Creating Mission Models

using SysML and Cameo Systems Modeler

John DeHart -AVIAN Inc.

John DeHart

2023 CATIA MBSE CYBER EXPERIENCE SYMPOSIUM

Hello, I’m John. I have worked in aerospace industry for +25yrs. I have been a part of several major aircraft development programs including 737-X, Cessna 650, B-2, F-22, and many missile programs including Tac-Tom, Javelin, JASSM, LRASM, NLOS, there are a lot more… I just forget.

Most of my career has been in structures, controls, multidiscipline, simulation, and now as a systems engineer.

M.S. Systems Engineering, Georgia Tech
B.S. MET, Georgia Southern

Outside of modeling and sim, I enjoy making home brew, and I have a neat greenhouse specializing in lettuce.

2023 MBSE CYBER EXPERIENCE SYMPOSIUM

CATIA

avian®

Your future, simplified.

Creating Mission Models using SysML and Cameo Systems Modeler

John K. DeHart
jdehart@avian.com

About AVIAN

Avian is a leading Service-Disabled, Veteran Owned Small Business (SDVOSB) with all the responsiveness and reachability of a small business and the resources of a large one (≈ 250 employees)

- Our Capability
  - Provide skilled modelers at all levels of expertise Provide an initial “Assist Team” to assess capability needs Provide a functional environment with use of floating Cameo licenses Ability to produce modelers through organic training environment.

- AVIAN’s Involvement in MBSE
  - Direct support to the NAVAIR / NAWCAD Systems Engineering Transformation (SET) team to build the processes and methodology necessary to implement MBSE at the organization level.
  - Direct support to SET for MBSE courseware development and training.

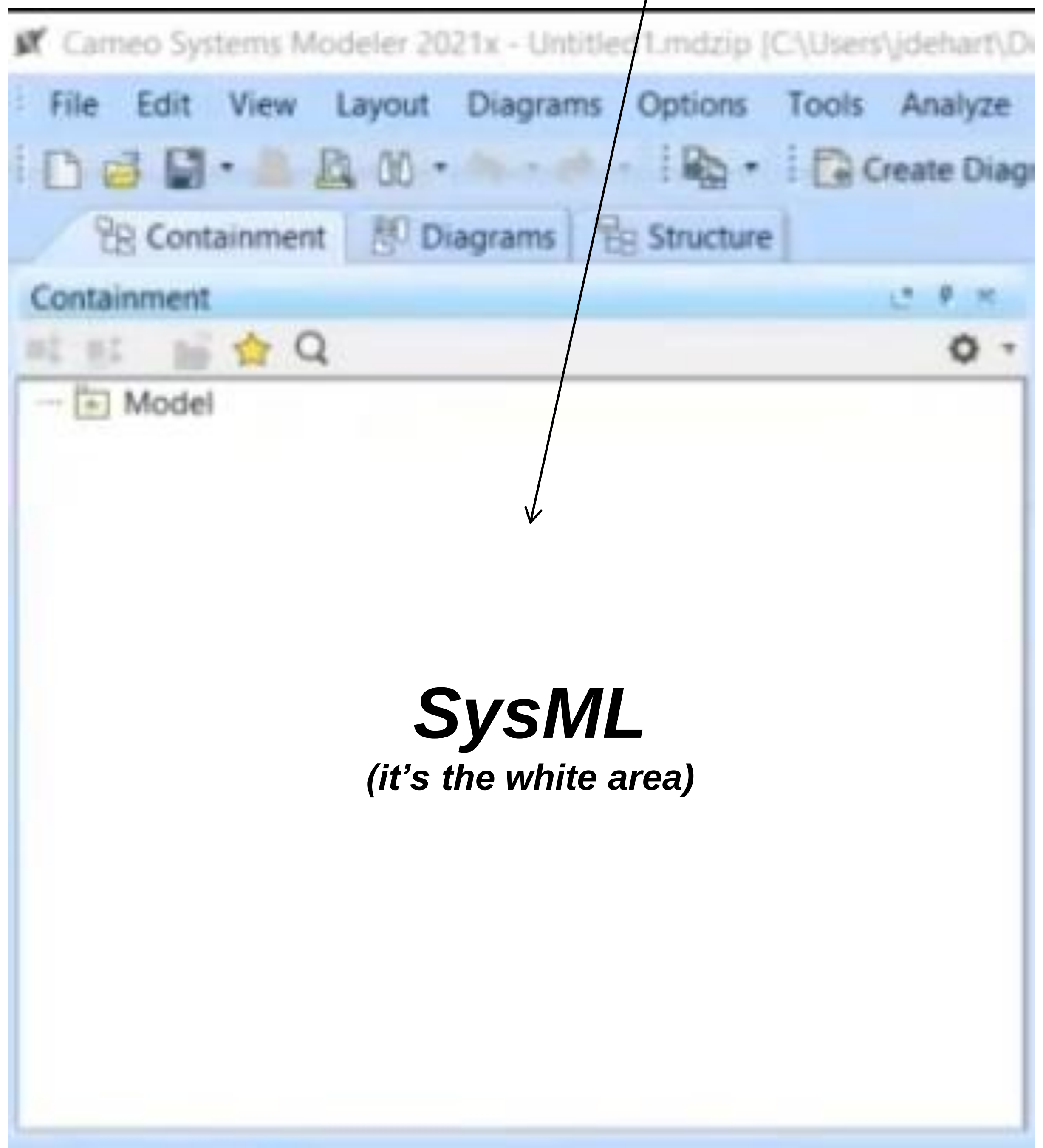
Introductions

- Mission Modeling in SysML
  - What is the purpose of a model? → To Inform
  - Inform a document, simulation, analysis, etc.
  - Think of it like playing Risk..
    - Set the board up → play the game→ analyze your loss

- Agenda
  - Some general modeling observations
  - Demo - Example mission laydown model
  - Demo - Example arrow
  - Wrap-up

