34th Annual INCOSE international symposium

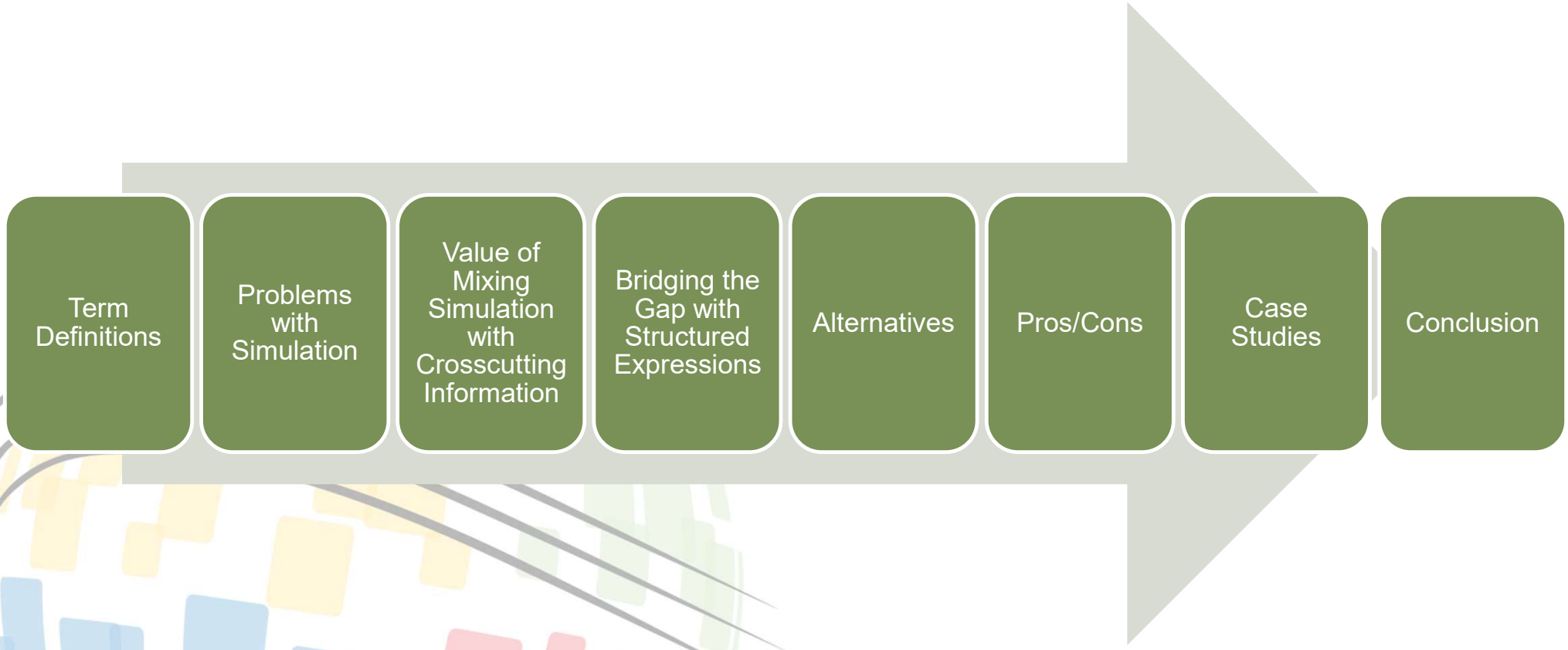
hybrid event

Dublin, Ireland
July 2 - 6, 2024

Enhancing Capabilities of Parametric Diagrams with Opaque Behaviors

By Sean Densford, Daniel Brookshier, and Barry Papke

Agenda


Term Definitions
Problems with Simulation
Value of Mixing Simulation with Crosscutting Information
Bridging the Gap with Structured Expressions
Alternatives
Pros/Cons
Case Studies
Conclusion

Terms and Definitions

Simulation Information

- Model data that is available during the runtime of a simulation

Cross-cutting Information

- Model data that is calculated from other model information, such as number of requirements

Structured Expression

- A tool feature in Cameo that allows model parsing of cross-cutting information based on relationships, logic, and/or code

Opaque Behavior

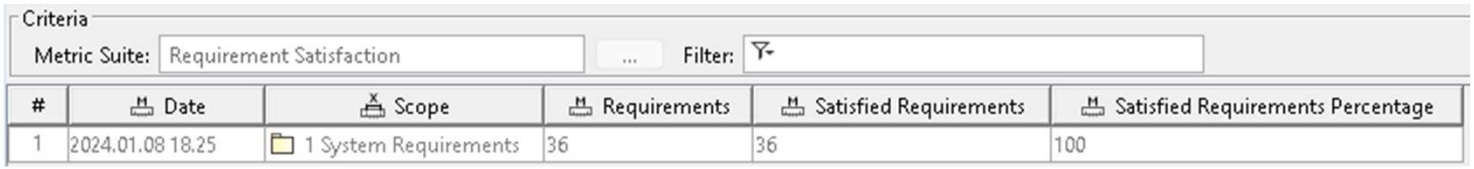
- A reusable model element that holds scripts and/or Structured Expressions

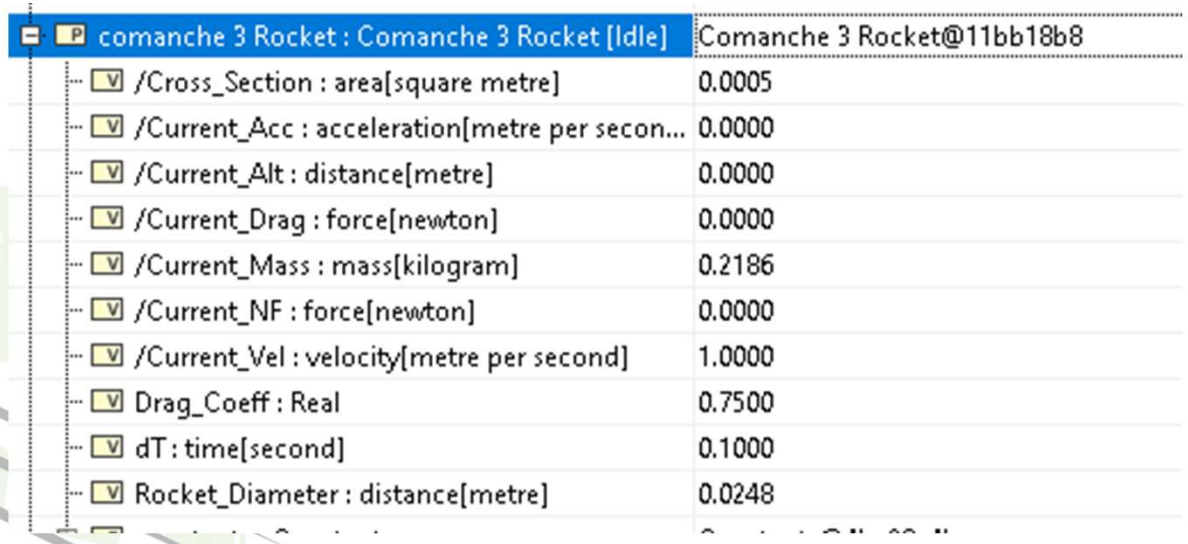
Opaque Expression

- An expression that allows another language instead of UML/SysML to be used for evaluation

