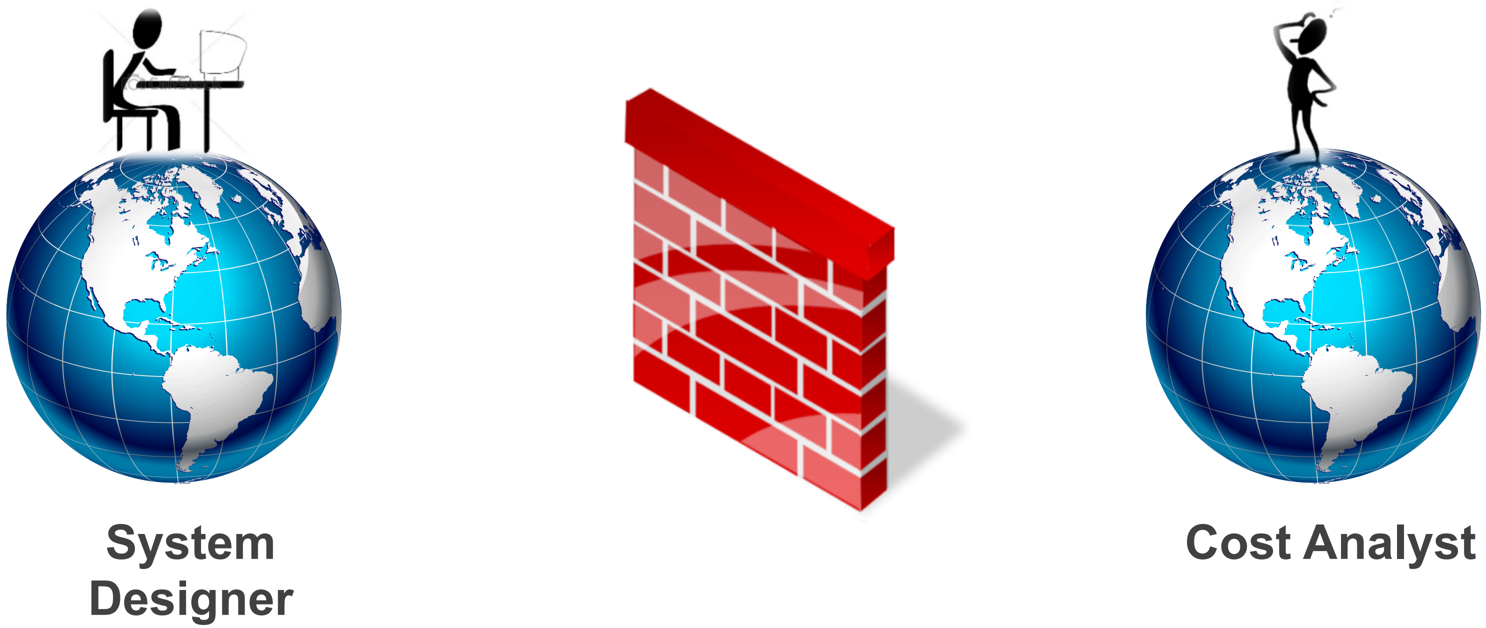
27th annual INCOSE international symposium
Adelaide, Australia
July 15 - 20, 2017

ENGINEERS AUSTRALIA SESA

Enabling Repeatable SE Cost Estimation with COSYSMO and MBSE

Problem Statement



- Two stove-piped worlds today: system architecting vs. cost estimation
  - Lack of traceability of data
  - Lack of ability to conduct cost trades (CAIV or DTC)
  - Lack of early visibility to the economic effect of system architecture

Introduction

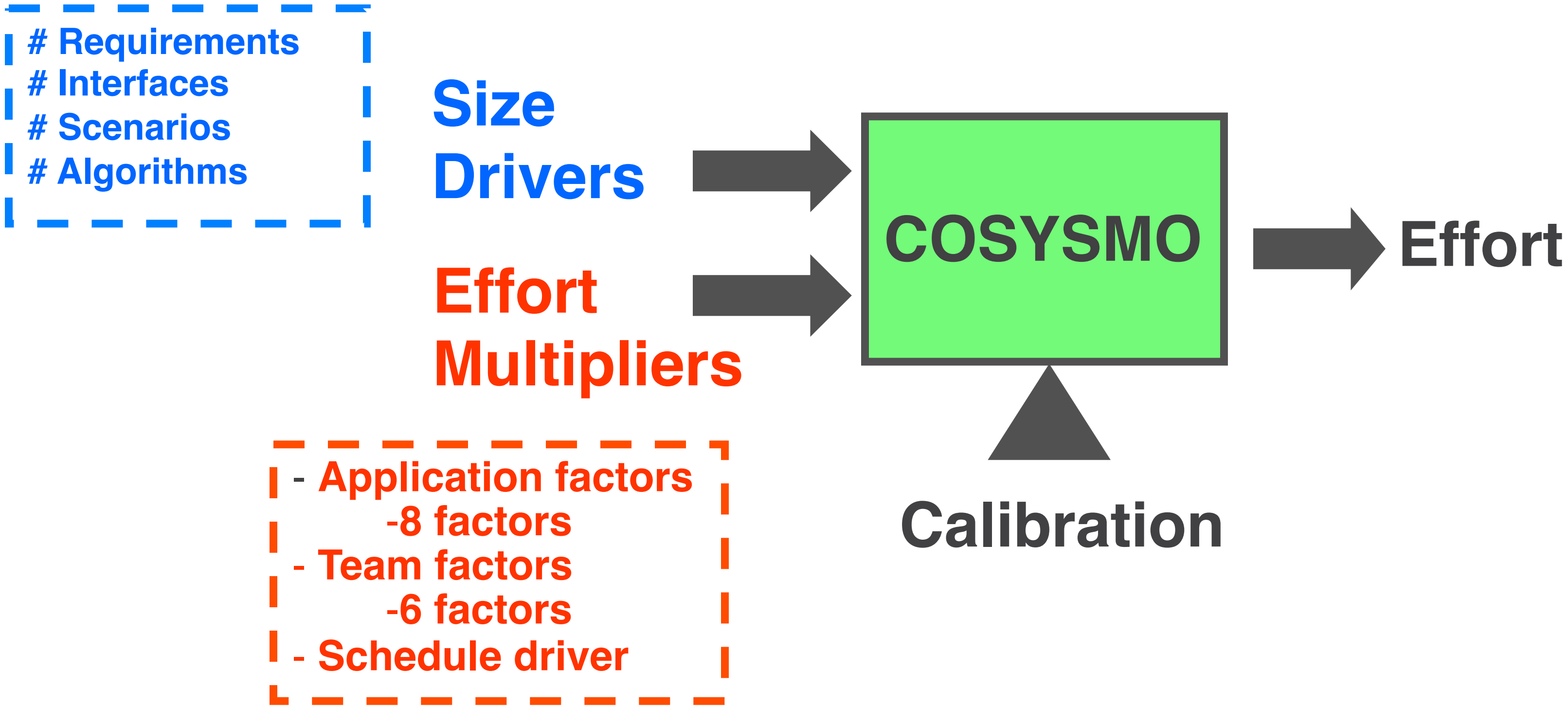
- This paper proposes a concept of integrating parametric cost estimating with Model Based Systems Engineering (MBSE)
  - An approach of integrating COSYSMO cost estimating relationship with SysML modeling environment
  - Demonstrated use case for a practical implementation
  - Feasibility of parametric cost estimating as a natural extension of Model-Based Systems Engineering

Background

- 2005
  - The Constructive Systems Engineering Cost Model (COSYSMO) was originally introduced by Ricardo Valerdi (PhD Dissertation, University of Southern California)
- 2008-2015
  - COSYSMO was extended by the Wang, Fortune, Valerdi and others with the General Reuse Framework to address reuse in systems engineering activity
- 2016
  - The Generalized Reuse Framework –Strategies and the Decision Process for Planned Reuse was presented by Dr. Gan Wang at 26th Annual INCOSE International Symposium (IS 2016) Edinburgh, Scotland

COSYSMO – Parametric Cost Model for Systems

4 Size Drivers and 14 Cost Drivers



Basic Cost Estimating Relationship (CER)

$$PH_{NS} = A \cdot SS^{E} \cdot CEM$$

- ***PH_{NS}*** = systems engineering effort in person-hours under nominal schedule
- ***A*** = productivity constant, typically derived from historical project data
- ***SS*** = system size, determined by the four size drivers
- ***E*** = nonlinearity for the productivity curve, representing a diseconomy of scale
- ***CEM*** = composite effort multiplier, determined by the fourteen cost drivers

Basic Cost Estimating Relationship (CER)

$$PH_{NS} = A \cdot SS^{E} \cdot CEM$$

- ***PH_{NS}*** = systems engineering effort in person-hours under nominal schedule
- ***A*** = productivity constant, typically derived from historical project data
- ***SS*** = system size, determined by the four size drivers
- ***E*** = nonlinearity for the productivity curve, representing a diseconomy of scale
- ***CEM*** = composite effort multiplier, determined by the fourteen cost drivers

This presentation will focus on generation of the System Size (SS) component of the COSYSMO equation.

