Strategies for Aggregating Digital Twins

John Morris, Duncan Gibbons, Joe Gregory, Greg Mocko

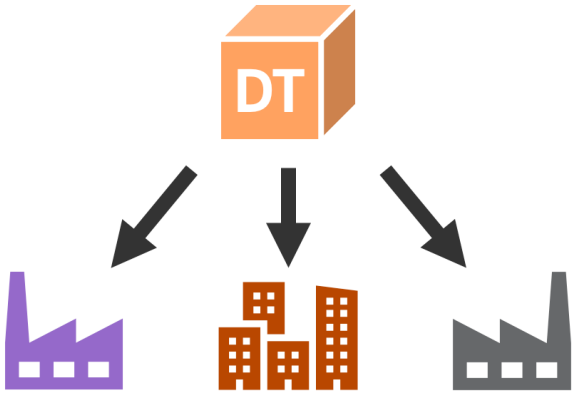
9 April 2026
Conference on Systems Engineering Research

THE UNIVERSITY OF ALABAMA IN HUNTSVILLE
Stellenbosch UNIVERSITY IYUNIVESITHI UNIVERSITEIT
THE UNIVERSITY OF ARIZONA
CLEMSON

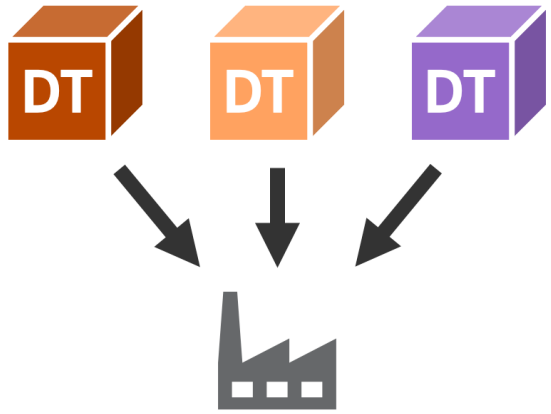
Motivation: Understanding complex systems requires integration of heterogeneous information and models

A digital twin is a system whose states and behavior inform concerning another system of interest

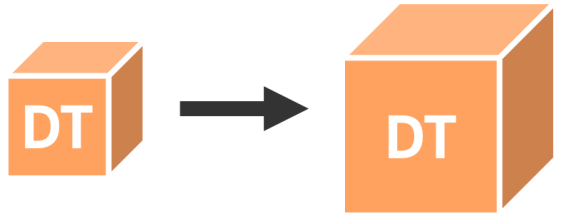
Interoperable digital twins allow us to understand complex systems



Redeployable
DT function in multiple organizations

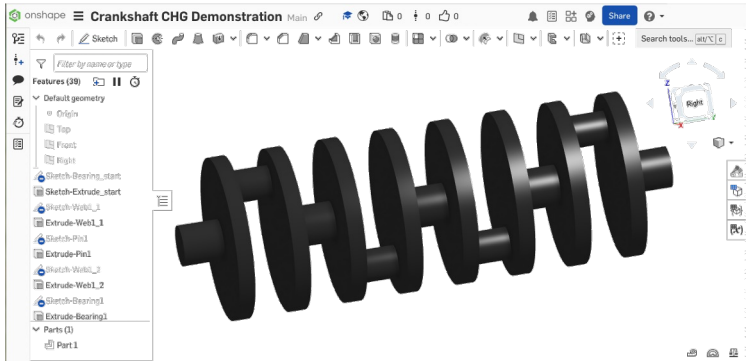


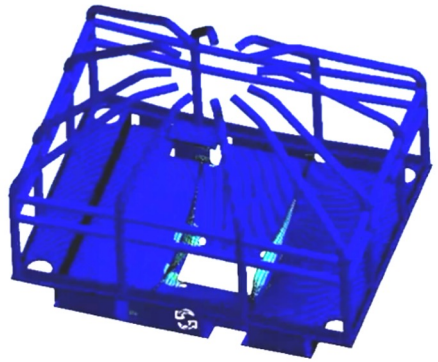
Modular
DTs interface with other DTs

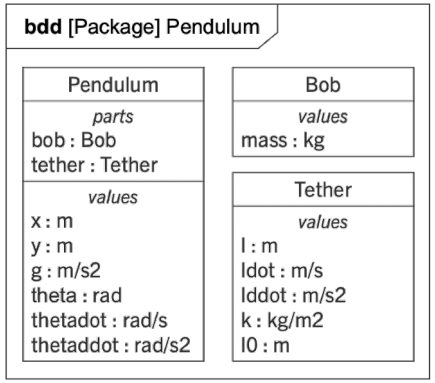


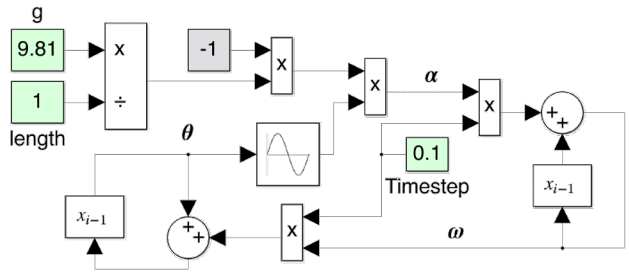
Scalable
DTs grow with a company's resources

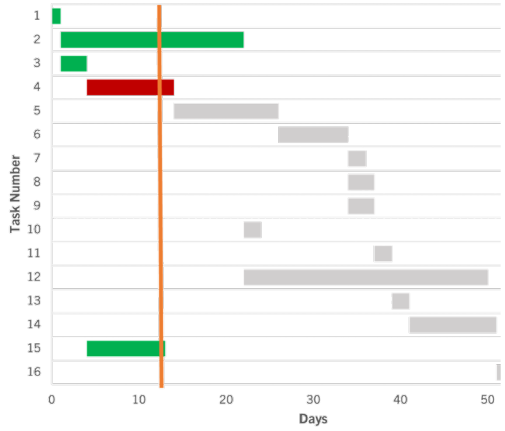
Each individual digital twin provides a view of a subsystem

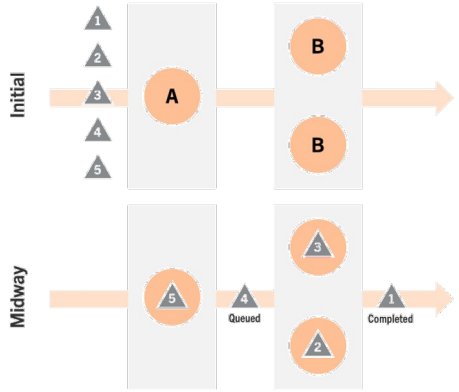


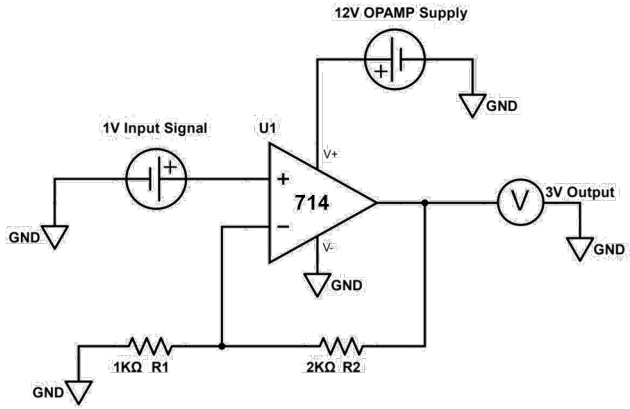


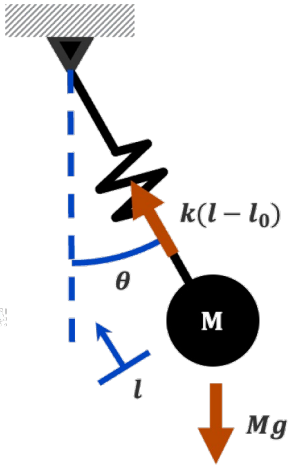




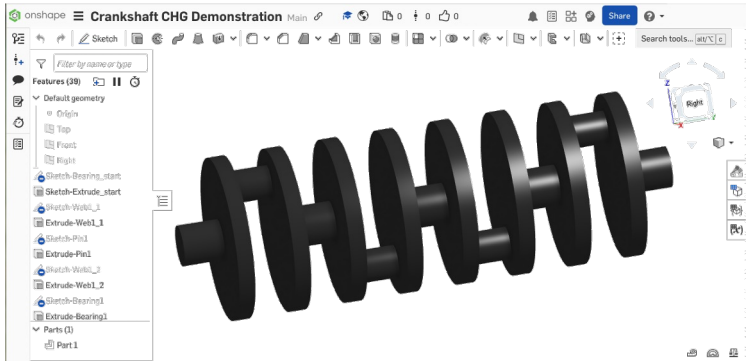


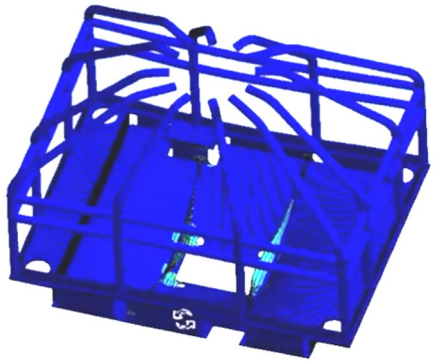


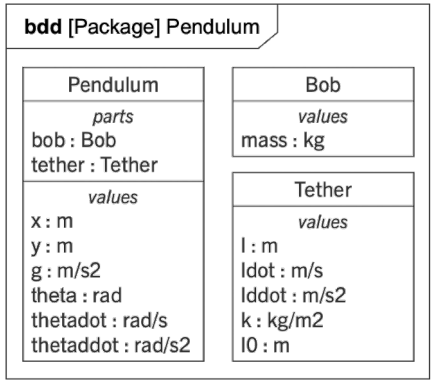


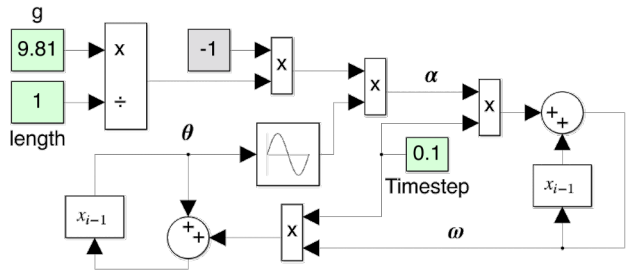


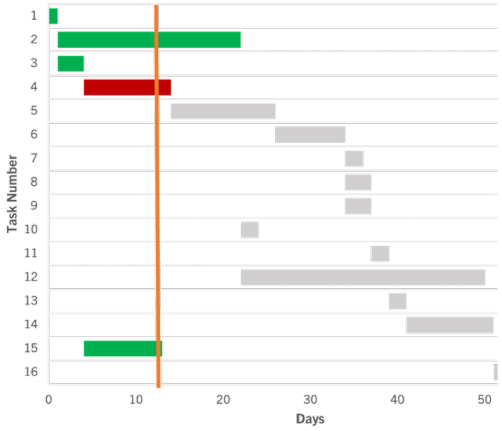
The aggregation of subsystem twins describes a supersystem