Difference Between Cross-cutting Information and Simulation Information

- Cross-cutting information is calculated from model information, such as the number of requirements that are satisfied
- Simulation Information is calculated from values during runtime

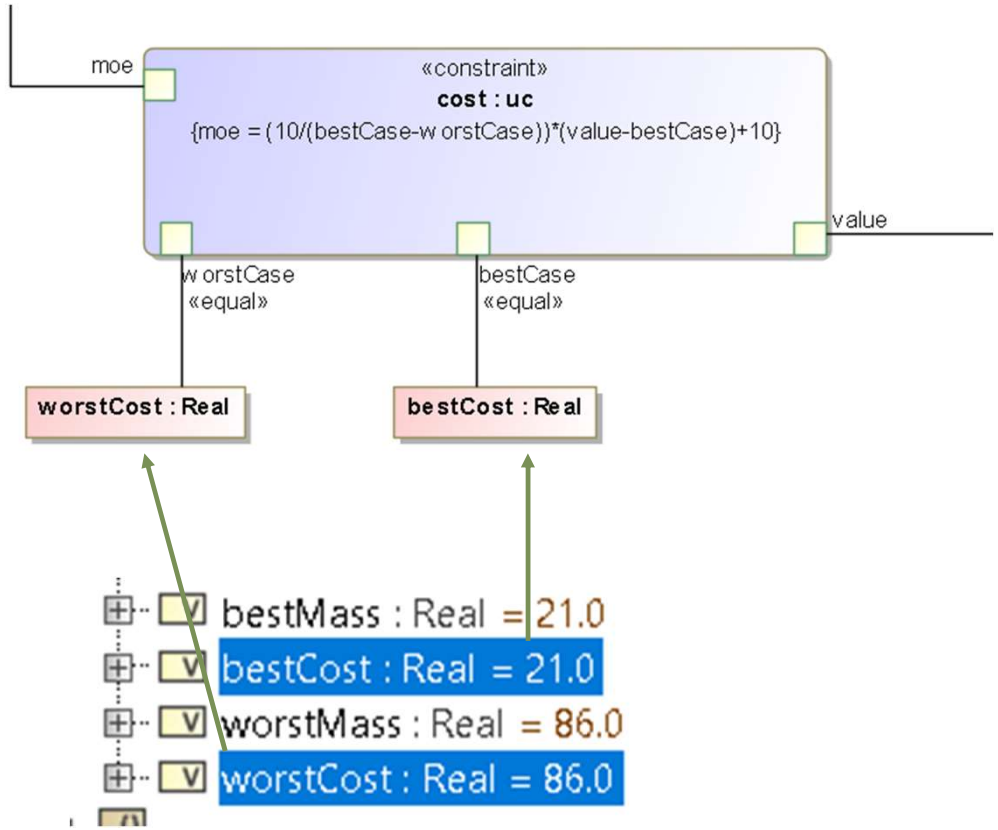




Snippets From Cameo 24x

Problem with Simulation

- There is a disconnect between Parametric Diagram Simulation and Model Data



What if you want to calculate this based off other model values at simulation?

What is the Value of Integrating the Two?

Cross-cutting information in simulation

- Calculate value properties based off real time model metadata
- Automate taking information from multiple sources to set values

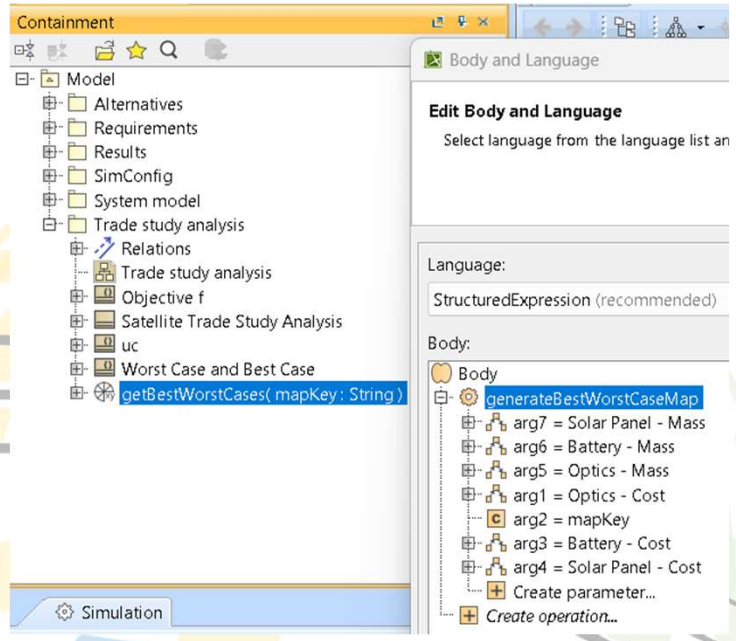
Bridging the Gap: Simple Example

1. Create an Opaque Behavior using a Structured Expression
2. Create a Constraint Block with desired value properties
3. Create a Constraint with a script to invoke the opaque behavior

This 3-Step Pattern can be Applied to Other Tools and Models
Example Based on Trade Study Tutorial

Bridging the Gap: Structured Expression

1. Create an Opaque Behavior using a Structured Expression

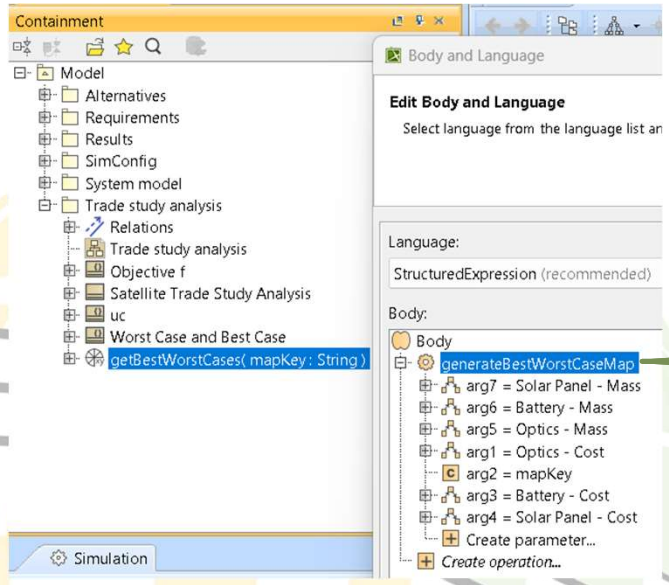


- arg1 – Metachain to get Optics Cost
- arg2 – mapKey input to function
- arg3 – Metachain to get Battery Cost
- arg4 – Metachain to get Solar Panel Cost
- arg5 – Metachain to get Optics Mass
- arg6 – Metachain to get Battery Mass
- arg7 – Metachain to get Solar Panel Mass

