

Practical Methods for Robust, Interoperable Digital Twins

John Morris

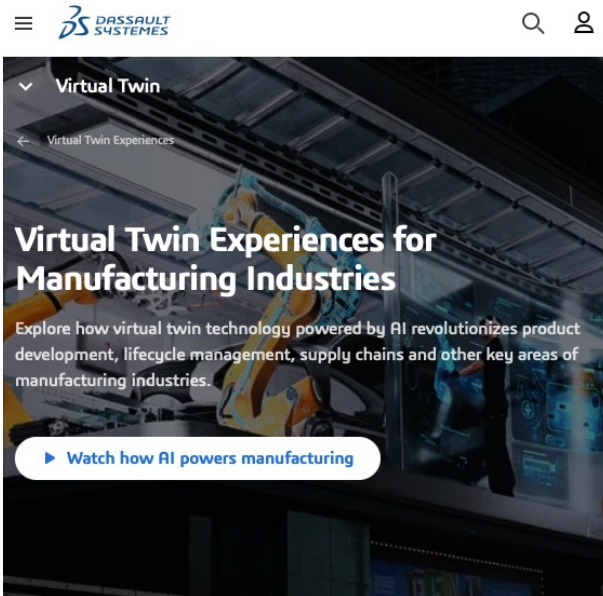
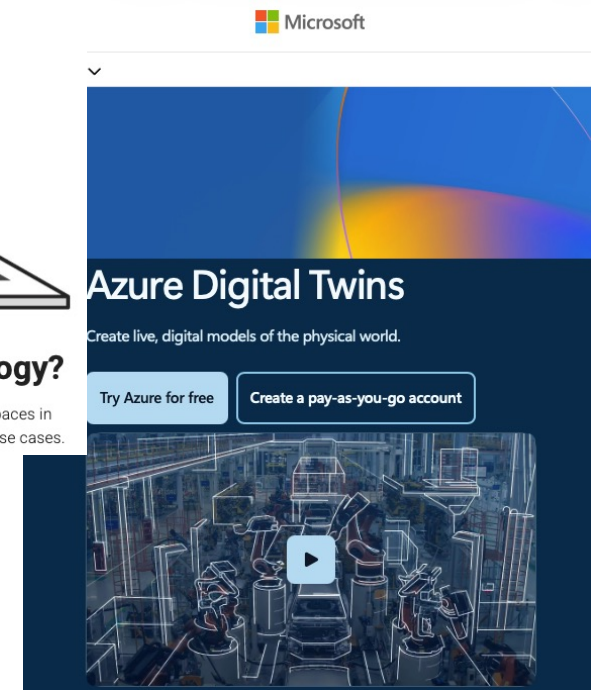
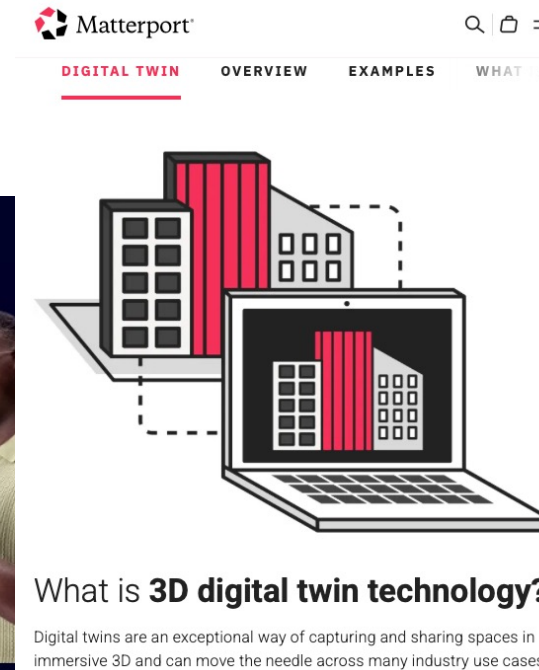
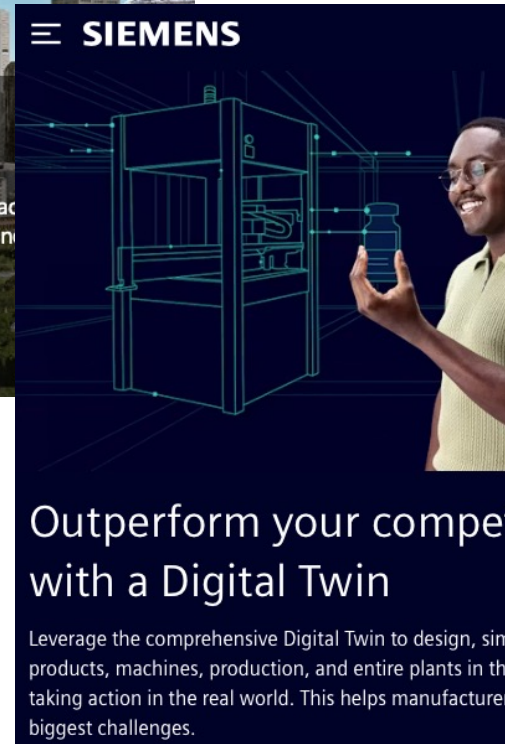
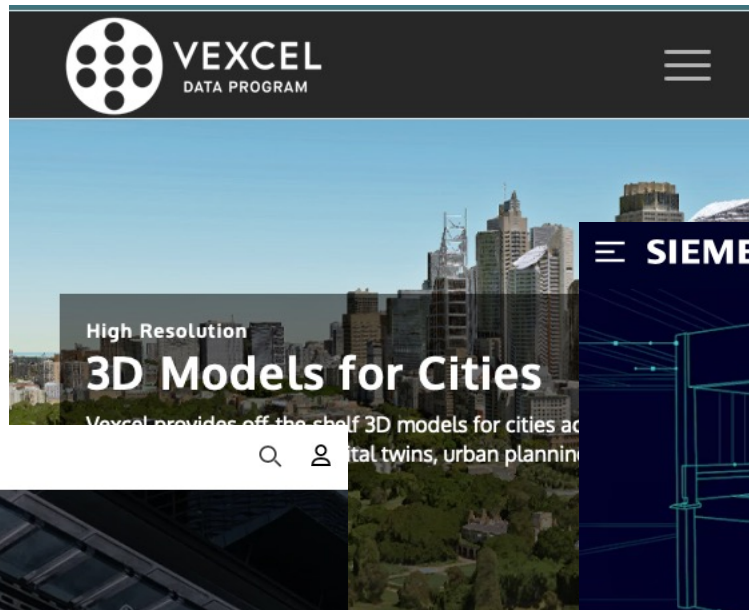
23 Aug 2026

Assistant Professor | Dept. of Industrial & Systems Engineering and Engineering Management
The University of Alabama in Huntsville



To industry, a digital twin is a synchronized virtual representation of some physical system

Digital twins need some pedantism

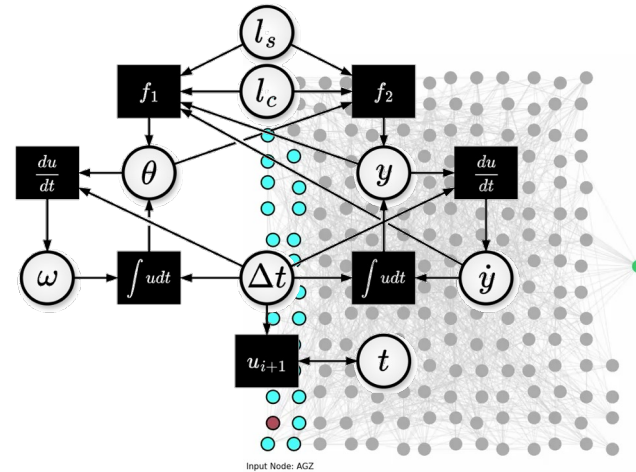


Systems theory provides the frameworks needed to define digital twins

Define digital twins from a systems perspective



Show frameworks for next-generation DTs



Uses and limitations with current DTs



Demonstration

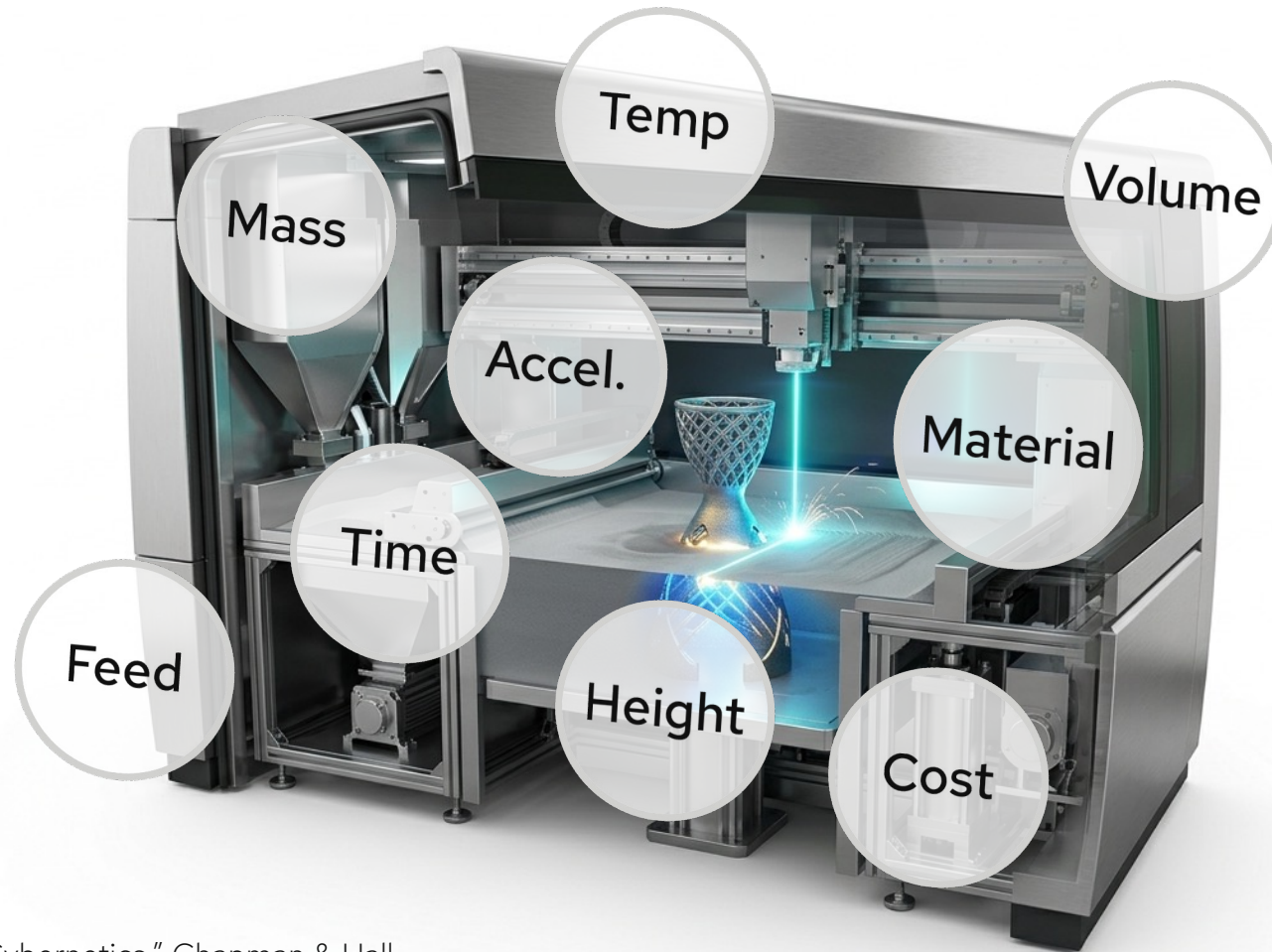
What is a Digital Twin

Robust, Interoperable Digital Twins

A system is something that *is*; it is *identifiable* because it behaves different than something else

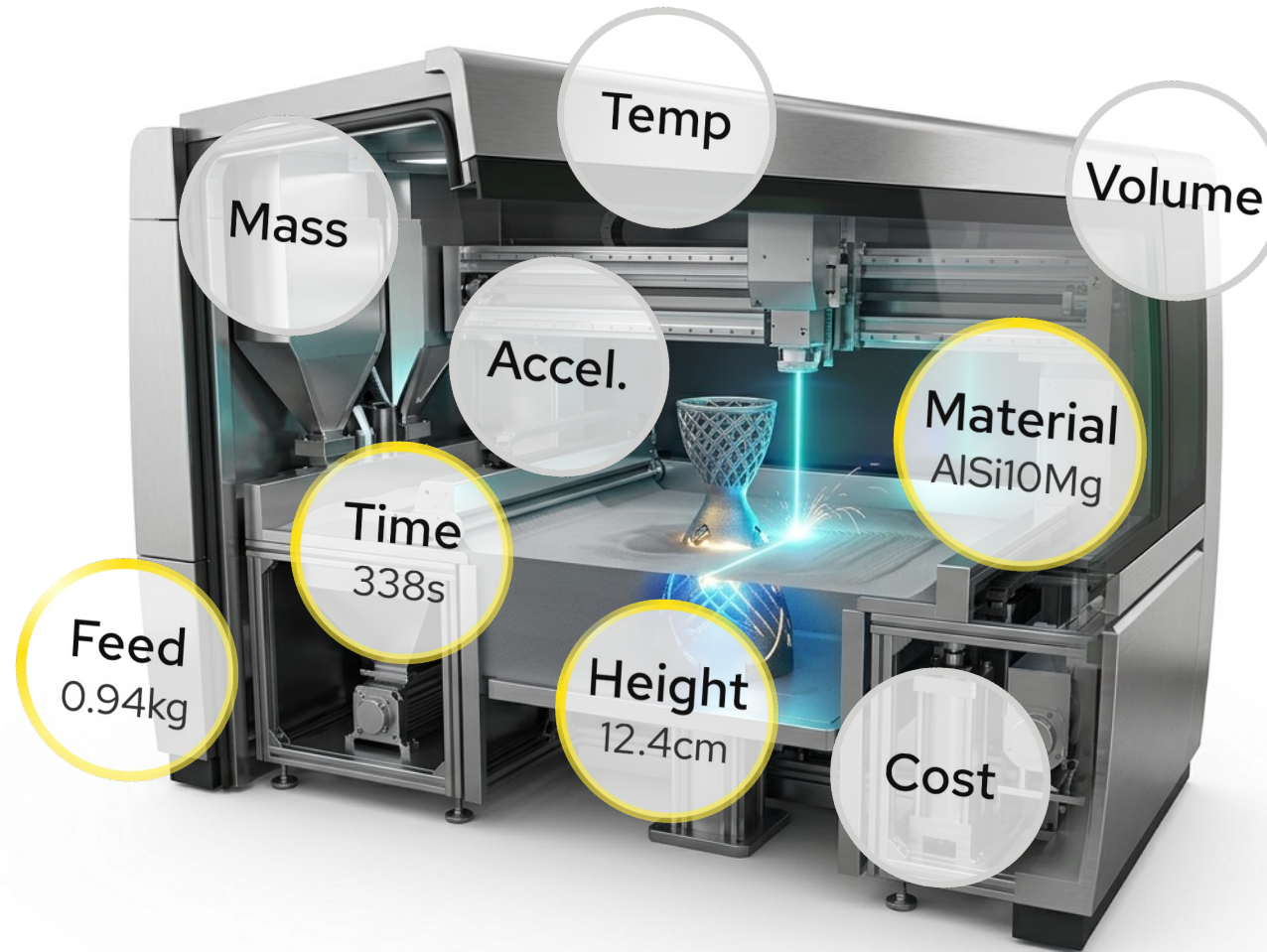


“A system is not a thing but a list of variables” – W. Ross Ashby

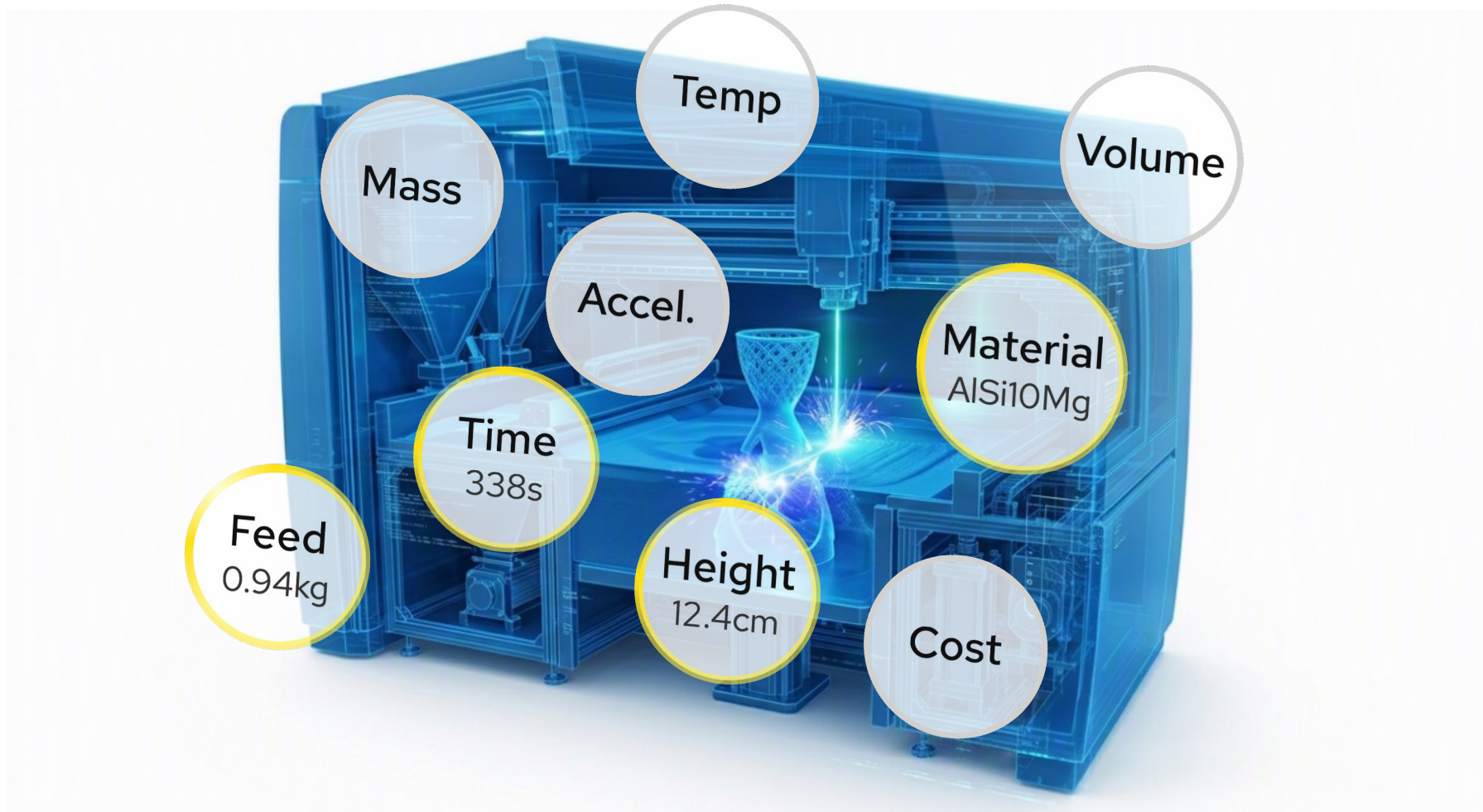


Giving a value to a variable is known as observation

Observations can be given by measurement or simulation



The value of a digital twin is providing observations for a physical system



The only role of a digital twin is to provide observations for some System of Interest (SOI)

