2026 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
DIGITAL ENGINEERING / SYSTEMS ENGINEERING (DE/SE) TECHNICAL SESSION
AUGUST 11-13, 2026 - NOVI, MICHIGAN

MODEL-BASED PRODUCT LINE TRADE STUDIES WITH POWER BI VISUALIZATION

Gary Kliczinski[1], Ryan Pykor[1]

[1]STC, an Arcfield company, 14295 Park Meadow Drive, Suite 100, Chantilly, Virginia, USA

ABSTRACT

Modern automotive platforms must serve multiple market segments amid rapid technological change and stringent regulation, making early concept selection a multi-criteria, product-line problem. This paper presents a repeatable workflow that integrates Model-Based Product Line Engineering (MBPLE), Multi-Criteria Decision Analysis (MCDA), and enterprise visualization. A Systems Modeling Language (SysML) 150% vehicle architecture in Cameo captures powertrain, chassis, and Advanced Driver Assistance Systems (ADAS) variability plus baseline and segment-specific requirements. Parametric models compute vehicle-level attributes (e.g., cost, mass, braking capacity, detection performance, Technology Readiness Level (TRL)) and enforce segment limits. A multi-attribute value theory (MAVT) framework model is implemented as constraint blocks to normalize attributes and aggregate stakeholder-elicited weights into an overall score per configuration. Cameo Trade Study sweeps the design space, exports a configuration–criteria dataset, and Power Business Intelligence (BI) dashboards enable interactive cost–value views, requirement compliance, and shortlist comparisons. Selected concepts are fed back into Cameo as feature configurations, improving traceability, scalability across product lines, and alignment between engineering models and management decisions.

Citation: G. Kliczinski, R.Pykor, “Model-Based Product Line Trade Studies with Power BI Visualization,” In *Proceedings of the Ground Vehicle Systems Engineering and Technology Symposium* (GVSETS), NDIA, Novi, MI, Aug. 11-13, 2026.

1. INTRODUCTION

Automotive Original Equipment Manufacturers (OEM) increasingly deliver families of vehicles rather than single-point designs, while facing rapid technology change and mounting regulatory and stakeholder pressures. These conditions amplify the importance of early concept selection: architectural decisions made in the front-end drive downstream cost, mass, performance, and feasibility across multiple market segments, and the rationale for those decisions must be defensible. As Peterson and Schindel emphasize, modern engineered systems operate amid rising complexity and uncertainty, making informal or intuitiondriven decisions difficult to justify at scale.[5]

MCDA provides a structured way to compare alternatives using explicit criteria, scales/value functions, and preference weighting, enabling transparent ranking and sensitivity exploration. In practice, however, many trade studies remain document-centric, relying on spreadsheets and slide decks that are hard to maintain and weakly traceable to the engineering assumptions that generate the numbers. Model-Based Systems Engineering (MBSE) mitigates this by using SysML models as a coherent, updateable source of truth for structure, requirements, and parametric relationships. Model-Based Product Line Engineering (MBPLE) further extends this benefit across configurable product lines using feature models and a 150% architecture. Yet model-centric results often still fail to communicate effectively with decision-makers who expect interactive, management-friendly views.

This paper addresses that gap with an integrated MBPLE–MCDA workflow coupled to Power BI visualization to preserve traceability while improving decision usability. The paper then reviews related work, details the method and case application, discusses limitations and adoption considerations, and concludes with future directions.

2. BACKGROUND

2.1. Decision Analysis and MCDA For Concept Selection

Decision analysis frames concept selection as comparing alternatives against criteria using stakeholder preferences that define what "better" means. MCDA formalizes this by representing each candidate architecture as a vector of criterion scores (e.g., cost, mass, safety, user experience) and using a value model to produce an overall preference ordering. Bouyssou et al. stress that MCDA quality depends on well-constructed criteria—operational, non-redundant, and discriminating—and an appropriate aggregation model [1][2]. Additive models compute a single value/utility score for ranking many options, while outranking approaches support screening by testing whether one alternative is "not worse" on most criteria while tolerating local disadvantages—often mirroring engineers' dominance checks.

In practice, MCDA is frequently implemented as a weighted-sum or multi-attribute value function: criteria are normalized to a common scale (e.g., 0–1) and combined using weights reflecting relative importance [1]. This structure aligns with familiar tools like Pugh matrices and morphological analysis, but makes assumptions about scales, tradeoffs, and priorities explicit rather than ad hoc [12][13][14].

MCDA's benefits are structured tradeoffs, transparency, and repeatability; its costs are effort in defining criteria, eliciting weights, and communicating the model. Embedding MCDA within model-based engineering and dashboards can strengthen defensibility and accessibility for non-specialists.