Calculated Real Time During Simulation

Bridging the Gap: Scripting a Map

1. Create an Opaque Behavior using a Structured Expression

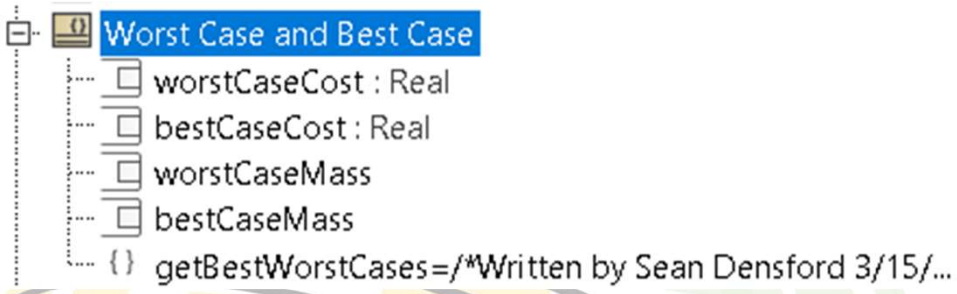


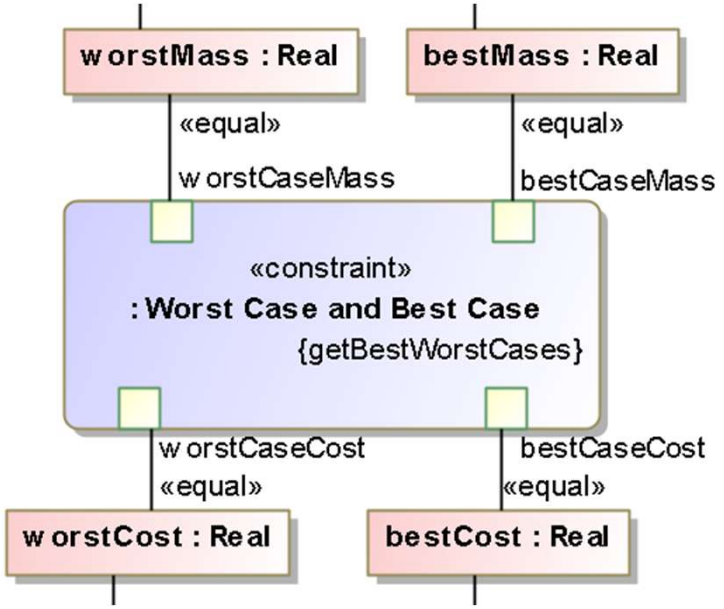

```
//Define a map to hold the Key as the name of each array and the Value
as the size of each array
//Sum worst by min() and best by max()

def costMassMap =
['worstCost':arg1.value.max()+arg3.value.max()+arg4.value.max(),'bestCos
t':arg1.value.min()+arg3.value.min()+arg4.value.min(),
'worstMass':arg5.value.max()+arg6.value.max()+arg7.value.max(),'bestMass
':arg5.value.min()+arg6.value.min()+arg7.value.min()]

//Call the map using arg2 string from the user
costMassMap[arg2]
```

Groovy Script Builds a Callable map with the Cost and Mass

Bridging the Gap: Binding the mapKeys

2. Create a Constraint Block with desired value properties

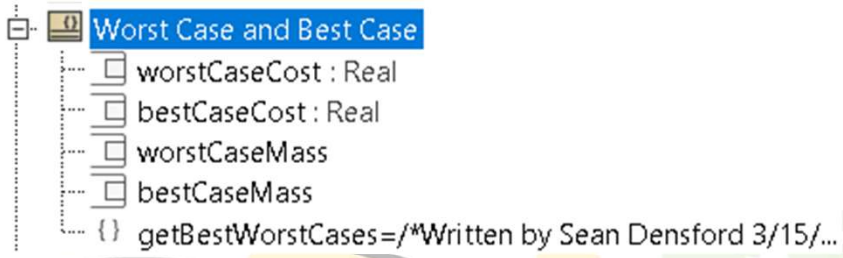




Value Properties must be Bound to each Constraint Parameter

Bridging the Gap: Expressing the Opaque Behavior

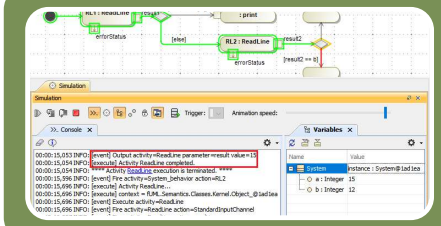
3. Create a Constraint with a script to invoke the opaque behavior




```
/*
The following are a set of calls that use a key (i.e worstCost,
bestCost, etc) to grab the corresponding
value from the map getBestWorstCases and sets them equal to the
corresponding constraint parameter to be
binded to the value property
*/

worstCaseCost = ExpressionHelper.call(behaviorExpression,
"worstCost");
bestCaseCost = ExpressionHelper.call(behaviorExpression,
"bestCost");
worstCaseMass = ExpressionHelper.call(behaviorExpression,
"worstMass");
bestCaseMass = ExpressionHelper.call(behaviorExpression,
"bestMass");
```