Containment Trees, Models, and Instances

This is where we make a model

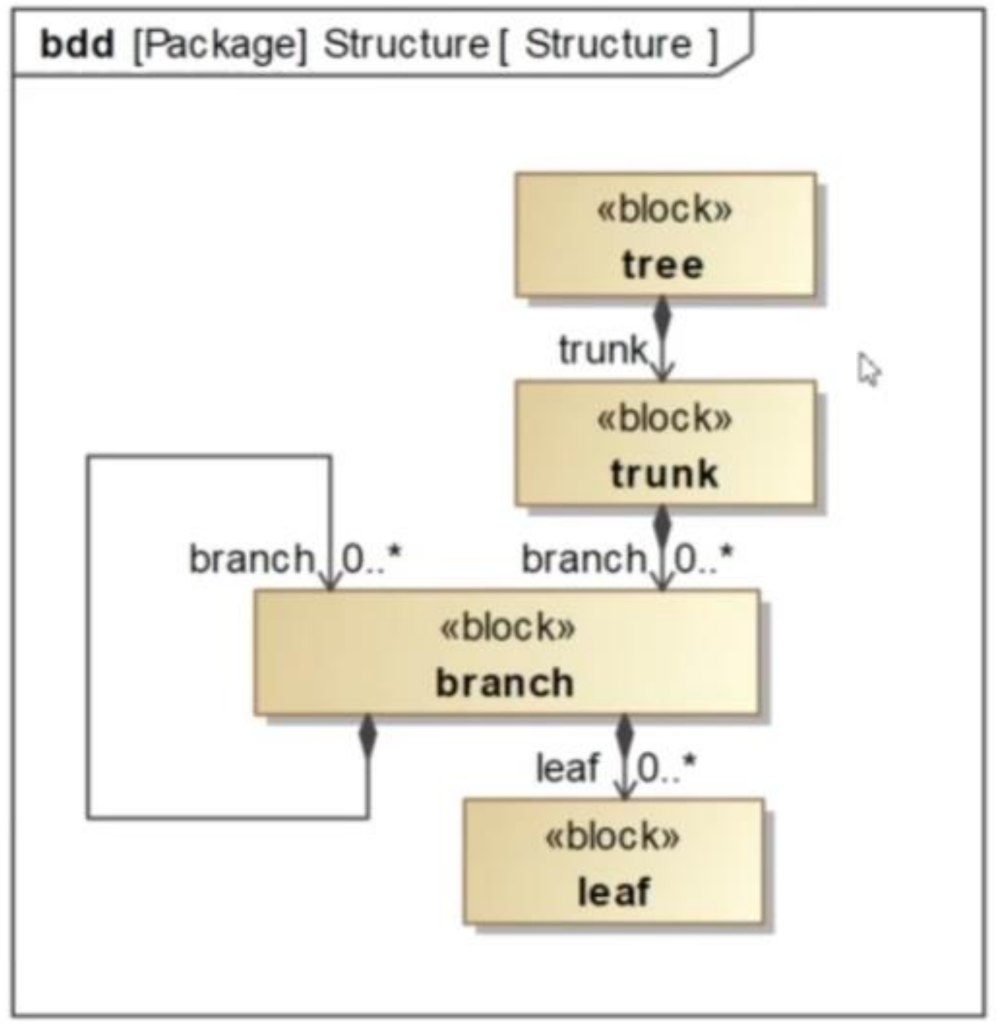


SysML

(it's the white area)

Instance of

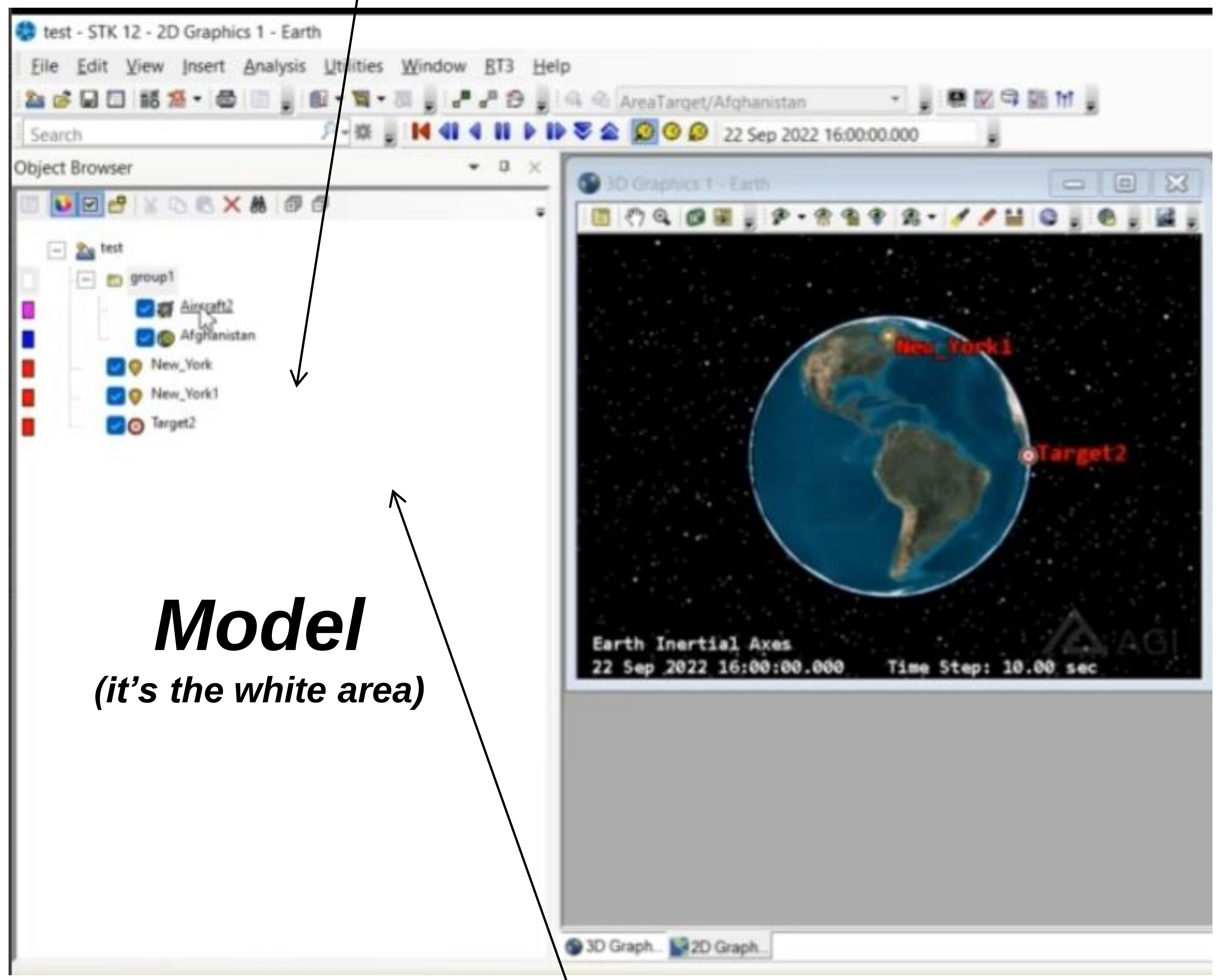
A model view



STK, Catia, etc

An Instance

This is where we make an instance



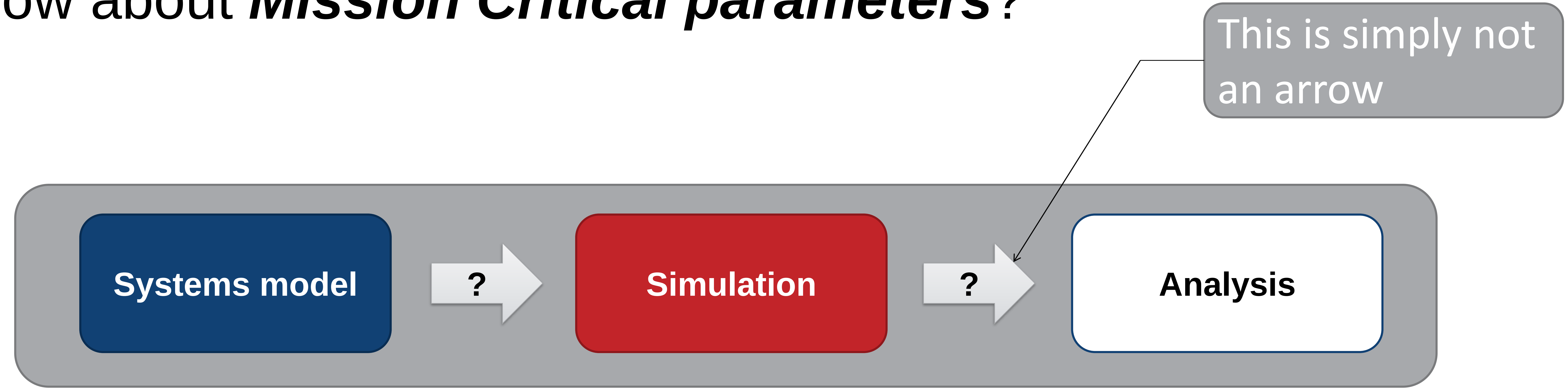
Model

(it's the white area)

When I start to build the mission model, I'm looking for a way to build this type of instance from the SysML model.

Defining Purpose, Objectives, Goals, and Scope

- Before we start modeling, we need to define a clear purpose, objective, goals, and scope
  - What is the model informing? (a document, a sim, an analysis)
  - Do we need to inform structure? How does one do this?
  - How about ***Mission Critical parameters***?

