Integration of Parametric Cost Estimation with System Architecture Modeling

... It’s a dirty job but someone has to do it

Dr. Gan Wang, BAE Systems
Mr. Barry Papke, No Magic, Inc. (3DS)

33rd International Forum on COCOMO and Systems/Software Cost Modeling
November 5-6, 2018
Washington, DC
Hey! You’ve got your cost model on my architecture model!!

No! You’ve got your architecture model on my cost model!!
Hey! Your cost estimate is way out of whack!!

No! Your architecture has to be realistic and affordable!!
“Chasm” Between the Two Worlds

Architecture Development vs. Cost Estimation

Separated by the Same Data!
When Do Systems Engineers Care About Cost Estimating?

Cost baseline is part of the technical baseline throughout the project lifecycle!
Why Parametric Cost Estimation?

• An established, often preferred estimating methodology
  • Traceability to design and historical data
  • Objective measures of validity
• Ideal for early lifecycle before detailed design date exists
  • Simple, quick, low cost
  • Easily adjusted for changes
  • Statistical measures of risk
• Effective for what-ifs, trade studies
  • Exploration of architecture options (trades)
  • “What-if” sensitivity analyses
• Validation for more detailed, labor intensive methods (e.g. bottoms-up.)
The Goal: Integrated System Design and Cost Estimation

"Single Source of Truth" – Extending the digital thread into the cost domain!
What do we mean when we say MBSE?

Model Based Systems Engineering

- Model/data repository provides a single source of truth!
  - Cost model is another model
  - Estimate is another piece of data within repository
For This Work, We Used...

Parametric Cost Model

SysML Modeling Environment

The Principle Applies to All Modeling Tools
### COSYSMO CONOPS

#### 4 Size Drivers and 14 Cost Drivers...

- **Size Drivers**
  - Application factors
    - 8 factors
  - Team factors
    - 6 factors

- **Effort Multipliers**

- **Calibration**
  (Historical Data)

$$PH_{Total} = A_{DWR} \cdot SS_{DWR} \cdot CEM_{DWR} + A_{DFR} \cdot SS_{DFR} \cdot CEM_{DFR}$$

$$SS_{DWR} = \sum_k \left( \sum_e w_e \Phi_{e,k} + w_{a,k} \Phi_{a,k} + w_{d,k} \Phi_{d,k} \right)$$

$$SS_{DFR} = \sum_k \left( \sum_q w_q \Psi_{e,k} + w_{a,k} \Psi_{a,k} + w_{d,k} \Psi_{d,k} \right)$$
COSYSMO Cost Modeling Process – In General

These steps can be naturally achieved in an MBSE modeling environment...
What We Have Established...

... COSYSMO Size Drivers Are Directly Embedded in System Models

- Number of System Requirements
- Number of System Interfaces
- Number of Critical Algorithms
- Number of Operational Scenarios

Each weighted by:
1) Levels of complexity
2) Categories of reuse

Example: Define Scope of the “System of Interest”

Example SOI
Maintain the Levels of Abstraction, Consistency and Traceability

• Example SOI Context Level
  • 300 REQ
  • 9 IF
  • 11 ALG
  • 3 SCN
Maintain the Levels of Abstraction, Consistency and Traceability

- Example SOI Level 1
  - 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.
The Question is How to Apply Counting/Classification Rules

Degrees of Reuse

- “Generalized Reuse Framework”

Level of Complexity

- “Easy”
- “Nominal”
- “Difficult”
# Development for Reuse (DFR) Process

<table>
<thead>
<tr>
<th>Category</th>
<th>Required Activities</th>
<th>Delivering (for reuse)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No DFR</td>
<td>• N/A</td>
<td>• Little / accidental</td>
</tr>
<tr>
<td>Conceptualized For Reuse</td>
<td>• Analysis</td>
<td>• Functional &amp; Logical architecture</td>
</tr>
<tr>
<td></td>
<td>• Architecture development</td>
<td></td>
</tr>
<tr>
<td>Designed For Reuse</td>
<td>• Analysis</td>
<td>• Physical design of system</td>
</tr>
<tr>
<td></td>
<td>• Architecture</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• System design</td>
<td></td>
</tr>
<tr>
<td>Constructed For Reuse</td>
<td>• Design</td>
<td>• Implemented system or component</td>
</tr>
<tr>
<td></td>
<td>• Build</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• Unit test</td>
<td></td>
</tr>
<tr>
<td>Validated For Reuse</td>
<td>• Design</td>
<td>• Validated and deployed system or</td>
</tr>
<tr>
<td></td>
<td>• Build</td>
<td>component</td>
</tr>
<tr>
<td></td>
<td>• System test</td>
<td></td>
</tr>
</tbody>
</table>