2.2. Product Line Engineering

Model-Based Product Line Engineering (MBPLE) merges product line engineering with MBSE by organizing a product family's core assets—requirements, architecture, behavior, and verification—around features rather than individual products [7]. Instead of maintaining separate models for each derivative, MBPLE uses a single shared model that captures what is common and what varies, with explicit rules for selecting variability into concrete products.

MBPLE centers on feature models and a 150% architecture. Feature models define customer-visible and technical options (e.g., powertrain, chassis, ADAS, infotainment) plus constraints (mandatory/optional, alternatives, cross-tree rules). The 150% SysML architecture is the superset containing all blocks, connections, behaviors, and requirements that could appear in any product. A configuration model maps a valid feature selection to a specific subset of the 150% model, creating a product instance and enabling consistent reuse and evaluation across the family.

For automotive applications, MBPLE supports variability across product, lifecycle, abstraction layers, and system hierarchy, treating the vehicle portfolio as an analyzable product line rather than isolated designs [7].

2.3. MCDA and Risk Applications

Ren's edited volume *Multi-Criteria Decision Analysis for Risk Assessment and Management* shows that MCDA has become a core approach for risk-focused decisions when probability-based metrics alone are inadequate [3]. Across domains, MCDA structures integrate severity, likelihood, detectability, and organizational impacts, and several chapters augment or replace traditional Failure Mode and Effects Analysis (FMEA) by addressing limitations of the Risk Priority Number (RPN) through richer weighting and preference models [3]. Examples include modified RPN formulations that use methods such as the Best–Worst Method to derive more defensible weights, and fuzzy Multi-Criteria Decision Making (MCDM) variants (e.g., Z-numbers, grey numbers) that represent linguistic judgments and incomplete information by modeling uncertainty in consequences and preferences [3][4]. Collectively, this literature highlights MCDA's flexibility for combining qualitative and quantitative evidence and improving transparency in prioritization [3][4].

Early-phase automotive concept selection is inherently risk-laden (technology, cost, and market risks). While probabilistic or fuzzy risk scoring could be integrated with MCDA as suggested in this body of work [3][4], this paper applies value-based MCDA on expected performance and treats uncertainty qualitatively, reserving explicit risk computation as future work building on these approaches.