Four Sizing Drivers

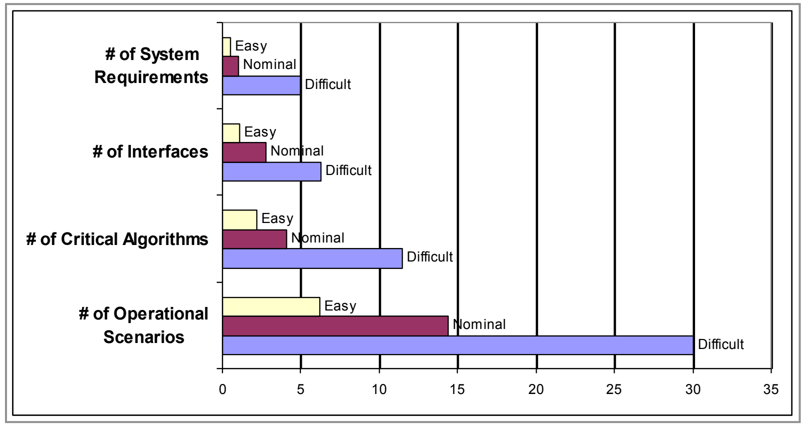
REQ	• Number of System Requirements
IF	• Number of System Interfaces
ALG	• Number of Critical Algorithms
SCN	• Number of Operational Scenarios

Each weighted by:

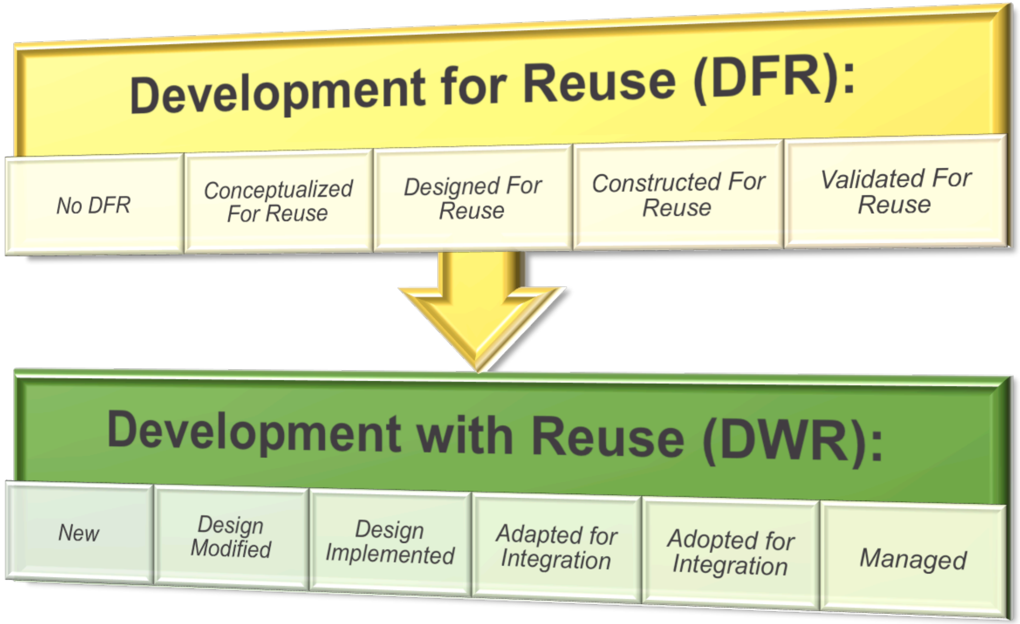
1) **Levels of complexity**
2) **Degrees of reuse**

Driver Counting/Classification Rules

- **Levels of Complexity**
  - "Easy"
  - "Nominal"
  - "Difficult"



- **Degrees of Reuse**
  - "Generalized Reuse Framework"



COSYSMO 3.0 with the Generalized Reuse Framework

Total Project Effort = DWR Effort + DFR Effort

$$PM_{DWR+DFR} = A_1 \cdot \left[\sum_k \left(\sum_r w_r \left(w_{e,k}\Phi_{e,k} + w_{n,k}\Phi_{n,k} + w_{d,k}\Phi_{d,k} \right) \right) \right]^{E_1} \cdot CEM_1$$

$$+ A_2 \cdot \left[\sum_k \left(\sum_q w_q \left(w_{e,k}\Psi_{e,k} + w_{n,k}\Psi_{n,k} + w_{d,k}\Psi_{d,k} \right) \right) \right]^{E_2} \cdot CEM_2$$

Where:

- **PM_{DWR}** = effort in Person Hours/Months (Nominal Schedule)
- **A_1** = DWR constant derived from historical project data
- **k** = {REQ, IF, ALG, SCN}
- **r** = {***New, D. Modified, D. Implemented, Adapted for Int., Adopted for Int., Managed***}
- w_r = weight for defined levels of size driver reuse
- w_x = weight for “easy”, “nominal”, or “difficult” size driver
- Φ_x = quantity of “k” size driver
- **E_1** = represents diseconomy of scale in DWR
- **CEM_1** = composite effort multiplier for DWR

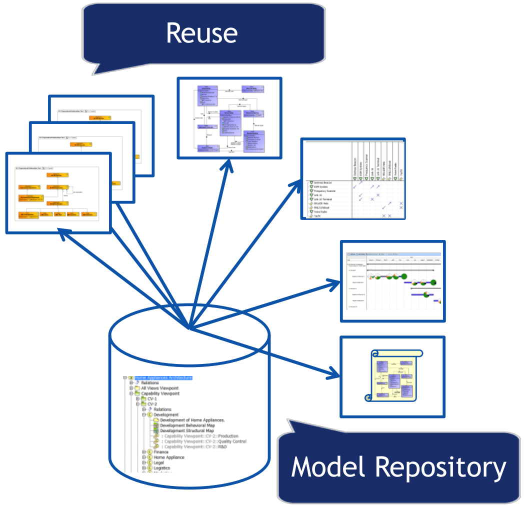
Where:

- **PM_{DFR}** = effort in Person Hours/Months (Nominal Schedule)
- **A_2** = DFR constant derived from historical project data
- **k** = {REQ, IF, ALG, SCN}
- **q** = {***No DFR, Conceptualized, Designed, Constructed, Validated***}
- w_q = weight for defined levels of size driver reuse
- w_x = weight for “easy”, “nominal”, or “difficult” size driver
- Φ_x = quantity of “k” size driver
- **E_2** = represents diseconomy of scale in DFR
- **CEM_2** = composite effort multiplier for DFR

Wang, Gan. 2016. “The Generalized Reuse Framework - Strategies and the Decision Process for Planned Reuse.” *INCOSE International Symposium Volume 26, Issue 1*, July: 175-189.

What Do We Mean When We Say MBSE?

