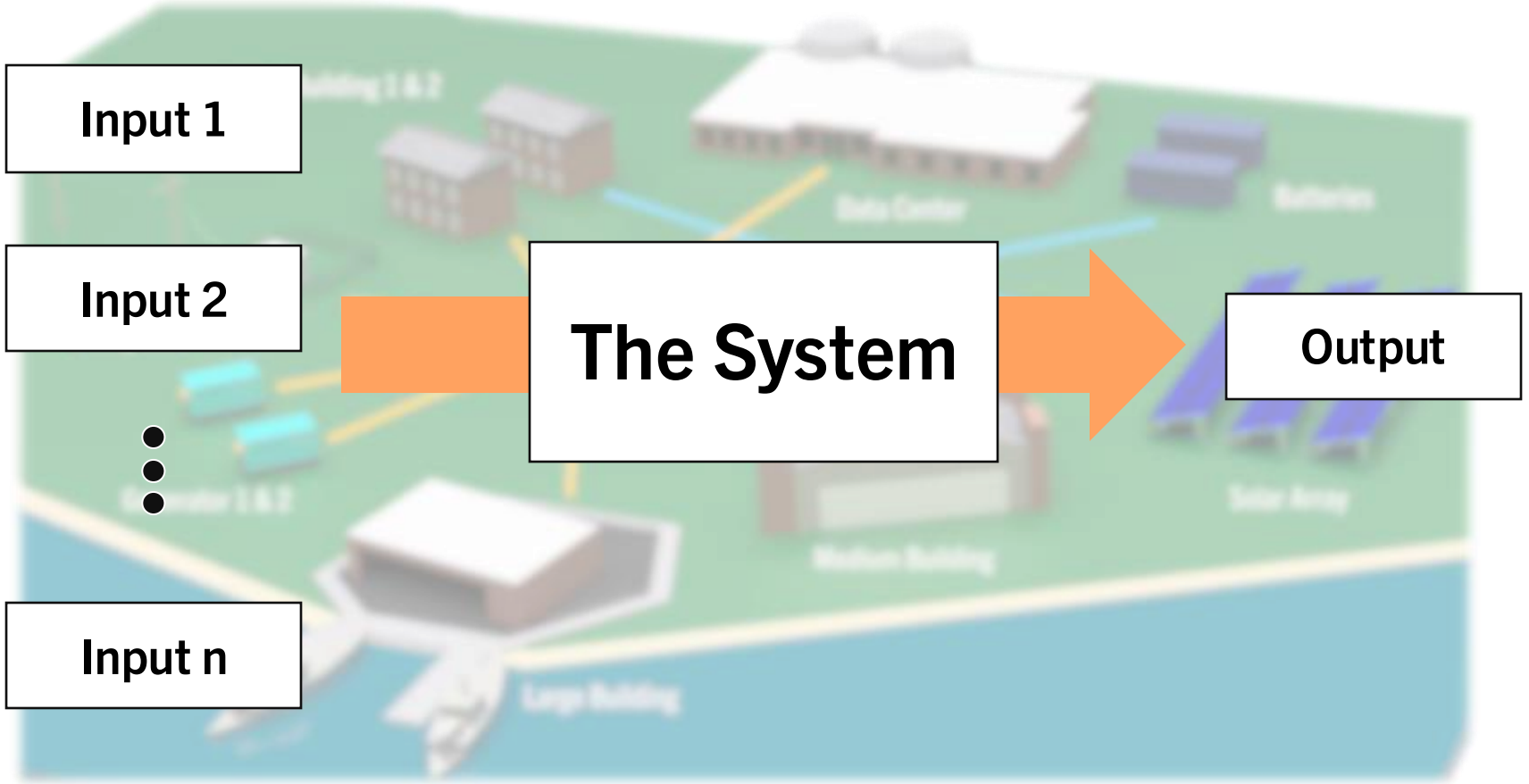
Linear equation solvers

Dynamic simulation

File input/output

Database queries

Hypergraphs of behavioral constraints capture the relationships of things



More information available at our documentation

Please reach out to jhmrrs@clemson.edu

PLMC

CEDAR
Clemson Engineering Design Applications and Research

CLEMSON