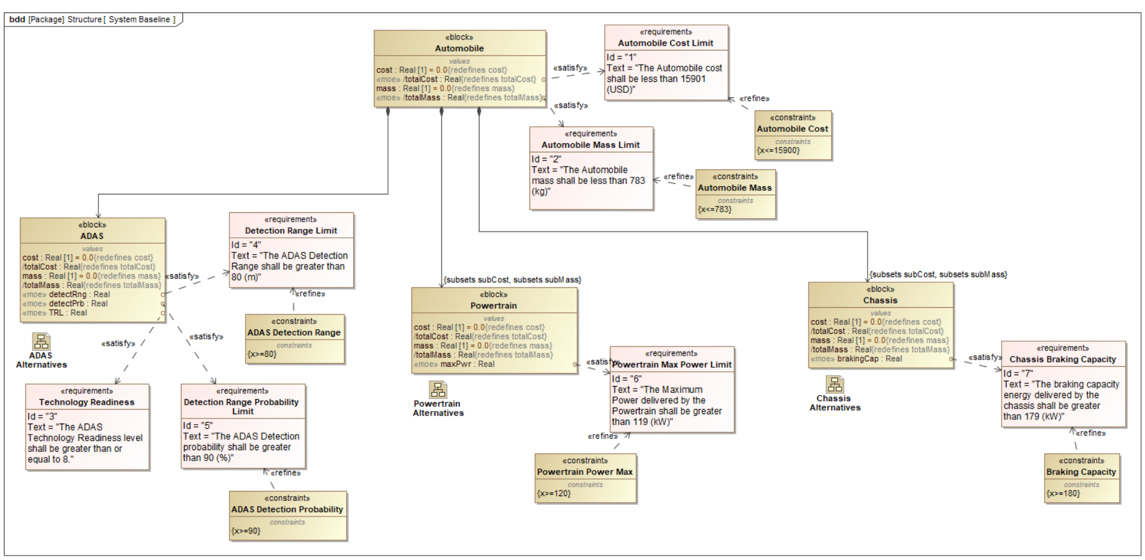


Figure 1: System Baseline Block Definition Diagram

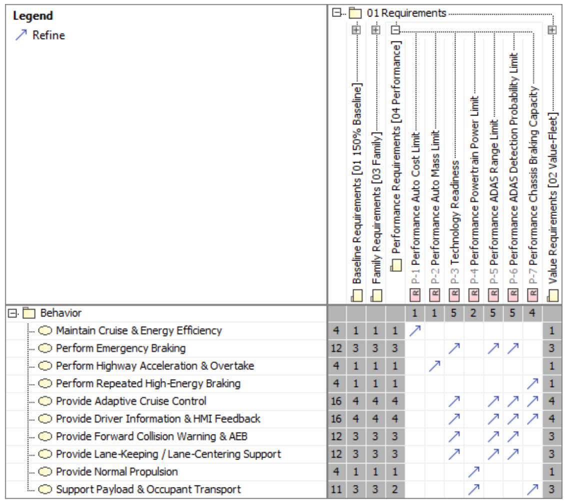


Figure 2: Use Case to Performance Segment Requirements Mapping

2.4. Tradespace Visualization

MBSE-based decision-support in clean hydrogen shows how visualization can make complex models decision-ready. Lawrence et al. use an Innoslate system model as a single repository that generates multiple stakeholder views—spider diagrams for requirement/architecture hierarchies, operational concept diagrams, parametric performance views—and supplements them with evaluation matrices and multi-criteria charts to compare alternatives across technical, economic, and sustainability criteria, enabling tradespace reasoning instead of spreadsheet interpretation [6]. Because the same model drives all views, changes to requirements, constraints, or design elements propagate immediately, supporting traceability and "what-if" exploration [6].

However, this literature—including hydrogen examples and MBPLE dashboards such as Forlingieri's—typically assumes users work inside specialized engineering environments that require modeling fluency [6][7]. As a result, tradespace and traceability visualizations remain less accessible to executives who expect enterprise BI dashboards (e.g., Power BI/Tableau). This paper argues for bridging model-centric rigor and "board-ready" decision views by streaming MBSE/MBPLE tradespace data into enterprise BI, improving communicability and adoption without losing traceability [6][7].

2.5. Identified Gaps

Prior work reveals three gaps at the MBSE–MBPLE–MCDA intersection. First, MBSE studies often use SysML mainly to describe architecture and compute attributes (cost, mass, performance) while decisions occur in external spreadsheets or ad-hoc scoring, leaving MCDA structures (criteria, value functions, preferences) outside the model [6]. Second, MBPLE advances feature models, 150% architectures, and configuration reuse, but offers limited stakeholder-friendly visualization for selecting among many valid configurations [7]. Third, MCDA literature rarely preserves traceability from scores back to architecture, requirements, and behavioral assumptions [1][2], despite calls to reconnect decision analysis and systems engineering [5]. These gaps motivate an integrated, scalable, model-grounded, visually accessible method.

3. INTEGRATING MBPLE, MBSE, AND VISUALIZATION

3.1. Define the Decision Problem

The integrated framework starts by defining the decision problem directly in the SysML model. In the automotive case, the task is to select, for each market segment (Fleet, Mainstream, Performance), one or more architecture configurations drawn from the 150% product line: six powertrain variants, six chassis/braking variants, and six ADAS variants, each with cost, mass, performance, and technology-readiness properties. Baseline system limits (automobile cost/mass, powertrain cost/power, ADAS detection range/probability, braking capacity, ADAS TRL) are captured in Figure 1 System Baseline Block Definition Diagram and the requirement table; segment-specific requirement sets refine these limits. The ten vehicle use cases in Figure 2 (e.g., Provide Normal Propulsion, Perform Emergency Braking, Provide Adaptive Cruise Control) further refine the same requirements and highlight the impacted operational behaviors.

The decision question is: *Given the segment specific requirements and cross vehicle use cases, which combinations of powertrain, chassis, and ADAS alternatives should be selected per segment to maximize value while satisfying all limits?*

To answer it, the method builds a compact objectives hierarchy consistent with MCDA guidance for operational, non-redundant, preference-relevant criteria [1,3]. The overall goal (maximize program value) decomposes into four objectives: Cost & Resource Efficiency, Vehicle Performance & Capability, Safety/Compliance, and Technical Risk.

Each objective is mapped to SysML value properties: *totalCost, totalMass, maxPower, brakingCap, detectRng, detectProb,* and *TRL*. These attributes form the evaluation vector per configuration, computed via Cameo parametric diagrams and later normalized and aggregated using MCDA [1,3].