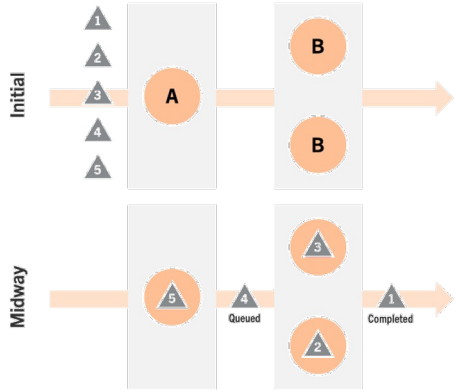


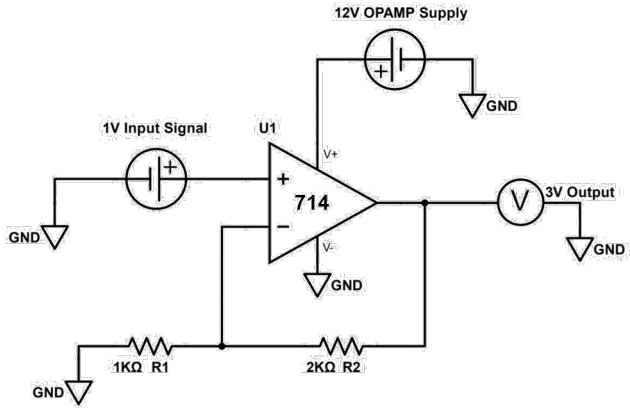


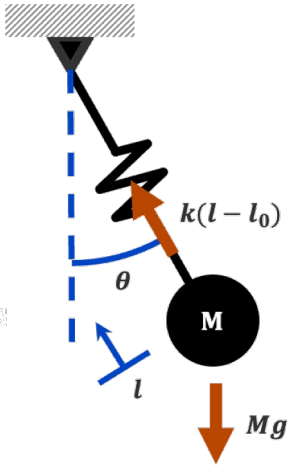




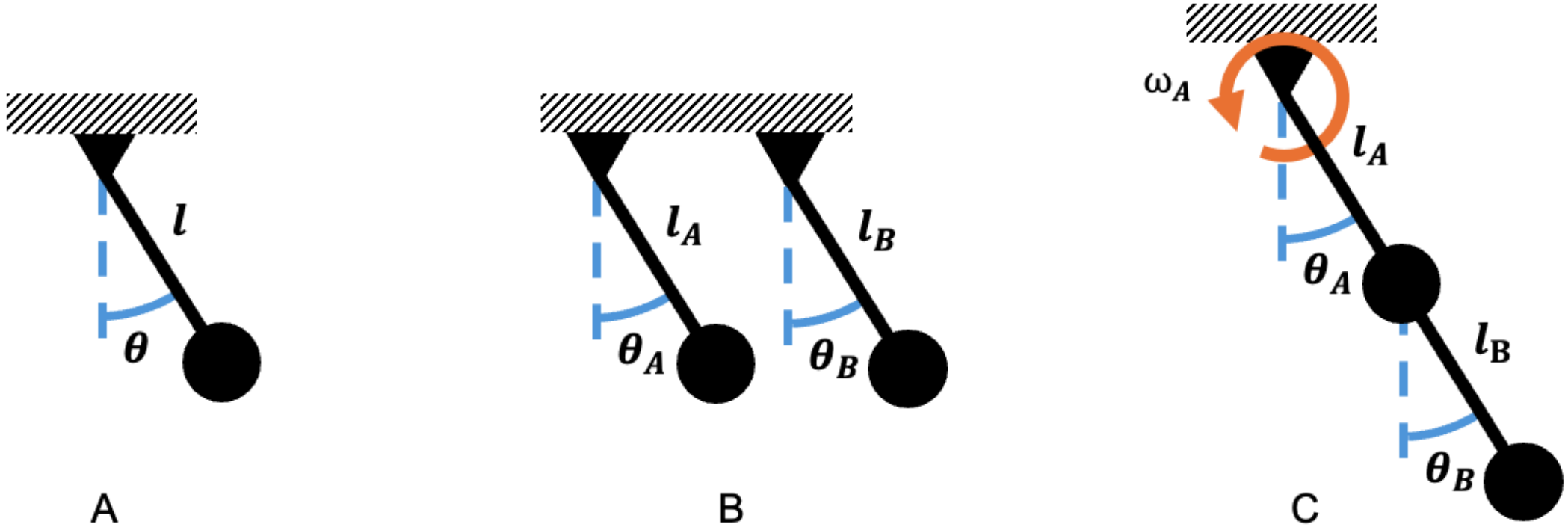




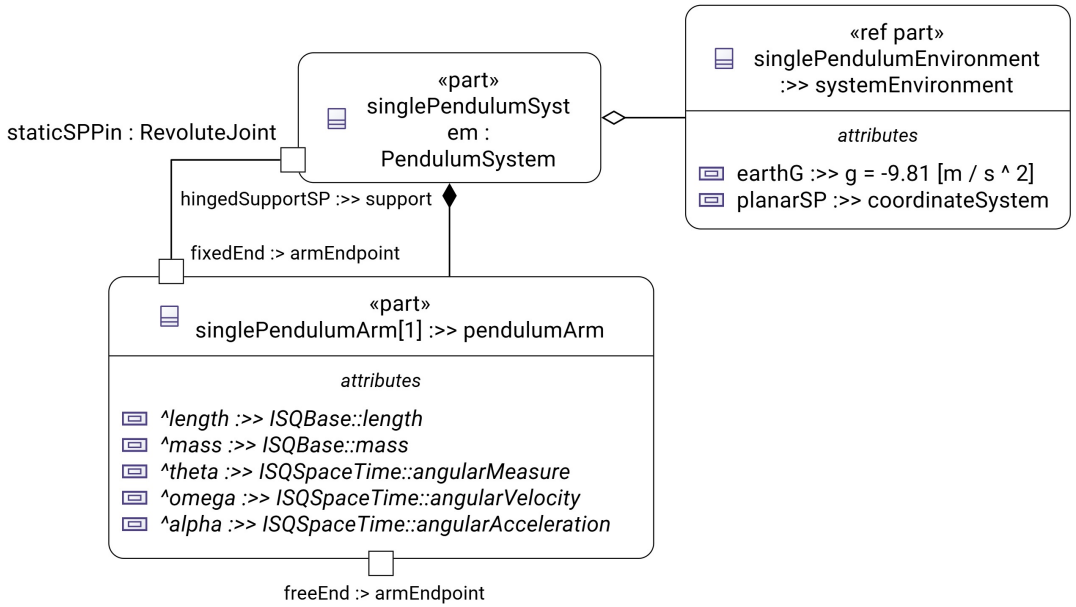


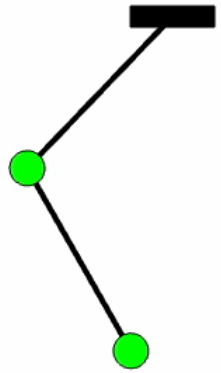


Pendulum case study considers the composition of subsystems



Two sides of model use: semantics and behavior

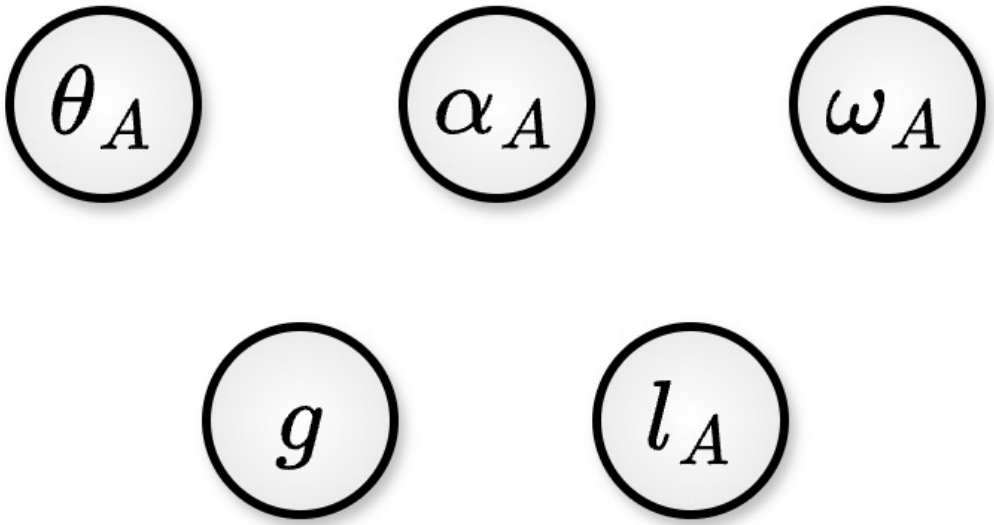


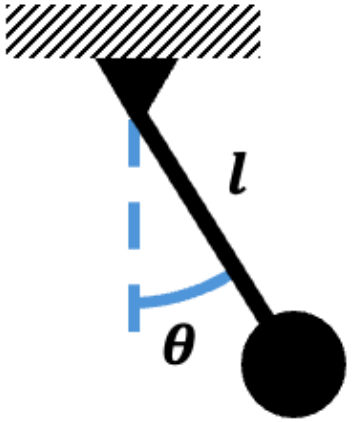


Semantics: what the system is

Behavior: what the system does

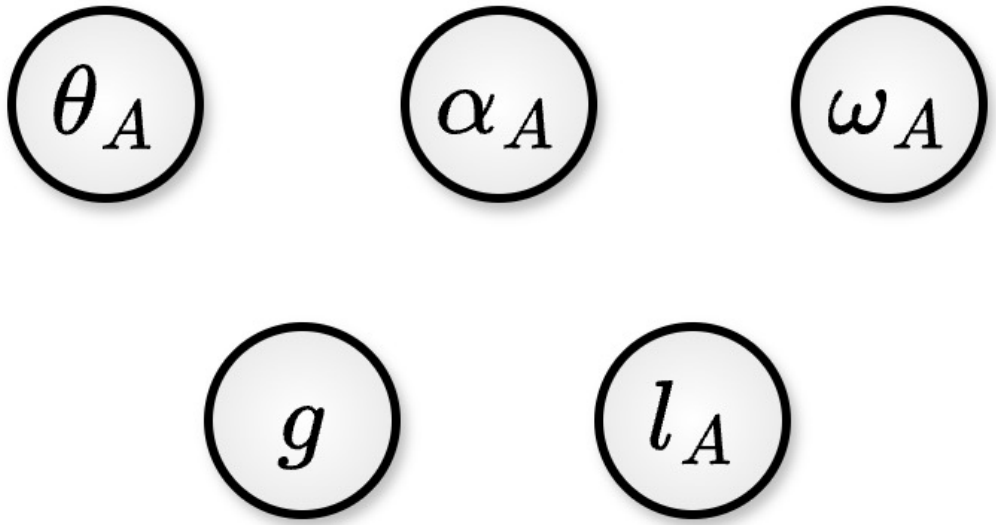
Behavior is given by constraints on the system state-space