Monitor:

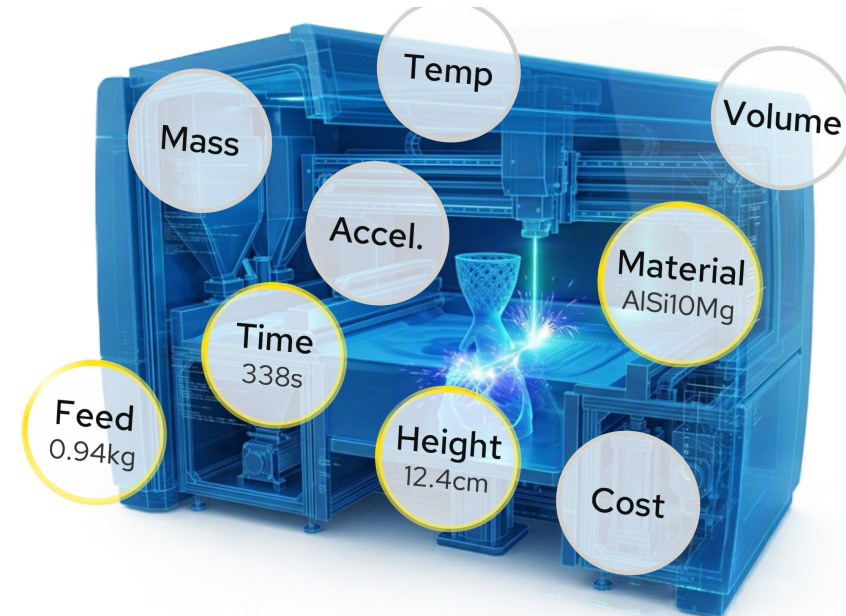
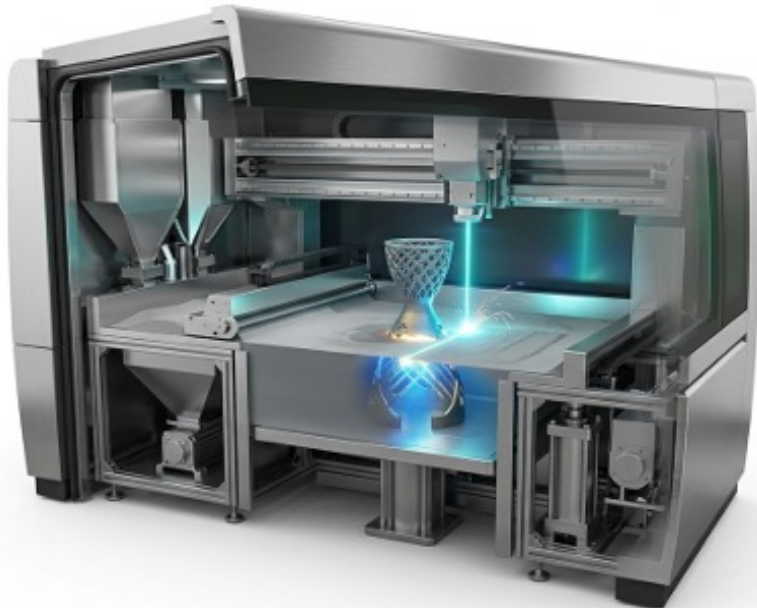
Measure SOI

Analyze:

Process measurements from SOI

Predict:

Use measurements to simulate SOI



The value of a digital twin is providing information about the SOI

How can you use information about some System of Interest?

**Predictive
Maintenance**

**Performance
Forecasting**

Fault Traversal

Role Optimization

Automation

Risk Analysis

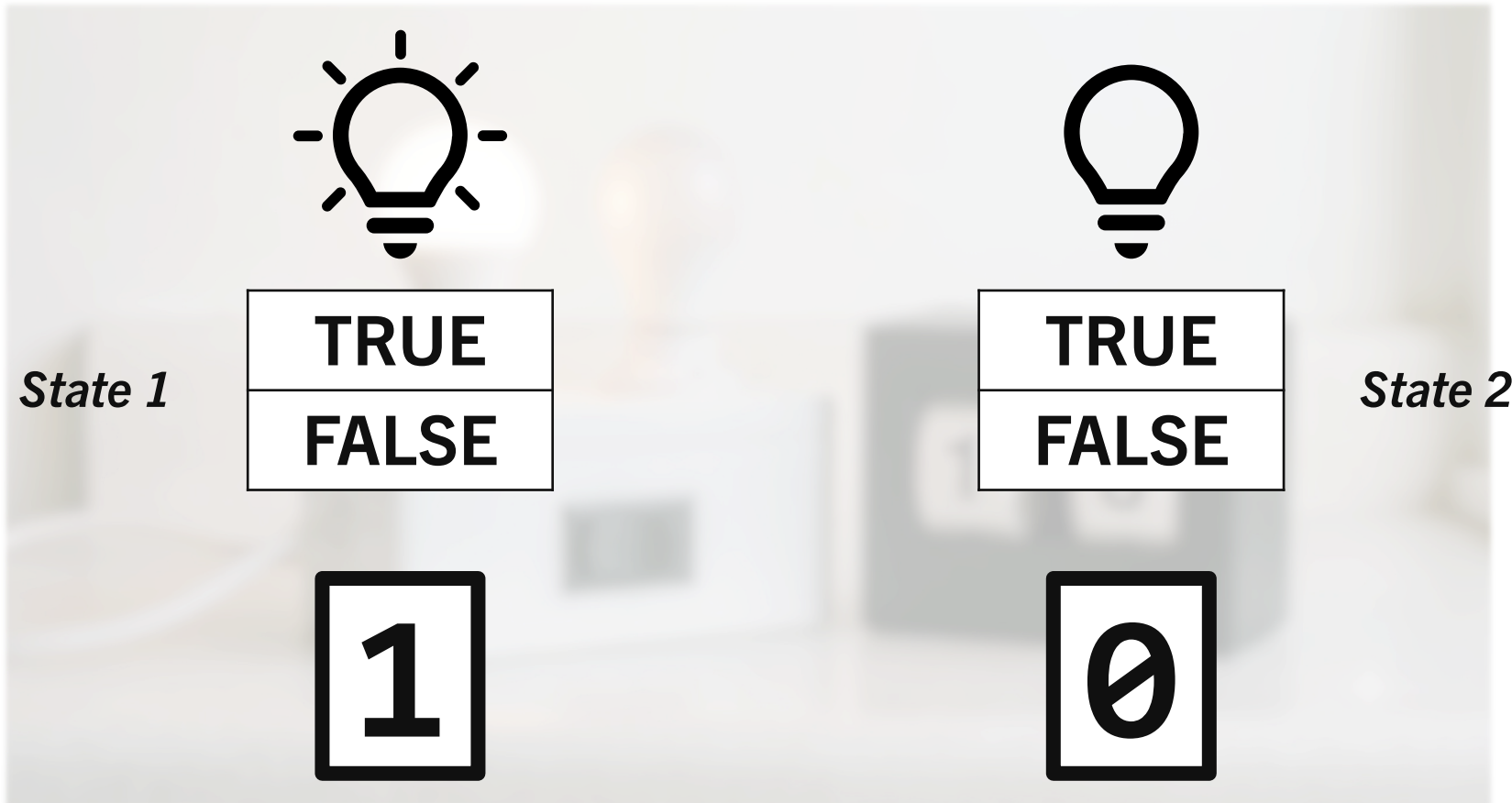
**Trade-Space
Exploration**

**Informed
Decision-Making**

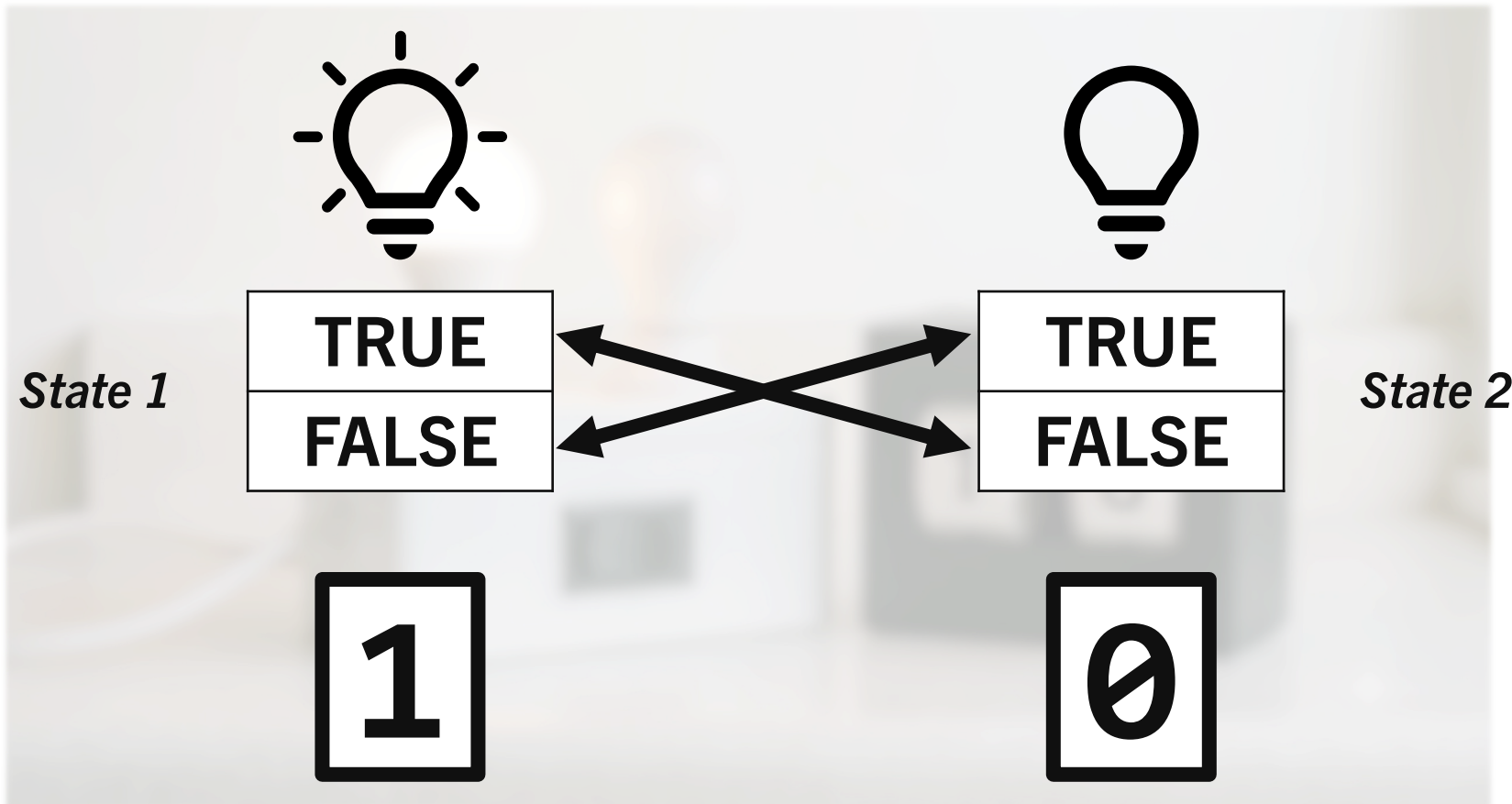
A digital twin is a system configured to enable observation of some analogous system



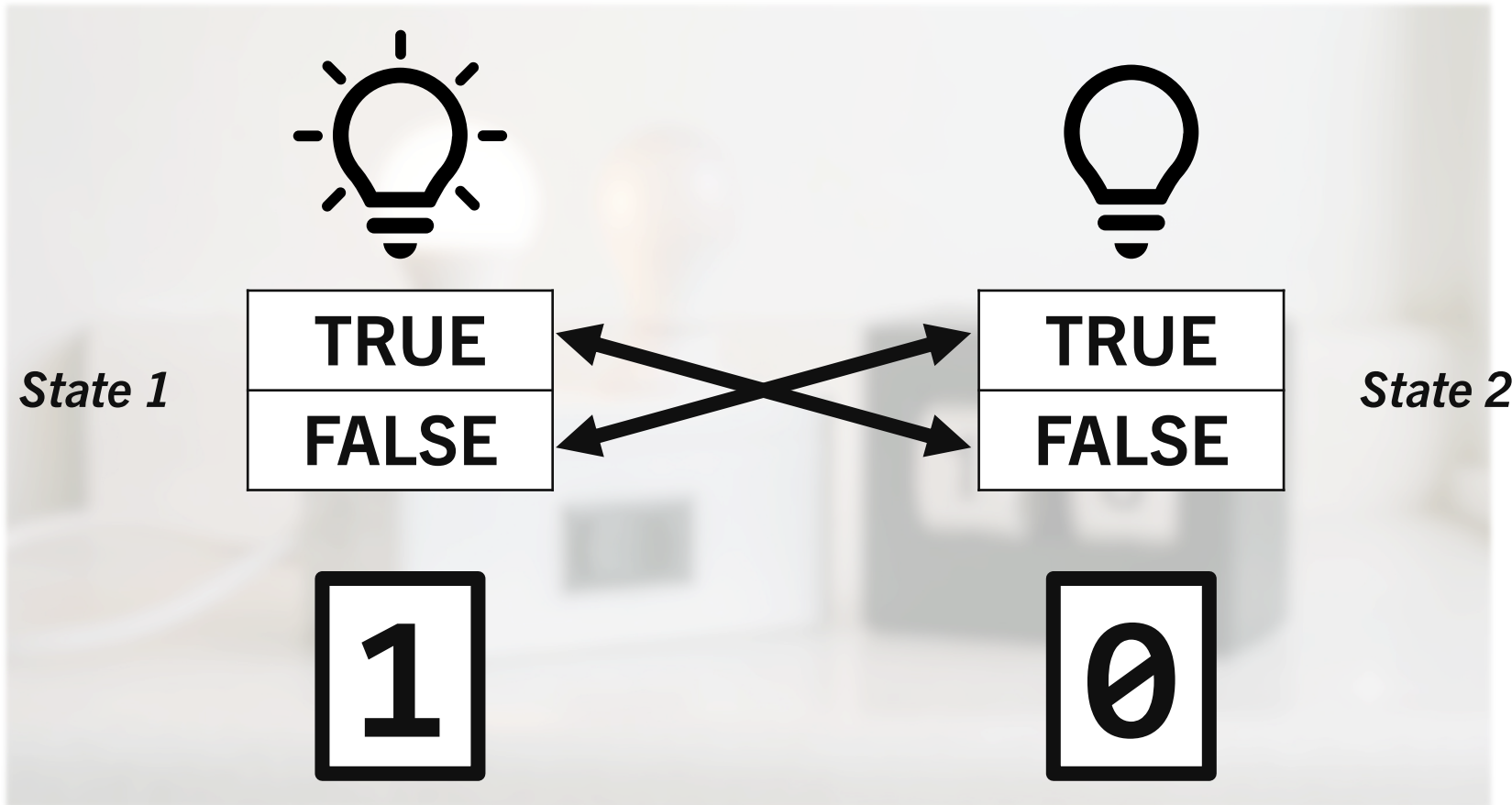
Both systems are scoped to a similar set of variables