![Activity-based Model](image)
## Development with Reuse (DWR) Process

<table>
<thead>
<tr>
<th>Category</th>
<th>Required Activities</th>
<th>Leveraging (existing)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>New</strong></td>
<td>• Develop anew</td>
<td>• New concept</td>
</tr>
<tr>
<td></td>
<td>• Revamp of existing</td>
<td></td>
</tr>
<tr>
<td><strong>Design Modified</strong></td>
<td>• Design &amp; implement from logical architecture</td>
<td>• Logical/functional architecture</td>
</tr>
<tr>
<td><strong>Design Implemented</strong></td>
<td>• Implement from design</td>
<td>• Physical design of system</td>
</tr>
<tr>
<td></td>
<td>• Build-to-print</td>
<td>• Built system or component</td>
</tr>
<tr>
<td><strong>Adapted for Integration</strong></td>
<td>• Adapt from existing implementation</td>
<td>• Build system or component</td>
</tr>
<tr>
<td></td>
<td>• Tailor to integrate</td>
<td></td>
</tr>
<tr>
<td><strong>Adopted for Integration</strong></td>
<td>• Integrate per instructions</td>
<td>• Build system or component</td>
</tr>
<tr>
<td></td>
<td>• V&amp;V testing</td>
<td></td>
</tr>
<tr>
<td><strong>Managed</strong></td>
<td>• Manage</td>
<td>• Integrated &amp; verified system or component</td>
</tr>
</tbody>
</table>
COSYSMO 3.0 with the Generalized Reuse Framework

Total Project Effort = DWR Effort + DFR Effort

\[
PM_{DWR+DFR} = A_1 \cdot \left[ \sum_k \left( \sum_r w_r (w_{e,k} \Phi_{e,k} + w_{n,k} \Phi_{n,k} + w_{d,k} \Phi_{d,k}) \right) \right]^{E_1} \cdot CEM_1 \\
+ A_2 \cdot \left[ \sum_k \left( \sum_q w_q (w_{e,k} \Psi_{e,k} + w_{n,k} \Psi_{n,k} + w_{d,k} \Psi_{d,k}) \right) \right]^{E_2} \cdot CEM_2
\]

Where:
- \( PM_{DWR} \) = effort in Person Hours/Months (Nominal Schedule)
- \( A_1 \) = DWR constant derived from historical project data
- \( k = \{\text{REQ, IF, ALG, SCN}\} \)
- \( r = \{\text{New, D. Modified, D. Implemented, Adapted for Int., Adopted for Int., Managed}\} \)
- \( w_r \) = weight for defined levels of size driver reuse
- \( w_x \) = weight for “easy”, “nominal”, or “difficult” size driver
- \( \Phi_x \) = 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
- \( k = \{\text{REQ, IF, ALG, SCN}\} \)
- \( q = \{\text{No DFR, Conceptualized, Designed, Constructed, Validated}\} \)
- \( w_q \) = weight for defined levels of size driver reuse
- \( w_x \) = weight for “easy”, “nominal”, or “difficult” size driver
- \( \Phi_x \) = quantity of “k” size driver
- \( E_2 \) = represents diseconomy of scale in DFR
- \( CEM_2 \) = composite effort multiplier for DFR
Create COSYSMO Profile and Metrics Rules (Non-Recurring)

- COSYSMO sizing elements 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 set 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:
\[(5+6) \times 3 \times 4 = 132\]
Apply Reuse Categories

- Application 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.
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 created 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:
  - Export of element tables and count externally.

The metrics tables show the history of metrics calculations.

A documentation column can be added to record rational and other data for each metric calculation.
A New CONOPS: An Integrated COSYSMO-SysML Modeling Use Case

- **Develops design artifacts and cost estimate simultaneously**

- **Performs quick-turn, “what-if” DTC analysis**
A New Process: Estimating as an Integral Part of System Modeling

Four (4) Integrated Modeling Activities:

**Activity 1 – Create COSYSMO Profile (one-time configuration)**
- SysML Profile Package
- DWR, DFR and Complexity Stereotypes and Counting Metrics
- Create once and reuse in any project

**Activity 2 – Develop System Architecture**
- Unchanged from current practice
- Application of modeling guidelines ensure consistent definition across projects

**Activity 3 – Identify Cost Model Inputs**
- Enabled by tool automation and SysML constructs
- Assigns Reuse Category and Complexity

**Activity 4 – Generate Estimate**
- Execution of the cost model is unchanged
- Cost model inputs are now part of the system architecture definition

These Activities Are Performed as Part of an Augmented System Modeling Process

Activities 2 & 3: we do this already, sort of…

Activities 1 & 4: This is what’s new!