J. C. Willems, “The Behavioral Approach to Open and Interconnected Systems,” IEEE Control Systems Magazine (2007)

Behavior is given by constraints on the system state-space

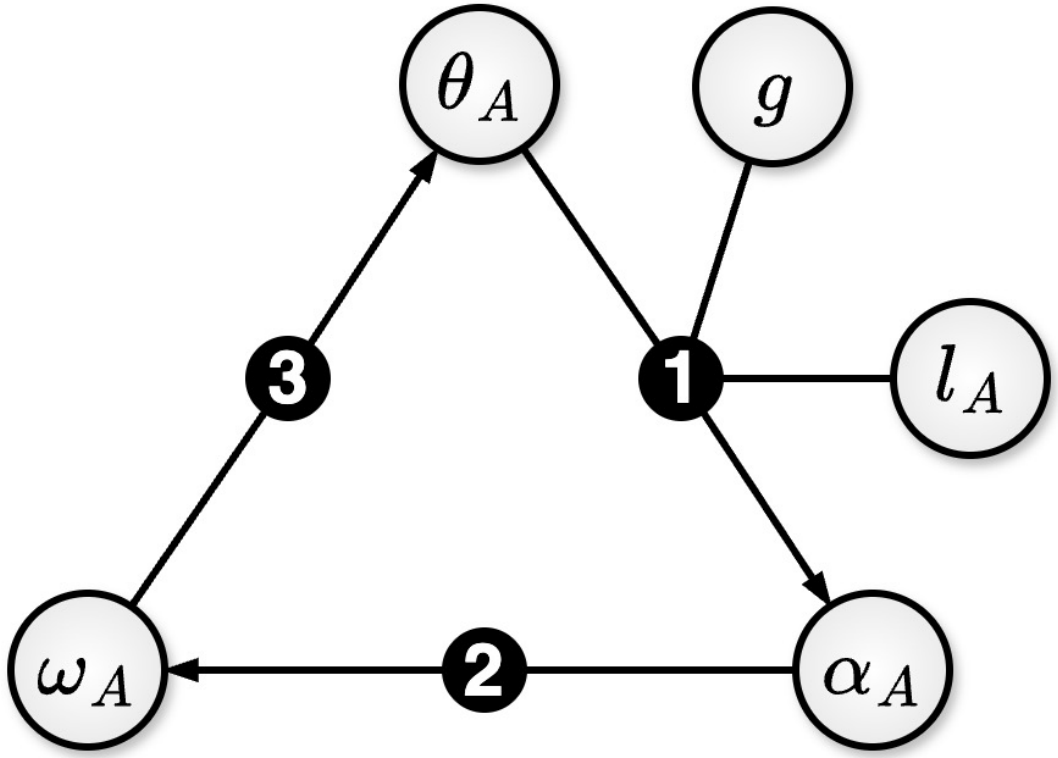


$$\alpha = \frac{g}{l} sin\,\theta$$

$$\omega^{i+1} = \omega^i + \alpha\Delta t$$

$$\theta^{i+1} = \theta^i + \omega\Delta t$$

Behavior is given by constraints on the system state-space

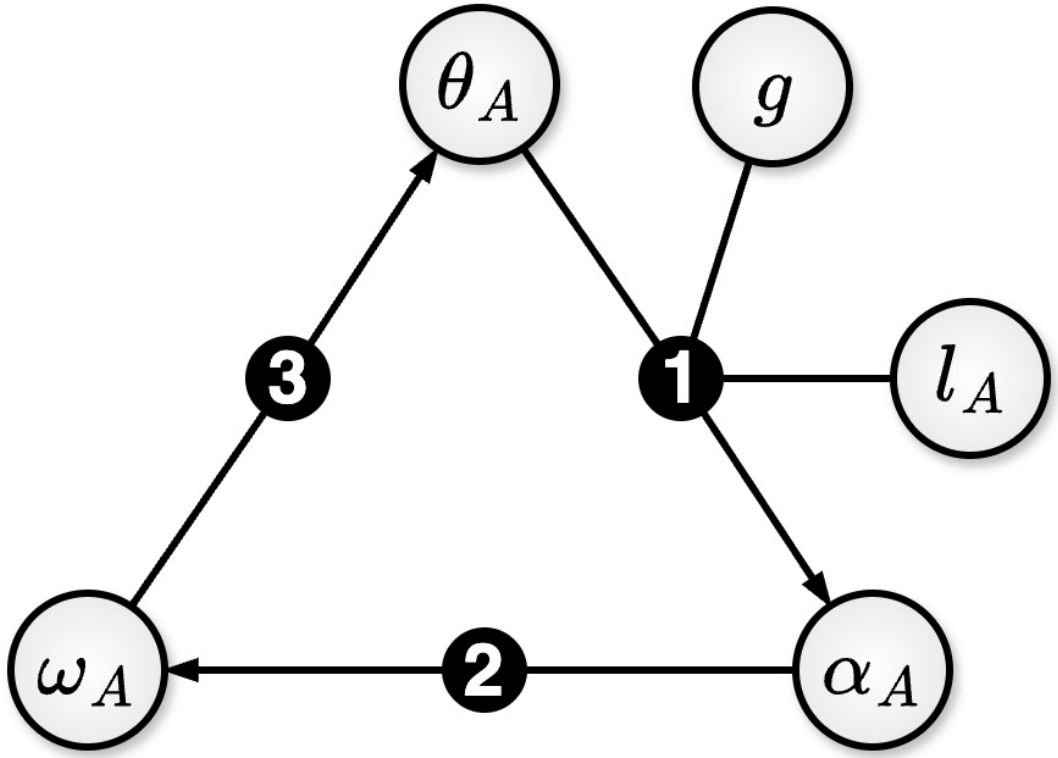


$$\alpha = \frac{g}{l} sin\,\theta \quad (1)$$

$$\omega^{i+1} = \omega^i + \alpha \Delta t \quad (2)$$

$$\theta^{i+1} = \theta^i + \omega \Delta t \quad (3)$$

Behavioral constraints collectively form a Constraint Hypergraph



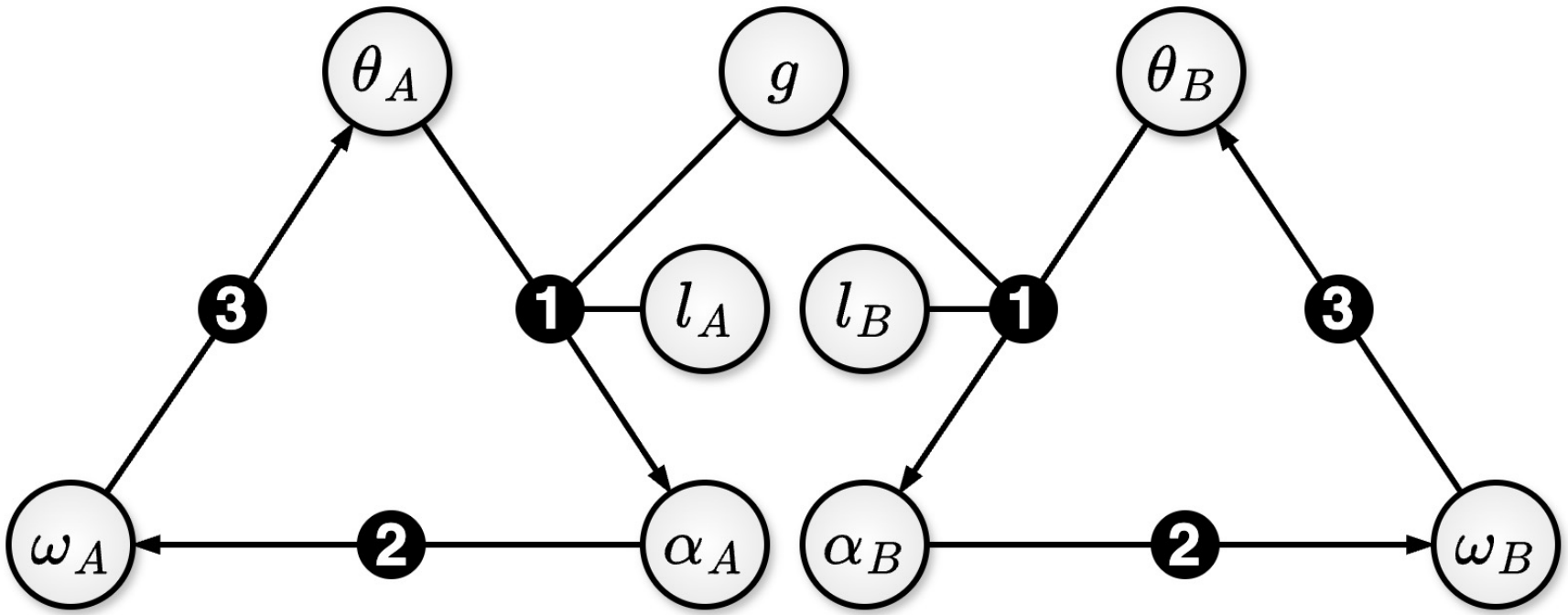
$$\alpha = \frac{g}{l} sin\,\theta \quad (1)$$

$$\omega^{i+1} = \omega^i + \alpha\Delta t \quad (2)$$

$$\theta^{i+1} = \theta^i + \omega\Delta t \quad (3)$$

Morris, J., Mocko, G., & Wagner, J. (2025). Unified System Modeling and Simulation via Constraint Hypergraphs. *J. Comput. Inf. Sci. Eng.*

Constraint Hypergraphs can be immediately combined to show the behavior of a supersystem