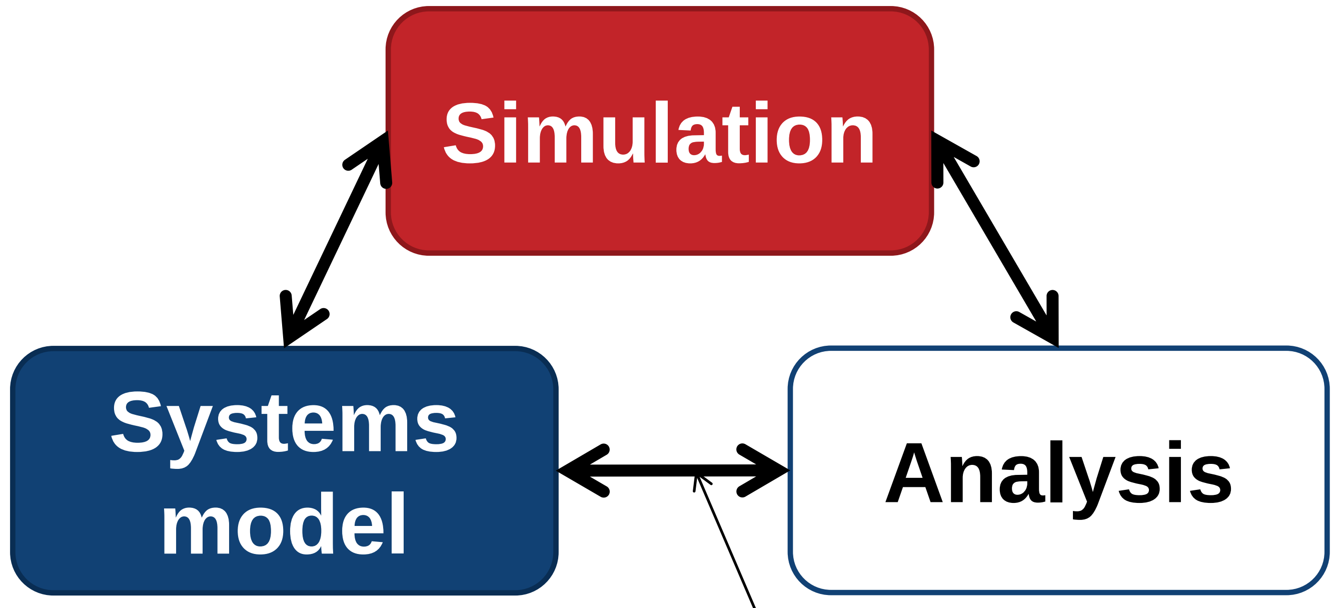


Creating Informed and Informing Models

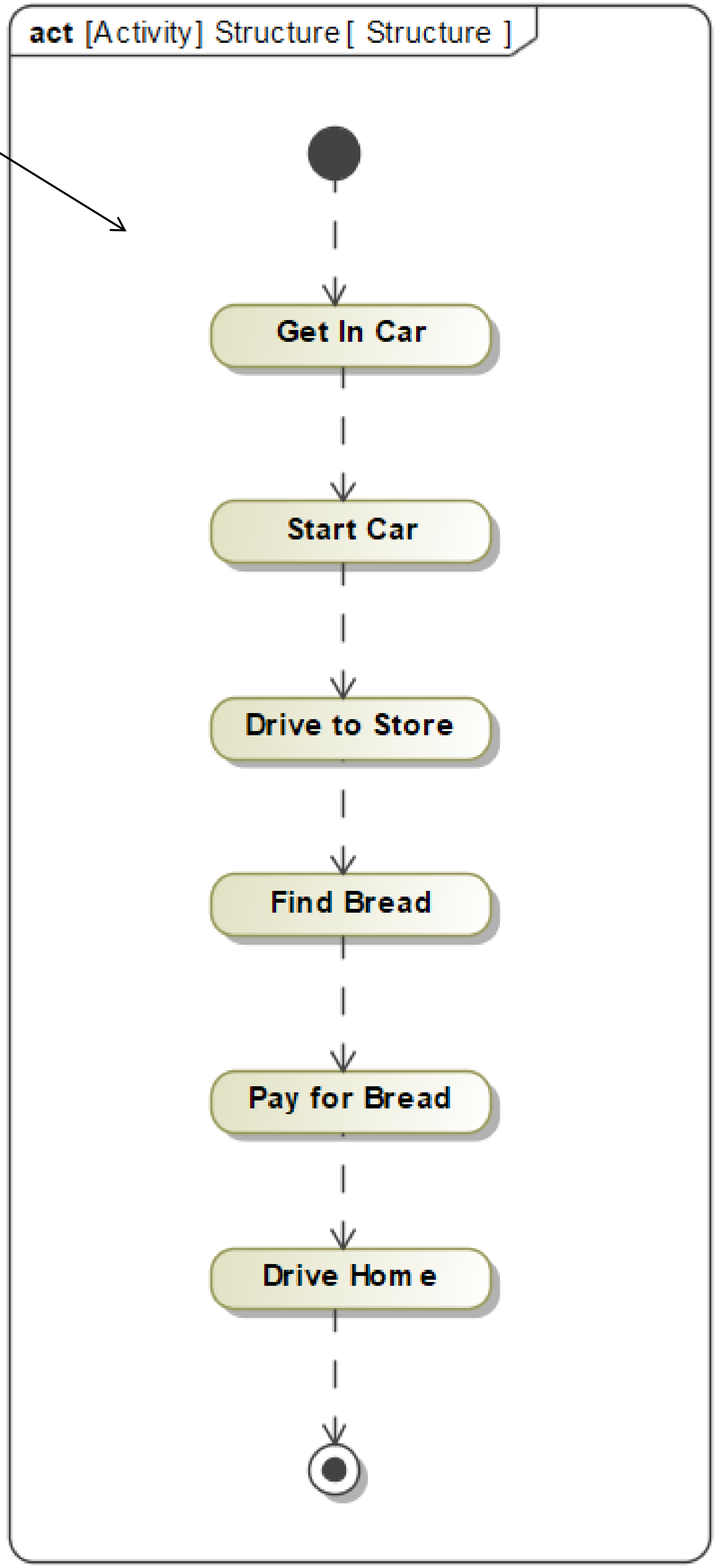
- Models should **inform** and **be informed** by other models to avoid creating malformed models
- SE provides ***Mission Critical Information***
  - – *What information does the simulation need?*

<u>Well All of it…</u>

What does this model inform?