3.2. Build the Product Line

Step two represents the vehicle family as a 150% product-line architecture in Cameo, avoiding separate models per variant. A single SysML model contains all admissible Powertrain, Chassis, and ADAS combinations (Figure 3) and shared vehicle-level properties (cost, mass, braking capacity, detection performance, technology readiness, etc.). In the block definition diagrams for Powertrain Alternatives, Chassis Alternatives, and ADAS Alternatives, each subsystem has a root block (Powertrain, Chassis, ADAS) whose value properties redefine vehicle attributes (cost, mass, maxPwr, brakingCap, detectRng, detectProb, TRL) plus specialized blocks representing options (e.g., ICE_Base, HEV_LowVolt, BEV_Standard; BaseSteel, PerformanceAlu, AdaptiveDynamics; CamOnly_LDW, CamRadarLidar_L2, SurroundLidar_L2Plus). The root blocks and their specializations constitute the 150% architecture: every block that could appear in any product is present, with variation captured by specialization and redefined

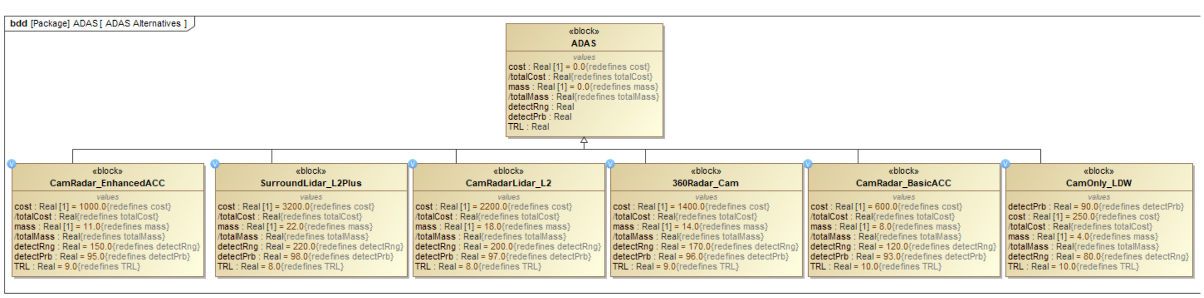


Figure 3: ADAS Alternatives Block Definition Diagram

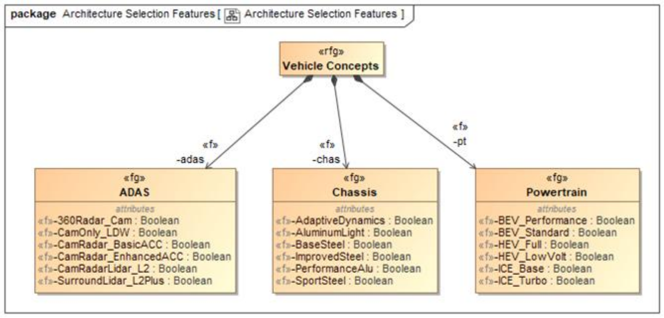


Figure 4: Architecture Selection Features Class Diagram

value properties—shared assets configured into product instances in MBPLE terms [7].

Cameo's PLE plugin then ties architecture to the decision via feature groups. The Market Segments feature group defines four Boolean features—Baseline, Value-Fleet, Family, Performance—matching the baseline and segment-specific requirement sets. These act as context selectors, controlling which requirement packages and use-case variants apply during trade study execution.

The Architecture Selection Features hierarchy (Figure 4) decomposes into ADAS, Chassis, and Powertrain groups with Boolean features named after alternatives (e.g., CamOnly_LDW, CamRadar_BasicACC, CamRadarLidar_L2, SurroundLidar_L2Plus; BaseSteel, PerformanceAlu, AdaptiveDynamics; ICE_Base, HEV_Full, BEV_Performance).

3.3. Concept Alternatives as Features

With the 150% architecture and feature groups defined, concrete alternatives are generated and evaluated using Cameo Trade Study Analysis (rather than the PLE configurator). In the "Alternatives" Block Definition Diagrams (BDD), the Powertrain, Chassis, and ADAS alternative blocks generalize their respective root blocks and redefine shared value properties (cost, mass, maxPwr, brakingCap, detectRng, detectProb, TRL, etc.). The Trade Study groups these into option sets: six powertrain variants (ICE_Base, ICE_Turbo, HEV_LowVolt, HEV_Full, BEV_Standard, BEV_Performance), six chassis variants (BaseSteel, ImprovedSteel, AluminumLight, SportSteel, PerformanceAlu, AdaptiveDynamics), and six ADAS variants (CamOnly_LDW, CamRadar_BasicACC, CamRadar_EnhancedACC, 360Radar_Cam, CamRadarLidar_L2, SurroundLidar_L2Plus). The engine enumerates the cross-product and evaluates the parametric roll-up to vehicle attributes (totalCost, totalMass, brakingCapability, detectProb, etc.).

Requirements are controlled by the PLE feature model. Requirement packages (Figure 5) for Baseline, Value-Fleet, Family (Mainstream), and Performance use presence conditions tied to Market Segments Boolean features. Enabling a segment feature activates only its requirements, allowing the same design space to be evaluated under Fleet, Family, and Performance limits without duplicating models. See "Value-Fleet Requirements" in the Appendix.