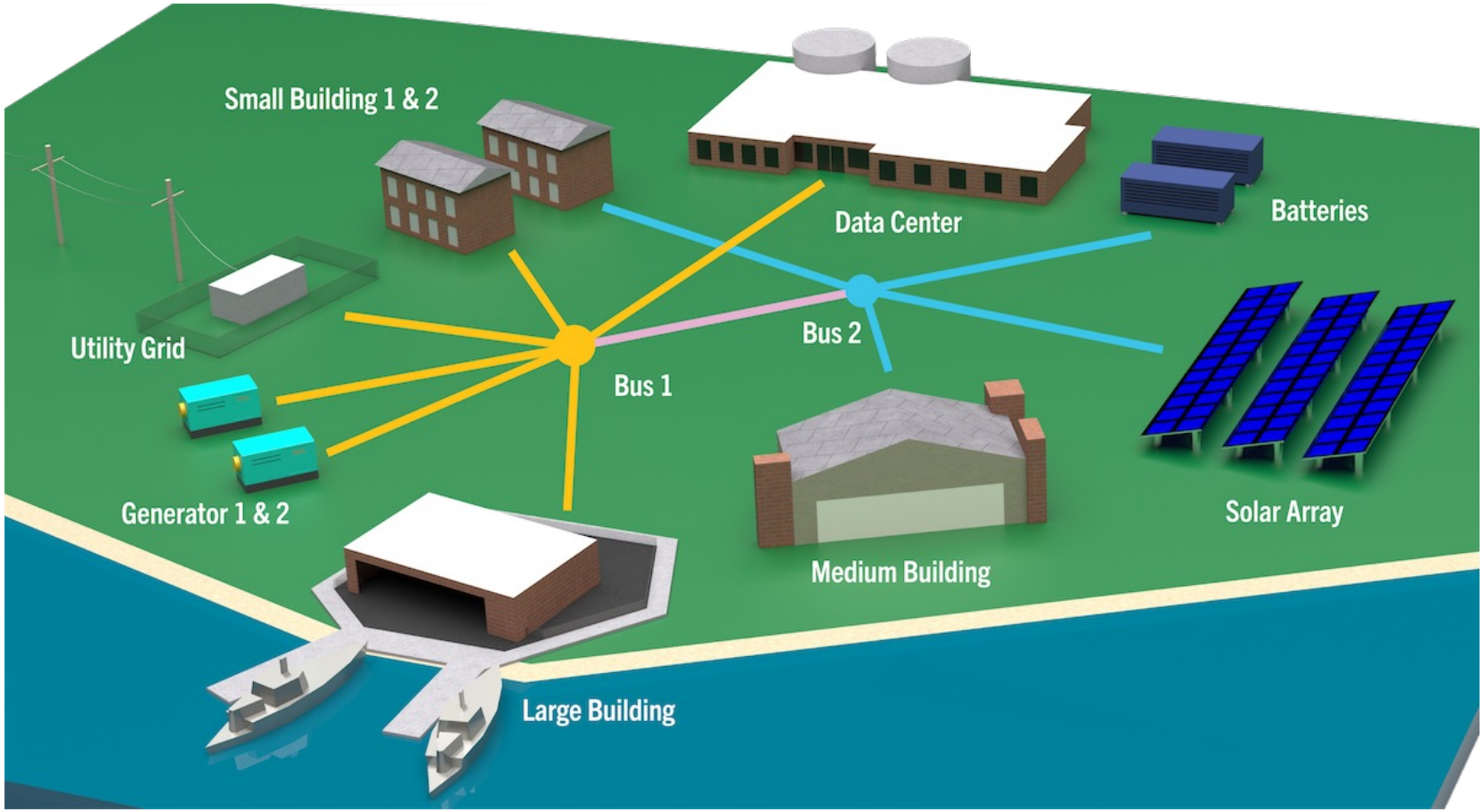
$$\alpha = \frac{g}{l} sin\,\theta \quad (1)$$

$$\omega^{i+1} = \omega^i + \alpha \Delta t \quad (2)$$

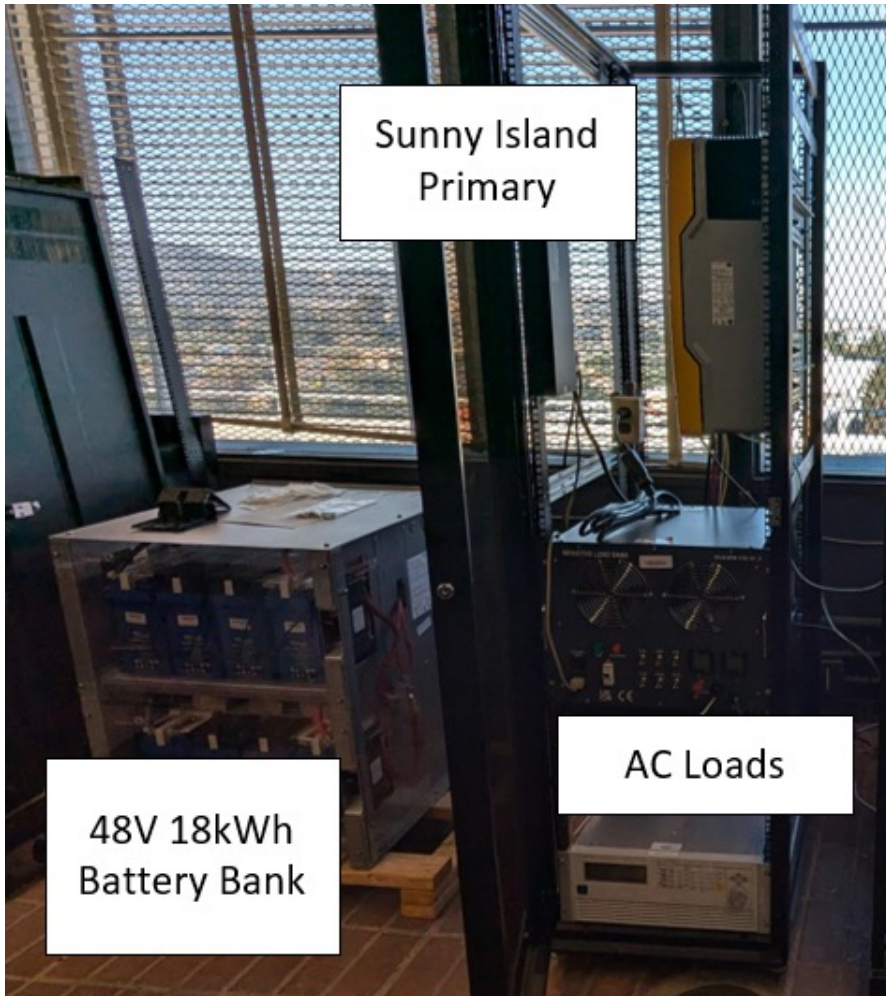
$$\theta^{i+1} = \theta^i + \omega \Delta t \quad (3)$$

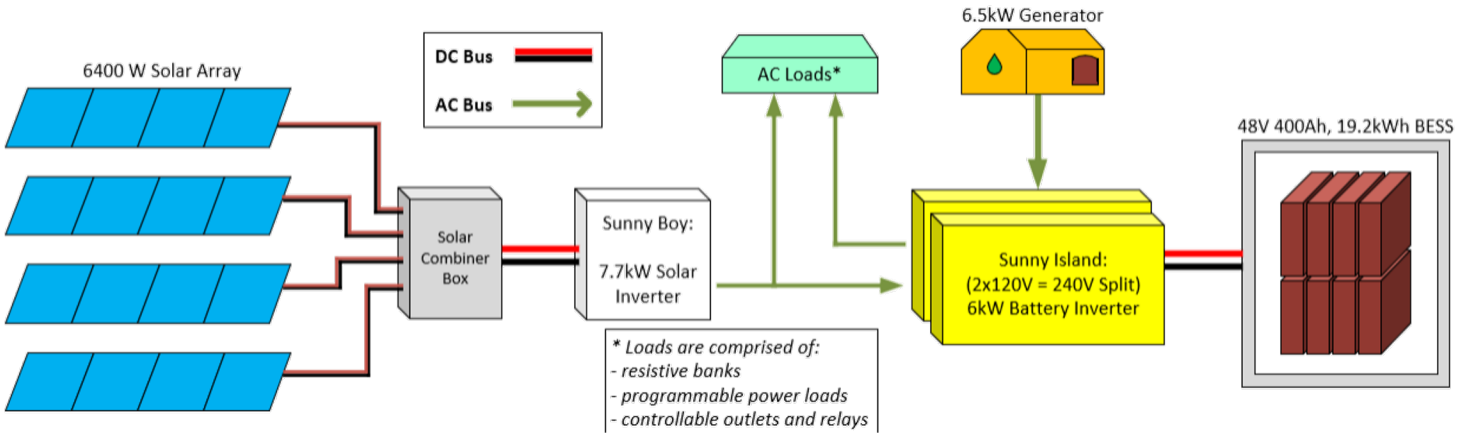
Morris, J., Mocko, G., & Wagner, J. (2025). Unified System Modeling and Simulation via Constraint Hypergraphs. *J. Comput. Inf. Sci. Eng.*

Demonstration: Microgrid Digital Twin



Physical Microgrid at Naval Postgraduate School



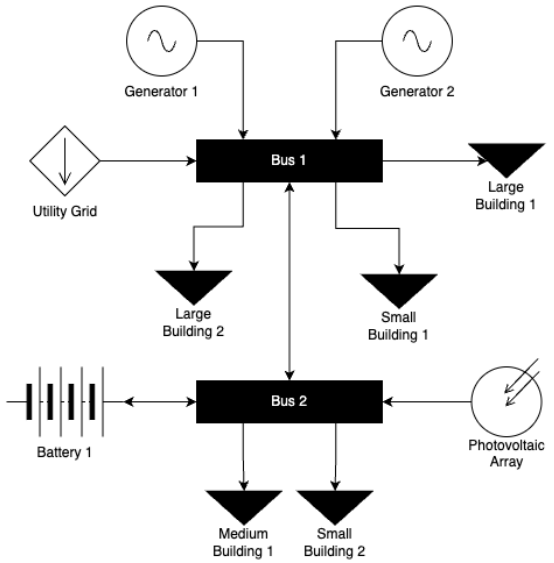


Previous Imperative Modeling of Microgrid

MG_LP_Test_v10.m

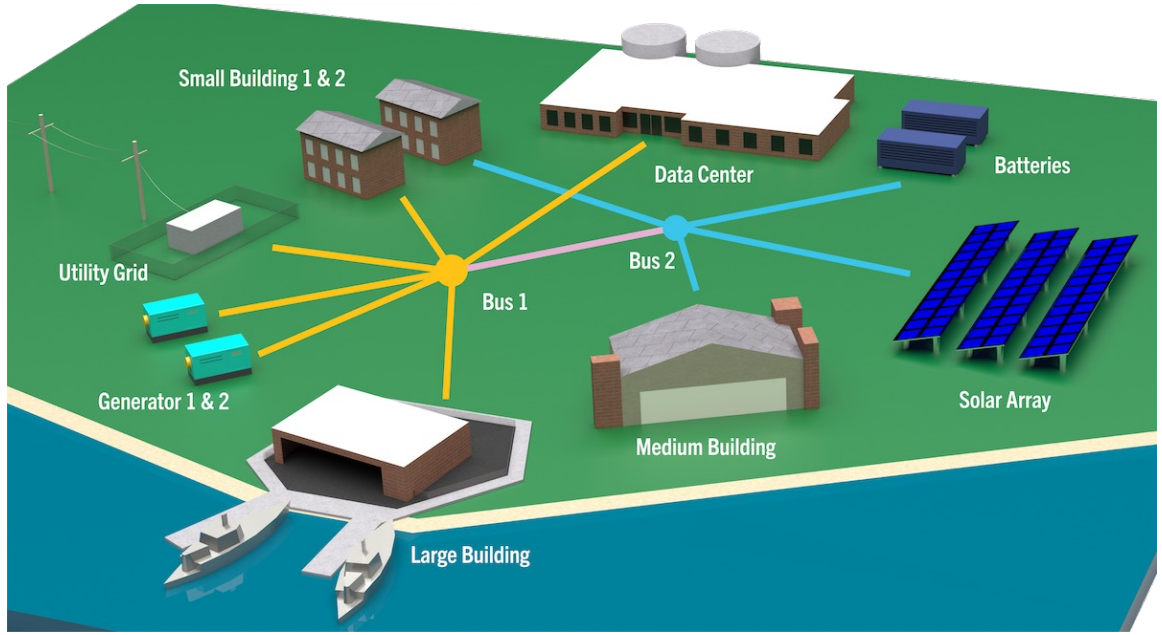
/Users/john_morris/Documents/Clemson/Research/MicrogridHg/archived_model/MG_LP_Test_v