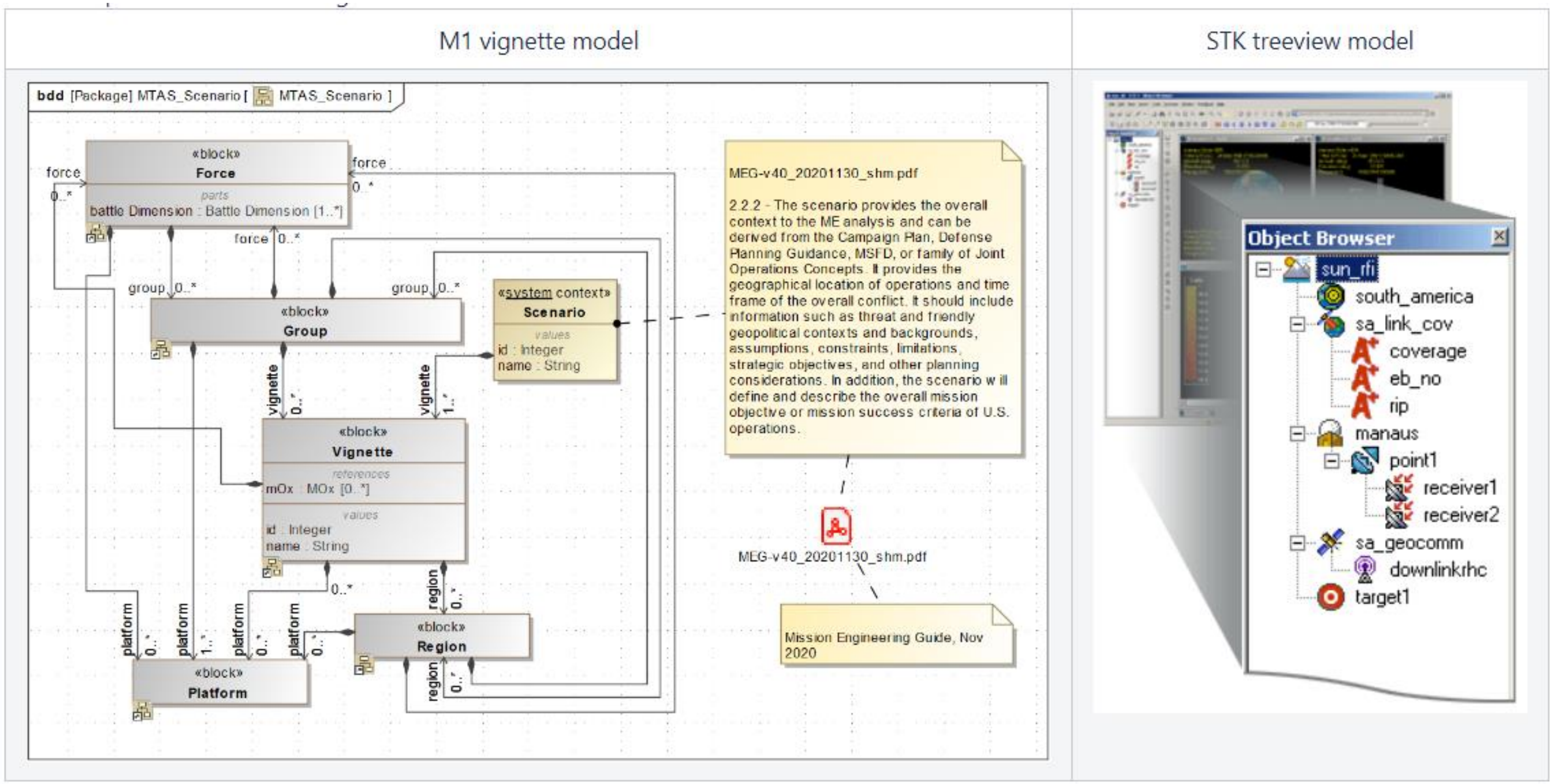


Not just data, but data and form



Treeview and Model Instances

- In software development we have a Treeview
- Typical for most engineering software
- We can mimic this method when building systems models that inform
- Particularly useful when informing the mission laydown



MEG-v40_20201130_shm.pdf

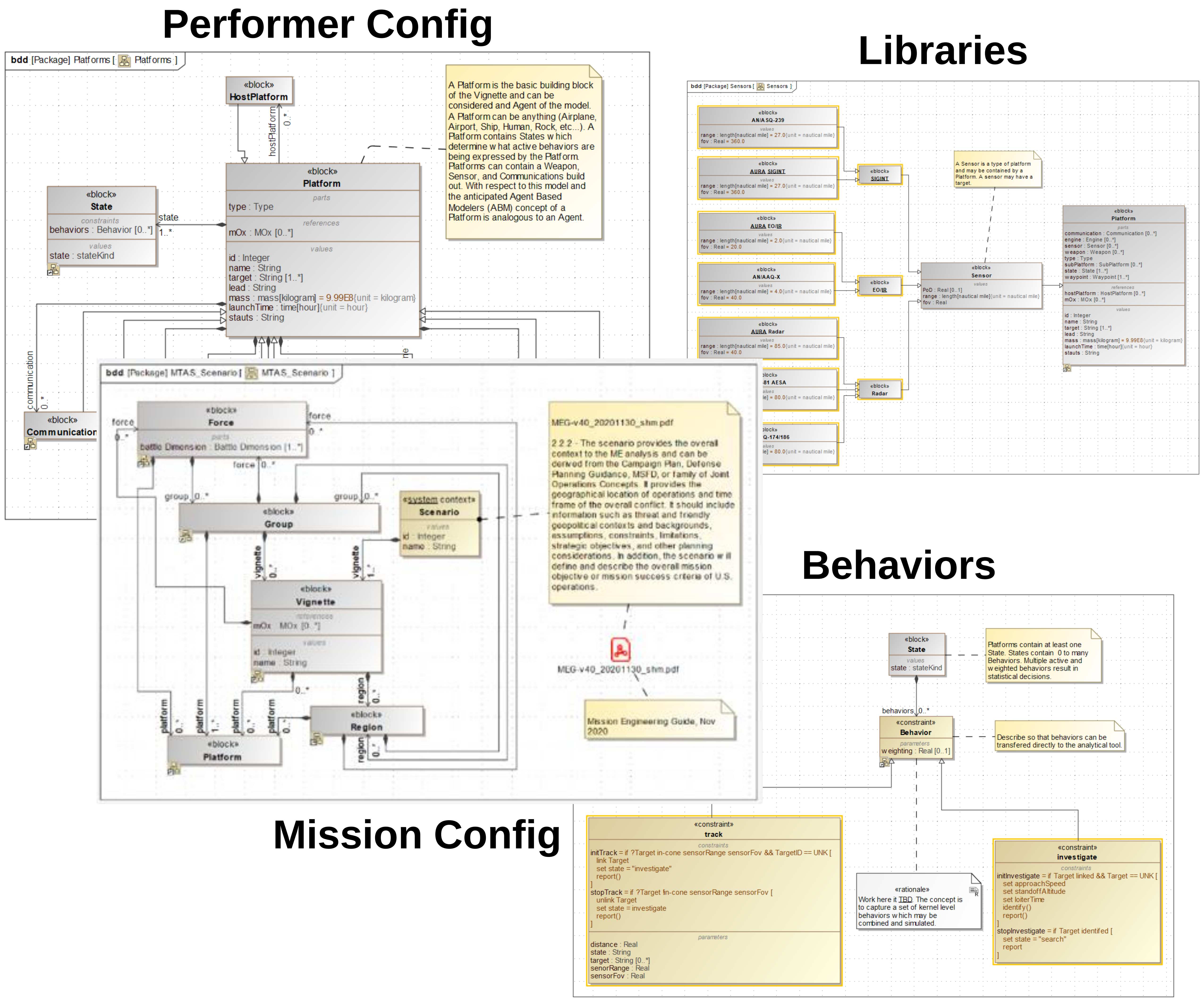
2.2.2 - The scenario provides the overall context to the ME analysis and can be derived from the Campaign Plan, Defense Planning Guidance, MSFD, or family of Joint Operations Concepts. It provides the geographical location of operations and time frame of the overall conflict. It should include information such as threat and friendly geopolitical contexts and backgrounds, assumptions, constraints, limitations, strategic objectives, and other planning considerations. In addition, the scenario w ill define and describe the overall mission objective or mission success criteria of U.S. operations.

Mission Engineering Guide, Nov 2020

To inform, the systems model can simply take the form of a ***TreeView***

Thinking About Model Structure

- What structure do we need to model a mission? mission model, performer, behaviors, libraries…
- If we are planning on simulating using an ABM what should the SE model look like?



A Platform is the basic building block of the Vignette and can be considered and Agent of the model. A Platform can be anything (Airplane, Airport, Ship, Human, Rock, etc...). A Platform contains States w hich determine w hat active behaviors are being expressed by the Platform. Platforms can contain a Weapon, Sensor, and Communications build out. With respect to this model and the anticipated Agent Based Modelers (ABM) concept of a Platform is analogous to an Agent.

A Sensor is a type of platform and may be contained by a Platform. A sensor may have a target.

MEG-v40_20201130_shm.pdf

2.2.2 - The scenario provides the overall context to the ME analysis and can be derived from the Campaign Plan, Defense Planning Guidance, MSFD, or family of Joint Operations Concepts. It provides the geographical location of operations and time frame of the overall conflict. It should include information such as threat and friendly geopolitical contexts and backgrounds, assumptions, constraints, limitations, strategic objectives, and other planning considerations. In addition, the scenario w ill define and describe the overall mission objective or mission success criteria of U.S. operations.