Groovy Script Expresses the Opaque Behavior

Alternatives



Create a simulation behavior that matches the structured expression

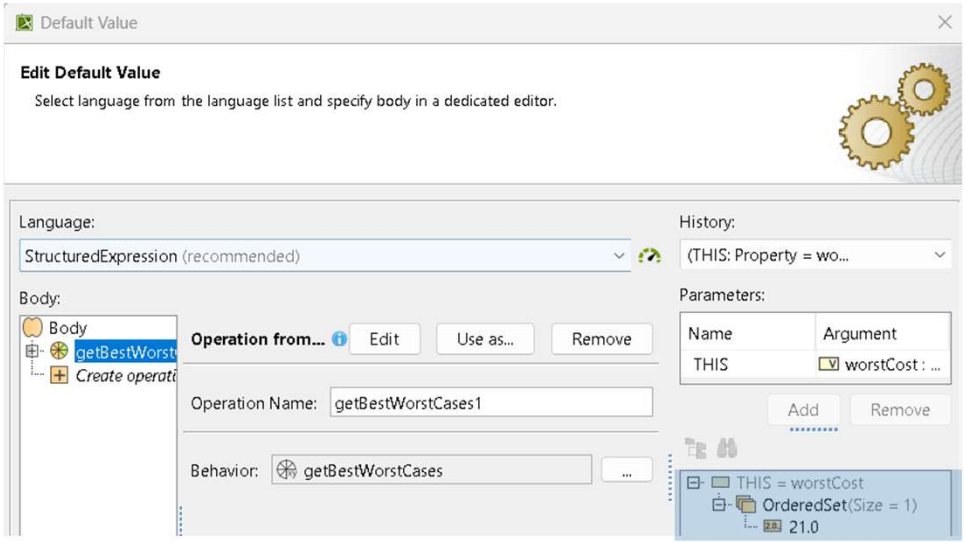
- Complex scripts

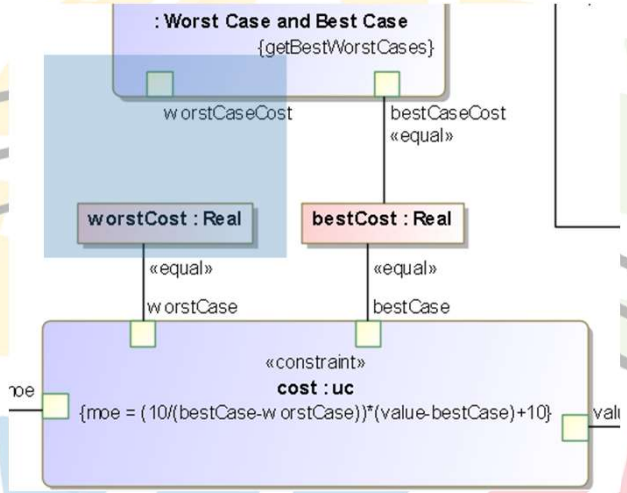
Manual

Direct use of complex scripts

Alternative: Why Can't We Directly Use a Structured Expression?

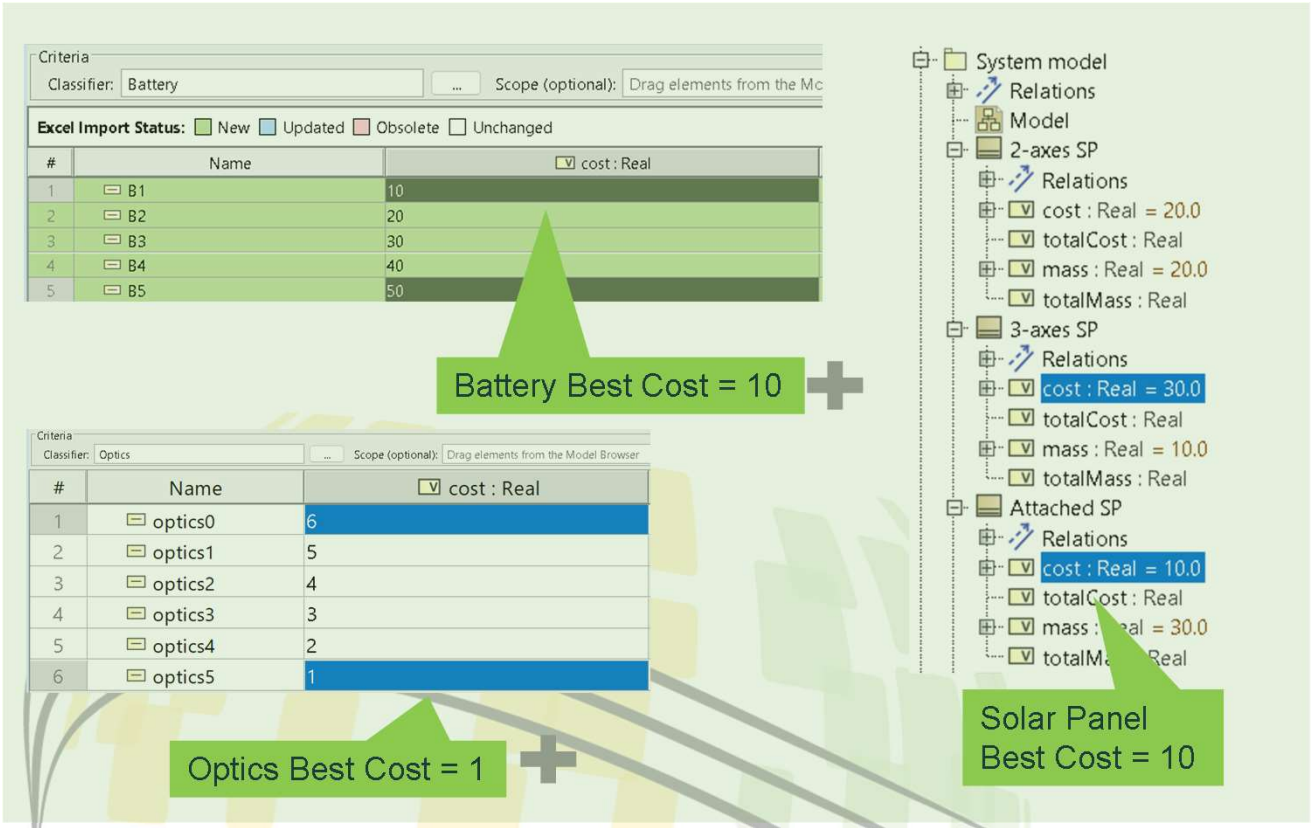
- Default Value
  - Works in evaluation, but not Simulation
  - Nothing to tell the model to express the opaque behavior

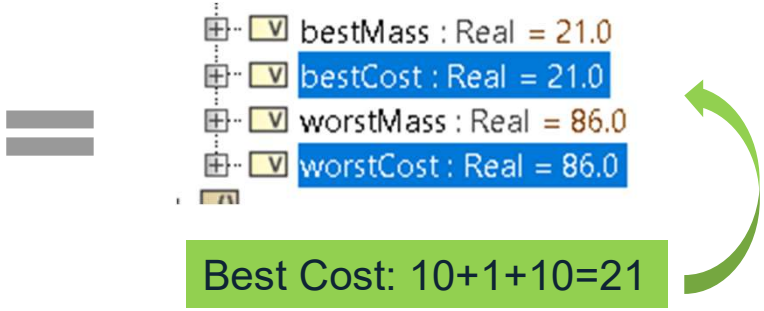




TradeStudy Satellite Trade Study Analysis is started.
90 of 90 alternatives evaluated for Satellite Trade Study Analysis trade study (9 sec).
Winning configuration: : optics3, : #2, : 3-axes SP
Winning score = 0.0
WARN: 35 alternatives obtained the same winning score.