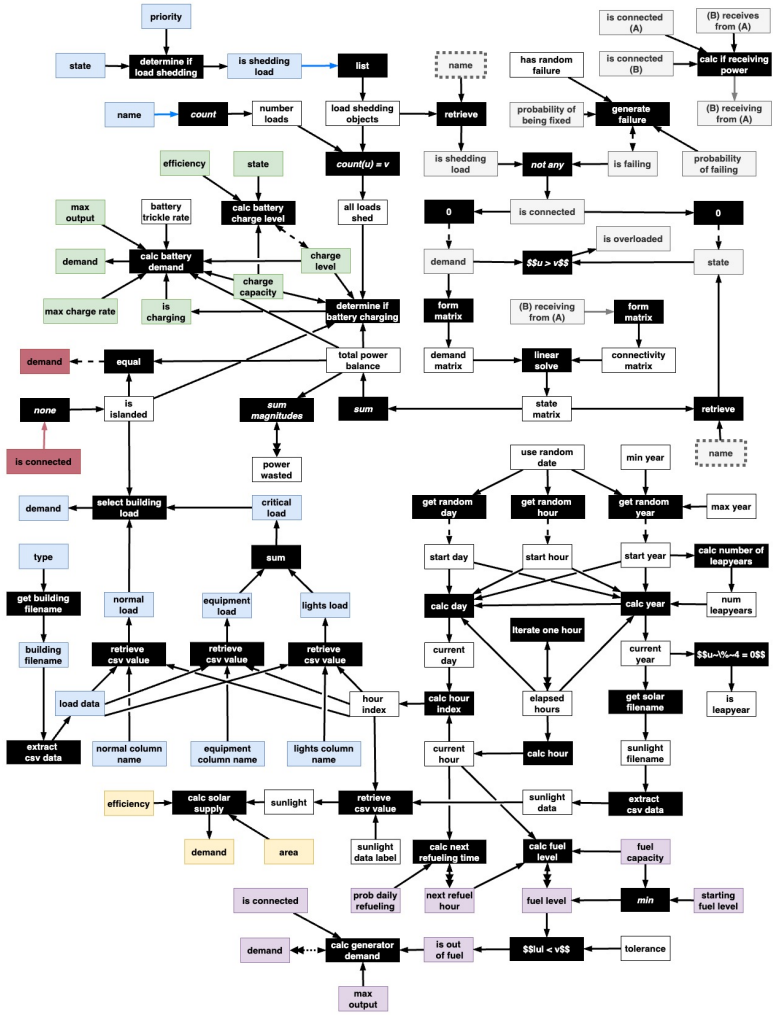
```
423         %Initial Solve
424         x(:,n) = linsolve(A2,b);
425
426         %Determine if generator demand exceeds generat
427         Gen_Demand = -x(pos.Gens,n);
428
429         if any(Gen_Demand > [Gens.Capacity]')
430             %If the B2 to B1 Buss line or BT1 line is
431             %exhausted cannot utilize ESS to make up g
432             if MG_State.B2_B1(n) == false || MG_State.
433                     || BT1_Charge(n) < 0
434                 % Below line was useful for debugging
435                 % disp("Gens Overloaded at time step "
436                 Overload.Gens = true;
437             else
438                 %Otherwise set Generators at full outp
439                 %make up unmet demand
440                 A2(6,:) = pos.Gens & MG_State{n,:};
441                 b(6) = -MG_State{n,{'B1_GEN1' 'B1_GEN2
442                 x(:,n) = linsolve(A2,b);
443             end
444         %If Generator demand is negative, then generat
445         %charge batteries with excess PV generation
446         elseif any(Gen_Demand < 0)
447             A2(6,:) = pos.Gens;
448             b(6) = 0;
449             x(:,n) = linsolve(A2,b);
450         end
451
452         % Check for Battery Output Exceeded
453         % Add 0.1 Due to Rounding Errors in Linear Solver
454         if -x(contains(LoadVars,'BT1'),n) - 0.01 > BT1.Output * (BT1_Charge(n) > 0)
455             Overload.BT1 = true;
456             % Below line was useful for debugging in single runs. Commented out
```



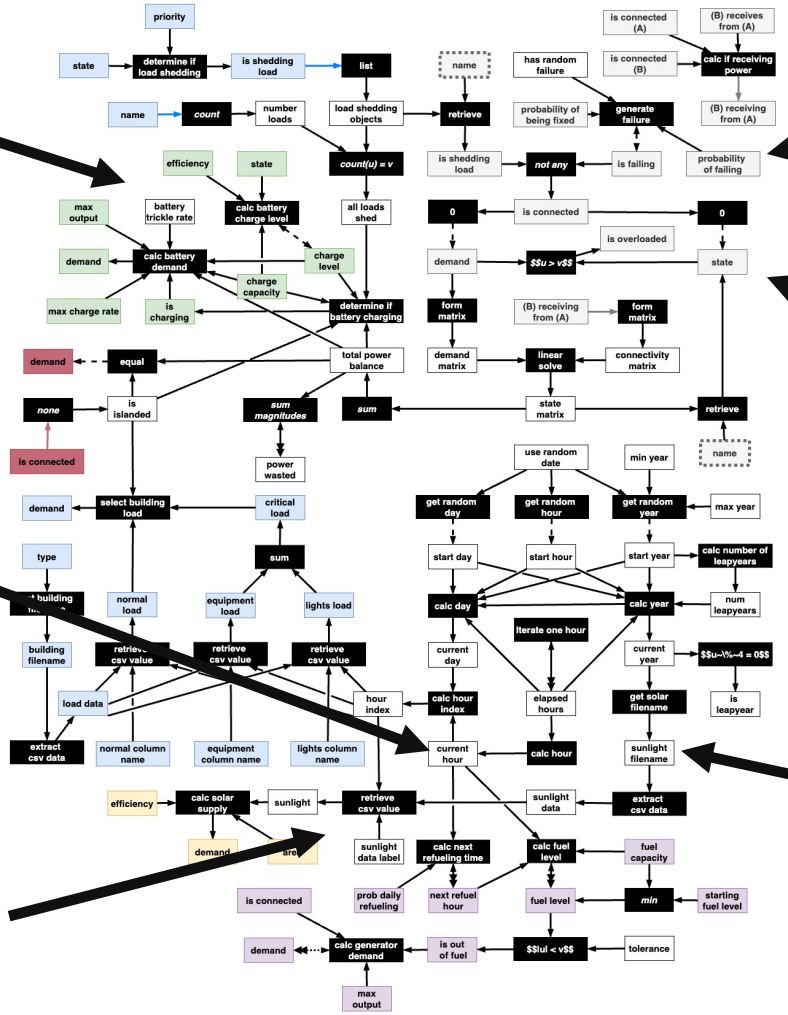
C. J. Peterson, "Systems Architecture Design and Validation Methods for Microgrid Systems," Master's thesis, Naval Postgraduate School, Monterey, CA, 2019

Transformed CHG contains 594 nodes and 334 edges





Includes all kinds of simulation and modeling



Algebraic rules

Stochastic models

Linear equation solvers

Dynamic simulation

File input/output

Database queries

Declarative simulation provided by ConstraintHg

ConstraintHg

constrainthg 0.3.2

```
pip install constrainthg
```

Kernel for building and simulating constraint hypergraphs.

Navigation

- Project description
- Release history
- Download files

Project description

ConstraintHg

JOSS 10.21105/joss.09131 | docs passing | test/linter passing | release v0.3.2 | last commit february

ConstraintHg

Search

SOFTWARE

- Quickstart
- Tutorial
- API
  - Node
  - Edge
  - Hypergraph
  - Relations Module
- About
- Demos
- Repository

CONSTRAINT HYPERGRAPHS

- CHG Overview
- Learn About CHGs

Quickstart

Use PIP to install ConstraintHg into your Python environment:

```
pip install constrainthg
```

From there you'll want to import the library into your Python script. This is a pretty typical method to use:

```
from constrainthg.hypergraph import Node, Hypergraph
import constrainthg.relations as R
```

Simple Demo

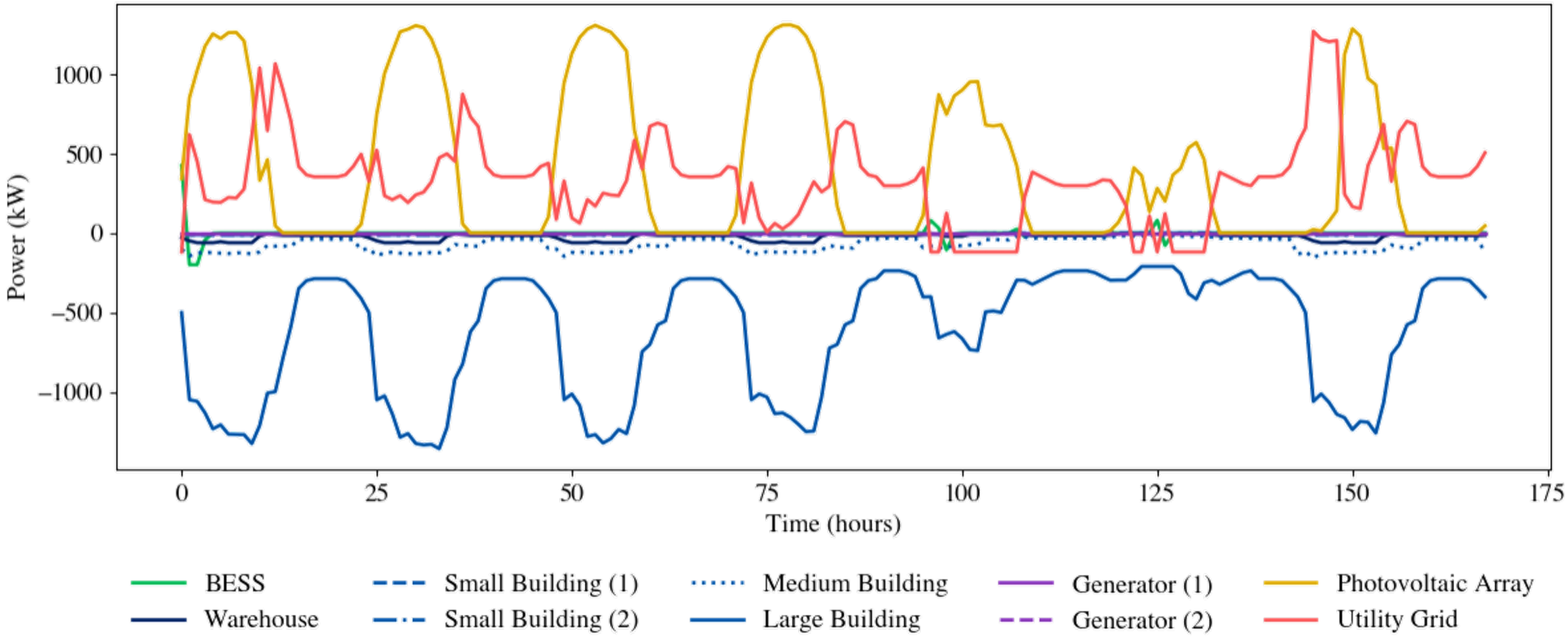
Note that this demo is found in demos/demo_basic.py