Platforms contain at least one State. States contain 0 to many Behaviors. Multiple active and w eighted behaviors result in statistical decisions.

Describe so that behaviors can be transfered directly to the analytical tool.

Work here it TBD. The concept is to capture a set of kernel level behaviors w hich may be combined and simulated.

```
constraints
initTrack = if ?Target in-cone sensorRange sensorFov && TargetID == UNK [
  link Target
  set state = "investigate"
  report()
]
stopTrack = if ?Target !in-cone sensorRange sensorFov [
  unlink Target
  set state = investigate
  report()
]
parameters
distance : Real
state : String
target : String [0..*]
senorRange : Real
sensorFov : Real
```

```
constraints
initInvestigate = if Target linked && Target == UNK [
  set approachSpeed
  set standoffAltitude
  set loiterTime
  identify()
  report()
]
stopInvestigate = if Target identifed [
  set state = "search"
  report
]
```

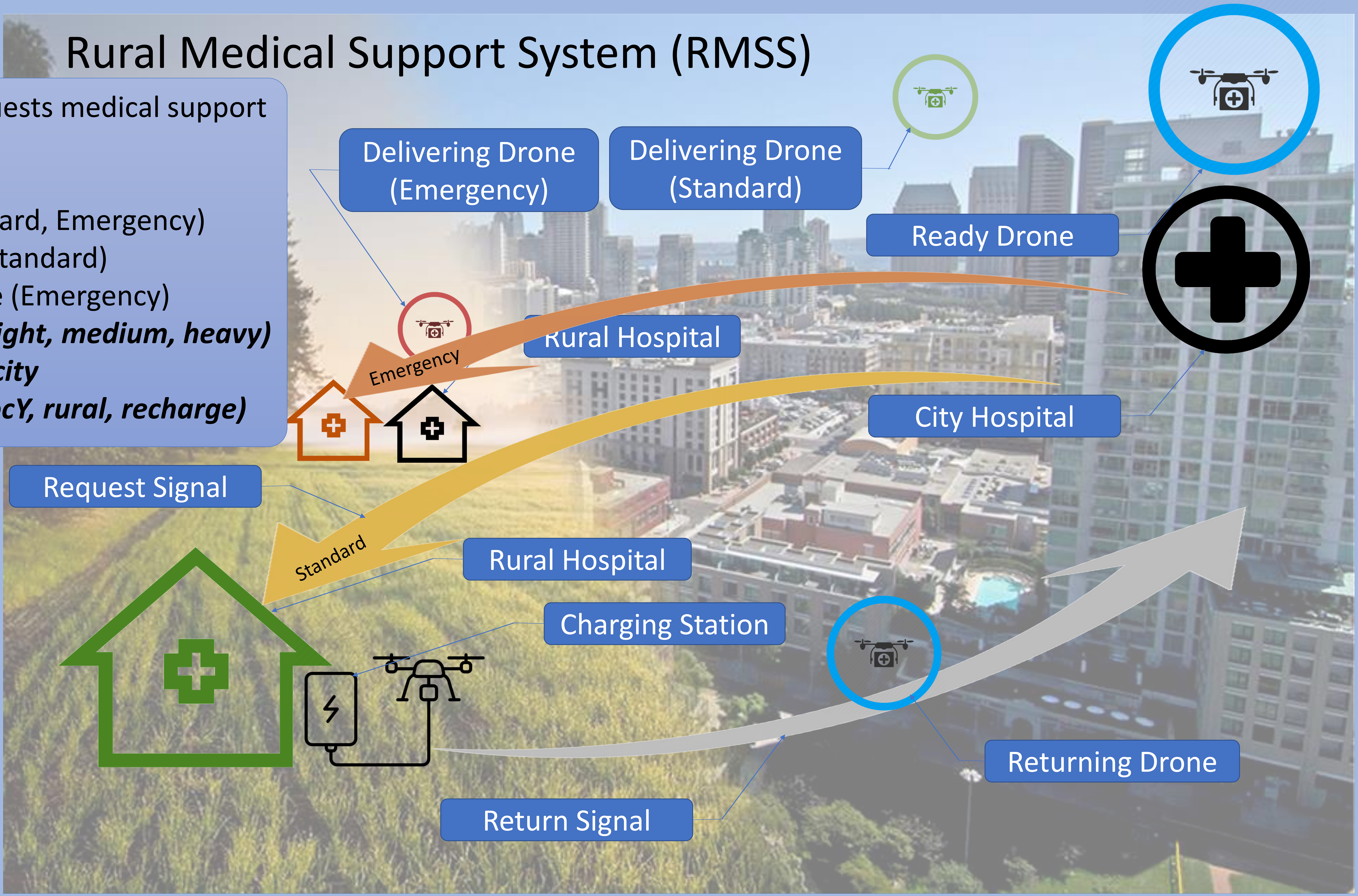
Demo - Building a Simple Mission Model

- Demonstrate building a mission model
  - Use a remote medical support system as an example

Rural Medical Support System (RMSS)

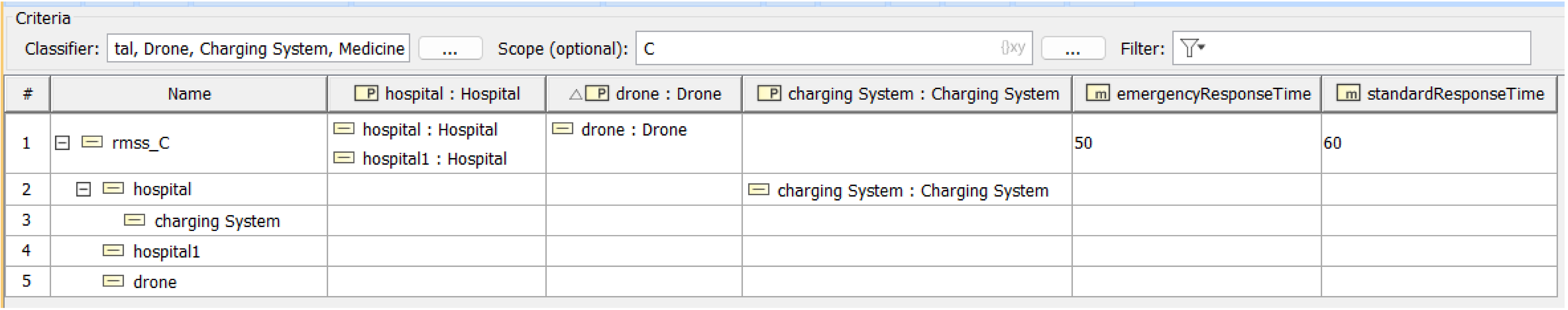
- Rural Hospital requests medical support (medicine)
- Request queue
  - Signals (Standard, Emergency)
  - FIFO queue (Standard)
  - Priority queue (Emergency)
- ***Multi-size drone (light, medium, heavy)***

  Cargo capacity
- ***Hospitals (LocX, LocY, rural, recharge)***



Demo - Creating Instance Tables

- Demo creating instance tables for drones, hospitals, and medicine
  - Show to configure an instance table to behave as a treeview for building instances quickly