Alternative: Manual Process





3 Different Places to Look for Data for Manual Calculation

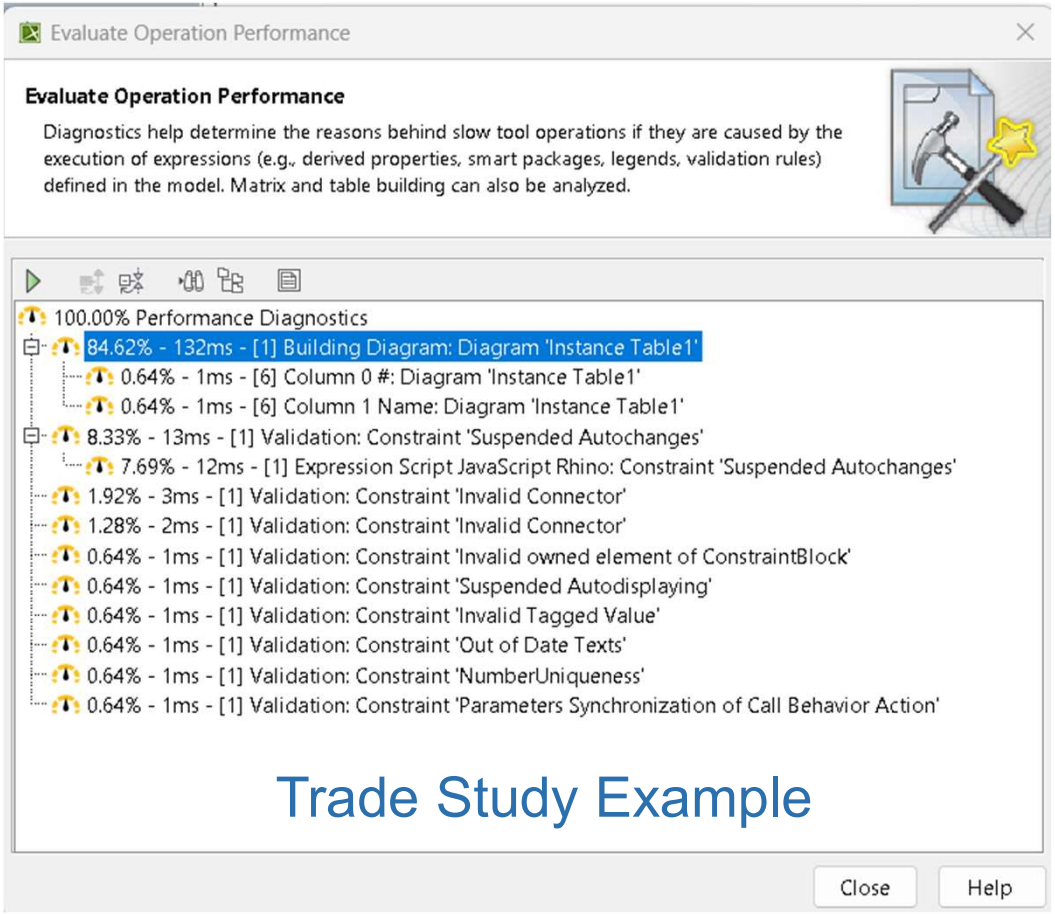
Pros/Cons of this Method - Performance

Pros

- Avoids need to search the model manually for all values
- Automates Calculation Process

Cons

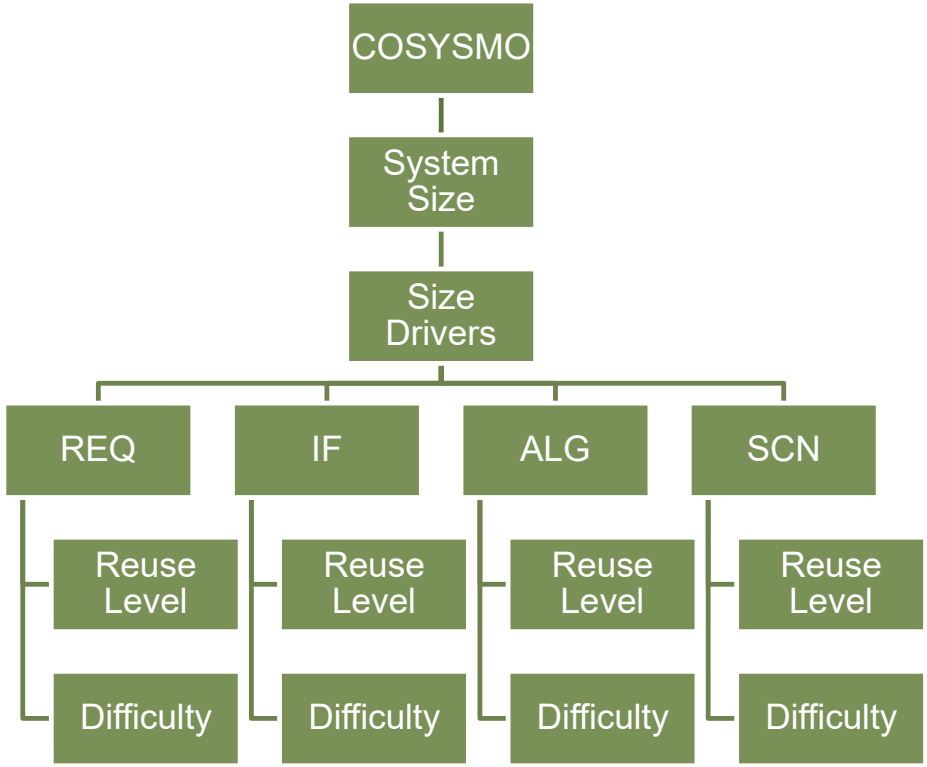
- Code could be optimized; this was not investigated thoroughly



Trade Study Example

Case Study – Parametric Cost Estimation

- The Constructive Systems Engineering Cost Model (COSYSMO) estimates the system engineering effort by calculating a value called System Size.
- System Size is determined by counting the number of size drivers:
  - system requirements (REQ)
  - system interfaces (IF)
  - system algorithms (ALG)
  - operational scenarios (SCN.)
- Each size driver has two properties that have weighted values: Reuse Level and Difficulty.



Case Study – Quick Background on COSYSMO

- There are 6 Design With Reuse (DWR) levels and 3 Difficulty levels in COSYSMO.
- Each level has a numerical weight.
- This results in 72 possible DWR categories (4 size drivers x 6 reuse levels x 3 complexity levels).
- There are rules to determine which REQ, IF, ALG, or SCN size driver model elements to include in the cost estimate.
- The system engineer needs to select the size driver model elements and assign the reuse level and difficulty level.
- **The problem:**
  - The only data in the model are the reuse and complexity levels (string value types).
  - The numerical weights are not value properties in the model (real value types.)