Let's build a basic constraint hypergraph of the following equations:

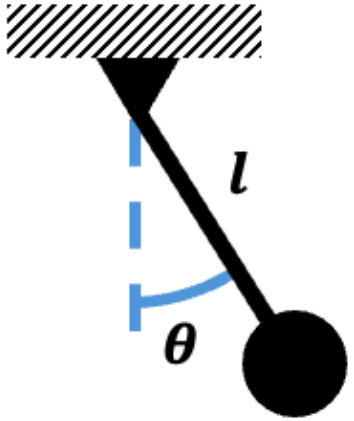
- $A + B = C$
- $A = -D$
- $B = -E$
- $D + E = F$
- $F = -C$

latest

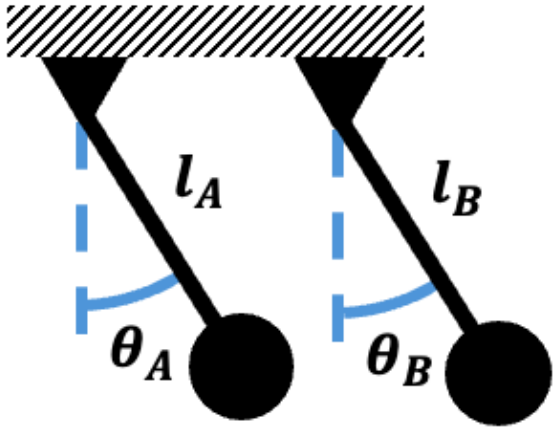
Autonomously observes and simulates the real system



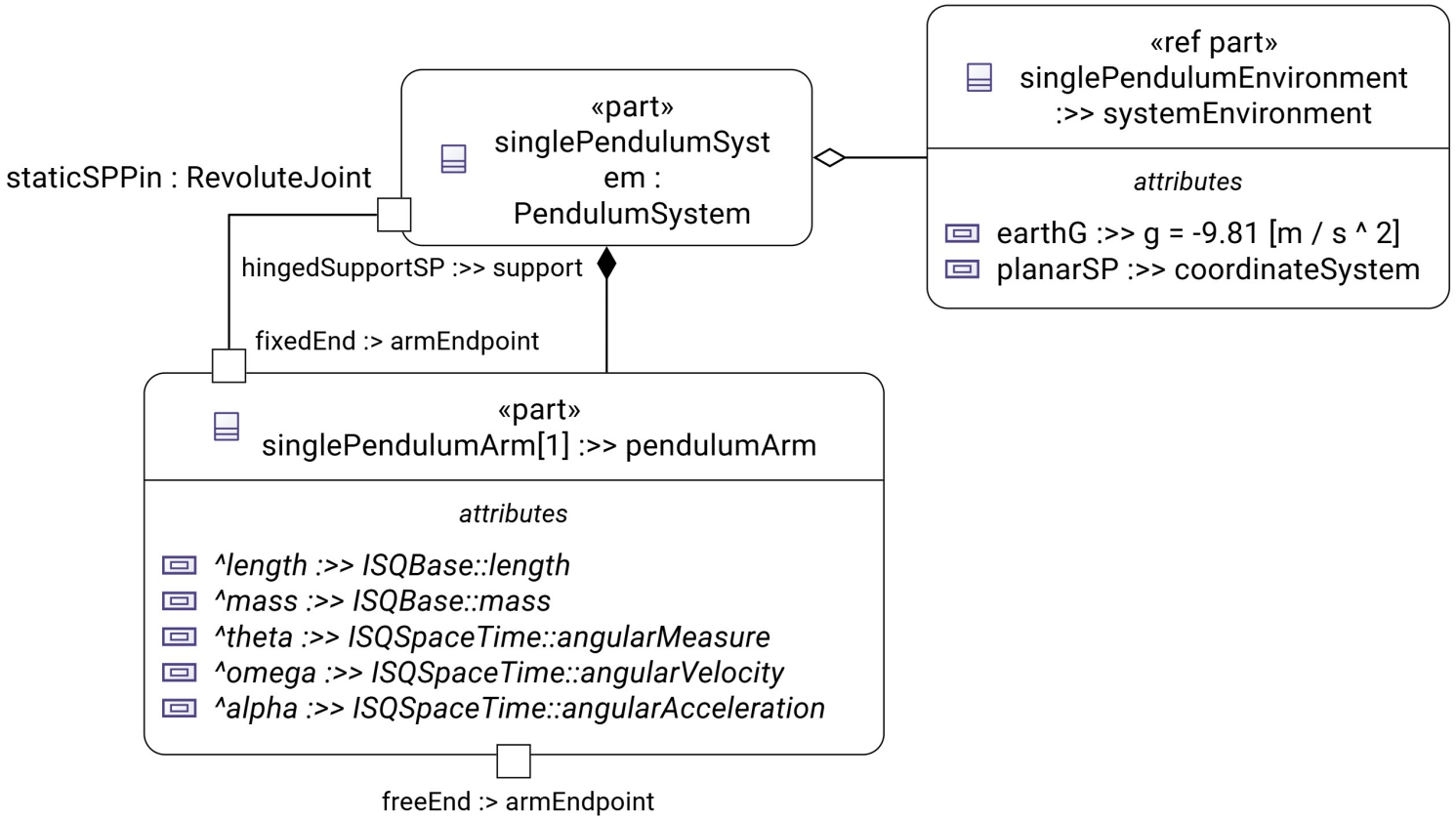
Goal is to show how aggregation reveals combined system behavior

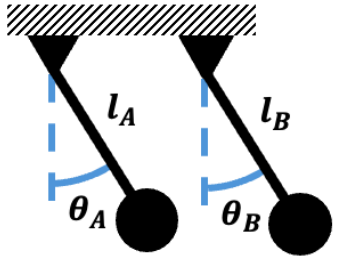


Case I: Trivial Aggregation

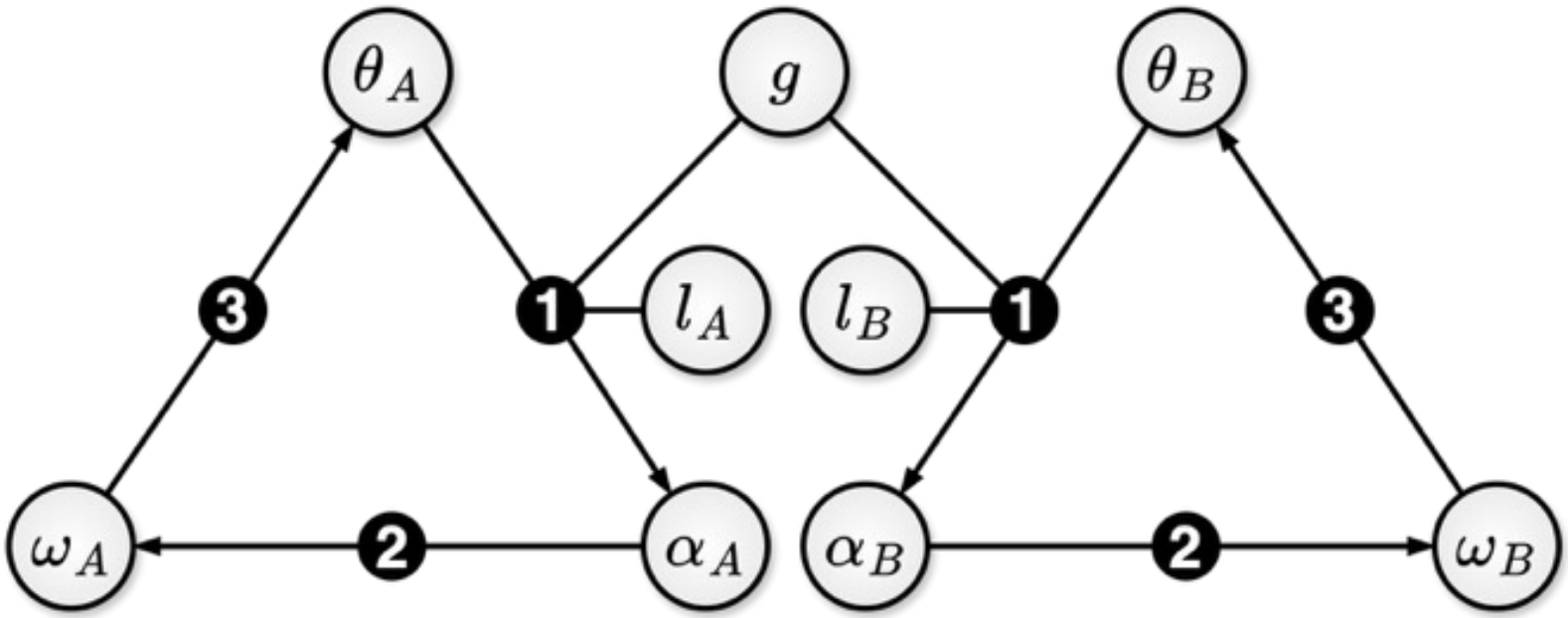


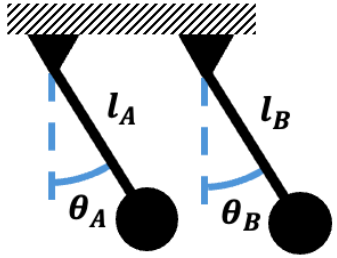
Semantics given via SysML



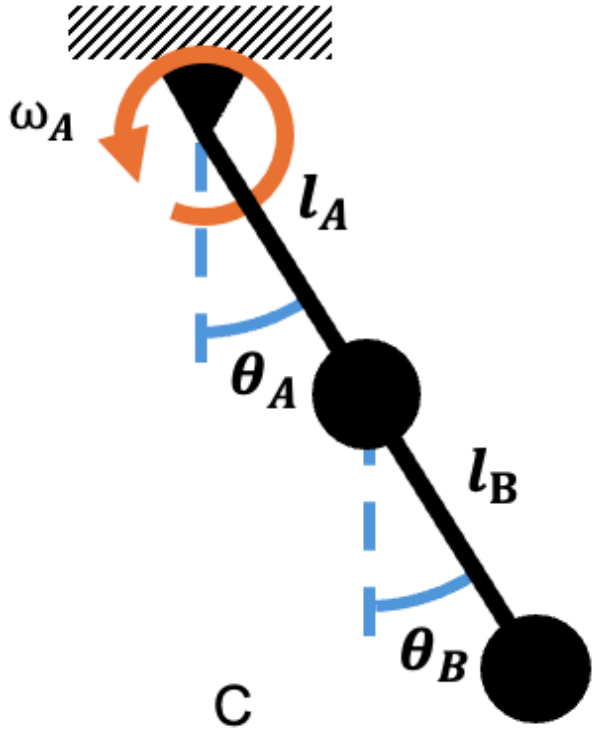


Lack of bidirectional arrows reveals no behavioral interaction between subsystems