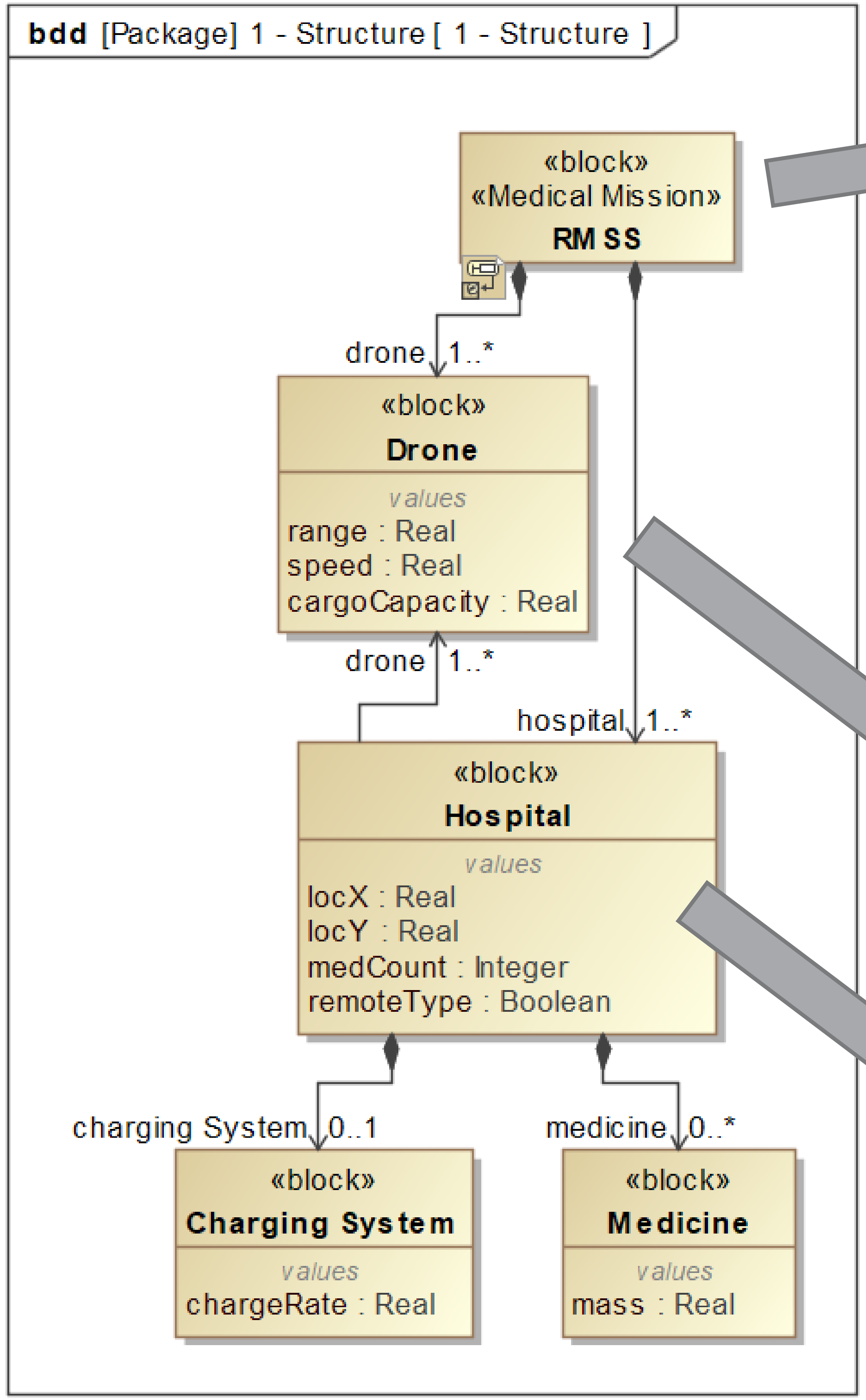


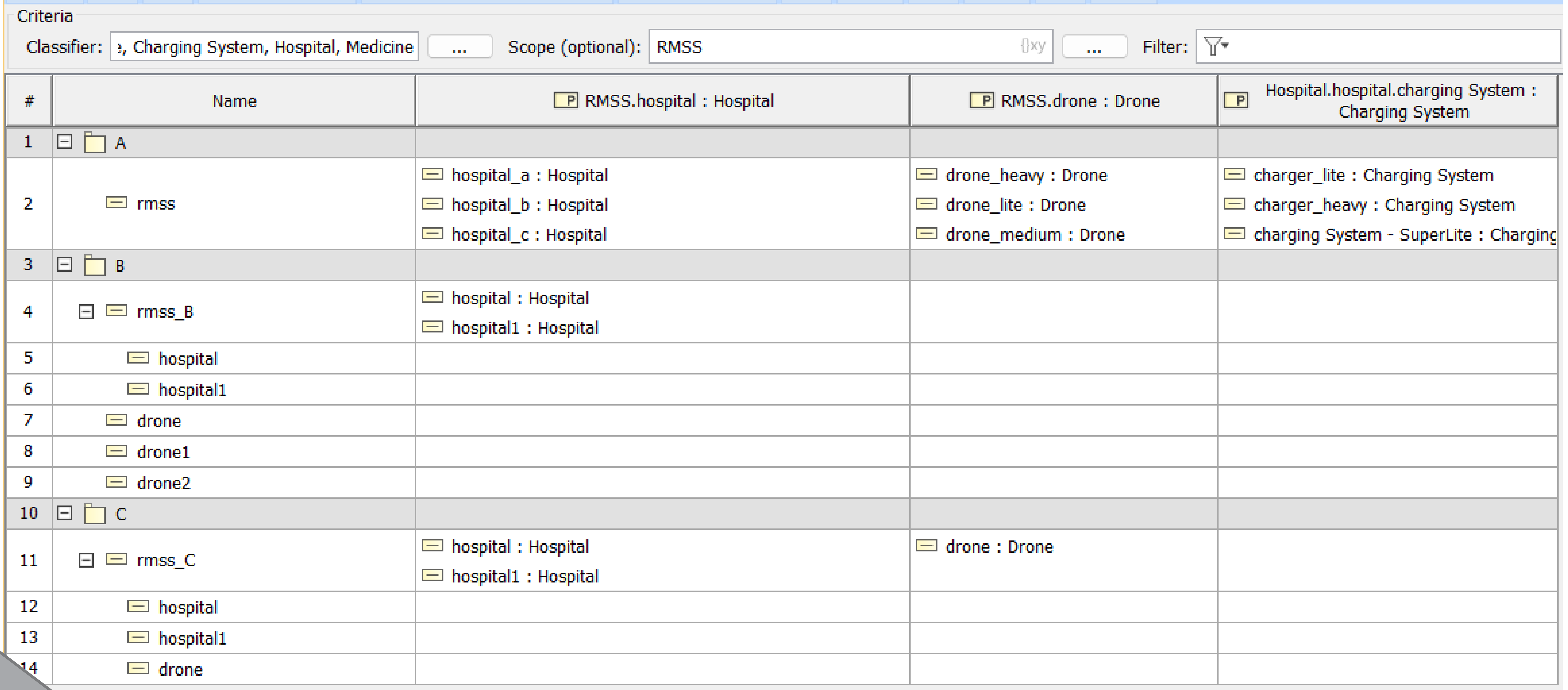
Criteria

Classifier: tal, Drone, Charging System, Medicine ... Scope (optional): C ... Filter:

#	Name	hospital : Hospital	drone : Drone	charging System : Charging System	emergencyResponseTime	standardResponseTime
1	rmss_C	hospital : Hospital hospital1 : Hospital	drone : Drone		50	60
2	hospital			charging System : Charging System		
3	charging System					
4	hospital1					
5	drone					