Difficulty

Size Driver Type	Easy	Nominal	Difficult
System Requirement (REQ)	0.5	1.0	4.5
System Interface (IF)	1.9	4.0	9.0
System Algorithm (ALG)	1.9	3.8	9.8
Operational Scenario (SCN)	6.4	13.6	26.3

Design With Reuse

	New	Design Modified	Design Implemented	Adapted for Integration	Adopted for Integration	Managed
REQ	1.00	0.67	0.56	0.43	0.39	0.22
IF	2.80	1.87	1.58	1.21	1.09	0.61
ALG	4.10	2.74	2.31	1.78	1.59	0.89
SCN	14.40	9.61	8.10	6.24	5.59	3.13

Case Study – Quick Background on COSYSMO

Once the system size has been calculated, it must be applied to the Cost Estimating Relationship (CER) equation:

$$SE\ Effort = A \cdot Size^{E} \cdot CEM$$

Where:

SE Effort = total systems engineering effort in person hours
A = productivity constant
Size = System Size
E = productivity curve coefficient
CEM = composite effort multiplier

Expanded to show the full calculation:

$$PH_{Total} = A_{DWR} \cdot \left[\sum_{k} \left(\sum_{r} w_r \left(w_{e,k} \Phi_{e,k} + w_{n,k} \Phi_{n,k} + w_{d,k} \Phi_{d,k} \right) \right) \right]^{E_{DWR}} \cdot CEM_{DWR}$$

The next several slides show the implementation of the process.

Case Study – Assign Reuse Level and Complexity

- Requirements Example
  - Selections are made by applying a reuse level and complexity as stereotypes:
    - Selected
      - True/False
    - Reuse Level
      - Adapted, Adopted, Managed, Implemented, Modified, New
    - Difficulty
      - Easy, Nominal, Difficult

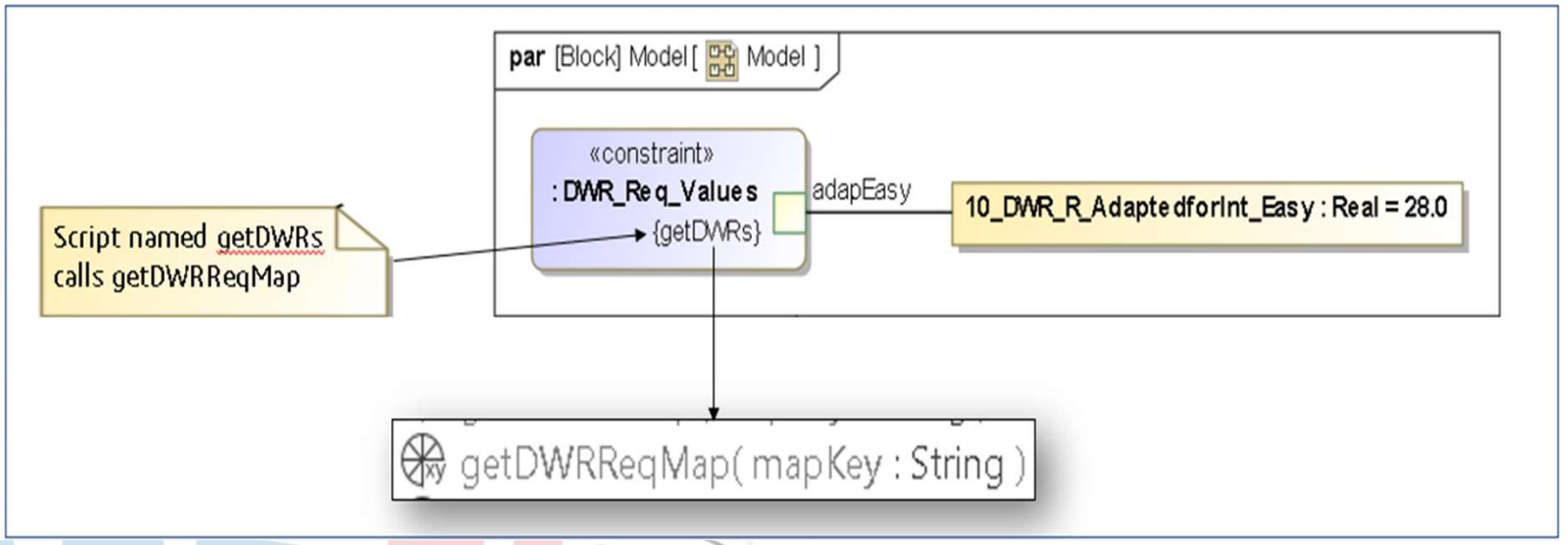
Criteria

Scope (optional): 1.1 Air Vehicle Spec ... Filter: Context (optional): Drag elements from the Model Brow

#	Id	Name	Text	Verify Method	DWR Selected	DWR_Categ...	DWR_Difficulty
1	AV-1	AV-1 System Requirements			<undefined>		
2	AV-2	AV-2 System Component Descriptions			<undefined>		
3	AV-2.1	AV-2.1 SAR/GMTI Capability			true	Adapted	Difficult
4	AV-2.2	AV-2.2 Communications Relay Capability			true	Adopted	Nominal
5	AV-2.4	AV-2.4 LOS and BLOS	LOS and BLOS command and control links shall (T) contain a primary and a secondary link.	Demonstrat	true	Managed	Nominal

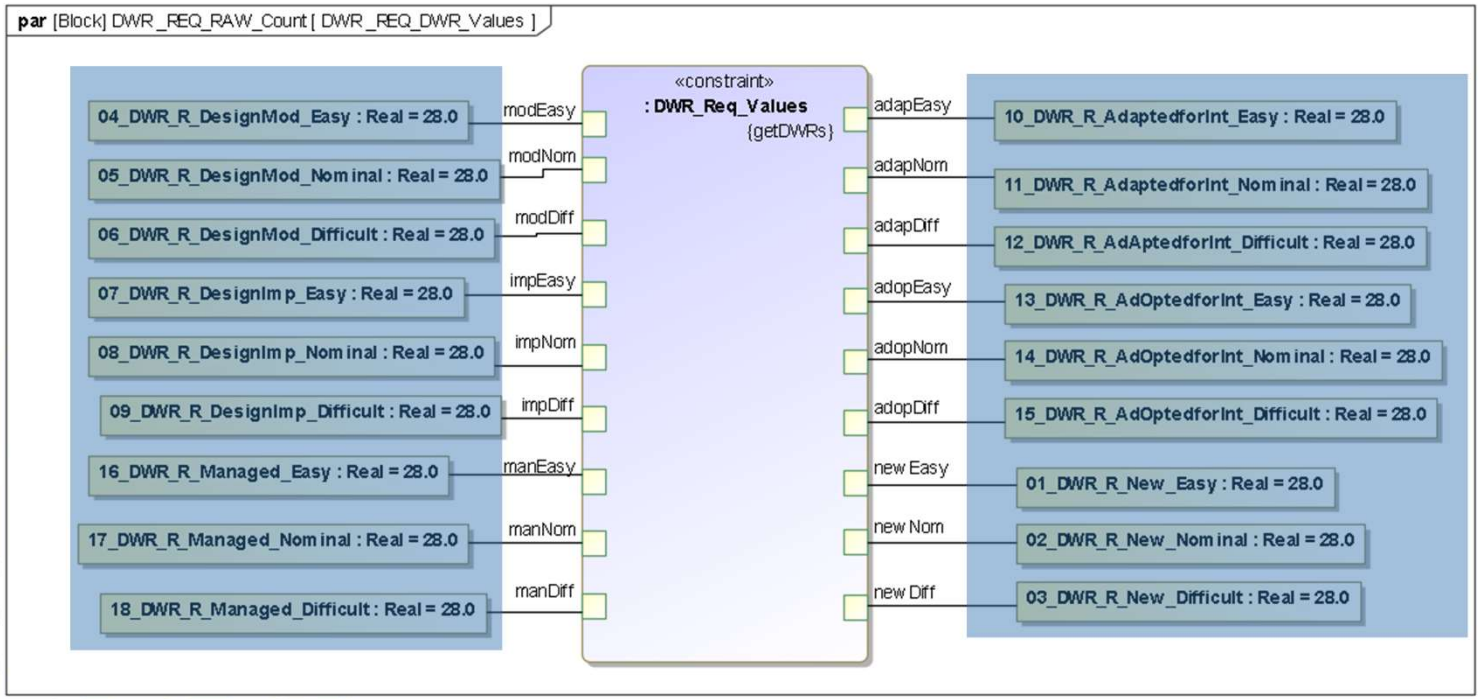
Case Study – Parse the Model for Size Drivers