Model Based Systems Engineering

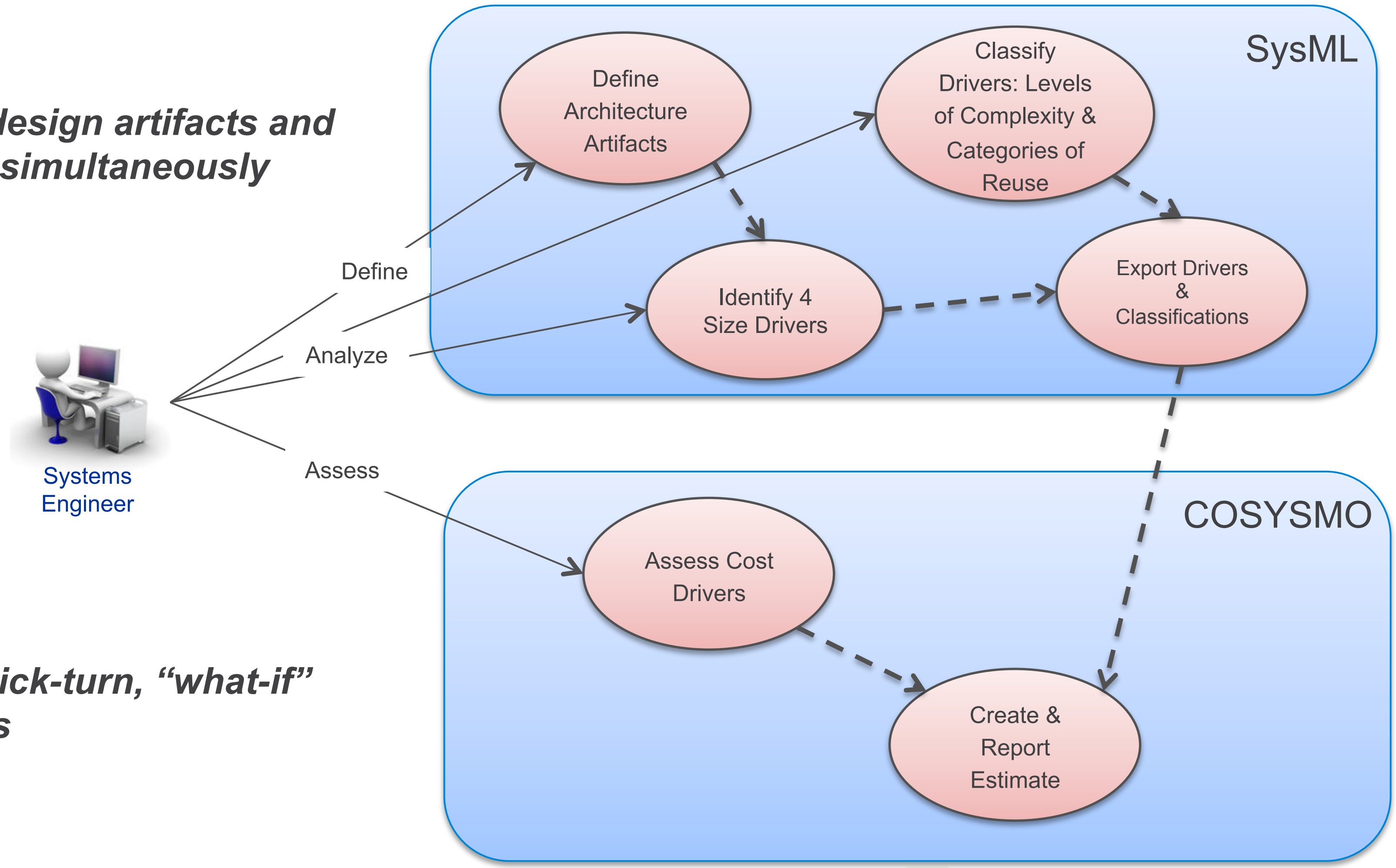


- Language
- Methodology
- Tool

- **Model/data repository provides a single source of truth!**
  - **Cost model is just another model**
  - **Size estimate is just another piece of data within repository**

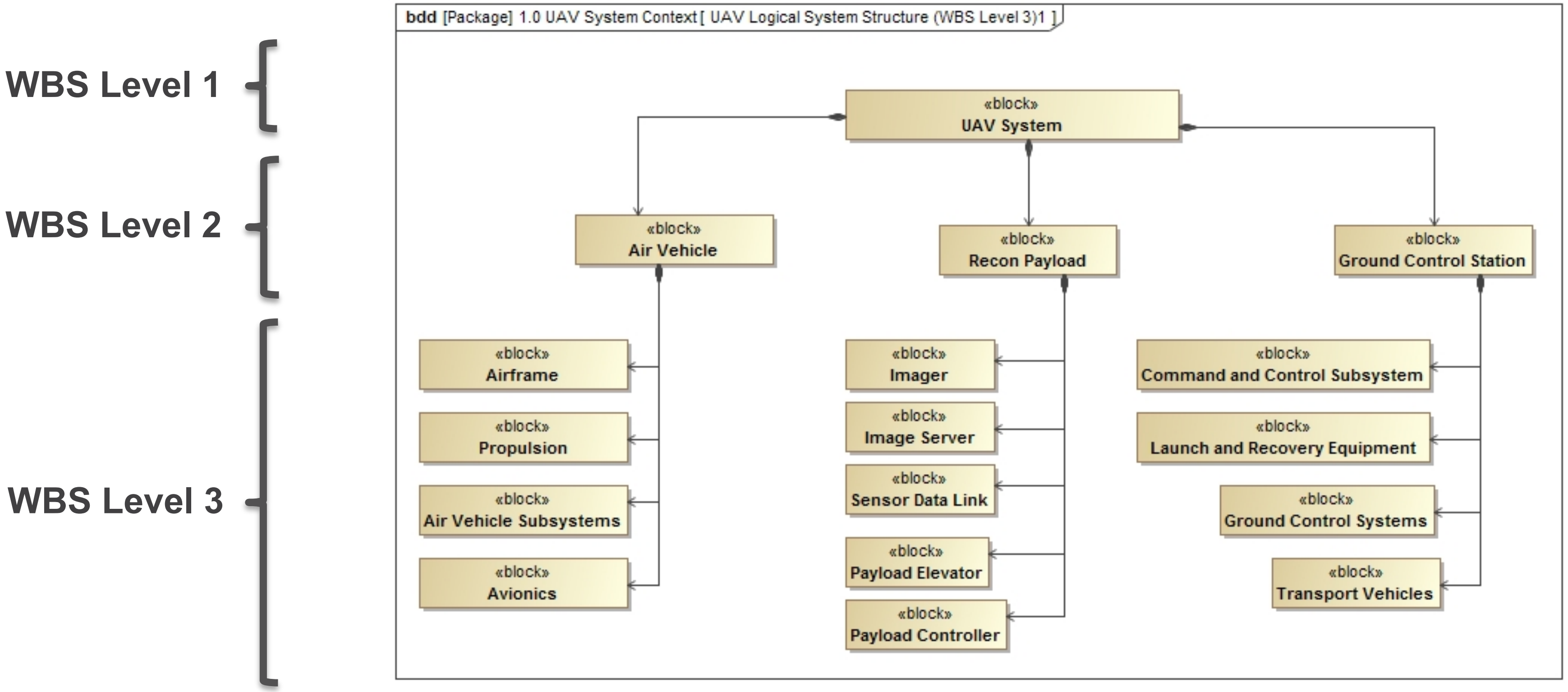
MBSE-Based Cost Modeling Process

- ***SE develops design artifacts and cost estimate simultaneously***



- ***Performs quick-turn, "what-if" DTC analysis***

Define the “System of Interest”

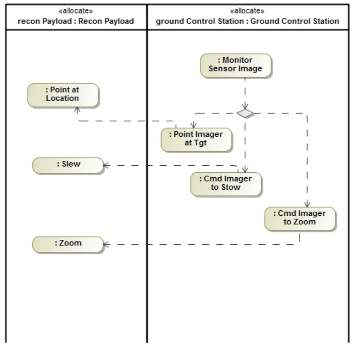


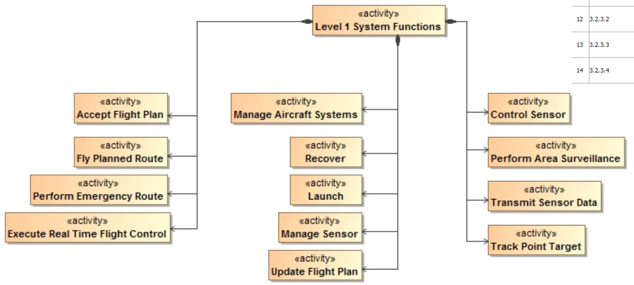
In SysML, system structure is defined through Block Definition Diagrams:

- **Systems, subsystems and components are represented by blocks**
- **System decomposition is represented by a “composition” relationship.**

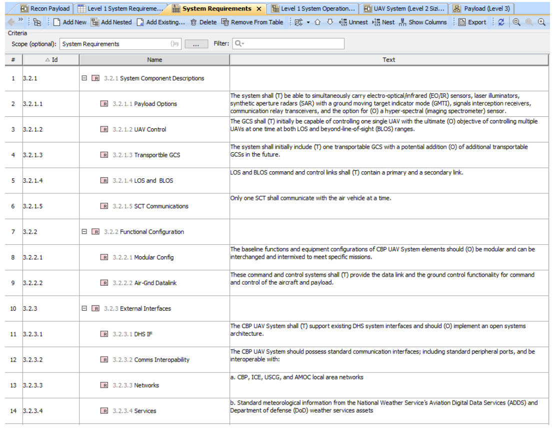
System Definition SysML vs Size Drivers

Algorithms





Requirements


Recon Payload | Level 1 System Requireme... | System Requirements | Level 1 System Operation... | UAV System (Level 2 Sizi... | Payload (Level 3)

Add New | Add Nested | Add Existing... | Delete | Remove From Table | Unnest | Nest | Show Columns | Export

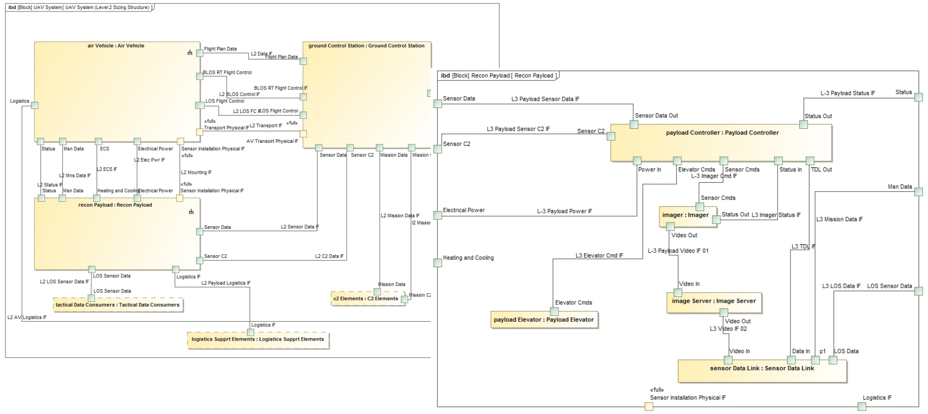
Criteria

Scope (optional): System Requirements Filter:

#	Id	Name	Text
1	3.2.1	3.2.1 System Component Descriptions	
2	3.2.1.1	3.2.1.1 Payload Options	The system shall (T) be able to simultaneously carry electro-optical/infrared (EO/IR) sensors, laser illuminators, synthetic aperture radars (SAR) with a ground moving target indicator mode (GMTI), signals interception receivers, communication relay transceivers, and the option for (O) a hyper-spectral (imaging spectrometer) sensor.
3	3.2.1.2	3.2.1.2 UAV Control	The GCS shall (T) initially be capable of controlling one single UAV with the ultimate (O) objective of controlling multiple UAVs at one time at both LOS and beyond-line-of-sight (BLOS) ranges.
4	3.2.1.3	3.2.1.3 Transportble GCS	The system shall initially include (T) one transportable GCS with a potential addition (O) of additional transportable GCSs in the future.
5	3.2.1.4	3.2.1.4 LOS and BLOS	LOS and BLOS command and control links shall (T) contain a primary and a secondary link.
6	3.2.1.5	3.2.1.5 SCT Communications	Only one SCT shall communicate with the air vehicle at a time.
7	3.2.2	3.2.2 Functional Configuration	
8	3.2.2.1	3.2.2.1 Modular Config	The baseline functions and equipment configurations of CBP UAV System elements should (O) be modular and can be interchanged and intermixed to meet specific missions.
9	3.2.2.2	3.2.2.2 Air-Gnd Datalink	These command and control systems shall (T) provide the data link and the ground control functionality for command and control of the aircraft and payload.
10	3.2.3	3.2.3 External Interfaces	
11	3.2.3.1	3.2.3.1 DHS IF	The CBP UAV System shall (T) support existing DHS system interfaces and should (O) implement an open systems architecture.
12	3.2.3.2	3.2.3.2 Comms Interopability	The CBP UAV System should possess standard communication interfaces; including standard peripheral ports, and be interoperable with:
13	3.2.3.3	3.2.3.3 Networks	a. CBP, ICE, USCG, and AMOC local area networks
14	3.2.3.4	3.2.3.4 Services	b. Standard meteorological information from the National Weather Service's Aviation Digital Data Services (ADDS) and Department of defense (DoD) weather services assets

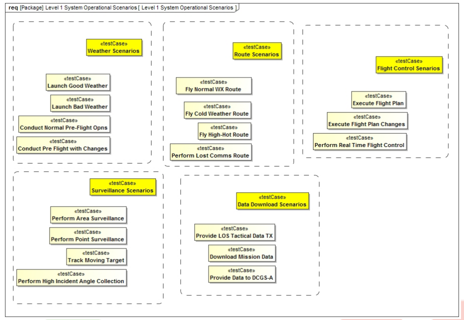
Requirements

Interfaces



Ports or Connectors

Scenarios

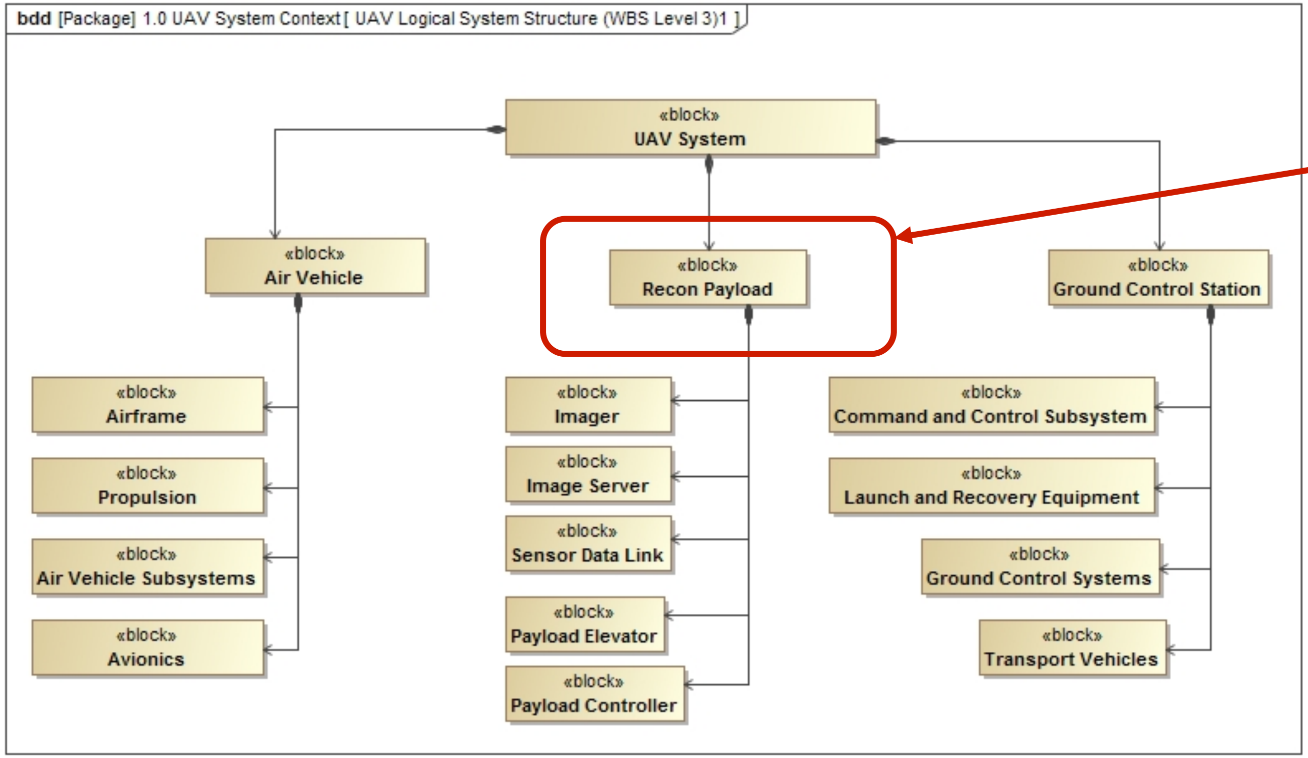


Test Cases or Use Cases

Behaviors:

- Activities
- Interactions
- State Machines

Define the “System of Interest”

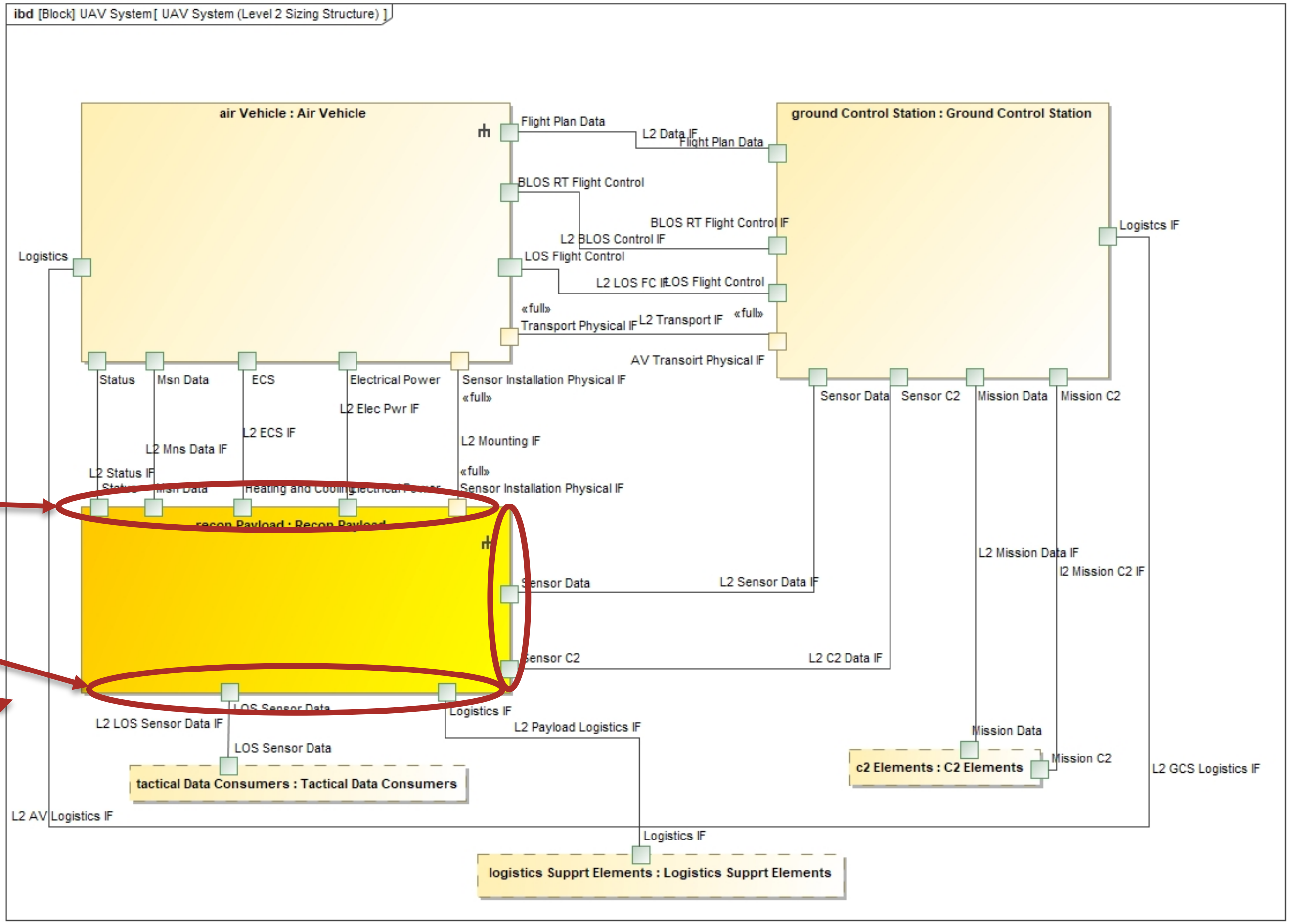


The system selected as the SOI for this example is the Recon Payload:

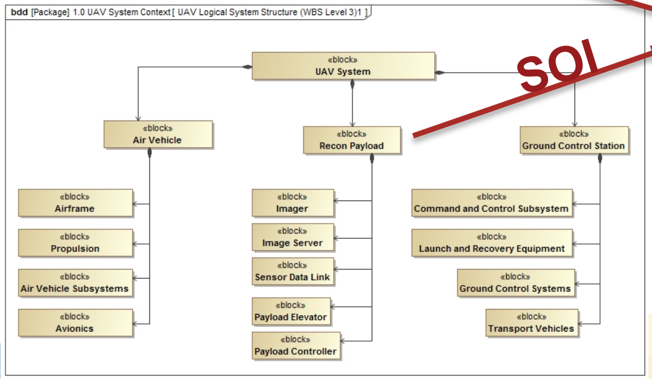
- WBS Level 2

Maintain the Levels of Abstraction, Consistency and Traceability

Internal Block Diagram showing interfaces between Level-2 Subsystems

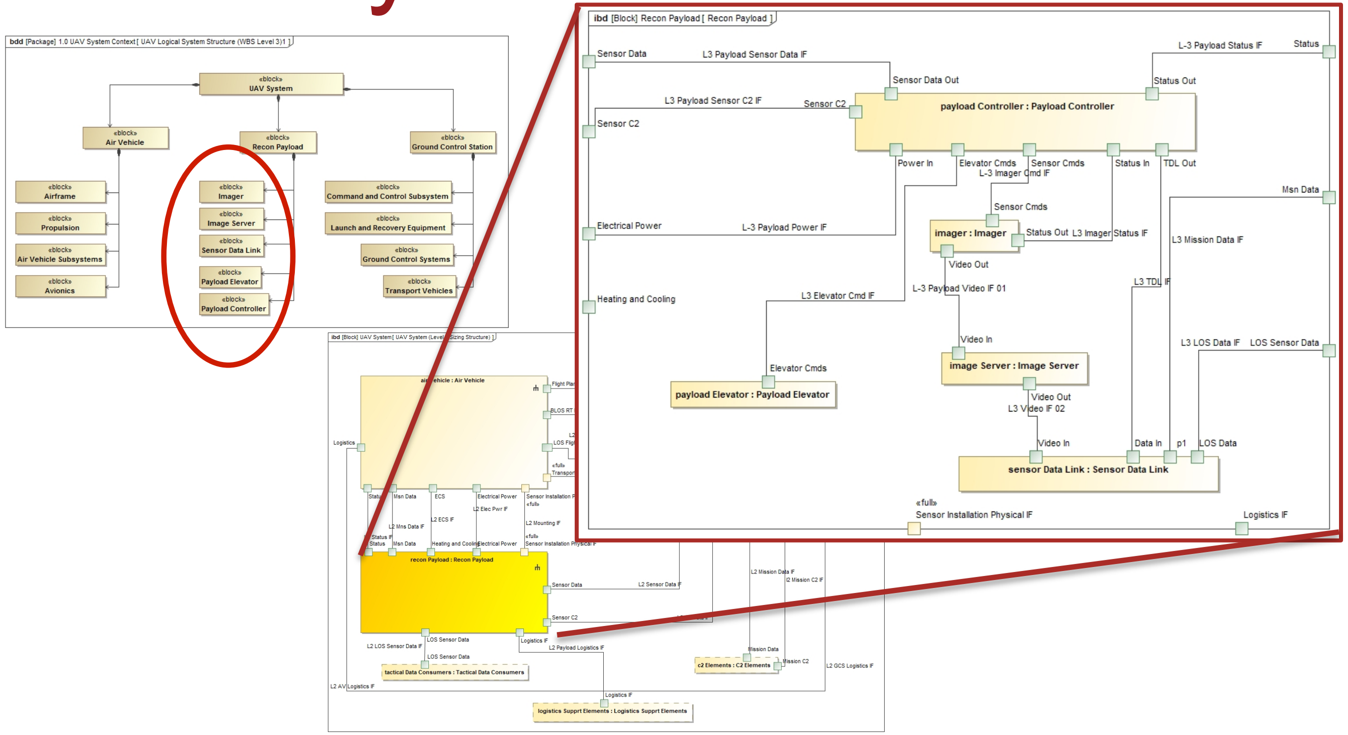


- Size Drivers are typically counted at the black box level for the SOI
- Example: Recon Payload
  - 300 REQ (Recon Payload System Spec)
  - 9 IF
  - 11 ALG
  - 3 SCN



Maintain the Levels of Abstraction, Consistency and Traceability

- At each level of abstraction, each component is defined with its REQ, ALG, IF, SCN.
- Example - Level 3 elements within the Recon Payload:
  - Payload Controller
    - 8 IF, 250 REQ, 8 ALG
    - 1 SCN
  - Payload Elevator
    - 2 IF, 100 REQ, 2 ALG, 2 SCN
  - Imager
    - 3 IF, 300 REQ, 6 ALG, 3 SCN
  - Image Server
    - …..
  - Sensor Data Link
    - …..



The level of abstraction chosen for the sizing estimate directly affects the quantity of sizing elements.

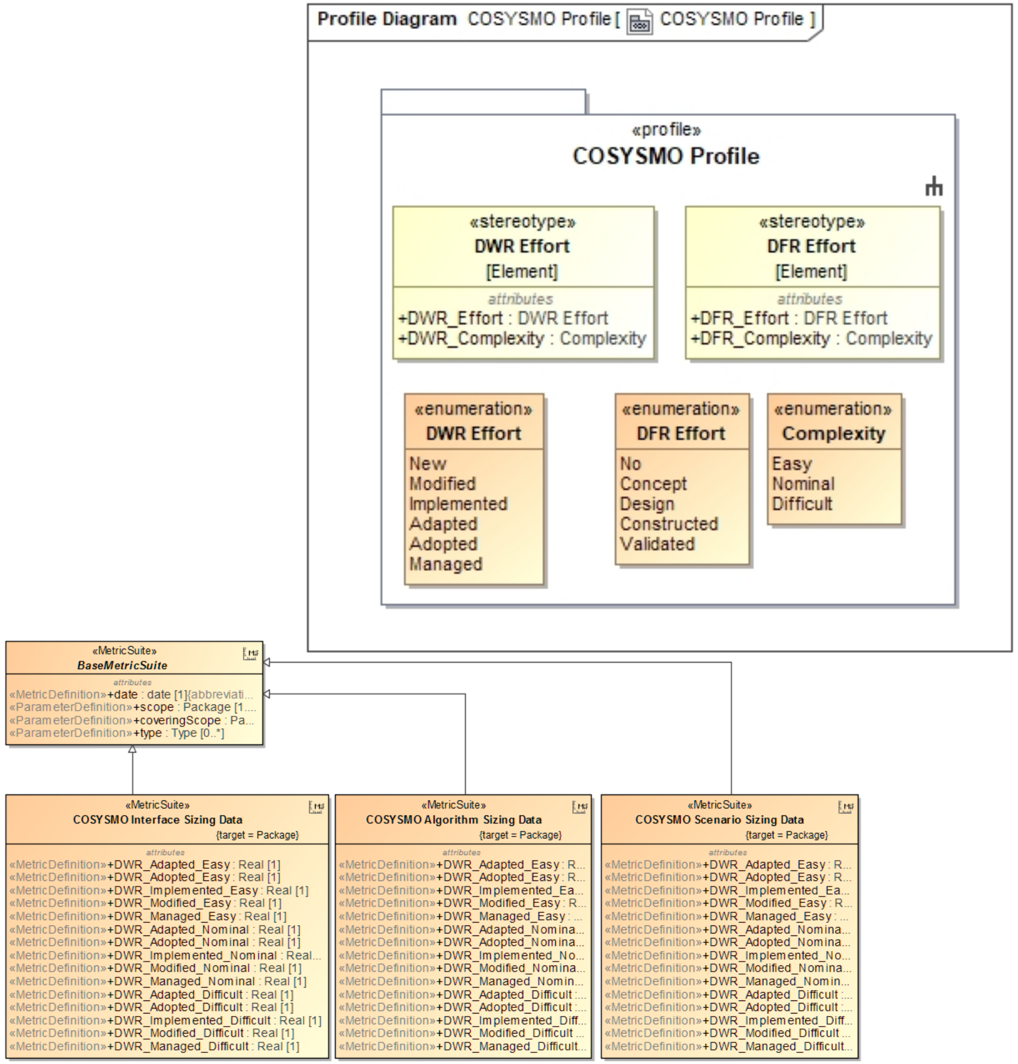
The key is to maintain consistency with the approach used across projects and with that used for calibration.

Applying Reuse Categories and Levels of Complexity

- Within a SysML model, reuse category and complexity are simply properties of model elements.
- There are multiple methods to assign properties to model elements in SysML.
  - The approach shown in this presentation takes advantage of specific tool features in the selected tool (MagicDraw™) that minimized the effort to assign and count sizing elements.
  - Other tools may have other features for defining model element properties and calculating model metrics.

Create COSYSMO Profile and Metrics Rules

- COSYSMO sizing properties are created as new stereotype elements:
  - Stereotypes are a core SysML feature
  - Defined in a Profile Package
- Metrics rules and measurements are a tool specific feature:
  - Multiple methods exist to determine the numbers of each sizing element
- The COSYSMO profile and metrics sets are created once as a separate project and reused:
  - The are applied (reused) on each new system project when generating sizing estimates



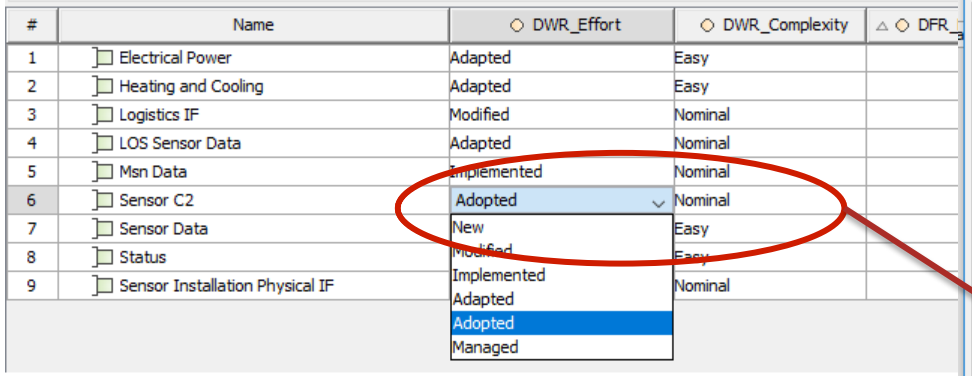
5 DFR Reuse Categories
6 DWR Reuse Categories
3 Levels of Complexity for each (Easy, Nominal, Difficult)
4 Sizing Elements Types (REQ, ALG, SCN and IF)