Demo - Mission Model Summary

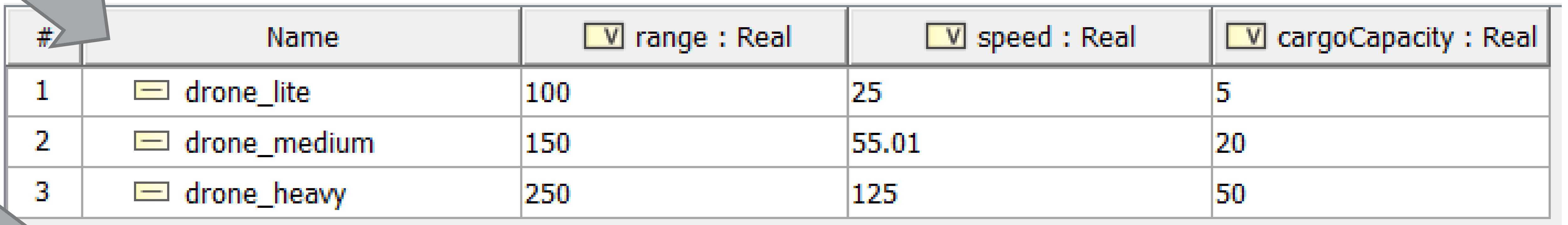




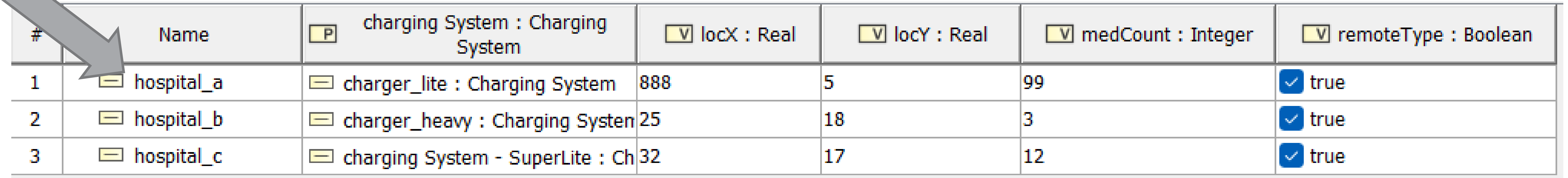
Criteria

Classifier: ɜ, Charging System, Hospital, Medicine ... Scope (optional): RMSS ... Filter:

#	Name	RMSS.hospital : Hospital	RMSS.drone : Drone	Hospital.hospital.charging System : Charging System
1	⊟ A			
2	rmss	hospital_a : Hospital hospital_b : Hospital hospital_c : Hospital	drone_heavy : Drone drone_lite : Drone drone_medium : Drone	charger_lite : Charging System charger_heavy : Charging System charging System - SuperLite : Charging
3	⊟ B			
4	⊟ rmss_B	hospital : Hospital hospital1 : Hospital		
5	hospital			
6	hospital1			
7	drone			
8	drone1			
9	drone2			
10	⊟ C			
11	⊟ rmss_C	hospital : Hospital hospital1 : Hospital	drone : Drone	
12	hospital			
13	hospital1			
14	drone			



#	Name	range : Real	speed : Real	cargoCapacity : Real
1	drone_lite	100	25	5
2	drone_medium	150	55.01	20
3	drone_heavy	250	125	50



#	Name	charging System : Charging System	locX : Real	locY : Real	medCount : Integer	remoteType : Boolean
1	hospital_a	charger_lite : Charging System	888	5	99	☑ true
2	hospital_b	charger_heavy : Charging Systen	25	18	3	☑ true
3	hospital_c	charging System - SuperLite : Ch	32	17	12	☑ true

Using Instances to Inform

- There are different ways to export models
  - – .mdzip: unzip and parse the xmi
  - – XML: great for git repo and text-based versioning
  - – API: perform CRUD operations on the model and access object
    - One big benefit of building skill with the API is that SysMLv2 is delivered with one right from the start potentially making transition easier
    - Add parts, delete parts, search the model, pull parameter sets, modify parameters, etc.

Using the API we can do almost anything that we can do in Cameo

Demo – Middleware (an Arrow) and API

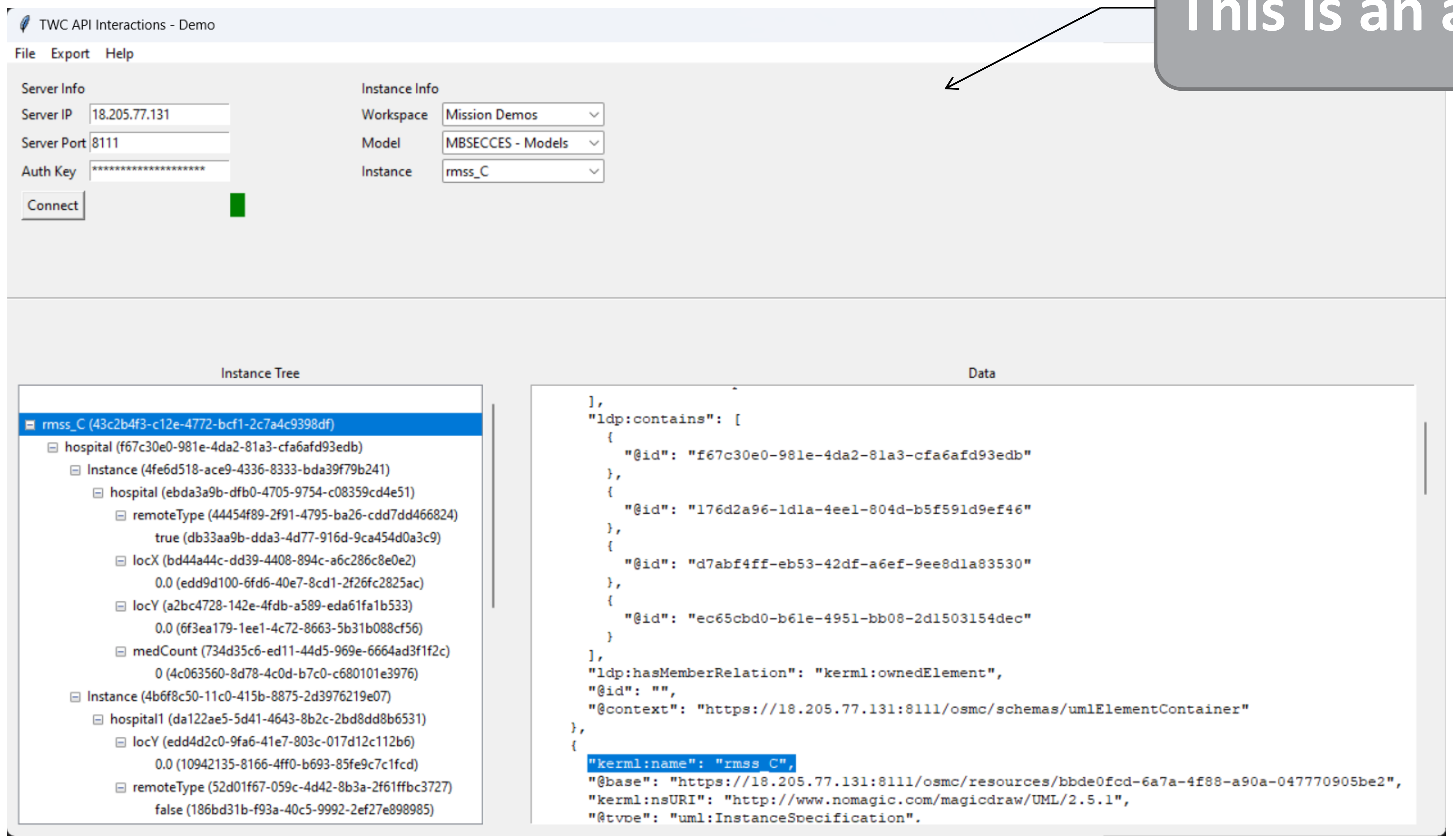
- Demonstrate accessing instances through the API and updating attributes
- Discuss the flexibility and CRUD operations the API offers

- All code here:
  - avianinc/2023-MBSE-Cyber-Experience-Symposium: Symposium materials and code (github.com)

Demo - Simulation Integration

- Export JSON file to demonstrate integration with Simulation
- Show how updating the instance affects the simulation