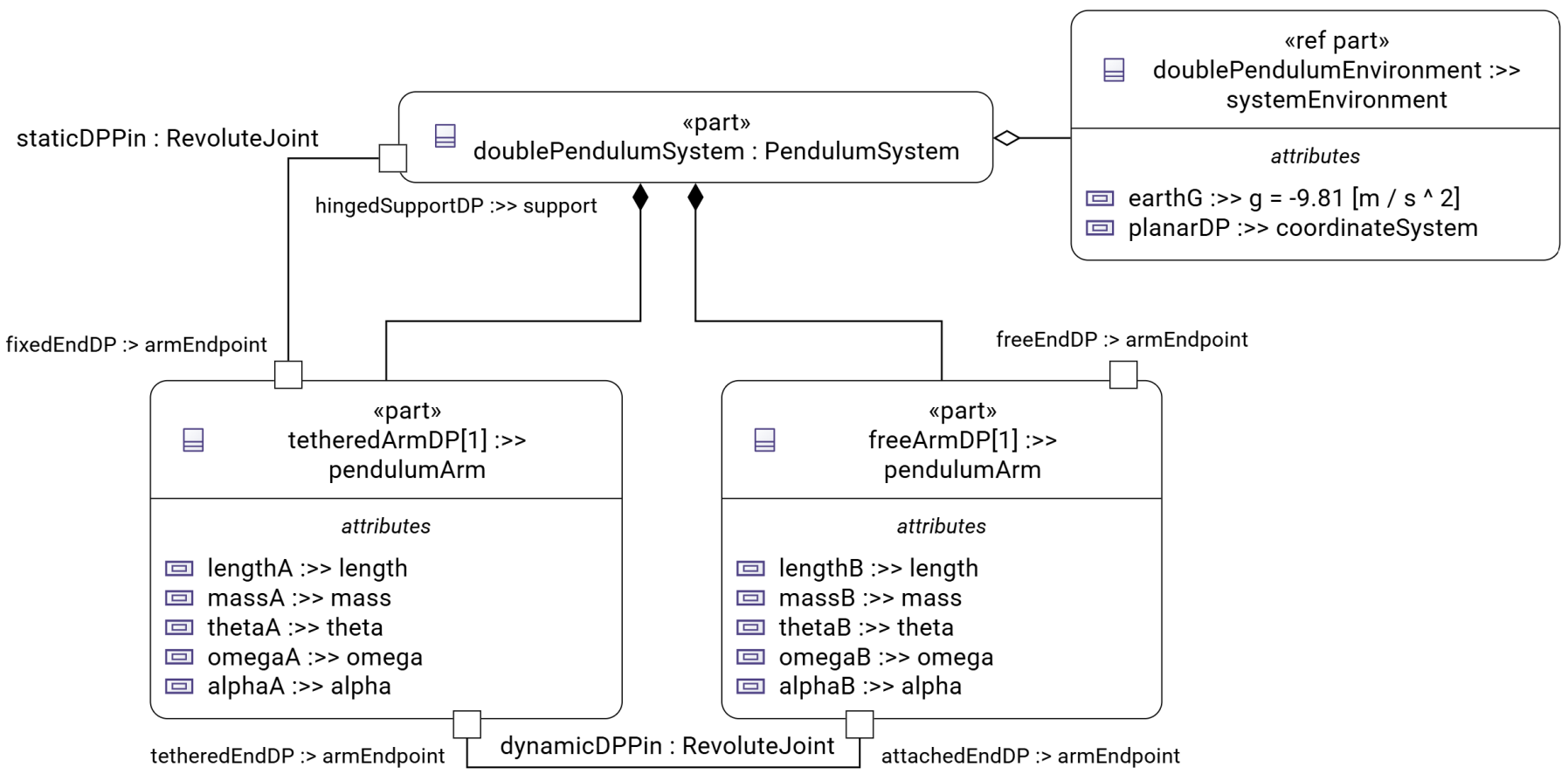


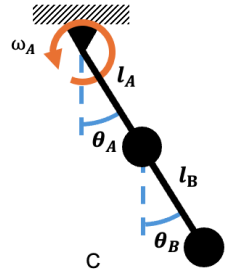


Case II: Emergent Behavior

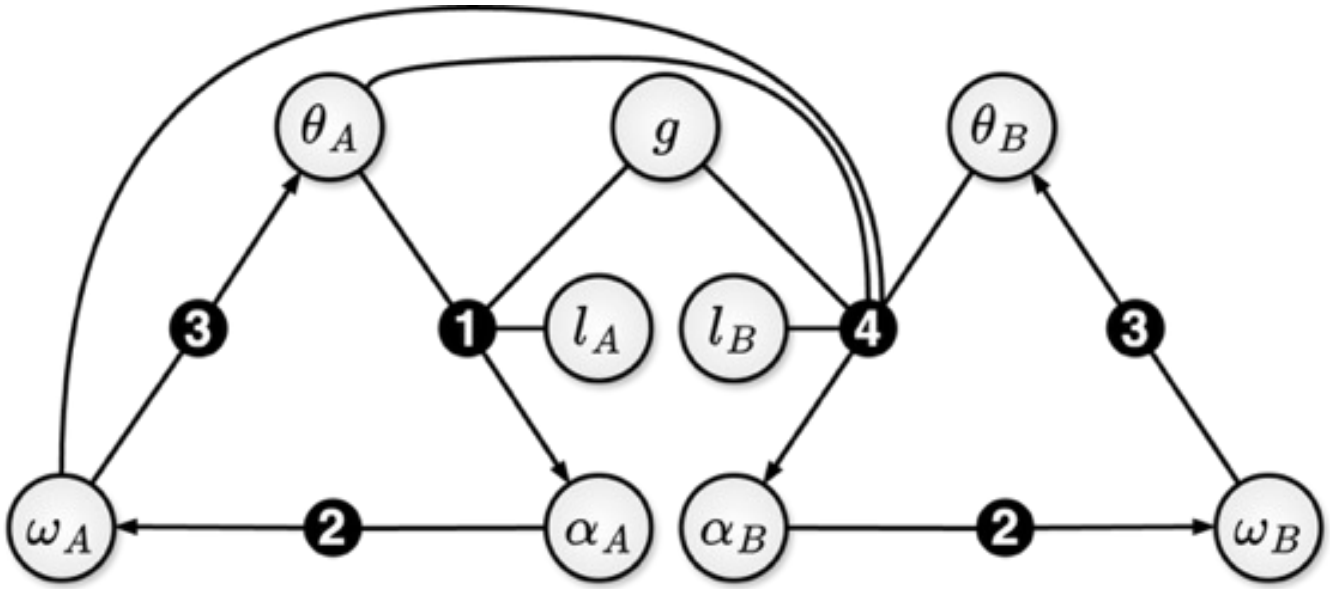


Dynamic behavior must be specified *a priori* in the descriptive model





Interactions between subsystems are given by manually prescribed behaviors



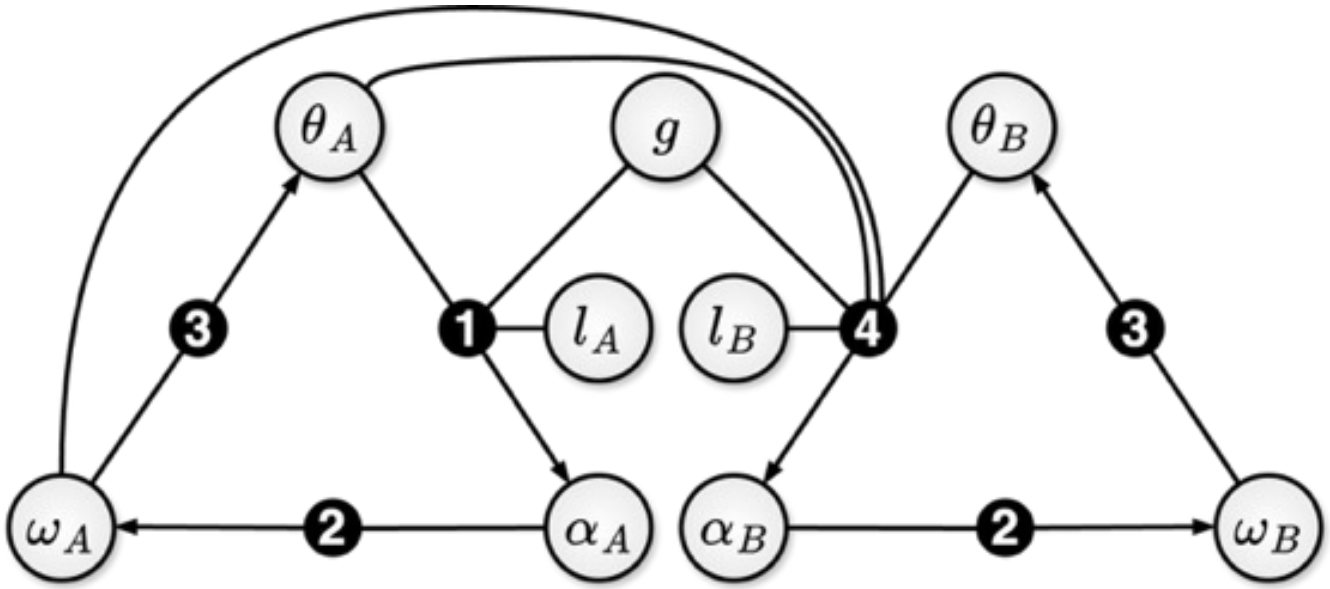
$$\alpha = \frac{g}{l} sin\,\theta \quad (1)$$

$$\omega^{i+1} = \omega^i + \alpha \Delta t \quad (2)$$

$$\theta^{i+1} = \theta^i + \omega \Delta t \quad (3)$$

$$\alpha_B = -cos\,\theta_B\,(\alpha_A\,cos\,\theta_A - \omega_A^2\,sin\,\theta_A) - sin\,\theta_B\,(\alpha_A\,sin\,\theta_A + \omega_A^2\,cos\,\theta_A + g) \quad (4)$$

Because the graph is the same as Case I, it cannot be automatically differentiated



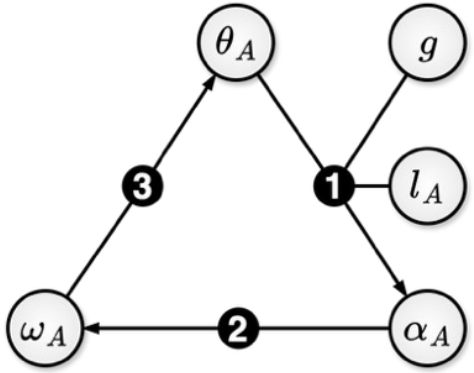
$$\alpha = \frac{g}{l} sin\,\theta \quad (1)$$