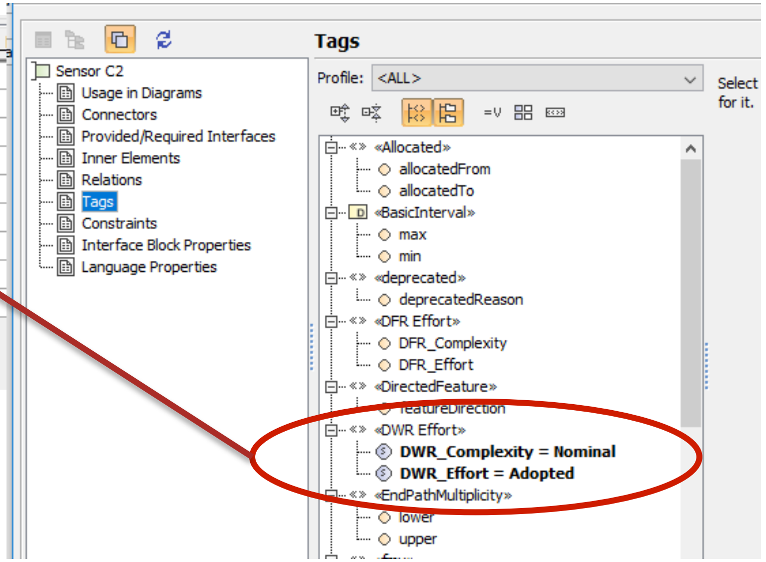
A potential of 132 individual pieces of sizing data:
(5+6) x 3 x 4 = 132

Apply Reuse Categories

- As part of the design analysis, the Systems Engineer identifies system elements (REQ, IF, ALG and SCN) that are to be included in the sizing estimate.
- Once elements are identified, assignment of re-use category and complexity is a trivial effort:
  - Create generic table and select element type and package scope.
  - Select the new stereotypes from the “show columns” pull-down.
  - Select the cell in the table and apply the reuse category and complexity.
  - Once selected, the tool applies the properties as tag values to the model element.

Interface Count Example





Properties selected in the table are actual properties of the model element.

Apply Reuse Categories

- The process of applying re-use categories and complexity level is repeated for each of the four sizing categories (REQ, IF, ALG and SCN)
- If requirements are managed in an external requirements management tool, sizing metrics for requirements can be easily calculated by applying properties in that tool, and using spreadsheets or other applications to sum each category.

Add New | Add Existing... | Export | 100%

Criteria

Element Type: Activity | Scope (optional): L2 System Functions | Filter:

#	Name	Allocated To	DWR_Effort	DWR_Complexity
1	enter area stare mode	Recon Payload	Adopted	Nominal
2	Monitor Sensor Status	Recon Payload	Adapted	Nominal
3	perform gyro alignment	Recon Payload	Adopted	Nominal
4	Point at Location	Recon Payload	Adapted	Nominal
5	record metadata	Recon Payload	Implemented	Nominal
6	record streaming image	Recon Payload	Adopted	Easy
7	Report Sensor Status	Recon Payload	Adapted	Nominal
8	Slew	Recon Payload	Adapted	Nominal
9	Store Search Plan	Recon Payload	Implemented	Nominal
10	Stow for landing	Recon Payload	Adopted	Nominal
11	Zoom	Recon Payload	Managed	Nominal

Criteria

Element Type: Test Case | Scope (optional): System Operational Scenarios | Filter:

#	Name	DWR_Effort	DWR_Complexity
1	Payload Stow and Deploy Scenarios	Adapted	Nominal
2	Sensor Visibility Scenarios	Adapted	Nominal
3	Target Tracking Scenarios		

Add New | Add Existing... | Export

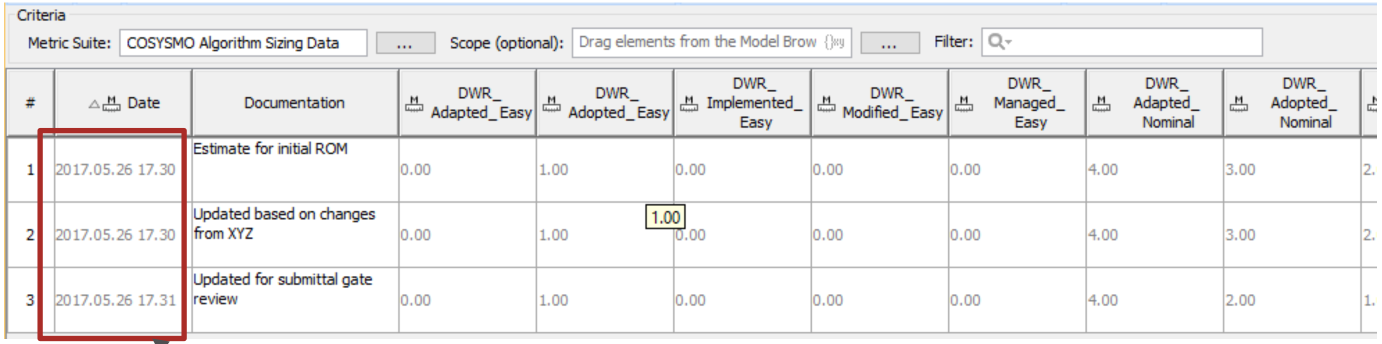
Criteria

Element Type: Full Port, Proxy Port | Scope (optional): Recon Payload

#	Name	DWR_Effort	DWR_Complexity
1	Electrical Power	Adapted	Easy
2	Heating and Cooling	Adapted	Easy
3	Heating and Cooling	Modified	Nominal
4	LOS Sensor Data	Adapted	Nominal
5	Msn Data	Implemented	Nominal
6	Sensor C2	Adopted	Nominal
7	Sensor Data	Adopted	Easy
8	Status	Adopted	Easy
9	Sensor Installation Physical IF	Modified	Nominal

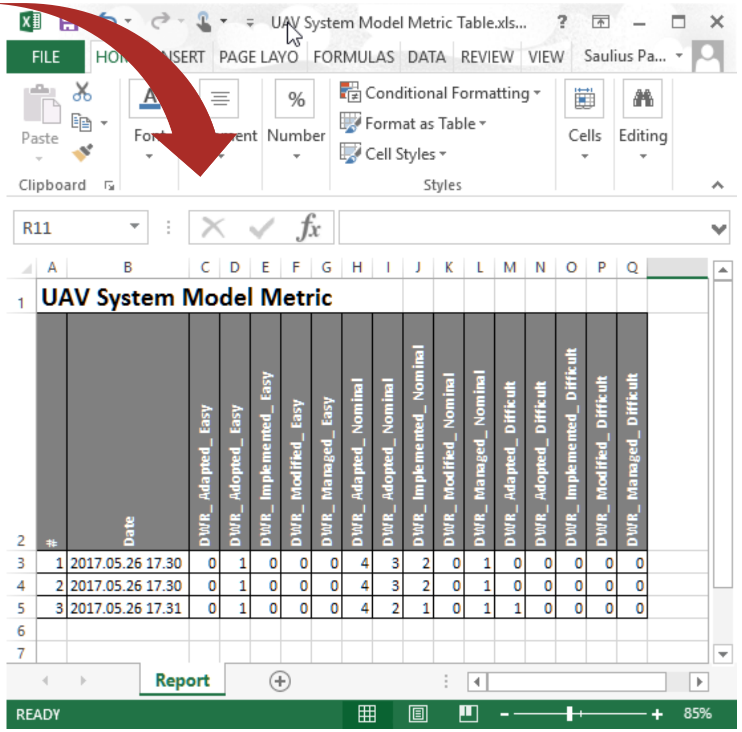
Run the Metrics Tool and Calculate Sizing Element Counts

- Run the metrics tool to generate a metrics table with counts for each reuse category/complexity combination.
  - Separate tables are create for each sizing element type (REQ, IF, ALG and SCN)
  - Metrics tables can be exported to Excel for input to the cost model.
- Depending on the tool, other methods may be available to determine sizing counts and export data.


Criteria

Metric Suite: COSYSMO Algorithm Sizing Data ... Scope (optional): Drag elements from the Model Brow ... Filter:

#	Date	Documentation	DWR_ Adapted_ Easy	DWR_ Adopted_ Easy	DWR_ Implemented_ Easy	DWR_ Modified_ Easy	DWR_ Managed_ Easy	DWR_ Adapted_ Nominal	DWR_ Adopted_ Nominal	
1	2017.05.26 17.30	Estimate for initial ROM	0.00	1.00	0.00	0.00	0.00	4.00	3.00	2.
2	2017.05.26 17.30	Updated based on changes from XYZ	0.00	1.00	0.00	0.00	0.00	4.00	3.00	2.
3	2017.05.26 17.31	Updated for submittal gate review	0.00	1.00	0.00	0.00	0.00	4.00	2.00	1.