Output is a per-segment configuration–criteria matrix: rows are subsystem

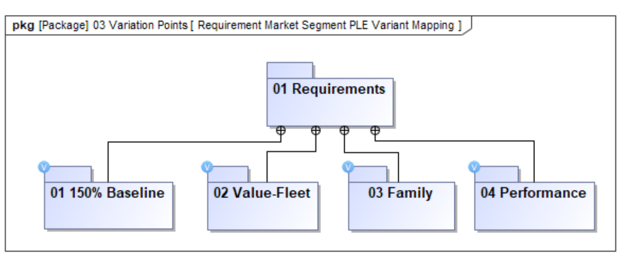


Figure 5: Requirement Market Segment PLE Variant Mapping Package Diagram

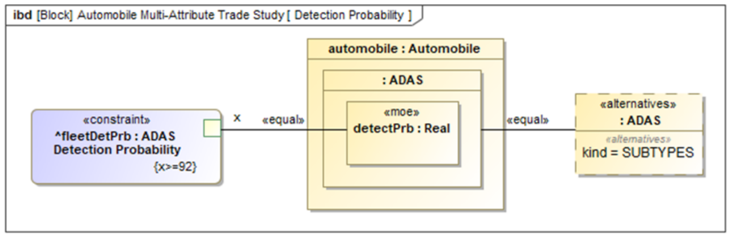


Figure 6: Automobile Multi-Attribute Trade Study Detection Probability Internal Block Diagram

combinations; columns include rolled-up attributes and active requirement pass/fail, supporting downstream MCDA, visualization, and traceability.

3.4. Compute Evaluation Attributes

Step four uses the SysML parametric model as the calculator for all trade-study evaluation attributes. Following Lawrence and Herber's model-driven decision-support pattern [6], each attribute in the multi-criteria decision model is implemented via constraint blocks instantiated in parametric diagrams and bound to subsystem properties.

For roll-ups such as totalCost and totalMass, constraint blocks like Automobile Cost and Automobile Mass (in the baseline system BDD) encode algebraic relations (e.g., vehicle cost = subsystem costs + overhead; vehicle mass = subsystem masses + integration mass). In the trade-study internal block diagrams, each constraint is instantiated as a constraint property and connected with «equal» bindings to Automobile.totalCost/totalMass and to the cost/mass value properties of the selected Powertrain, Chassis, and ADAS alternatives. When the Trade Study evaluates a configuration (e.g., HEV_LowVolt + PerformanceAlu + CamRadar_EnhancedACC), those values propagate through the bindings to compute vehicle-level totals.

The same pattern applies to non-aggregated attributes such as detection probability. In the Detection Probability Relationships BDD, the constraint block ADAS Detection Probability defines a parameter (e.g., x : Real with {x ≥ 90}). Baseline and segment refinements (Detection Range Probability Limit, Fleet ADAS Detection Probability Limit, Performance ADAS Detection Probability Limit) set different thresholds, satisfied by ADAS.detectPrb : Real. In Automobile Multi-Attribute Trade Study [Detection Probability] (Figure 6), a constraint property (e.g., «fleetDetPrb : ADAS Detection Probability {x >= 92}») is bound via «equal» connectors to the detectPrb of the nested ADAS part and the selected «alternatives» ADAS subtype. Thus, choosing CamOnly_LDW (detectPrb=90) vs CamRadarLidar_L2 (detectPrb=97) automatically evaluates compliance for the active segment. The same approach is used for detectRng, braking capacity, max power, and TRL, keeping computation and limit-checking in the model, while Trade Study only iterates combinations [6]. See "Performance Requirements Analysis Results" in the Appendix for an segmented snap shop of notional results.

3.5. Exported Data Transformation

After Trade Study sweeps the cross-product of Powertrain, Chassis, and ADAS alternatives under a selected market-segment requirement set, Cameo contains per-configuration vehicle attributes and pass/fail for each active requirement. Using Cameo's report Comma-separated Value (CSV) export, results are flattened into a table for Power BI: each row is one configuration; columns capture feature choices and evaluation fields. In the exported CSV, columns identify selections (e.g.,