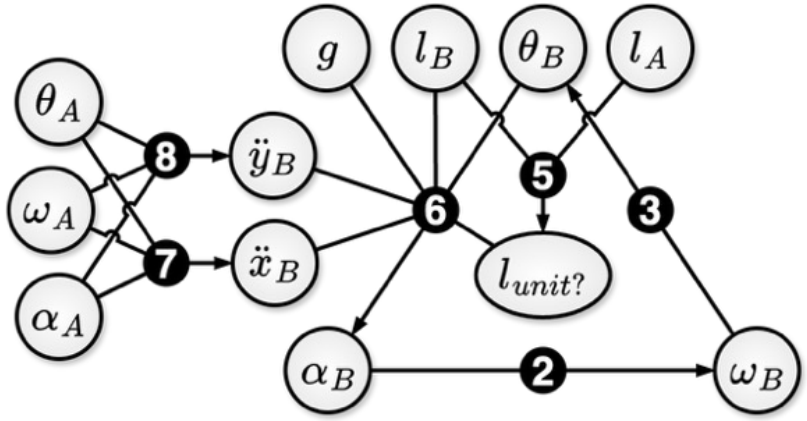
$$\omega^{i+1} = \omega^i + \alpha \Delta t \quad (2)$$

$$\theta^{i+1} = \theta^i + \omega \Delta t \quad (3)$$

$$\alpha_B = -cos\,\theta_B\,(\alpha_A\,cos\,\theta_A - \omega_A^2\,sin\,\theta_A) - sin\,\theta_B\,(\alpha_A\,sin\,\theta_A + \omega_A^2\,cos\,\theta_A + g) \quad (4)$$

Because the graph is the same as Case I, it cannot be automatically differentiated





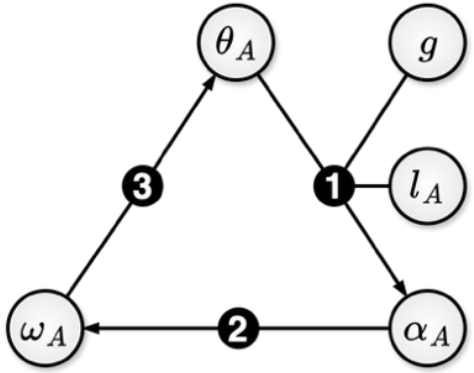
$$l_{unit?} \Leftarrow (l_A = 1 \wedge l_B = 1) \quad \textbf{5}$$

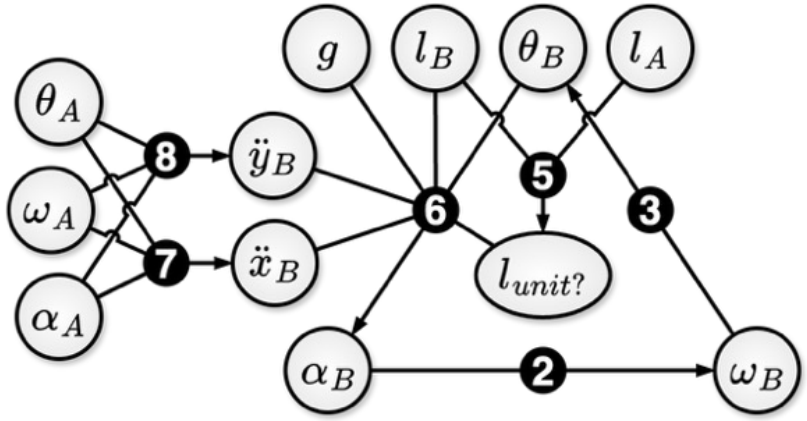
$$\alpha_B = -\ddot{x}_B \cos\theta_B - \sin\theta_B\,(\ddot{y}_B + g) \quad \textbf{6}$$

$$\ddot{x}_B = \alpha_A \cos\theta_A - \omega_A^2 \sin\theta_A \quad \textbf{7}$$

$$\ddot{y}_B = \alpha_A \sin\theta_A + \omega_A^2 \cos\theta_A \quad \textbf{8}$$

Because the graph is the same as Case I, it cannot be automatically differentiated





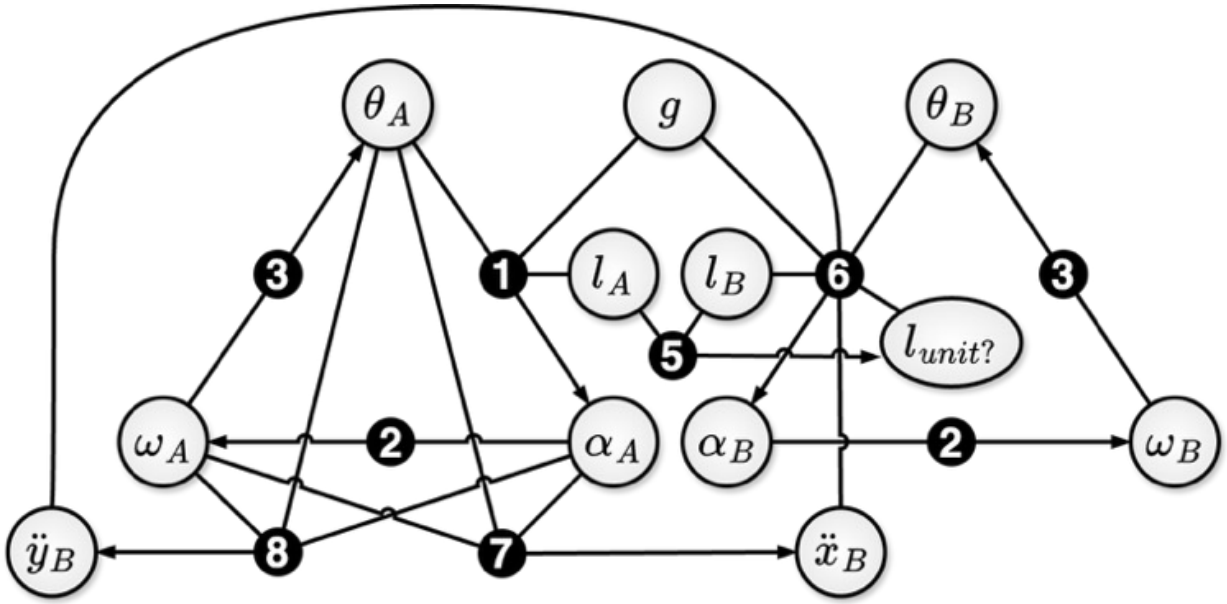
$$l_{unit?} \Leftarrow (l_A = 1 \wedge l_B = 1) \quad \textbf{5}$$

$$\alpha_B = -\ddot{x}_B \cos\theta_B - \sin\theta_B\,(\ddot{y}_B + g) \quad \textbf{6}$$

$$\ddot{x}_B = \alpha_A \cos\theta_A - \omega_A^2 \sin\theta_A \quad \textbf{7}$$

$$\ddot{y}_B = \alpha_A \sin\theta_A + \omega_A^2 \cos\theta_A \quad \textbf{8}$$

The graphs aggregate immediately and reveal the emergent chaotic behavior



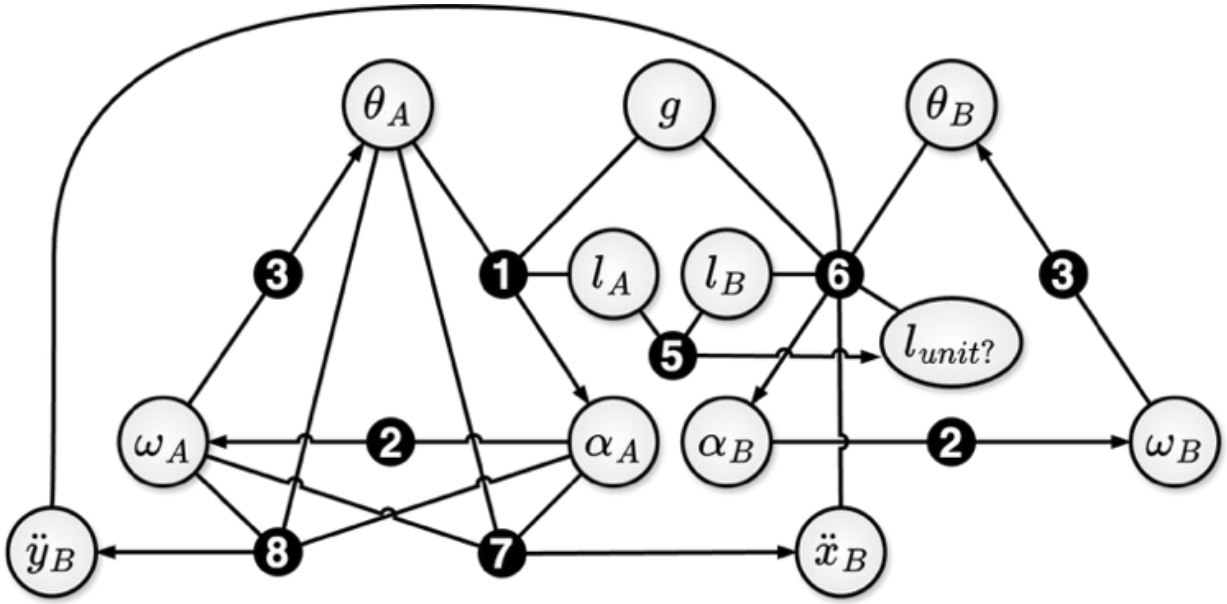
$$l_{unit?} \Leftarrow (l_A = 1 \wedge l_B = 1) \quad (5)$$

$$\alpha_B = -\ddot{x}_B \cos\theta_B - \sin\theta_B\,(\ddot{y}_B + g) \quad (6)$$

$$\ddot{x}_B = \alpha_A \cos\theta_A - \omega_A^2 \sin\theta_A \quad (7)$$

$$\ddot{y}_B = \alpha_A \sin\theta_A + \omega_A^2 \cos\theta_A \quad (8)$$

No new nodes are added in the merger, only edges



$$l_{unit?} \Leftarrow (l_A = 1 \wedge l_B = 1) \quad (5)$$

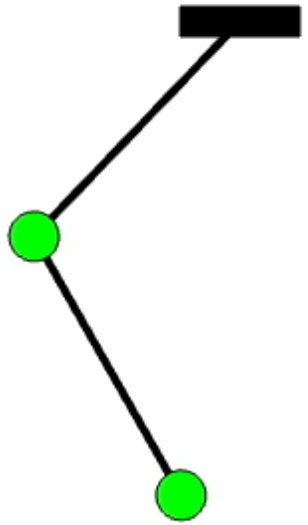
$$\alpha_B = -\ddot{x}_B \cos\theta_B - \sin\theta_B\,(\ddot{y}_B + g) \quad (6)$$

$$\ddot{x}_B = \alpha_A \cos\theta_A - \omega_A^2 \sin\theta_A \quad (7)$$

$$\ddot{y}_B = \alpha_A \sin\theta_A + \omega_A^2 \cos\theta_A \quad (8)$$

Takeaways

- Systems can be aggregated through a CHG by merging models along common variables (nodes)
- Problems arise from a lack of specificity in the supersystem model
- Increasing specificity *a priori* such as with a descriptive model enables aggregational interoperability



Digital twin interoperability requires frameworks that can specify behaviors

Information about
Constraint Hypergraphs

Paper Posted Online

John Morris
john.h.morris@uah.edu