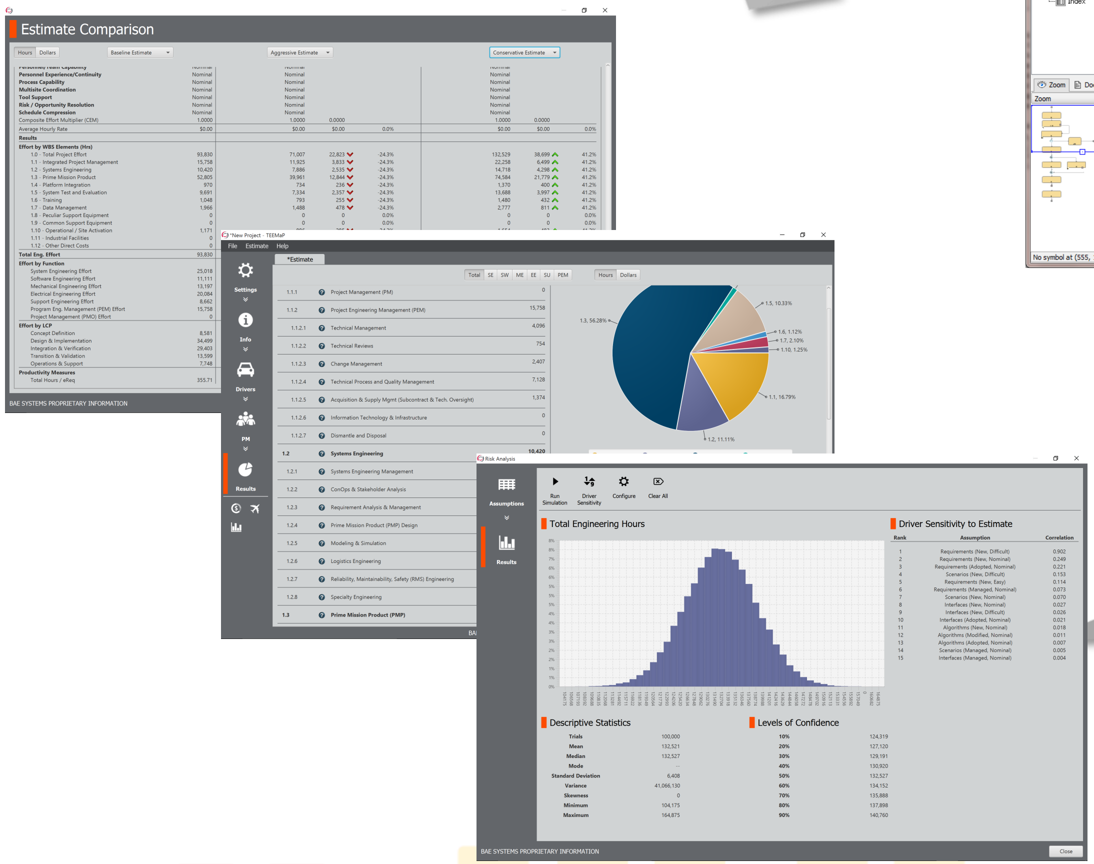
UAV System Model Metric

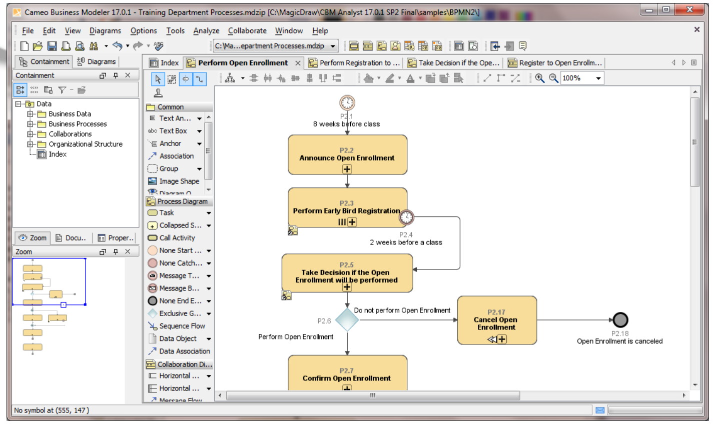
#	Date	DWR_Adapted_Easy	DWR_Adopted_Easy	DWR_Implemented_Easy	DWR_Modified_Easy	DWR_Managed_Easy	DWR_Adapted_Nominal	DWR_Adopted_Nominal	DWR_Implemented_Nominal	DWR_Modified_Nominal	DWR_Managed_Nominal	DWR_Adapted_Difficult	DWR_Adopted_Difficult	DWR_Implemented_Difficult	DWR_Modified_Difficult	DWR_Managed_Difficult
1	2017.05.26 17.30	0	1	0	0	0	4	3	2	0	1	0	0	0	0	0
2	2017.05.26 17.30	0	1	0	0	0	4	3	2	0	1	0	0	0	0	0
3	2017.05.26 17.31	0	1	0	0	0	4	2	1	0	1	1	0	0	0	0

The metrics tables shows the history of metrics calculations.

A documentation column can be added to record rational and other data for each metric calculation.

Run Cost Model and Analyze Estimate Result





Rapid "what-if" analysis

Developing Calibration Data

- Today, engineering organizations most likely will not have a database of cost data calibrated against COSYSMO sizing estimates.
- MBSE provides a practical approach to development of calibrated cost data:
  - Legacy systems with existing real cost data can be quickly modeled in SysML (to the level of detail required) to generate required sizing data.
  - The sizing data can then be applied through the calibration process to develop applicable CER parameters:

$$PH_{NS} = A \cdot SS^{E} \cdot CEM$$

- ***PH_{NS}*** = systems engineering effort in person-hours under nominal schedule
- ***A*** = productivity constant, typically derived from historical project data
- ***SS*** = system size, determined by the four size drivers
- ***E*** = nonlinearity for the productivity curve, representing a diseconomy of scale
- ***CEM*** = composite effort multiplier, determined by the fourteen cost drivers

Conclusion

- Parametric cost estimating can be seamlessly integrated into Model-Based Systems Engineering workflow:
  - Sizing data becomes a property of each model element
  - The model repository provides a single source of truth
  - Libraries can be created to maintain and revise reuse category and complexity consistent with changes in project lifecycles
- Required MBSE tool and language features are state of the practice:
  - All methods shown are existing features of the SysML language or the toolset

Key Benefits

- Seamless integration of cost estimation with the system design and modeling process:
  - Providing consistency and traceability.
    - Sizing data becomes a property of the model element.
  - Enabling rapid-turnaround “what-if” architecture trade analysis
  - Promotes Design-To-Cost.
  - Enables design reuse.
  - Economic impact early in system lifecycle and an integral part of architecture
  - Culture change for systems engineers:
    - Shift of mindset and right behavior in design
    - Systems engineering for economic goals
  - Application of Model Based Systems Engineering – LET THE TOOLS DO THE WORK

Future Work

- Develop Design Patterns and Guidelines for Sizing Estimation
  - Develop guidelines and standards for levels of abstraction, design patterns and identification of model elements that should be included or excluded from the sizing counts for calibration and cost estimation.
- Demonstrated Case Study
  - Present processes and lessons learned from application of parametric SE cost estimation on a real system
- Tool-Tool Data Exchange
  - Develop an export/report format that can be used as direct input to the calibrated cost estimation tools.
- Integrated Tool-Tool Interfaces
  - Develop a tool plug-in that launches the COSYSMO cost model from within the MBSE modeling tool and passes the data directly without an export/import process.
    - Includes COSYSMO profile, metrics definitions, etc

Questions?

Dr. Gan Wang, BAE Systems

Dr. Saulius Pavalkis, No Magic

Mr. Barry Papke, No Magic