Table 1: Model Data Transformation with Power Query

	Alternative_ID	adasAlt	autoCost_wt	autoCostComp	autoMassComp	AutoCost	AutoMass
1	Base_1	360Radar_Cam	0.1429	0.1104	0.06	pass	pass
2	Base_2	CamOnly_LDW	0.1429	0.1263	0.0704	pass	pass
3	Base_3	CamRadarLidar_L2	0.1429	0.0994	0.0559	pass	pass
4	Base_4	SurroundLidar_L2Plus	0.1429	0.0856	0.0518	pass	pass
5	Base_5	CamRadar_EnhancedACC	0.1429	0.1159	0.0631	pass	pass
6	Base_6	CamRadar_BasicACC	0.1429	0.1215	0.0663	pass	pass
7	Base_7	360Radar_Cam	0.1429	0.0414	0.0911	pass	pass
8	Base_8	CamOnly_LDW	0.1429	0.0573	0.1014	pass	pass
9	Base_9	CamRadarLidar_L2	0.1429	0.0304	0.087	pass	pass
10	Base_10	SurroundLidar_L2Plus	0.1429	0.0166	0.0828	pass	pass
11	Base_11	CamRadar_EnhancedACC	0.1429	0.0469	0.0942	pass	pass
12	Base_12	CamRadar_BasicACC	0.1429	0.0524	0.0973	pass	pass
13	Base_13	360Radar_Cam	0.1429	0.069	0.0704	pass	pass
14	Base_14	CamOnly_LDW	0.1429	0.0849	0.0807	pass	pass
15	Base_15	CamRadarLidar_L2	0.1429	0.058	0.0663	pass	pass
16	Base_16	SurroundLidar_L2Plus	0.1429	0.0442	0.0621	pass	pass
17	Base_17	CamRadar_EnhancedACC	0.1429	0.0745	0.0735	pass	pass
18	Base_18	CamRadar_BasicACC	0.1429	0.0801	0.0766	pass	pass
19	Base_19	360Radar_Cam	0.1429	0.0828	0.0911	pass	pass

PROPERTIES
Name
ResultsTable
All Properties

APPLIED STEPS
Source
Appended Query
Appended Query1
Appended Query2
Added Index
Changed Type1
Filtered Rows
Added Custom
Changed Type
Added Custom1
Renamed Columns1
Reordered Columns

ADAS=CamOnly_LDW, Chassis=AdaptiveDynamics, Powertrain=HEV_Full) plus an internal configuration ID/label. Requirement flags (perfAutoCost, perfAutoMass, perfBrakeCap, perfDetPrb, perfDetRng, perfMaxPwr, perfTRL) and optionally numeric attributes (totalCost, totalMass, brakingCap, detectProb, detectRng, maxPwr, TRL) are exported.

Next, the .csv export from Cameo is imported into Power BI through its *Get Data* feature. Upon import*, Power Query* is used to apply a series of transformations on the data, including changing data types, filtering out unnecessary values, and creating calculated columns that return additional data such as Configuration ID and Requirement Pass/Fail Status. These transformations, which are performed each time the data source is reimported or refreshed with new data, facilitate more efficient filtering, slicing, and visualization in the dashboards.(Table 1)

3.6. Interactive Decision Sessions

The final step uses Power BI tradespace dashboards as the focal point for iterative decision workshops. After each modeling cycle, the MBPLE team reruns Cameo Trade Study, regenerates the configuration–criteria CSVs, and refreshes the Power BI dataset as the next "sprint increment." Dashboards display, per segment, feasible configurations, value scores (under the current weight set), requirement-satisfaction status, and key choices (powertrain, chassis, ADAS). Analysts prep bookmarked views such as "Fleet Pareto Front," "Performance