- Analysis scripts were written to parse the model, count the number of model elements in each of the size driver “buckets,” and write the totals to their respective Size Driver Blocks.
- The analysis scripts are a combination of Stereotypes, Enumerations, and Opaque Behaviors.
- For example, the Opaque Behavior ‘getDWRReqMap’ calculates how many requirements there are in each reuse level and difficulty category.



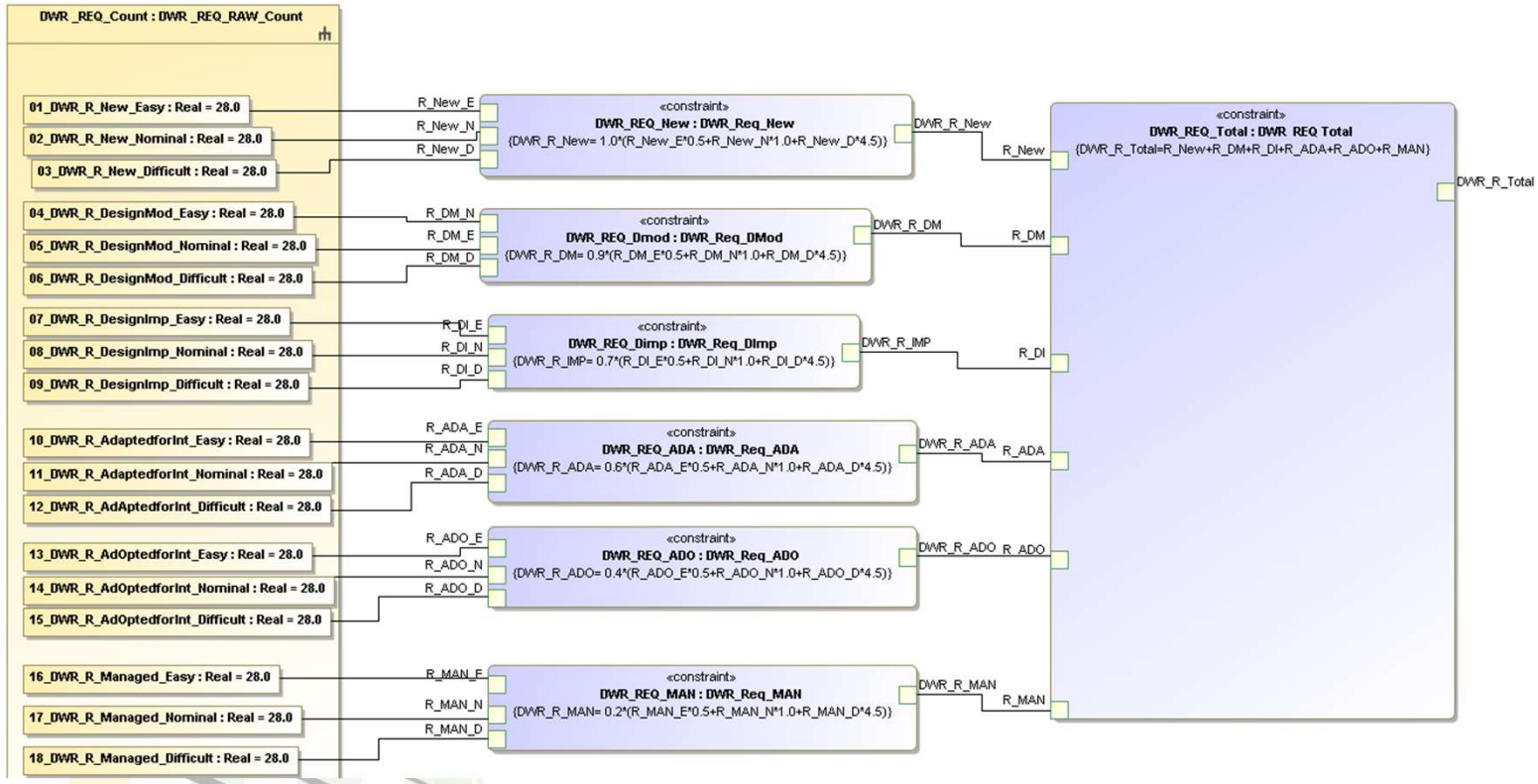
Case Study – Calculate the Totals for Each Size Driver

- A structured expression maps the requirement counts
- Each value property is bound to a constraint parameter corresponding to a mapKey