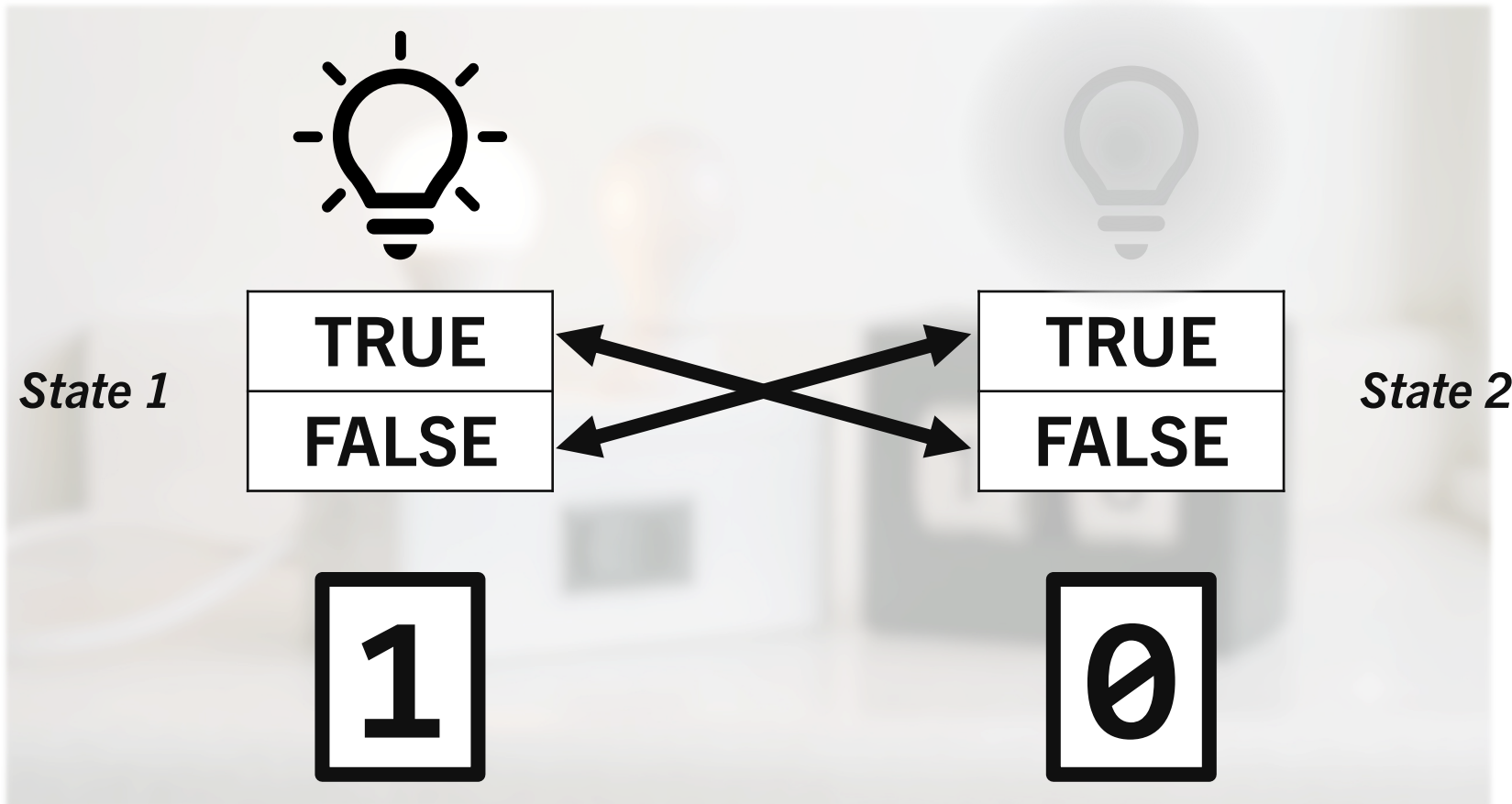
Both systems also share the same behavioral constraint



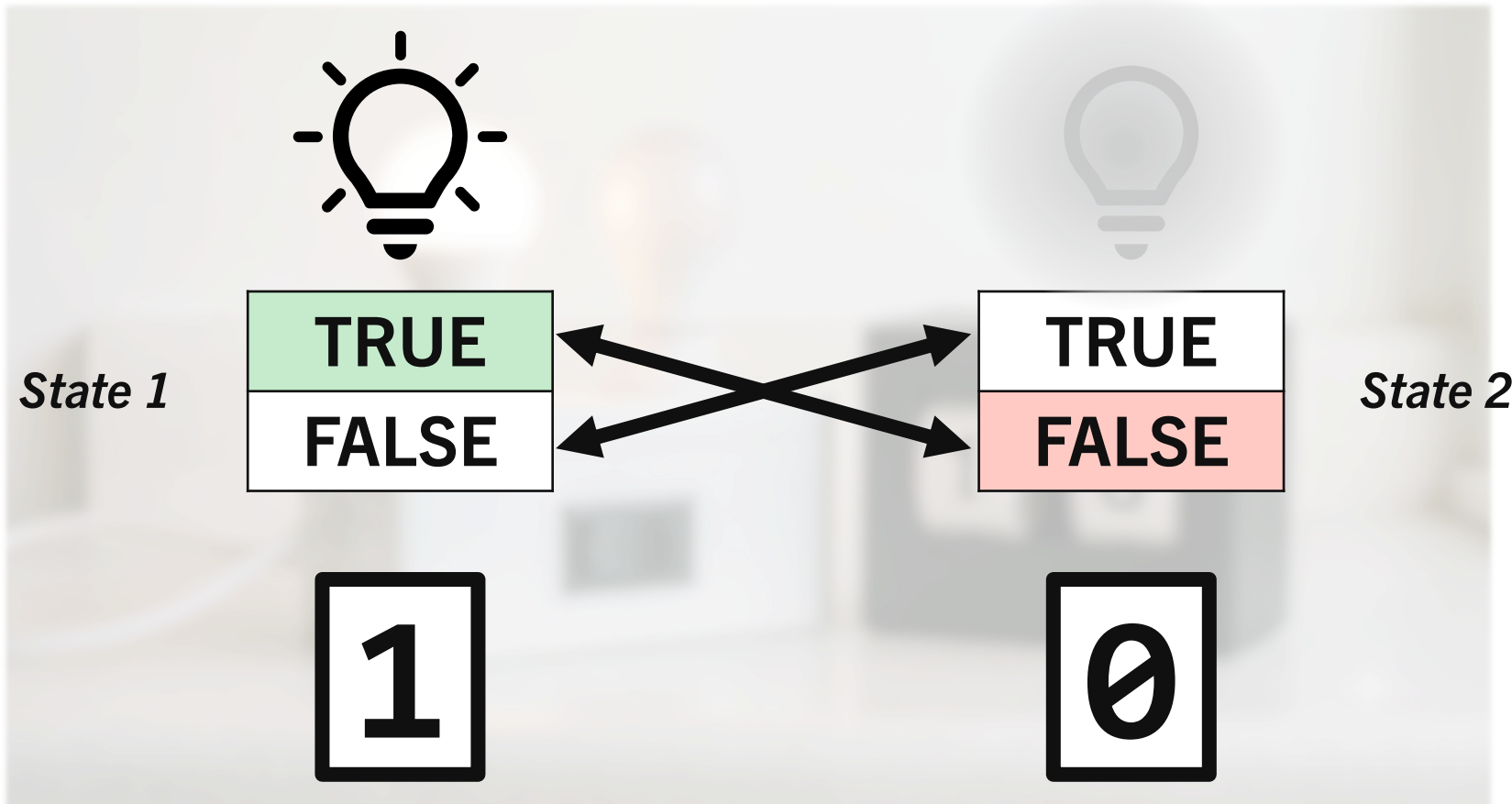
To simulate an observation requires knowledge of the system's behavior



We can simulate the SOI by measuring analogous outputs off the twin



We can simulate the SOI by measuring analogous outputs off the twin



A digital twin is a system configured to enable observation of some analogous system



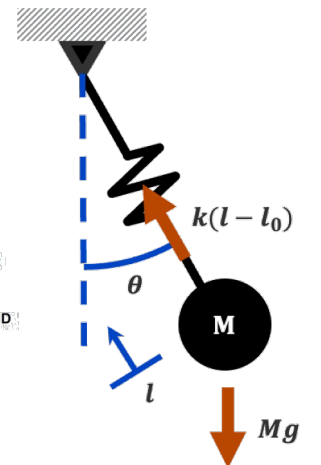
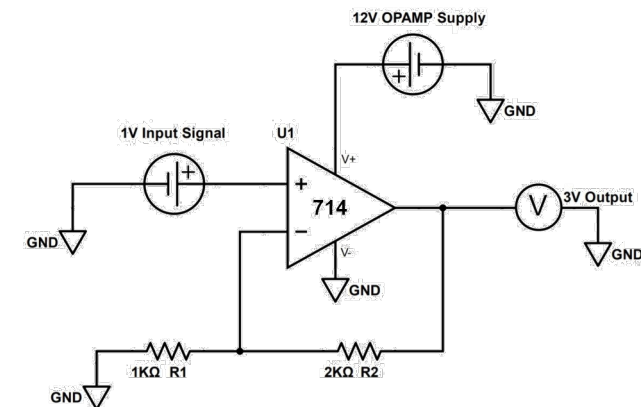
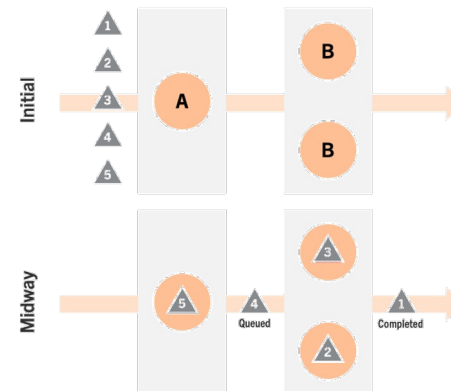
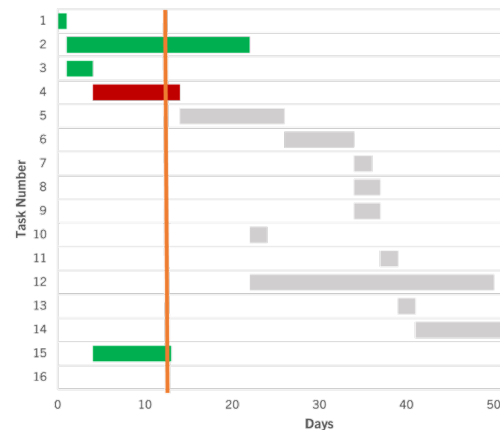
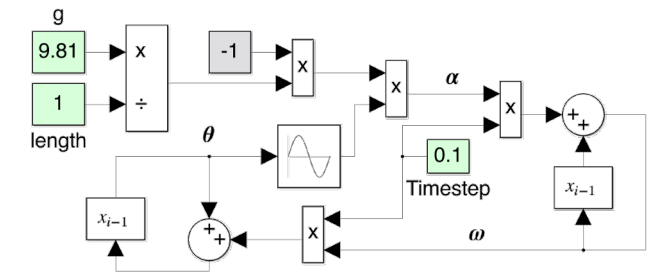
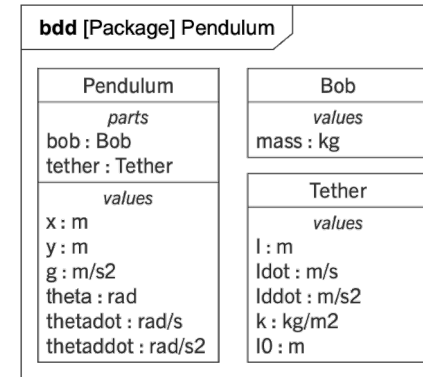
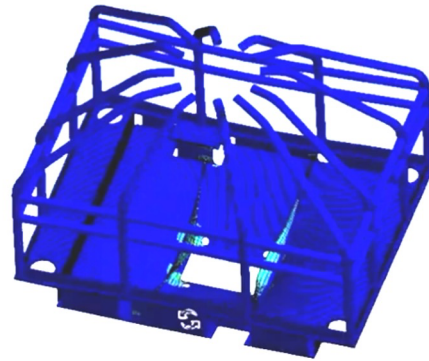
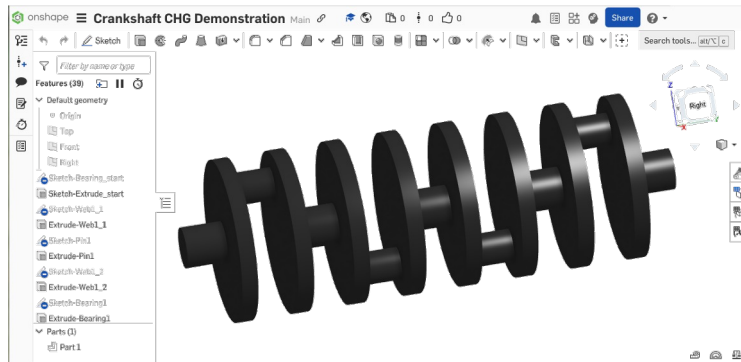
Synchronization mechanisms are not part of the digital twin



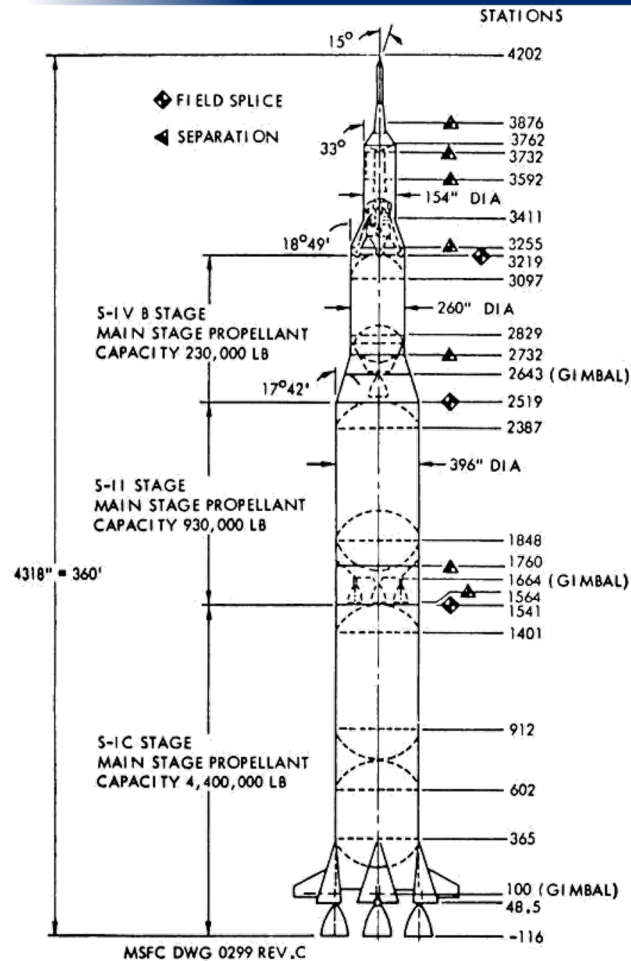


**Problems with digital twins
stem from maintaining meaning
when connecting information
for complex systems**

Understanding complex systems requires integration of heterogeneous information and models



Issues with modeling—such as uncertainty, verification, or ignorance—are not unique to digital twins