This is an arrow ☺



Server Info

Server IP: 18.205.77.131

Server Port: 8111

Auth Key: ********************

Instance Info

Workspace: Mission Demos

Model: MBSECCES - Models

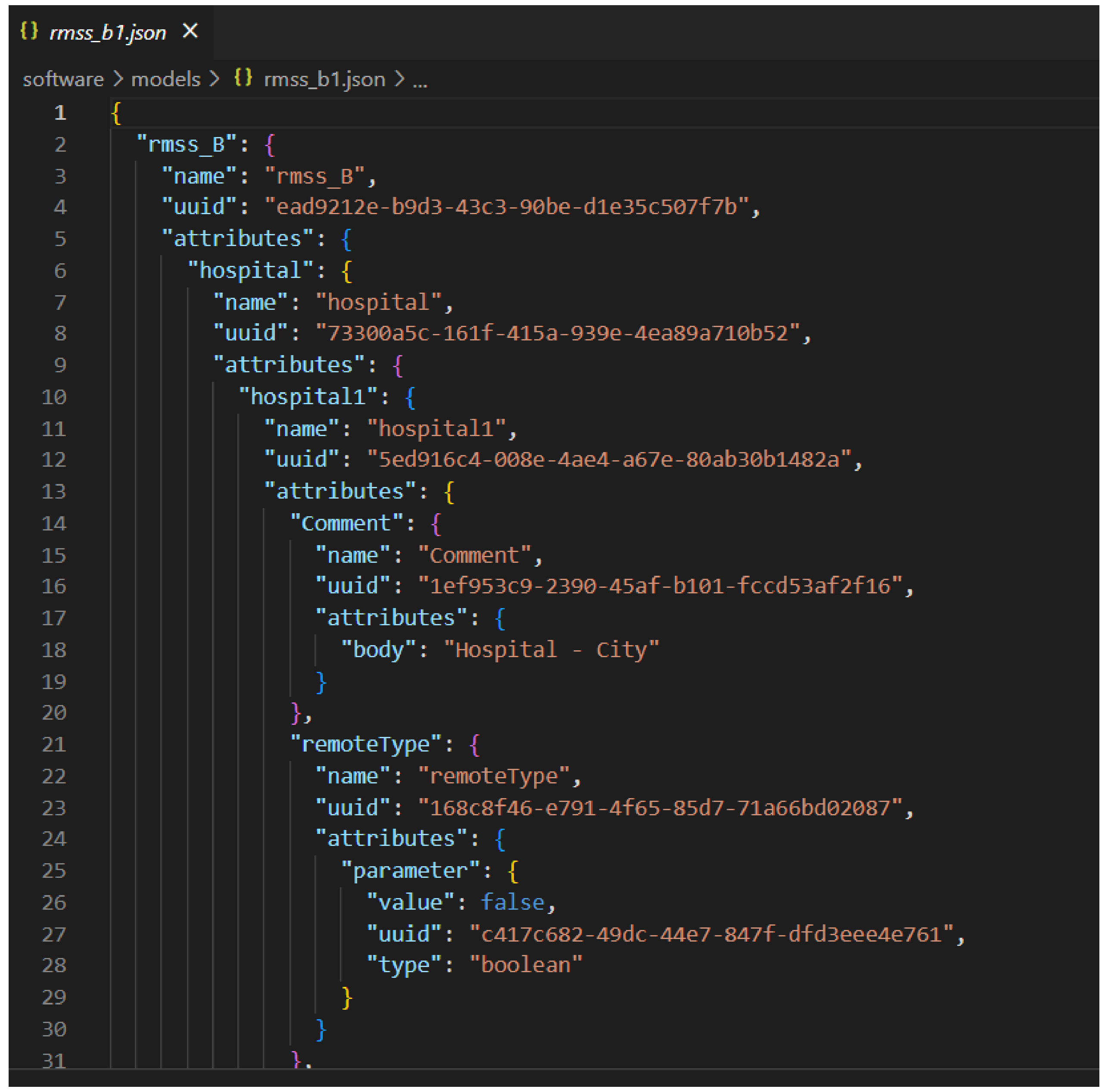
Instance: rmss_C

Instance Tree

```
⊟ rmss_C (43c2b4f3-c12e-4772-bcf1-2c7a4c9398df)
    ⊟ hospital (f67c30e0-981e-4da2-81a3-cfa6afd93edb)
        ⊟ Instance (4fe6d518-ace9-4336-8333-bda39f79b241)
            ⊟ hospital (ebda3a9b-dfb0-4705-9754-c08359cd4e51)
                ⊟ remoteType (44454f89-2f91-4795-ba26-cdd7dd466824)
                      true (db33aa9b-dda3-4d77-916d-9ca454d0a3c9)
                ⊟ locX (bd44a44c-dd39-4408-894c-a6c286c8e0e2)
                      0.0 (edd9d100-6fd6-40e7-8cd1-2f26fc2825ac)
                ⊟ locY (a2bc4728-142e-4fdb-a589-eda61fa1b533)
                      0.0 (6f3ea179-1ee1-4c72-8663-5b31b088cf56)
                ⊟ medCount (734d35c6-ed11-44d5-969e-6664ad3f1f2c)
                      0 (4c063560-8d78-4c0d-b7c0-c680101e3976)
        ⊟ Instance (4b6f8c50-11c0-415b-8875-2d3976219e07)
            ⊟ hospital1 (da122ae5-5d41-4643-8b2c-2bd8dd8b6531)
                ⊟ locY (edd4d2c0-9fa6-41e7-803c-017d12c112b6)
                      0.0 (10942135-8166-4ff0-b693-85fe9c7c1fcd)
                ⊟ remoteType (52d01f67-059c-4d42-8b3a-2f61ffbc3727)
                      false (186bd31b-f93a-40c5-9992-2ef27e898985)
```

Data

```
    ],
    "ldp:contains": [
      {
        "@id": "f67c30e0-981e-4da2-81a3-cfa6afd93edb"
      },
      {
        "@id": "176d2a96-1d1a-4ee1-804d-b5f591d9ef46"
      },
      {
        "@id": "d7abf4ff-eb53-42df-a6ef-9ee8d1a83530"
      },
      {
        "@id": "ec65cbd0-b61e-4951-bb08-2d1503154dec"
      }
    ],
    "ldp:hasMemberRelation": "kerml:ownedElement",
    "@id": "",
    "@context": "https://18.205.77.131:8111/osmc/schemas/umlElementContainer"
  },
  {
    "kerml:name": "rmss_C",
    "@base": "https://18.205.77.131:8111/osmc/resources/bbde0fcd-6a7a-4f88-a90a-047770905be2",
    "kerml:nsURI": "http://www.nomagic.com/magicdraw/UML/2.5.1",
    "@type": "uml:InstanceSpecification",
```



```
{
  "rmss_B": {
    "name": "rmss_B",
    "uuid": "ead9212e-b9d3-43c3-90be-d1e35c507f7b",
    "attributes": {
      "hospital": {
        "name": "hospital",
        "uuid": "73300a5c-161f-415a-939e-4ea89a710b52",
        "attributes": {
          "hospital1": {
            "name": "hospital1",
            "uuid": "5ed916c4-008e-4ae4-a67e-80ab30b1482a",
            "attributes": {
              "Comment": {
                "name": "Comment",
                "uuid": "1ef953c9-2390-45af-b101-fccd53af2f16",
                "attributes": {
                  "body": "Hospital - City"
                }
              },
              "remoteType": {
                "name": "remoteType",
                "uuid": "168c8f46-e791-4f65-85d7-71a66bd02087",
                "attributes": {
                  "parameter": {
                    "value": false,
                    "uuid": "c417c682-49dc-44e7-847f-dfd3eee4e761",
                    "type": "boolean"
                  }
                }
              },
```

Summary and Q&A

- Have a clear understanding of what a model is going to be used for ***prior*** to stating to model.
- Models ***inform*** and are ***informed by*** other models
- It can help to think of a model as a ***treeview*** *(instance)*
- The ***arrows*** are not easy, don’t forget the arrows
- Using the ***API*** is very powerful and should be a major part of any project.

- Questions? I open the floor for discussion… be gentle ☺

AVIAN Inc.

John DeHart (jdehart@avian.com)

Thank you!

Headquarters
301-866-2070
22111 Three Notch Road
Lexington Park, MD 20653

Panama City Beach
850-775-4904
1813 Thomas Drive
Panama City Beach, FL 32408