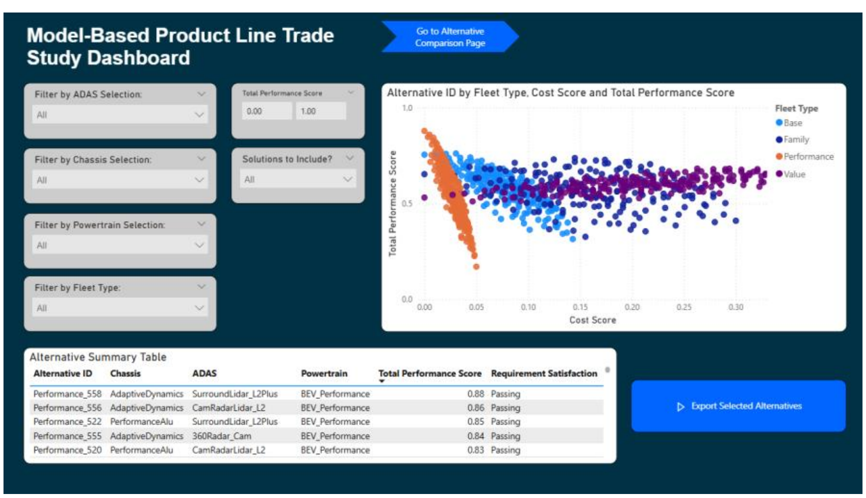


Figure 7: Interactive Concept Analysis and Selection Dashboard

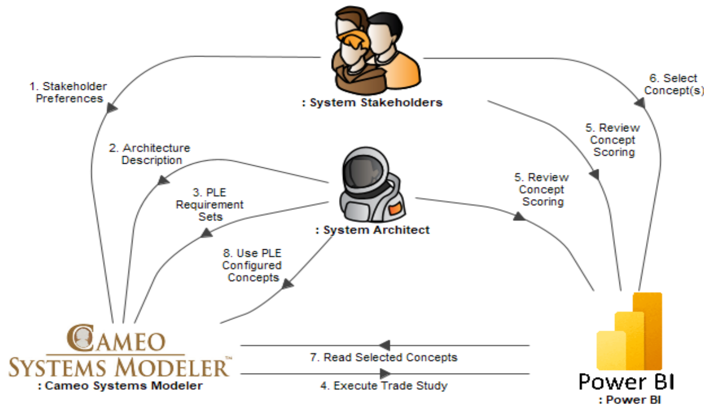


Figure 8: Closed-Loop Workflow Illustration

Shortlist," and "Cross-Segment Reuse Candidates." In workshops, stakeholders interact via filters/slicers (segments, technology options, requirement sets) and apply alternative weight sets live, refining shortlists in real time. (Figure 7) This supports an agile feedback loop [5][6] where assumptions and preferences are tested against model-based evidence. When the group converges (e.g., "Performance_558 for Performance Requirements" and "Fleet_476 for Fleet Requirements"), selections are saved as a "Selected Concepts" view and exported as a configuration CSV, then imported back into Cameo to instantiate configuration models and document the decision—closing the loop model → analysis → visualization → decision → updated model. See "Alternative Comparison" in the Appendix for additional dashboard views. A Power Automate button on the dashboard is used to export the configurations of interest, writing the configuration data to an external .csv file in the schema required for Cameo import.

3.7. End-to-End Workflow Summary

This section summarizes the integrated MBPLE–MCDA–visualization workflow as an end-to-end, repeatable process that closes the loop from stakeholder preferences to model-based concept selection and back to configuration management. Figure 8 illustrates this closed-loop workflow and the role handoffs between Cameo Systems Modeler (modeling, requirements, parametrics, trade study execution) and Power BI (interactive tradespace review and concept selection). The steps are:

1. Elicit stakeholder preference weights for the concept attributes used in the trade study (i.e., the objective hierarchy and weight sets that drive scoring).
2. Model the candidate architecture space by describing the architecture elements of interest—subsystem alternatives and their evaluation attributes—within the 150% SysML product-line model.
3. Define segment-aware requirement sets that reflect measures of effectiveness and apply PLE feature variation to activate the correct requirement package for each market segment.
4. Define the value model and execute Trade Study Analysis to enumerate alternative combinations, compute attributes via parametrics, and generate results data for each configuration.
5. Visualize and explore scoring results in the enterprise dashboard environment to review feasibility,

value scores, and key architectural choices.

6. Select concepts for further work, capturing shortlists (and the rationale/weight set context) as explicit selection views.
7. Import selected concepts back into Cameo as configuration inputs, preserving a stable key from the visualization dataset to the SysML model.
8. Conduct deeper analysis and/or expand the design space using PLE configuration selections (e.g., add alternatives, refine requirements, adjust value functions/weights), then iterate.

4. MCDA STRUCTURE

4.1. Value Model Definition

The MCDA layer builds directly on SysML-computed attributes and applies MAVT normalization and aggregation [1][2]. Each raw attribute is normalized to a [0,1] value scale using linear value functions implemented as reusable constraint blocks LinearMax (Equation 1) and LinearMin (Equation 2) See “Objective Function Scoring Diagram“ in Appendix. For less-is-better attributes (e.g., totalCost, totalMass), LinearMin maps the best (lowest feasible) value to 1 and the worst (highest) to 0 (e.g., autoCost : LinearMin, mass : LinearMin). For more-is-better attributes (e.g., brakCap, maxPwr, detectRng, detectPrb, TRL), LinearMax maps the minimum acceptable value to 0 and the maximum to 1; the Detection Probability Normalization diagram (Figure 9) shows detectPrb : LinearMax using bounds (detectPrb_min, detectPrb_max) to output a 0–1 score.

$$U(x) = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

$$U(x) = \frac{x_{max} - x}{x_{max} - x_{min}} \tag{2}$$

An objective : Objective Function constraint then computes a weighted additive score per configuration, taking normalized values and weights (autoCost_wt, mass_wt, detectRng_wt, detectPrb_wt, maxPwr_wt, brakCap_wt, trl_wt) and returning a single scalar in [0,1], consistent with MAVT [1][2].