Saturn V Reference Dimensions

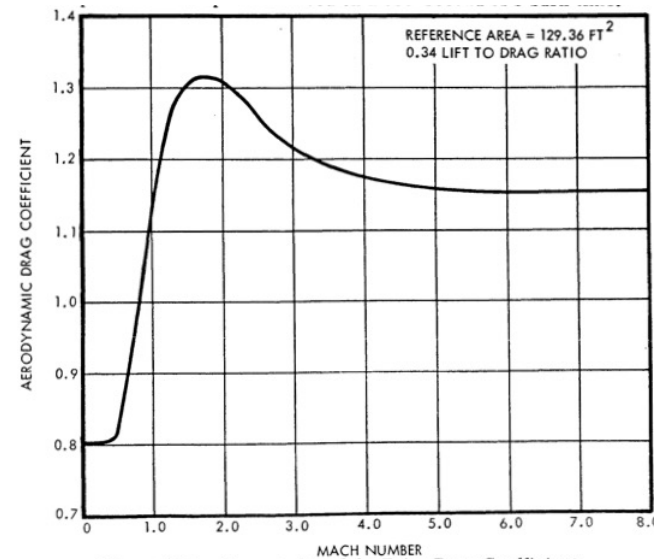
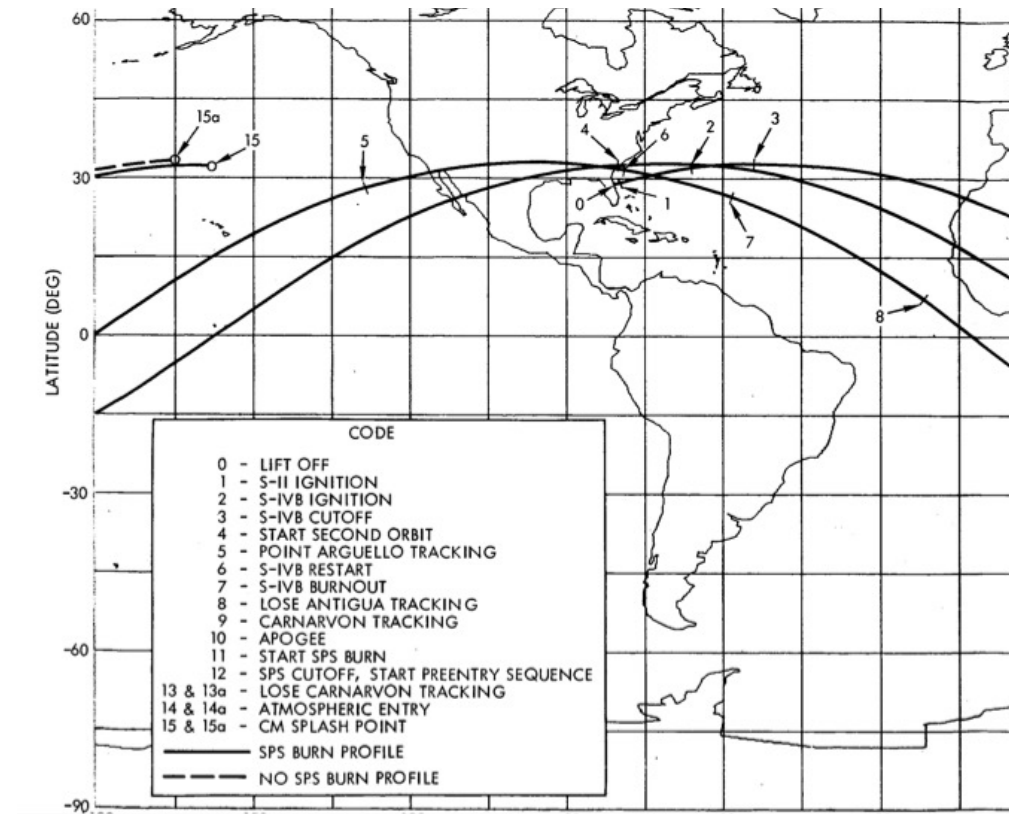
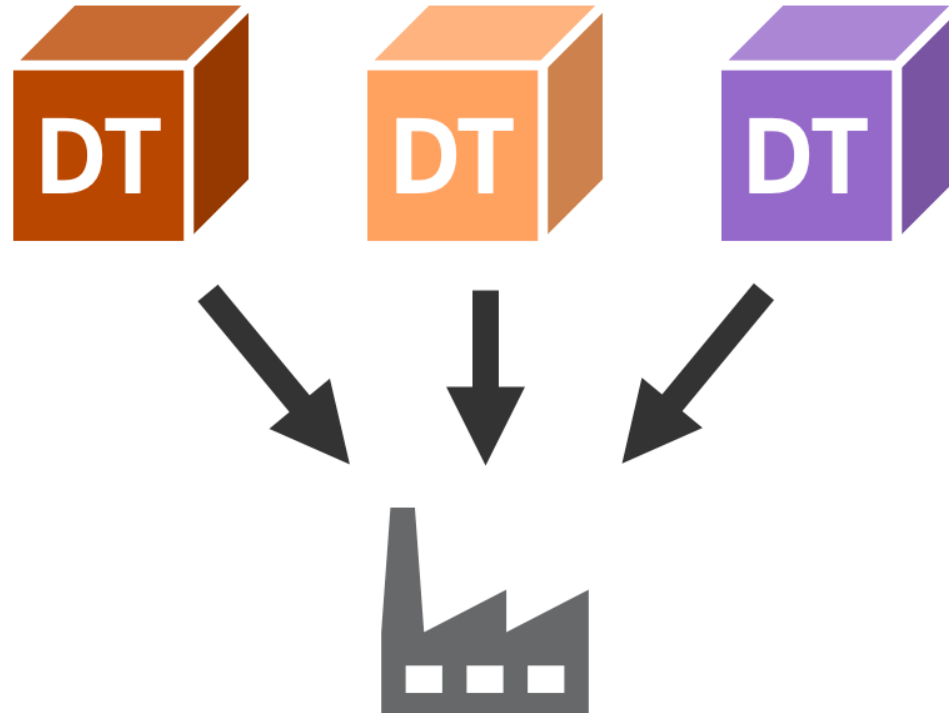


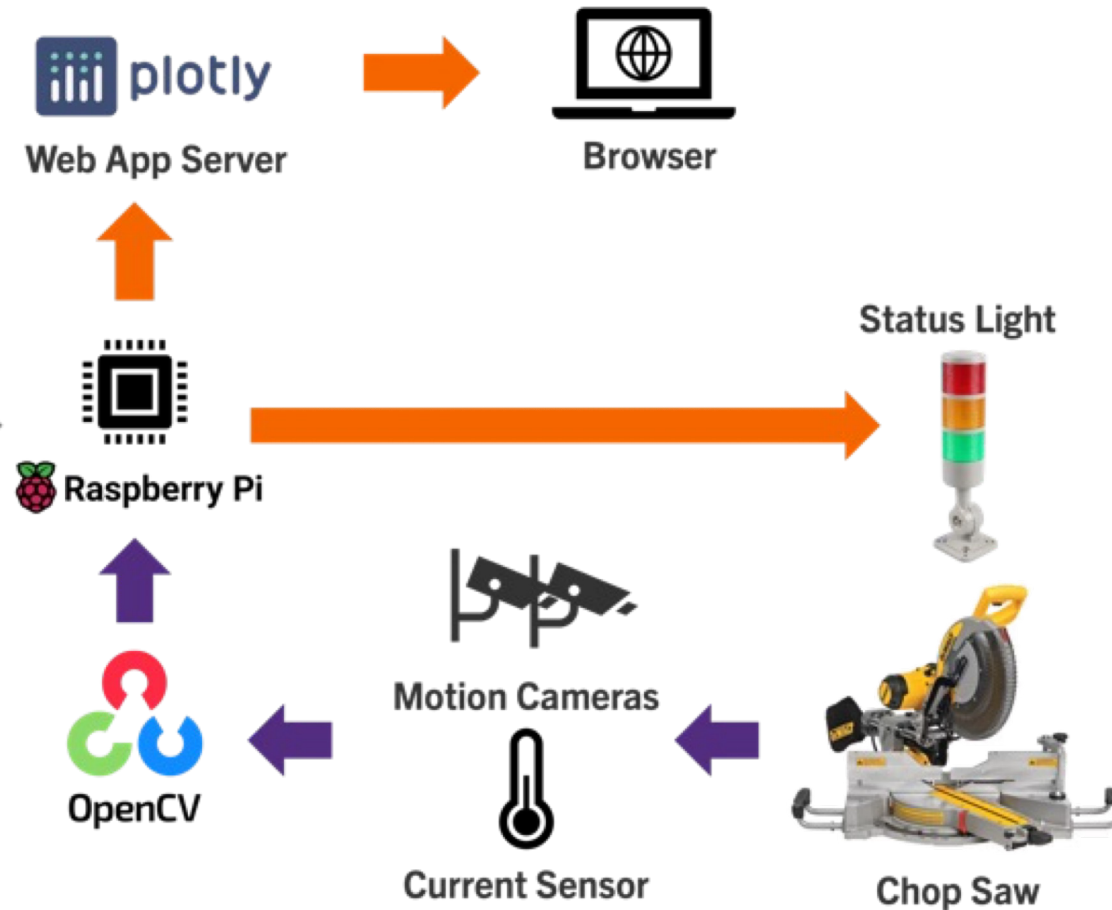
Figure 3-3. Command Module Trim Drag Coefficient



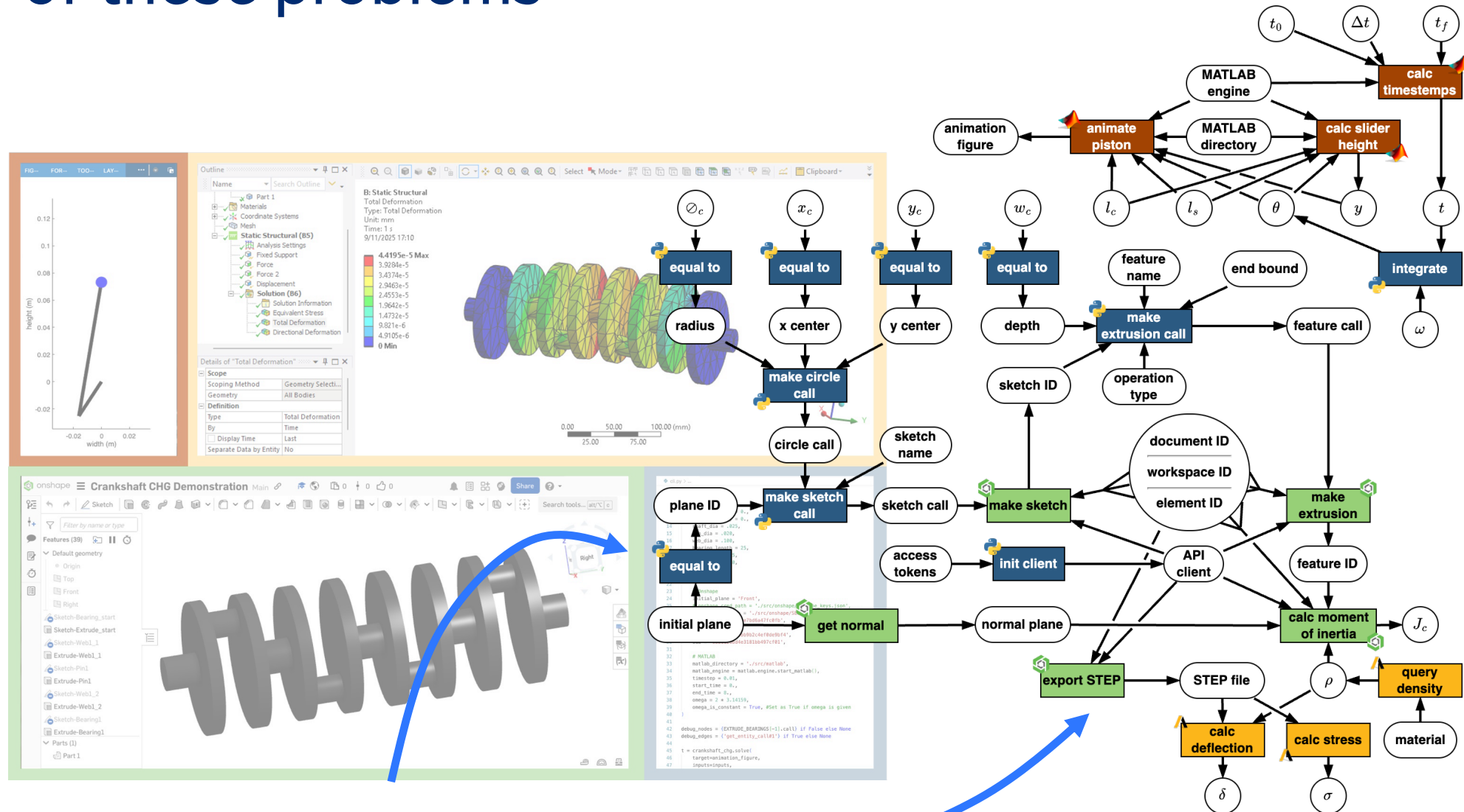
Emergent behavior arises when trying to combine subsystem digital twins



Complex behaviors require advanced software to solve, all which must be connected to make the digital twin



Using constraint hypergraphs (CHGs) mitigates many of these problems



independent software

Constraint Hypergraphs

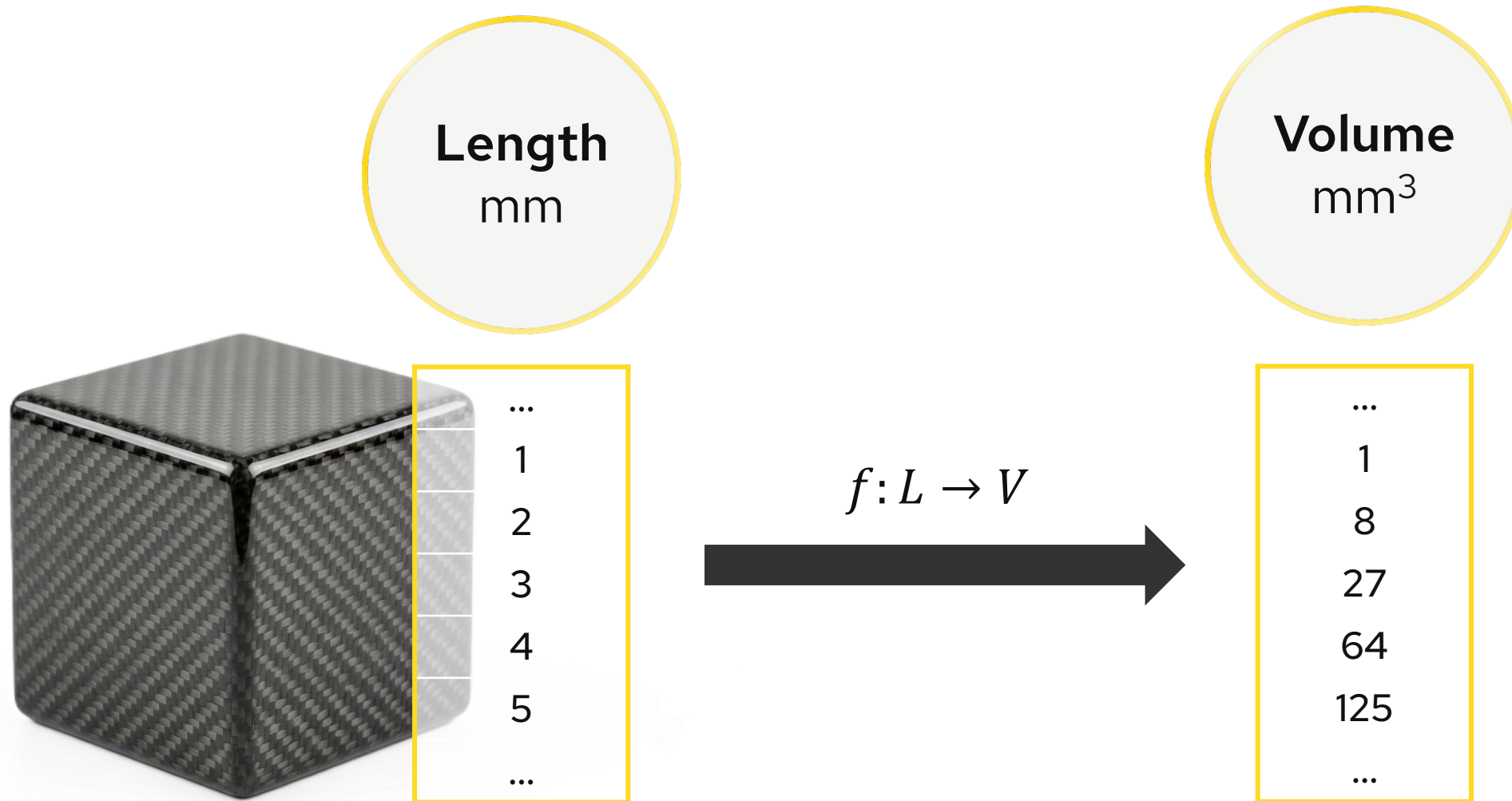
Robust, Interoperable Digital Twins

The behavior of a system is given by the relations, or constraints, between system variables

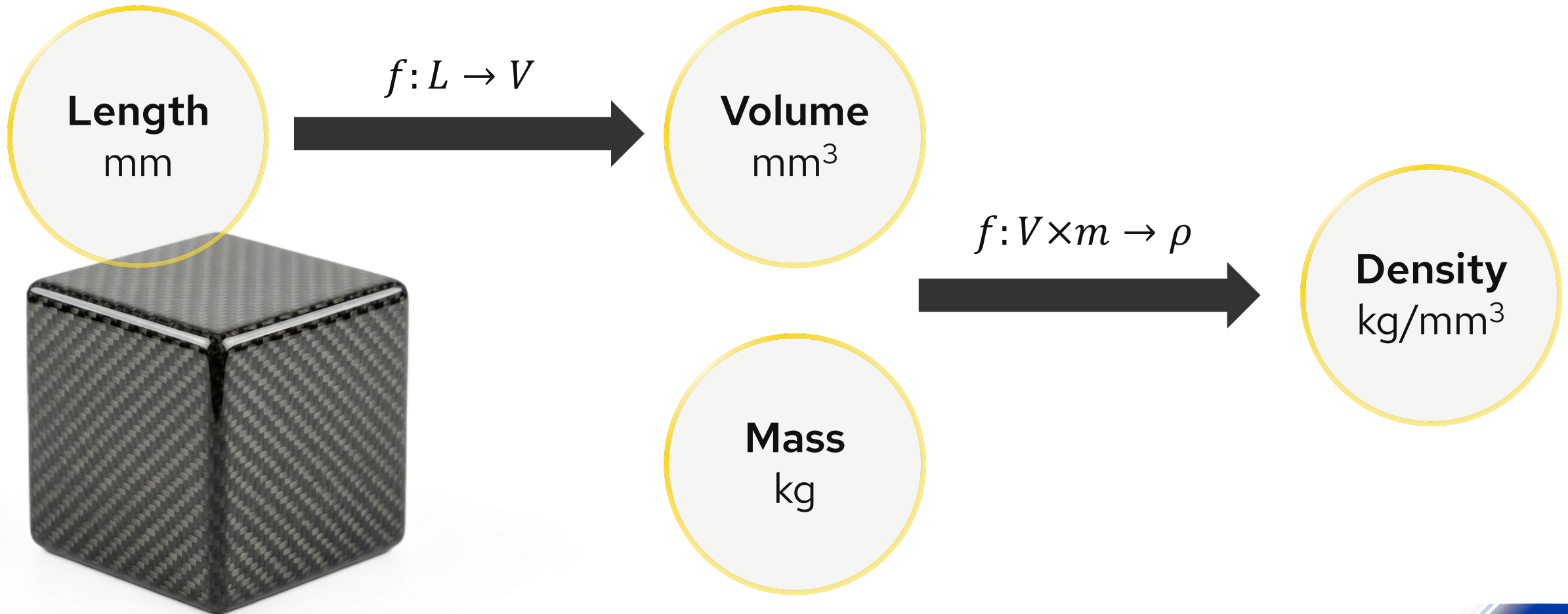
What is the behavior of this cube?