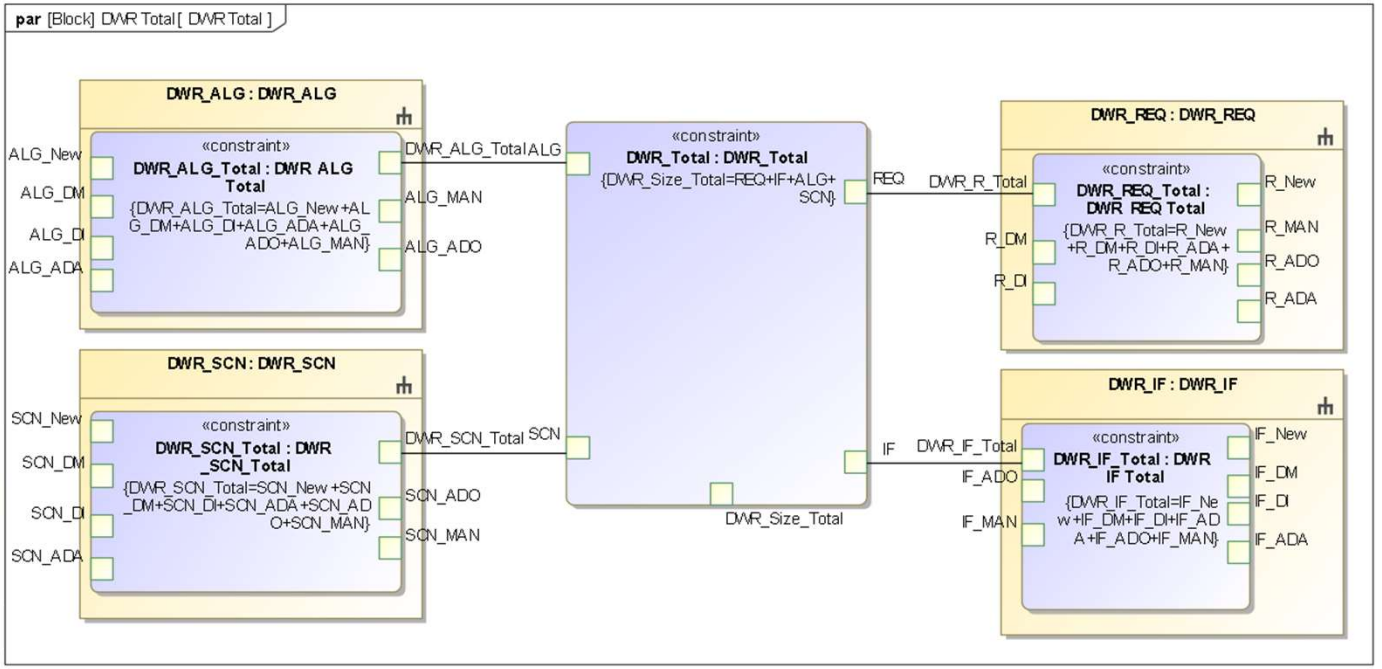
Case Study – Apply Reuse Level and Difficulty Weights

- Reuse level and difficulty weights are applied to each of the 6 reuse levels for each of the 4 size drivers using constraint blocks in a parametric diagram to compute each size driver total.



Case Study – Calculate Size Drivers and Apply CER coefficients

- The four size drivers are totaled and the resulting total SE effort is calculated, using constraint blocks in a parametric diagram as well.



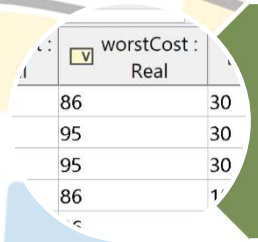
Case Study – Why is this so powerful?

None of the numerical values needed to calculate the total SE effort are in the model.

- The only data in the model are the reuse and difficulty values (strings.)
- There is no way to know which model elements need to be counted ahead of time.

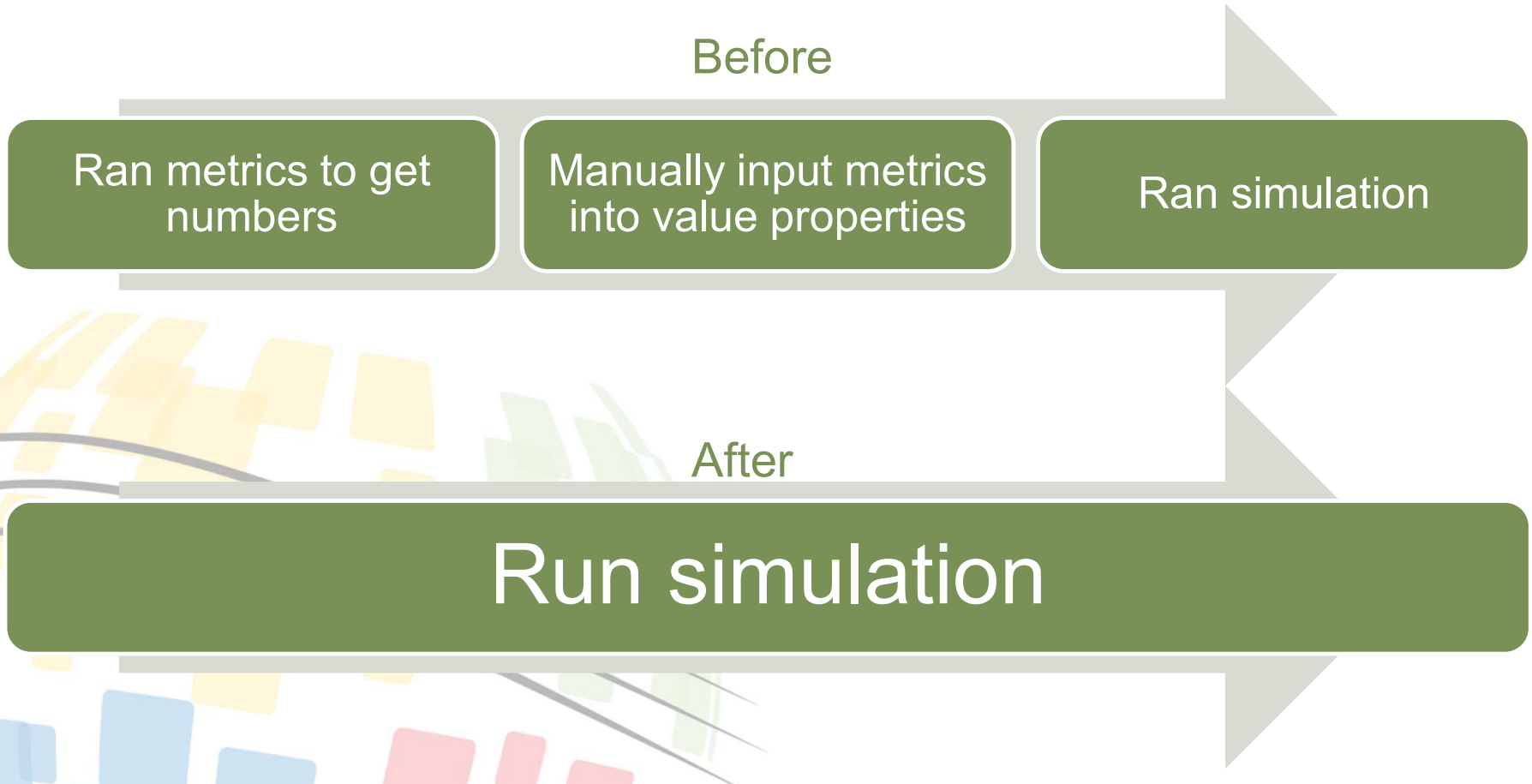
The entire set of scripts and parametric diagrams is implemented in a reusable profile.

- The profile also contains the reuse level and difficulty weights.
- When applied to any model, the scripts parse the model, count the elements and compute size driver totals and system size in memory at runtime.



The computed results can be written back to value properties if desired.

Case Study - COSYSMO Before and After



Conclusion

- Applications
  - Automate Manual Inputs
  - Mixing simulation data with cross-cutting data
- Case Studies
  - Automated Trade Study
  - COSYSMO
- What is your Use Case?

34th Annual INCOSE international symposium

hybrid event

Dublin, Ireland
July 2 - 6, 2024

www.incose.org/symp2024
#INCOSEIS