*Connecting the digital thread into the cost estimation domain!*
Activity 1: Create COSYSMO Profile (one-time event)

- **SysML Profile** package enables creation of stereotypes for
  - Reuse categories: DWR and DFR
  - Level of complexity: easy, nominal, difficult
- Once created, it can be reused in any new model
Activity 2: Develop System Architecture

Development of the SysML architecture model is unchanged*

* Projects must adopt consistent modeling standards to ensure repeatable sizing estimates that are consistent with calibration data.
Activity 3: Identify Cost Model Inputs

Iteratively with Step 2...

- Identify Sizing Elements
  - System of Interest (Count at the SOI Black Box Level):
    - Requirements
    - Interfaces
    - Algorithms
    - Operational Scenarios

- Determine Reuse Category and Complexity
  - DFR
  - DWR
  - Level of Complexity

- Count Sizing Elements (Automatic)
  - MBSE Tool features automate counting and collection of sizing data
  - A potential of 132 individual pieces of sizing data:
    - 5 DFR Reuse Categories
    - 6 DWR Reuse Categories
    - 3 Levels of Complexity
    - 4 Sizing Elements Types (REQ, ALG, SCN and IF)

- Generate Cost Drivers
  - 14 Cost Drivers:
    - Qualitative System characteristics
    - Assessed based on Behavior and Parametric Model Elements

- Review for Consistency
  - The key to accurate and repeatable parametric cost estimation
  - Tool features enable detailed review and analysis of sizing data

Requires System Domain Experience
Rely on tool automation
Rely on SysML language properties
Review Data for Consistency

Advanced query features enable comprehensive analysis of sizing inputs.

Example:
- The requirement that drove a specific critical algorithm should have similar DWR/DFR and complexity values as the SysML Activity that satisfies it.
New CONOPS: 4-Step Integrated Modeling/Estimating Process

1. Set up *COSYSMO* Profile (one-time)
   - SysML Stereotypes:
     - System of Interest
     - DWR
     - DFR
     - Easy, nominal, difficult

2. Develop System Architecture
   - Requirements
   - Structure
   - Behavior
   - Parametric

3. Identify & Classify Cost Drivers
   - Requirements
   - Interfaces
   - Algorithms
   - Scenarios

4. Review, Analyze & Report
   - System Architecture Views:
     - Requirements
     - Structure
     - Behavior
     - Parametric
   - Cost Estimate Views:
     - Hours, $
     - Schedule
A New Paradigm – Enabling “Single Source of Truth”
Integrated Modeling-Estimating Environment Enables Rapid Design Iteration and Optimization

- Sizing Data is a property of the architecture and maintained with the system model
- Alternatives can be quickly evaluated to achieve optimized design that meets:
  - Functional and Performance Requirements
  - Cost Targets
- Cost impacts can now be integrated into the systems engineering decision process
MBSE Allows Systems Engineers to Focus on the Important Things

- **Tasks for MBSE Toolset**
  - Maintain Sizing Data as part of the System Architecture
  - Provide efficient User Interface to apply Sizing Parameters
  - Automate Counting
  - Provide Cross Cutting Views for Analysis

- **Tasks for the Systems Engineer**
  - Design the System
  - Determine Sizing Elements
  - Determine Reuse Category and Complexity
  - Analyze Results
Conclusion with Perceived Benefits

- Integration of cost estimation with system modeling further extends the “digital thread” to the domain of economics – bridging a historical divide
  - Complete traceability from design to cost
  - **Repeatable** estimating with direct analysis/trade features
- Formalized development, integration, curation, and use of models for life cycle
  - **Early** system understanding
  - Reduced **cycle time** from design to cost, enabling to **earlier decision** making and **faster** time to market
- Enduring and authoritative “**Single source of truth**”
  - Reliable, trustworthy, and authoritative
  - Affordability and economic impact
  - Ultimately, better systems
About the Authors

**Barry Papke** is the Director of Professional Services for No Magic Inc. He has thirty-two years of systems engineering and operations analysis experience in the aerospace and defense industry across the entire systems engineering lifecycle from concept development through integration, test and post-delivery support.

**Gan Wang, Ph.D.,** is a Global Engineering Fellow at BAE Systems and the Chief Engineer for its Integrated Defense Solutions business. He has been actively engaged in the areas of systems engineering, particularly MBSE and MBE, engineering development processes and product engineering management, design reuse methodologies, and cost estimating and analysis. He has over 25 years of experience in complex system development involving software-intensive systems.
Thank You

Dr. Gan Wang, BAE Systems
Mr. Barry Papke, No Magic, Inc. (3DS)