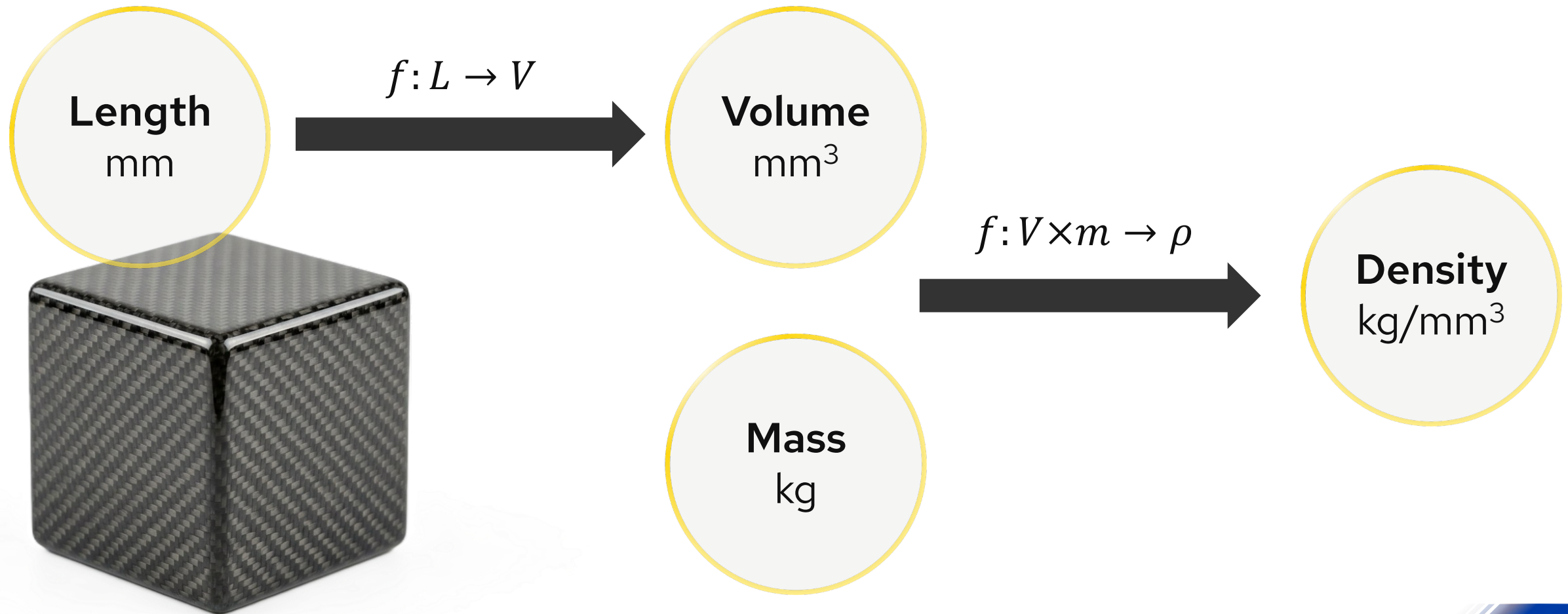
If variables are sets, then relationships can be represented as functions



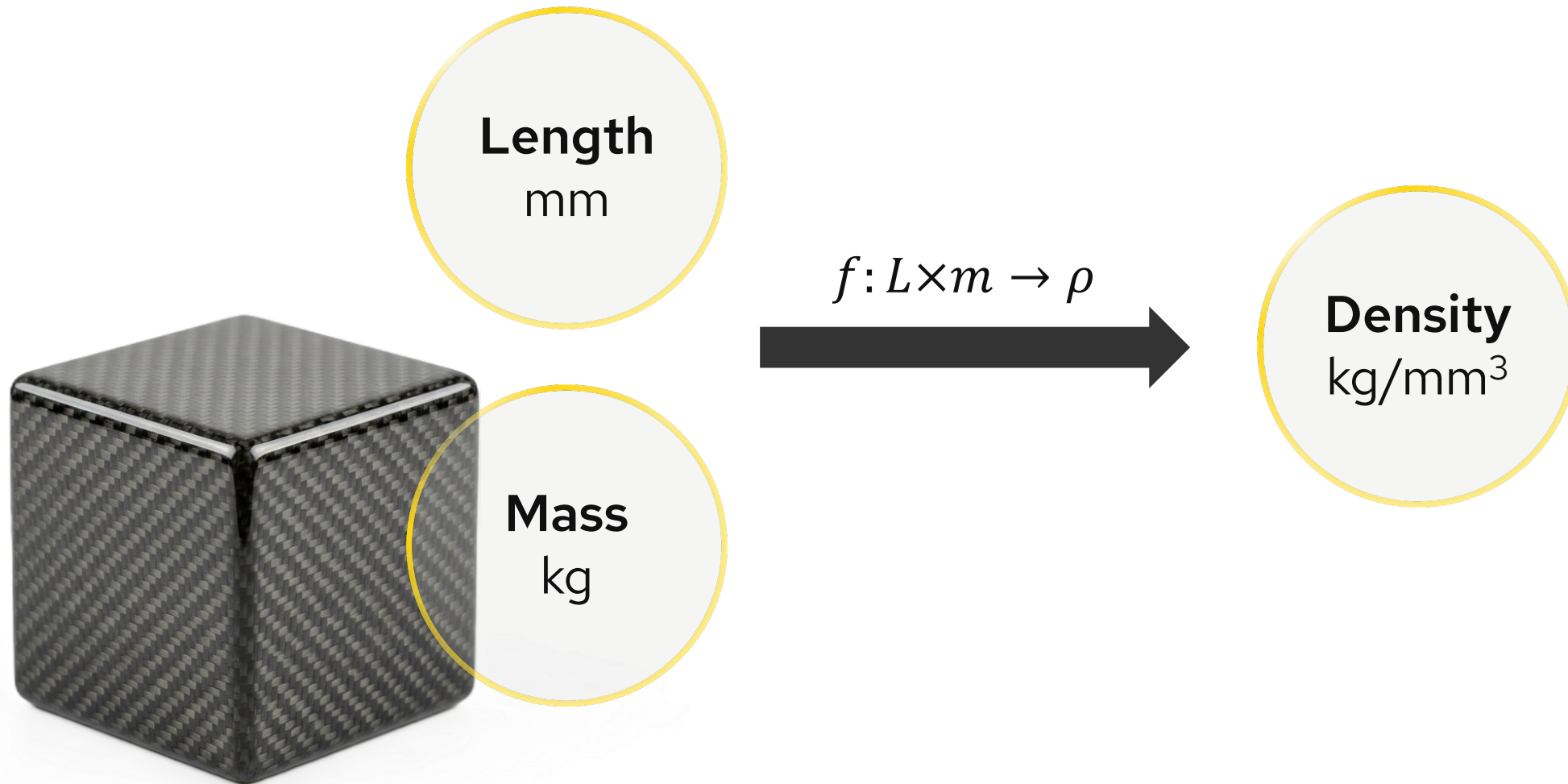
If variables are sets, then relationships can be represented as functions



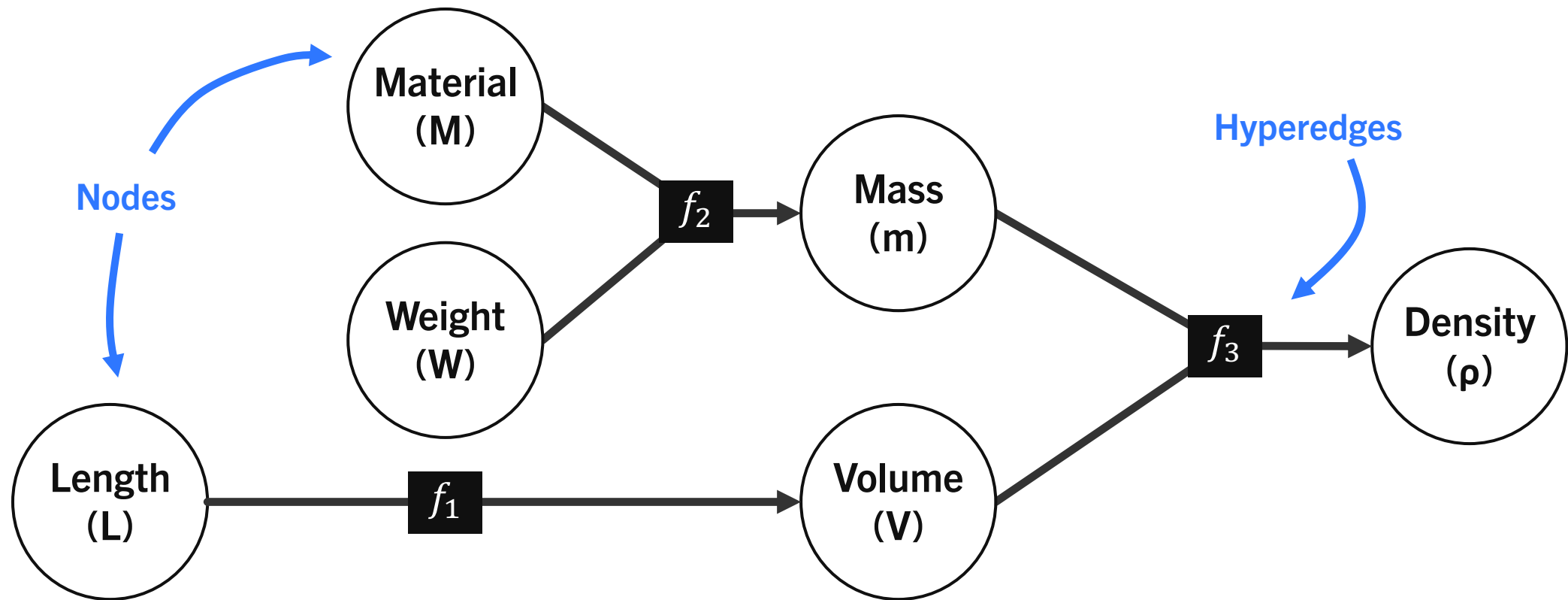
What inputs are needed to observe Density?



Functions compose to make new functions

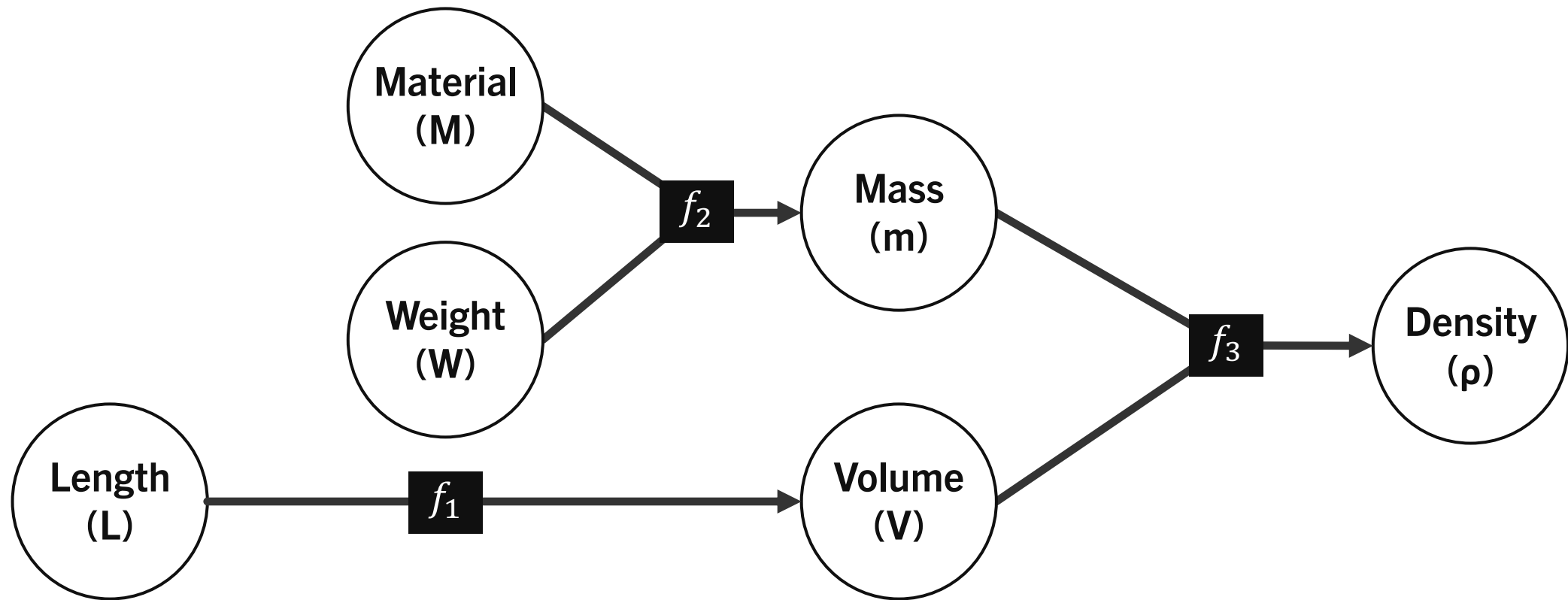


A constraint hypergraph is a mathematical structure consisting of all the variables (nodes) and behaviors (edges) modeled for a system

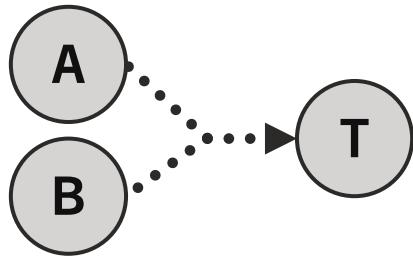


CHGs are purely functional, reducing models to their primitive constituents

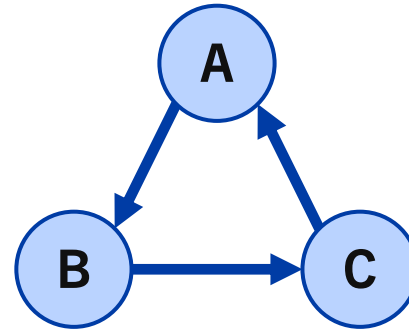
Consequently, all models can be represented as a CHG



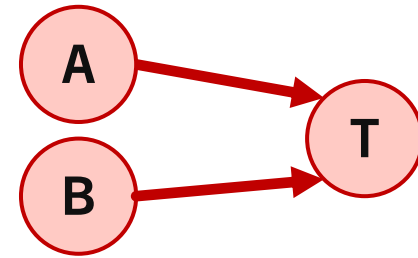
There are three structures in a CHG that complicate modeling



Partial Edges

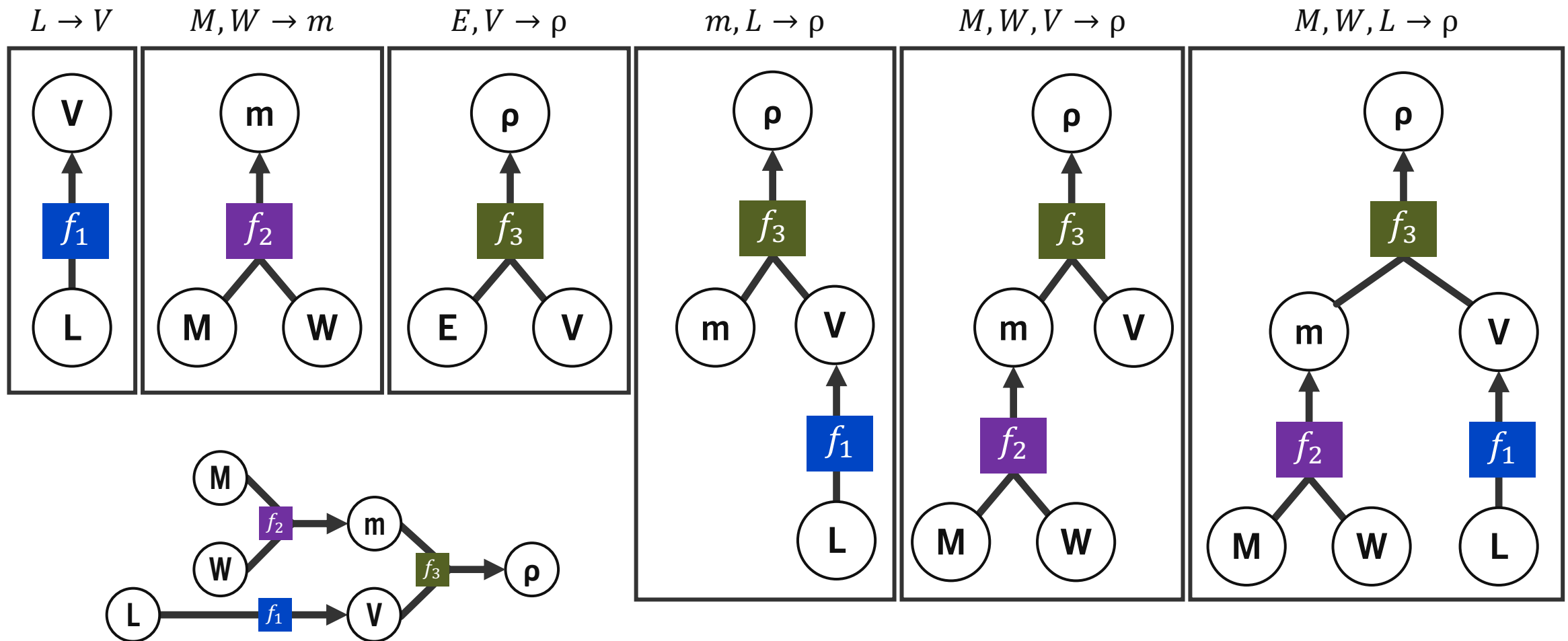


Cycles



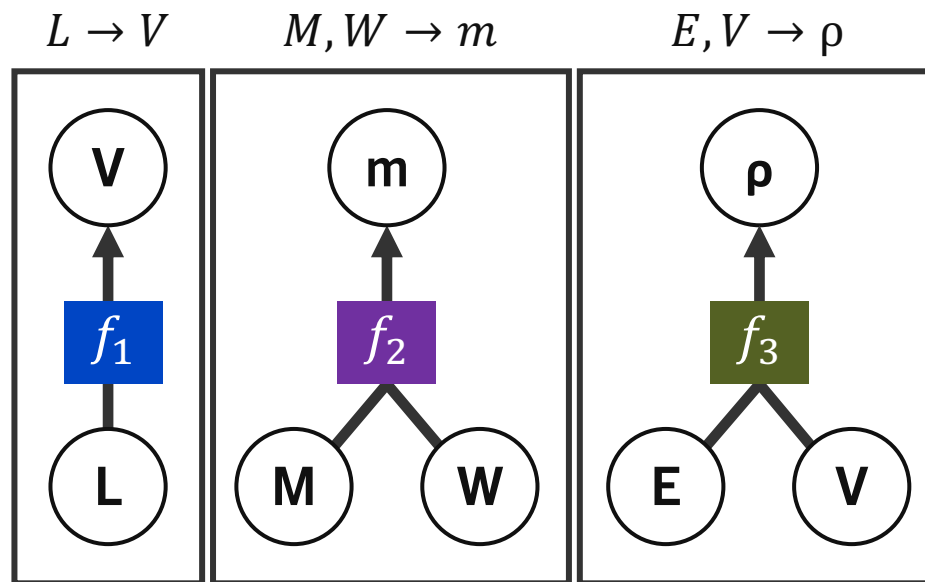
Multi-Edges

All paths through a CHG are a valid simulation

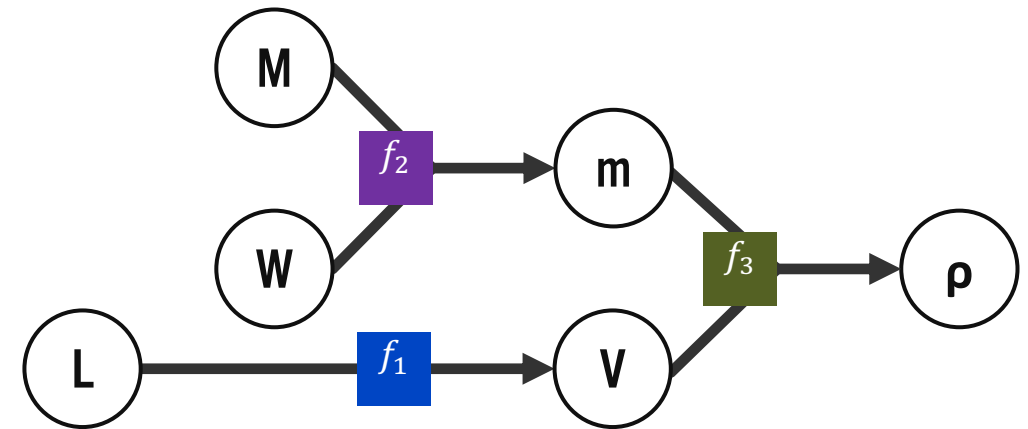


Pathfinding enables declarative simulation

CHGs are a fully simulatable framework



Imperative



Declarative

Declarative simulation provided by ConstraintHg



constrainthg 0.3.2

```
pip install constrainthg
```

Kernel for building and simulating constraint hypergraphs.

Navigation

- Project description
- Release history
- Download files

Project description

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



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:

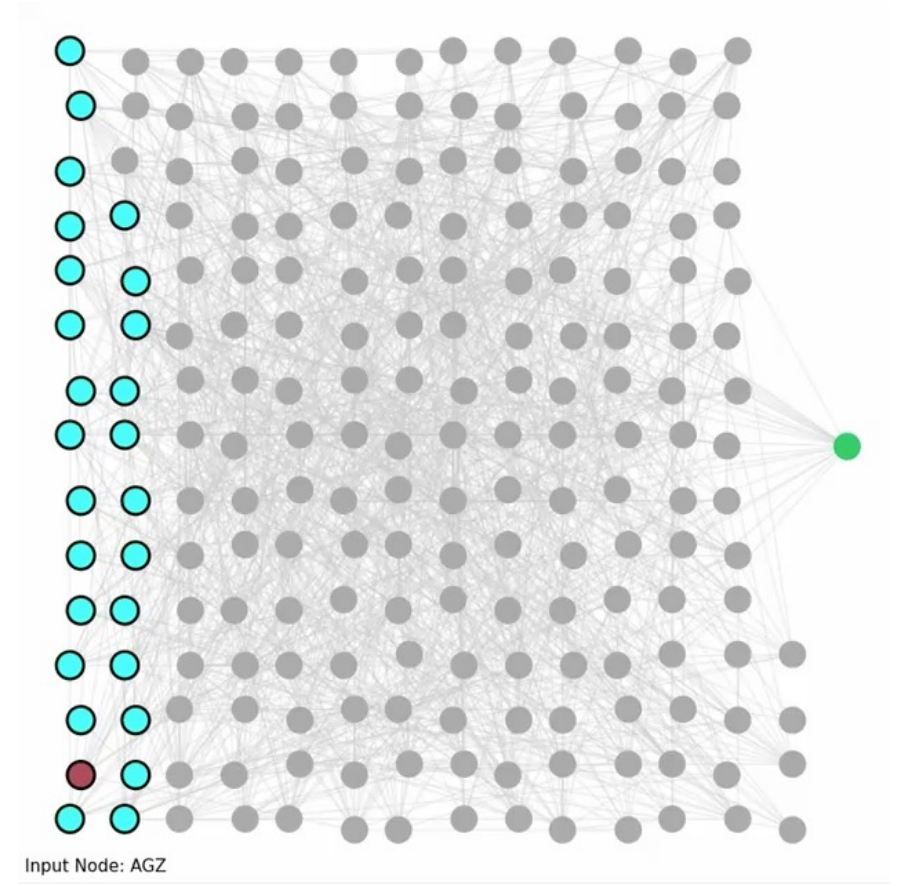
- $A + B = C$
- $A = -D$
- $B = -E$
- $D + E = F$
- $F = -C$

latest

ConstraintHg Functionality



- Provides methods for:
- Representing and visualizing CHGs
- Forming cycles, partial edges, and edge weights
- Pathfinding for declarative simulation
 - Extensive logging tools
- Merging CHGs



Process of making a CHG

1. Identify system facts (nodes)

Parameters, application variables, API tokens, etc.



3. Pass to CHG solver

Solver parses the CHG



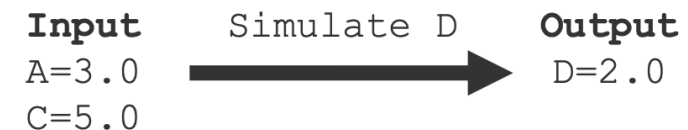
2. Form relations (edges) between nodes

Relations show how one node is determined by a set of other nodes

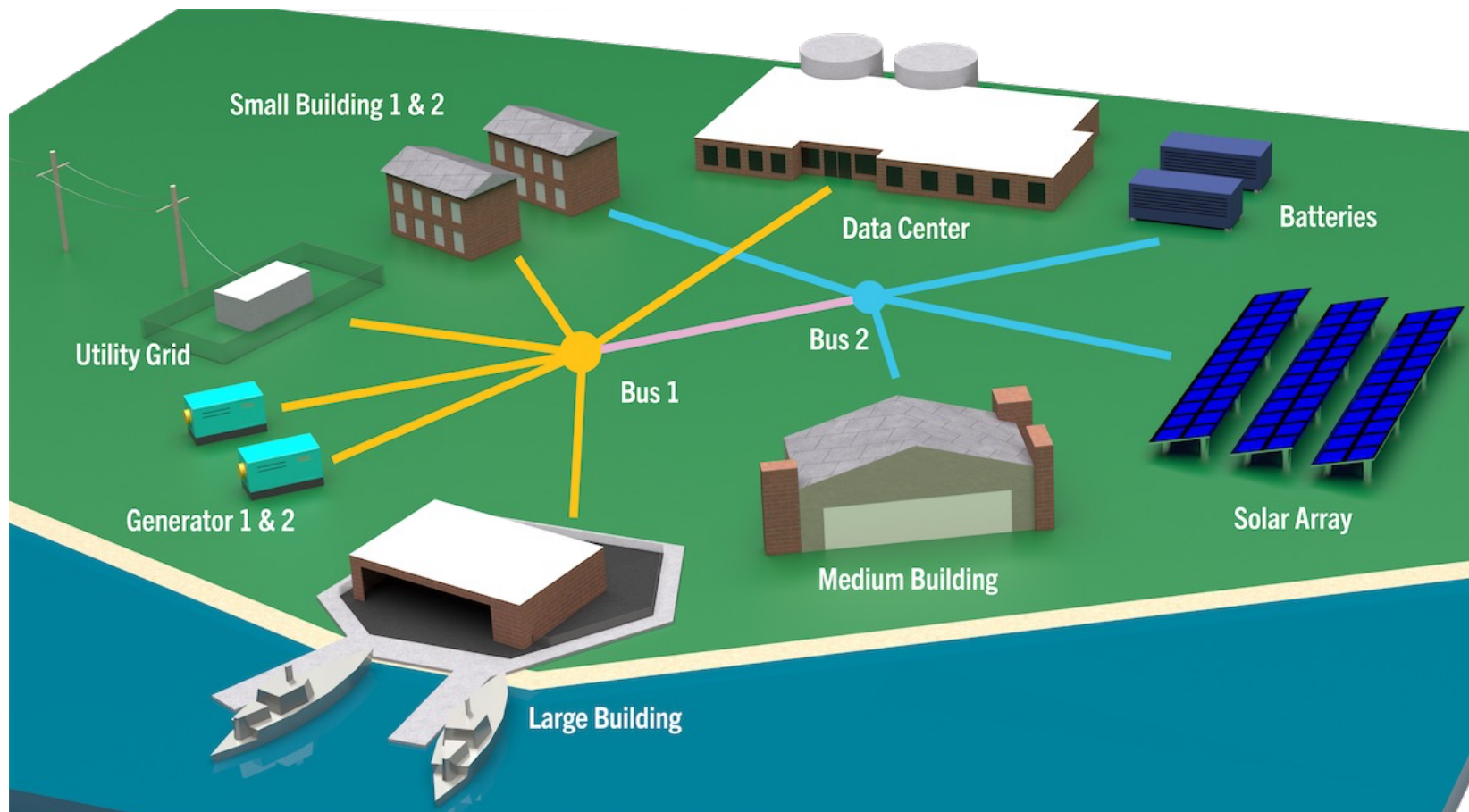


4. Request simulation

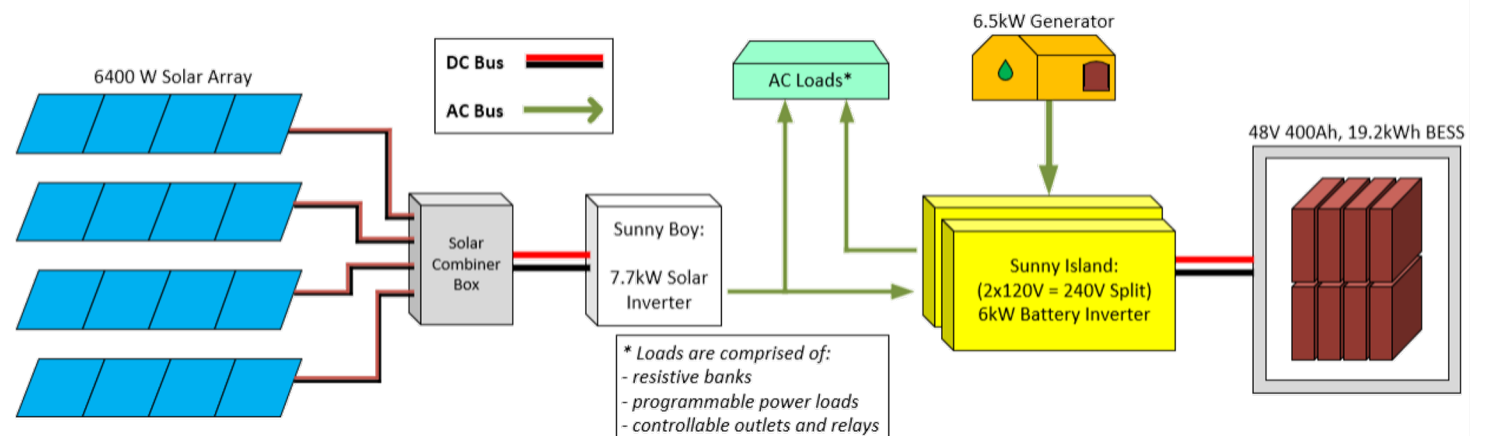
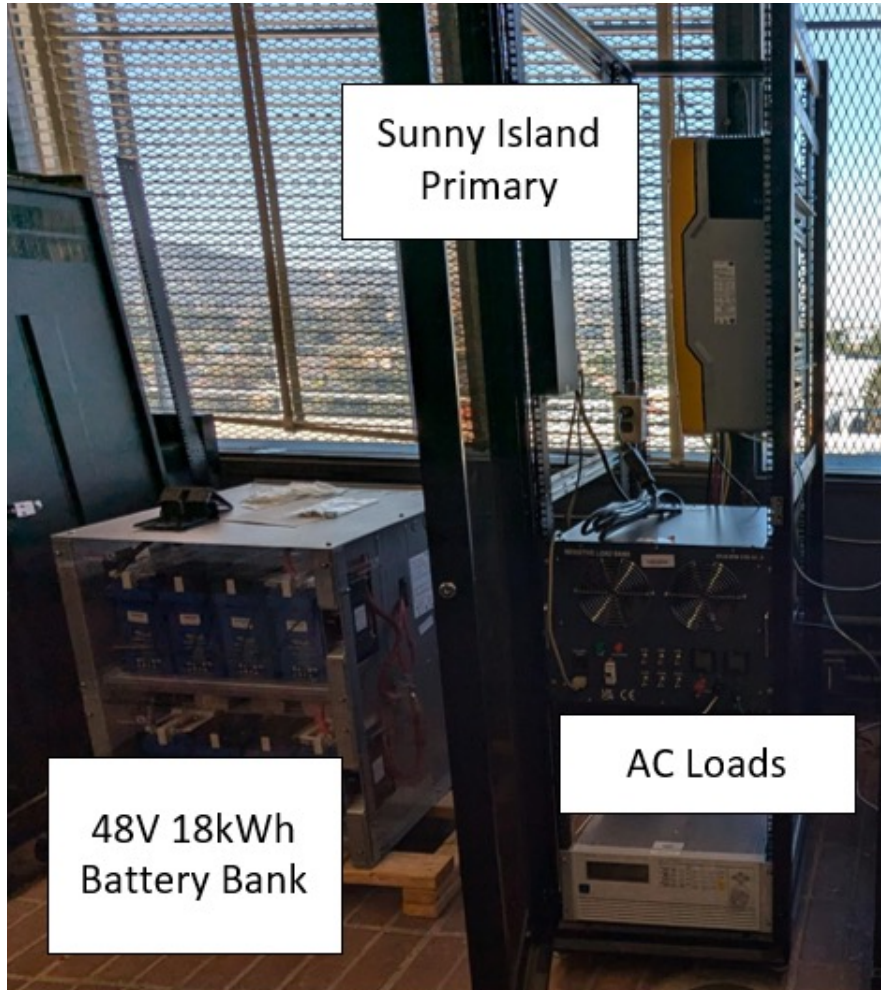
Solver simulates requested output by finding the shortest path mapping it to a set a known inputs



Demonstration: Microgrid Digital Twin



Physical Microgrid at Naval Postgraduate School



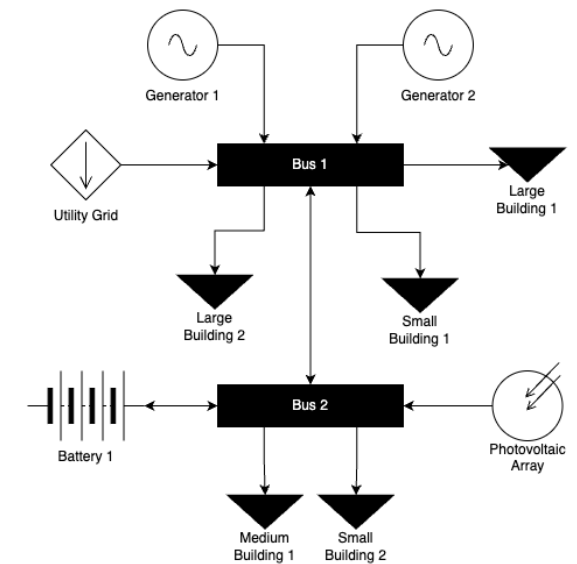
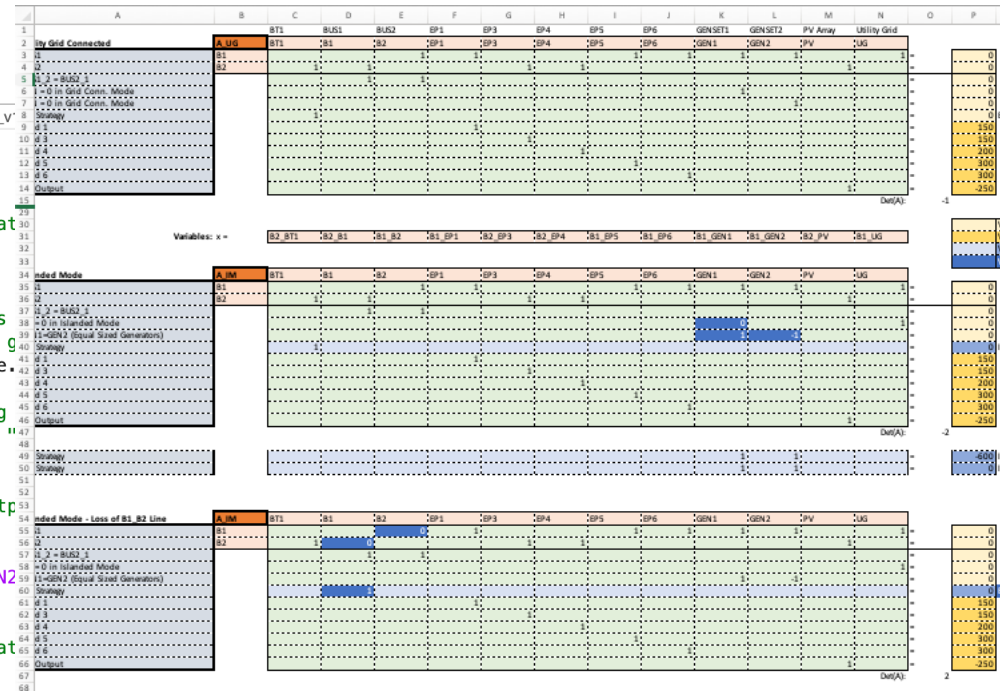
Previous Imperative Modeling of Microgrid

```

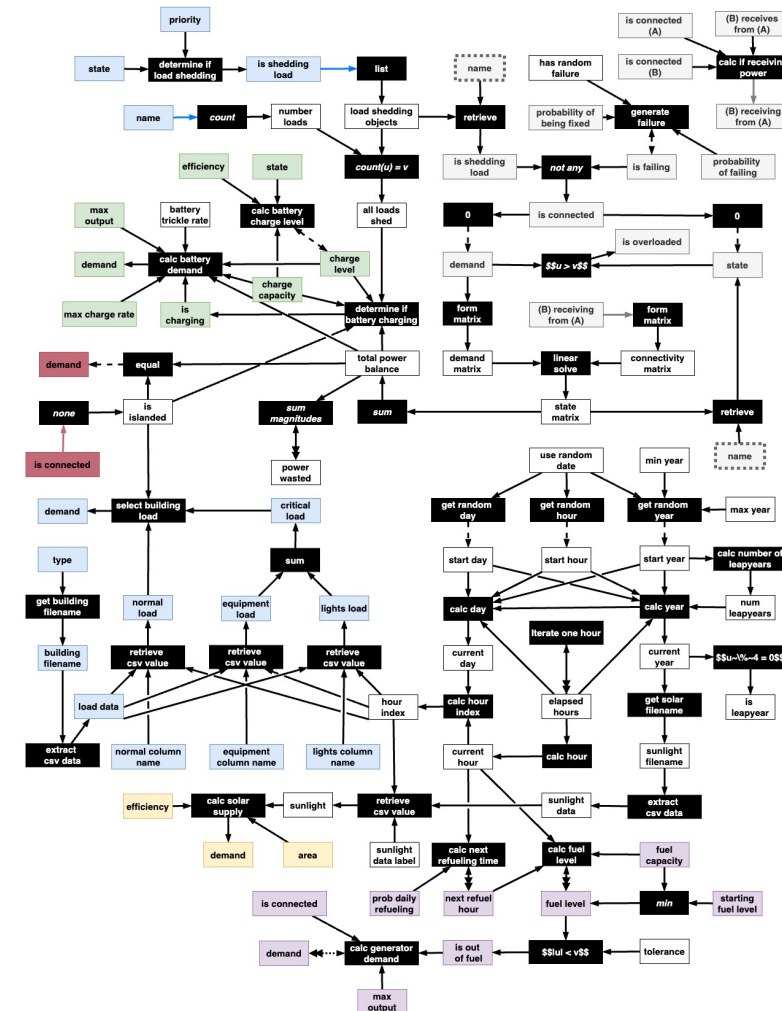
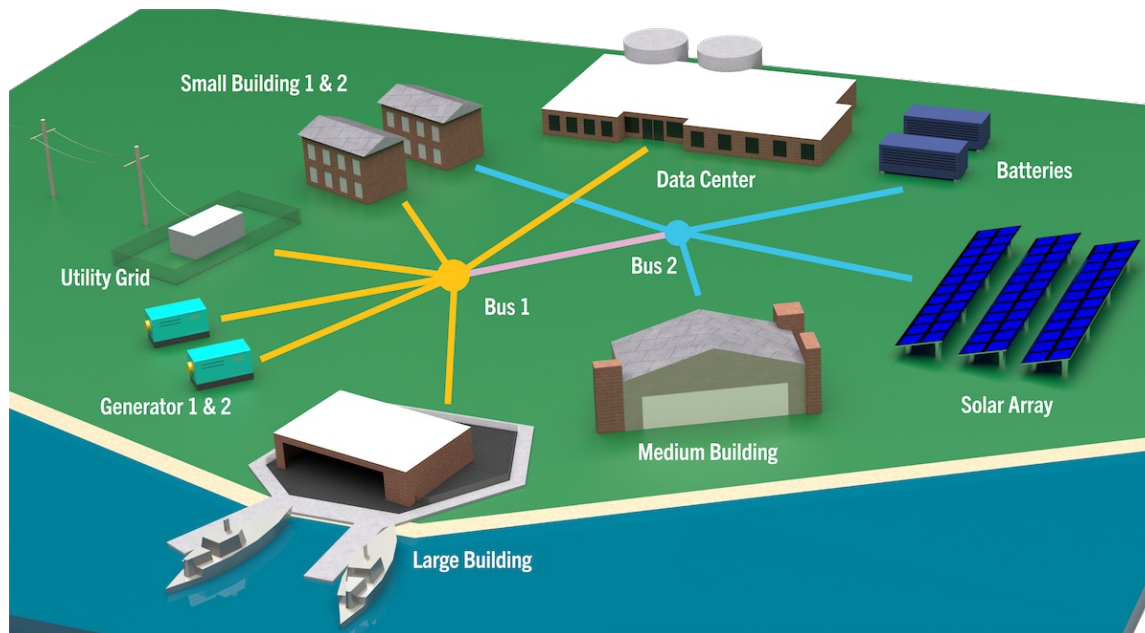
MG_LP_Test_v10.m ×
/Users/john_morris/Documents/Clemson/Research/MicrogridHg/archived_model/MG_LP_Test_v
% Initial Solve
x(:,n) = linsolve(A2,b);

%Determine if generator demand exceeds generation capacity
Gen_Demand = -x(pos.Gens,n);

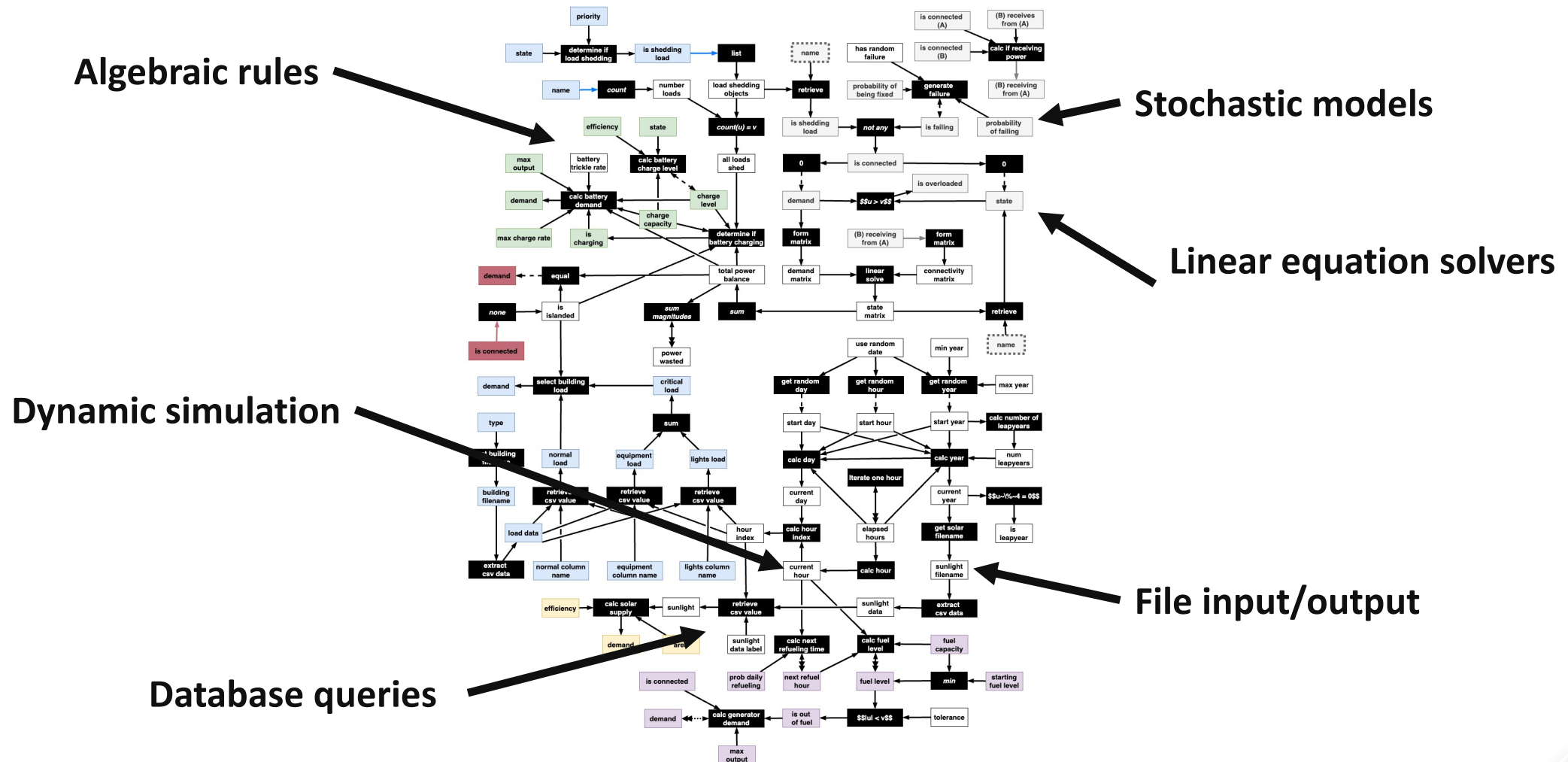
if any(Gen_Demand > [Gens.Capacity])
    %If the B2 to B1 Buss line or BT1 line is
    %exhausted cannot utilize ESS to make up generation
    if MG_State.B2_B1(n) == false || MG_State.BT1(n) == false
        % Below line was useful for debugging
        % disp("Gens Overloaded at time step "
        Overload.Gens = true;
    else
        %Otherwise set Generators at full output
        %make up unmet demand
        A2(6,:) = pos.Gens & n;
        b(6) = -MG_State(n,{'B1_GEN1' 'B1_GEN2' 'B1_GEN3' 'B1_GEN4' 'B1_GEN5' 'B1_GEN6' 'B1_GEN7' 'B1_GEN8' 'B1_GEN9' 'B1_GEN10' 'B1_GEN11' 'B1_GEN12' 'B1_GEN13' 'B1_GEN14' 'B1_GEN15' 'B1_GEN16' 'B1_GEN17' 'B1_GEN18' 'B1_GEN19' 'B1_GEN20' 'B1_GEN21' 'B1_GEN22' 'B1_GEN23' 'B1_GEN24' 'B1_GEN25' 'B1_GEN26' 'B1_GEN27' 'B1_GEN28' 'B1_GEN29' 'B1_GEN30' 'B1_GEN31' 'B1_GEN32' 'B1_GEN33' 'B1_GEN34' 'B1_GEN35' 'B1_GEN36' 'B1_GEN37' 'B1_GEN38' 'B1_GEN39' 'B1_GEN40' 'B1_GEN41' 'B1_GEN42' 'B1_GEN43' 'B1_GEN44' 'B1_GEN45' 'B1_GEN46' 'B1_GEN47' 'B1_GEN48' 'B1_GEN49' 'B1_GEN50' 'B1_GEN51' 'B1_GEN52' 'B1_GEN53' 'B1_GEN54' 'B1_GEN55' 'B1_GEN56' 'B1_GEN57' 'B1_GEN58' 'B1_GEN59' 'B1_GEN60' 'B1_GEN61' 'B1_GEN62' 'B1_GEN63' 'B1_GEN64' 'B1_GEN65' 'B1_GEN66' 'B1_GEN67' 'B1_GEN68' 'B1_GEN69' 'B1_GEN70' 'B1_GEN71' 'B1_GEN72' 'B1_GEN73' 'B1_GEN74' 'B1_GEN75' 'B1_GEN76' 'B1_GEN77' 'B1_GEN78' 'B1_GEN79' 'B1_GEN80' 'B1_GEN81' 'B1_GEN82' 'B1_GEN83' 'B1_GEN84' 'B1_GEN85' 'B1_GEN86' 'B1_GEN87' 'B1_GEN88' 'B1_GEN89' 'B1_GEN90' 'B1_GEN91' 'B1_GEN92' 'B1_GEN93' 'B1_GEN94' 'B1_GEN95' 'B1_GEN96' 'B1_GEN97' 'B1_GEN98' 'B1_GEN99' 'B1_GEN100' 'B1_GEN101' 'B1_GEN102' 'B1_GEN103' 'B1_GEN104' 'B1_GEN105' 'B1_GEN106' 'B1_GEN107' 'B1_GEN108' 'B1_GEN109' 'B1_GEN110' 'B1_GEN111' 'B1_GEN112' 'B1_GEN113' 'B1_GEN114' 'B1_GEN115' 'B1_GEN116' 'B1_GEN117' 'B1_GEN118' 'B1_GEN119' 'B1_GEN120' 'B1_GEN121' 'B1_GEN122' 'B1_GEN123' 'B1_GEN124' 'B1_GEN125' 'B1_GEN126' 'B1_GEN127' 'B1_GEN128' 'B1_GEN129' 'B1_GEN130' 'B1_GEN131' 'B1_GEN132' 'B1_GEN133' 'B1_GEN134' 'B1_GEN135' 'B1_GEN136' 'B1_GEN137' 'B1_GEN138' 'B1_GEN139' 'B1_GEN140' 'B1_GEN141' 'B1_GEN142' 'B1_GEN143' 'B1_GEN144' 'B1_GEN145' 'B1_GEN146' 'B1_GEN147' 'B1_GEN148' 'B1_GEN149' 'B1_GEN150' 'B1_GEN151' 'B1_GEN152' 'B1_GEN153' 'B1_GEN154' 'B1_GEN155' 'B1_GEN156' 'B1_GEN157' 'B1_GEN158' 'B1_GEN159' 'B1_GEN160' 'B1_GEN161' 'B1_GEN162' 'B1_GEN163' 'B1_GEN164' 'B1_GEN165' 'B1_GEN166' 'B1_GEN167' 'B1_GEN168' 'B1_GEN169' 'B1_GEN170' 'B1_GEN171' 'B1_GEN172' 'B1_GEN173' 'B1_GEN174' 'B1_GEN175' 'B1_GEN176' 'B1_GEN177' 'B1_GEN178' 'B1_GEN179' 'B1_GEN180' 'B1_GEN181' 'B1_GEN182' 'B1_GEN183' 'B1_GEN184' 'B1_GEN185' 'B1_GEN186' 'B1_GEN187' 'B1_GEN188' 'B1_GEN189' 'B1_GEN190' 'B1_GEN191' 'B1_GEN192' 'B1_GEN193' 'B1_GEN194' 'B1_GEN195' 'B1_GEN196' 'B1_GEN197' 'B1_GEN198' 'B1_GEN199' 'B1_GEN200' 'B1_GEN201' 'B1_GEN202' 'B1_GEN203' 'B1_GEN204' 'B1_GEN205' 'B1_GEN206' 'B1_GEN207' 'B1_GEN208' 'B1_GEN209' 'B1_GEN210' 'B1_GEN211' 'B1_GEN212' 'B1_GEN213' 'B1_GEN214' 'B1_GEN215' 'B1_GEN216' 'B1_GEN217' 'B1_GEN218' 'B1_GEN219' 'B1_GEN220' 'B1_GEN221' 'B1_GEN222' 'B1_GEN223' 'B1_GEN224' 'B1_GEN225' 'B1_GEN226' 'B1_GEN227' 'B1_GEN228' 'B1_GEN229' 'B1_GEN230' 'B1_GEN231' 'B1_GEN232' 'B1_GEN233' 'B1_GEN234' 'B1_GEN235' 'B1_GEN236' 'B1_GEN237' 'B1_GEN238' 'B1_GEN239' 'B1_GEN240' 'B1_GEN241' 'B1_GEN242' 'B1_GEN243' 'B1_GEN244' 'B1_GEN245' 'B1_GEN246' 'B1_GEN247' 'B1_GEN248' 'B1_GEN249' 'B1_GEN250' 'B1_GEN251' 'B1_GEN252' 'B1_GEN253' 'B1_GEN254' 'B1_GEN255' 'B1_GEN256' 'B1_GEN257' 'B1_GEN258' 'B1_GEN259' 'B1_GEN260' 'B1_GEN261' 'B1_GEN262' 'B1_GEN263' 'B1_GEN264' 'B1_GEN265' 'B1_GEN266' 'B1_GEN267' 'B1_GEN268' 'B1_GEN269' 'B1_GEN270' 'B1_GEN271' 'B1_GEN272' 'B1_GEN273' 'B1_GEN274' 'B1_GEN275' 'B1_GEN276' 'B1_GEN277' 'B1_GEN278' 'B1_GEN279' 'B1_GEN280' 'B1_GEN281' 'B1_GEN282' 'B1_GEN283' 'B1_GEN284' 'B1_GEN285' 'B1_GEN286' 'B1_GEN287' 'B1_GEN288' 'B1_GEN289' 'B1_GEN290' 'B1_GEN291' 'B1_GEN292' 'B1_GEN293' 'B1_GEN294' 'B1_GEN295' 'B1_GEN296' 'B1_GEN297' 'B1_GEN298' 'B1_GEN299' 'B1_GEN300' 'B1_GEN301' 'B1_GEN302' 'B1_GEN303' 'B1_GEN304' 'B1_GEN305' 'B1_GEN306' 'B1_GEN307' 'B1_GEN308' 'B1_GEN309' 'B1_GEN310' 'B1_GEN311' 'B1_GEN312' 'B1_GEN313' 'B1_GEN314' 'B1_GEN315' 'B1_GEN316' 'B1_GEN317' 'B1_GEN318' 'B1_GEN319' 'B1_GEN320' 'B1_GEN321' 'B1_GEN322' 'B1_GEN323' 'B1_GEN324' 'B1_GEN325' 'B1_GEN326' 'B1_GEN327' 'B1_GEN328' 'B1_GEN329' 'B1_GEN330' 'B1_GEN331' 'B1_GEN332' 'B1_GEN333' 'B1_GEN334' 'B1_GEN335' 'B1_GEN336' 'B1_GEN337' 'B1_GEN338' 'B1_GEN339' 'B1_GEN340' 'B1_GEN341' 'B1_GEN342' 'B1_GEN343' 'B1_GEN344' 'B1_GEN345' 'B1_GEN346' 'B1_GEN347' 'B1_GEN348' 'B1_GEN349' 'B1_GEN350' 'B1_GEN351' 'B1_GEN352' 'B1_GEN353' 'B1_GEN354' 'B1_GEN355' 'B1_GEN356' 'B1_GEN357' 'B1_GEN358' 'B1_GEN359' 'B1_GEN360' 'B1_GEN361' 'B1_GEN362' 'B1_GEN363' 'B1_GEN364' 'B1_GEN365' 'B1_GEN366' 'B1_GEN367' 'B1_GEN368' 'B1_GEN369' 'B1_GEN370' 'B1_GEN371' 'B1_GEN372' 'B1_GEN373' 'B1_GEN374' 'B1_GEN375' 'B1_GEN376' 'B1_GEN377' 'B1_GEN378' 'B1_GEN379' 'B1_GEN380' 'B1_GEN381' 'B1_GEN382' 'B1_GEN383' 'B1_GEN384' 'B1_GEN385' 'B1_GEN386' 'B1_GEN387' 'B1_GEN388' 'B1_GEN389' 'B1_GEN390' 'B1_GEN391' 'B1_GEN392' 'B1_GEN393' 'B1_GEN394' 'B1_GEN395' 'B1_GEN396' 'B1_GEN397' 'B1_GEN398' 'B1_GEN399' 'B1_GEN400' 'B1_GEN401' 'B1_GEN402' 'B1_GEN403' 'B1_GEN404' 'B1_GEN405' 'B1_GEN406' 'B1_GEN407' 'B1_GEN408' 'B1_GEN409' 'B1_GEN410' 'B1_GEN411' 'B1_GEN412' 'B1_GEN413' 'B1_GEN414' 'B1_GEN415' 'B1_GEN416' 'B1_GEN417' 'B1_GEN418' 'B1_GEN419' 'B1_GEN420' 'B1_GEN421' 'B1_GEN422' 'B1_GEN423' 'B1_GEN424' 'B1_GEN425' 'B1_GEN426' 'B1_GEN427' 'B1_GEN428' 'B1_GEN429' 'B1_GEN430' 'B1_GEN431' 'B1_GEN432' 'B1_GEN433' 'B1_GEN434' 'B1_GEN435' 'B1_GEN436' 'B1_GEN437' 'B1_GEN438' 'B1_GEN439' 'B1_GEN440' 'B1_GEN441' 'B1_GEN442' 'B1_GEN443' 'B1_GEN444' 'B1_GEN445' 'B1_GEN446' 'B1_GEN447' 'B1_GEN448' 'B1_GEN449' 'B1_GEN450' 'B1_GEN451' 'B1_GEN452' 'B1_GEN453' 'B1_GEN454' 'B1_GEN455' 'B1_GEN456' 'B1_GEN457' 'B1_GEN458' 'B1_GEN459' 'B1_GEN460' 'B1_GEN461' 'B1_GEN462' 'B1_GEN463' 'B1_GEN464' 'B1_GEN465' 'B1_GEN466' 'B1_GEN467' 'B1_GEN468' 'B1_GEN469' 'B1_GEN470' 'B1_GEN471' 'B1_GEN472' 'B1_GEN473' 'B1_GEN474' 'B1_GEN475' 'B1_GEN476' 'B1_GEN477' 'B1_GEN478' 'B1_GEN479' 'B1_GEN480' 'B1_GEN481' 'B1_GEN482' 'B1_GEN483' 'B1_GEN484' 'B1_GEN485' 'B1_GEN486' 'B1_GEN487' 'B1_GEN488' 'B1_GEN489' 'B1_GEN490' 'B1_GEN491' 'B1_GEN492' 'B1_GEN493' 'B1_GEN494'
```



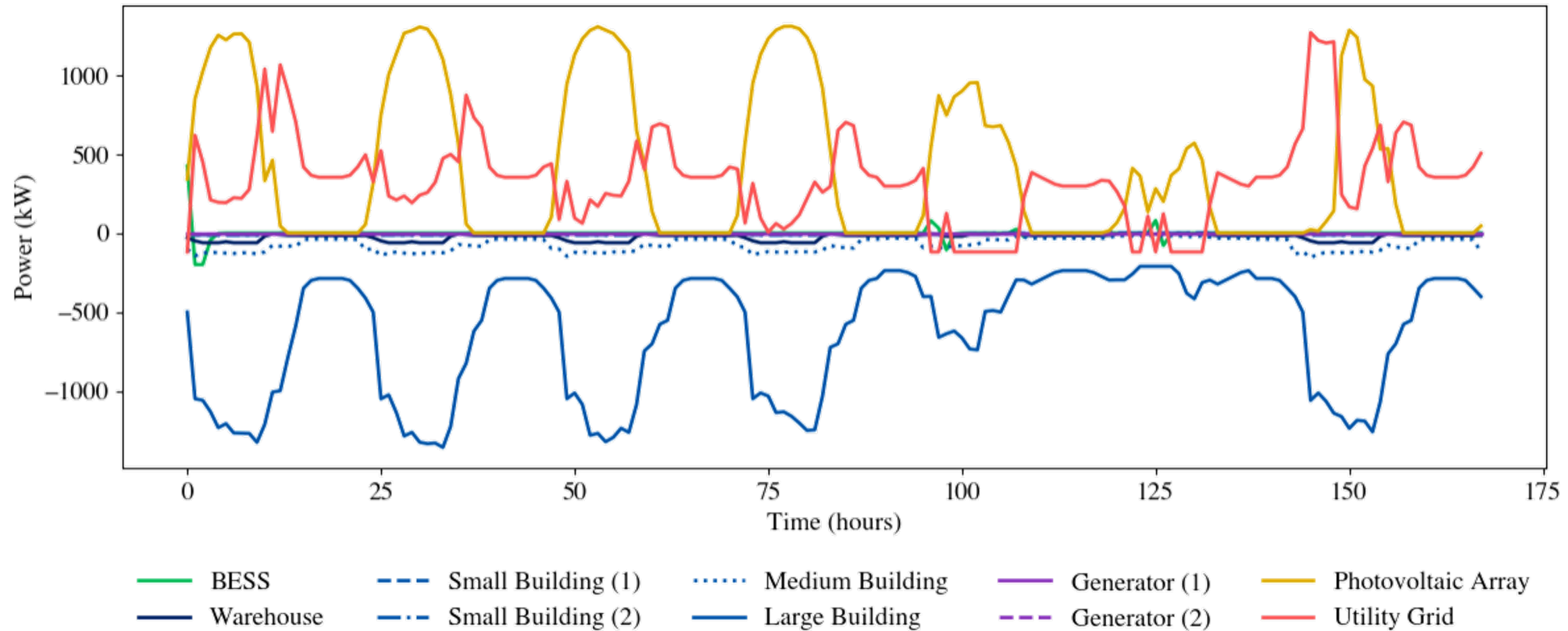
Transformed CHG contains 594 nodes and 334 edges



Includes all kinds of simulation and modeling



Autonomously observes and simulates the real system

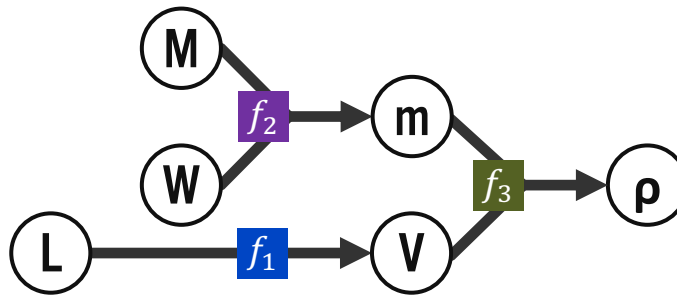


Digital twins require foundations that can handle modern challenges in modeling and simulation

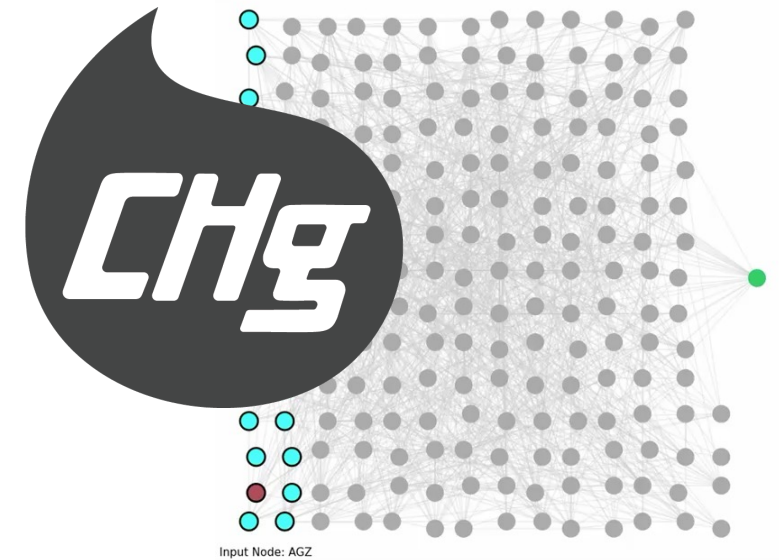
Digital twins are analogs of a real system



Twinned systems share the same CHG

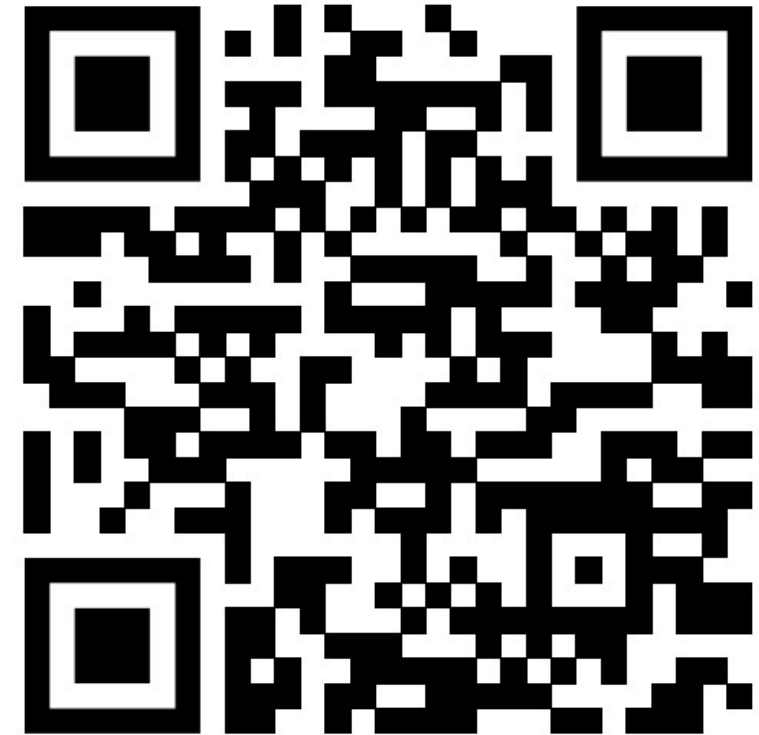
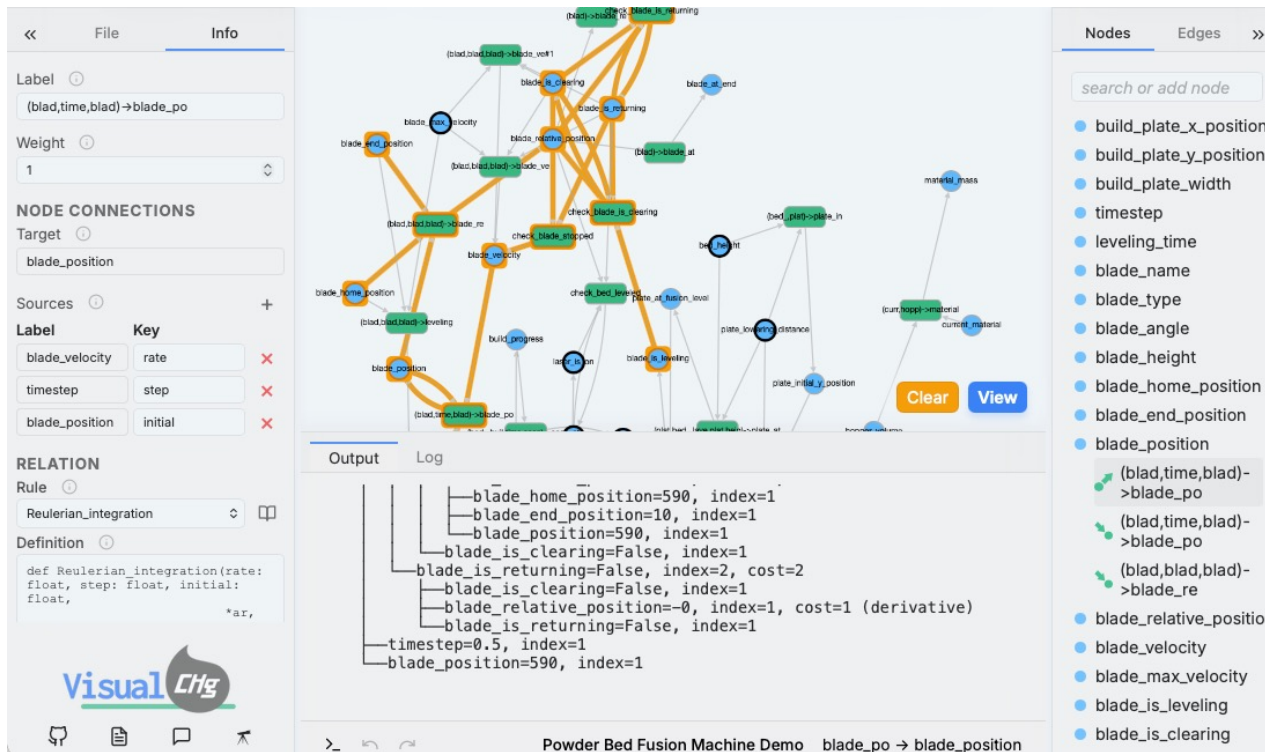


Pathfinding leads to autonomous observation



John Morris
john.h.morris@uah.edu

We can make our own digital twins using VisualCHG



visualchg.searchlaboratory.org