4.2. Preference Elicitation

The weight parameters in the objective function represent stakeholder preferences over the criteria. Rather than treating them as arbitrary tuning knobs, they are elicited using structured decision-management practices as advocated by Parnell and colleagues [14] (Table 2). In a facilitated workshop, stakeholders first review the objectives hierarchy (cost, performance, safety/risk, experience) and confirm that the selected

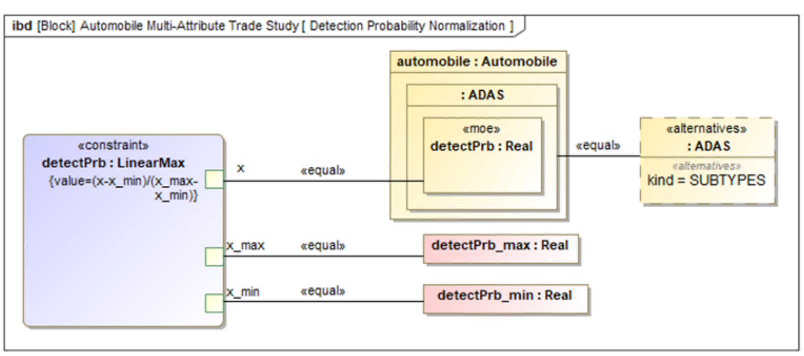


Figure 9: Detection Probability Normalization

attributes adequately represent their concerns. They then perform a simple swing-weighting exercise: for each attribute, they consider the "swing" from its worst to its best feasible level and judge how valuable that improvement is relative to the others. These judgements are captured as initial weights, normalized so they sum to one, and encoded in the SysML model as default parameter values for autoCost_wt, mass_wt, etc. If different stakeholder groups (e.g., finance, safety, marketing) have distinct viewpoints, the same structure supports multiple weight sets, enabling what-if analyses that explore how rankings change under alternative preference profiles.

4.3. Requirement Checks vs Value Scores

It is important to distinguish feasibility from desirability in the evaluation, echoing the feasibility/value dichotomy highlighted by Lawrence & Herber [6]. In this framework, each configuration is first evaluated against the segment-specific requirements using the constraint blocks described in Section 3 (e.g., Fleet, Family, Performance limits on cost, mass, detection probability, braking capacity, and TRL). These checks produce the pass/fail flags exported in the CSV tables. Only configurations that satisfy all mandatory requirements for a given segment are considered feasible; infeasible configurations are flagged in the dataset and can be filtered out or shown separately as "in need of design improvement."

The value score computed by the objective constraint is applied to both feasible and infeasible configurations, but its interpretation differs. For feasible configurations, the score directly supports preference ordering and trade-off analysis—higher scores indicate architectures that better align with stakeholder priorities. For infeasible ones, a high score highlights promising architecture configurations that currently violate one or more limits and might justify targeted design changes. Thus, the combination of requirement checks (feasibility) and weighted additive scoring (desirability) allows decision makers to see not only which configurations are acceptable today, but also where the most attractive opportunities for improvement lie, all while maintaining a clear link from value judgements back to the model-based attributes that underpin them.

4.4. Decision Data Traceability

In this workflow, decisions are explored in Power BI but recorded and enforced in Cameo, closing the MBPLE decision loop. In review boards, stakeholders navigate tradespace dashboards and may select, for example, "Performance_24 for the Performance segment." The choice is captured as a "Selected Concepts" view: a filtered subset of the fact table containing the configuration identifier, market segment,

Table 2: Performance Market Segment Value-Function Weights

#	Name	Default Value
1	perfAutoCost_wt	0.05
2	perfBrakCap_wt	0.25
3	perfDetectPrb_wt	0.1
4	perfDetectRng_wt	0.1
5	perfMass_wt	0.1
6	perfMaxPwr_wt	0.3
7	perfTRL_wt	0.1

Excel Import Status: New Updated Obsolete Unchanged

#	Name	adas. 360Radar_Cam : Boolean	adas. CamOnly_LDW : Boolean	adas. CamRadarLidar_L2 : Boolean	adas. CamRadar_BasicA... : Boolean	adas. CamRadar_Enhance... : Boolean	adas. SurroundLidar_L2Plus : Boolean
1	Performance_18	false	false	false	false	false	true
2	Performance_24	true	false	false	false	false	false
3	Fleet_240	false	false	false	false	false	true
4	Performance_560	false	false	false	false	false	true
5	Family_836	false	false	false	false	false	true
6	Family_324	false	false	false	false	false	true
7	Family_481	false	false	true	false	false	false
8	Fleet_493	false	false	true	false	false	false

Figure 10: Selected PLE Concepts Imported from PowerBI to Cameo

feature selections (Powertrain, Chassis, ADAS variants), final value scores, and key requirement-satisfaction indicators.

Power BI exports this as a CSV decision table where each row is a chosen concept and columns are Boolean feature/segment flags (e.g., PerfSegment=TRUE, PT_BEV_Performance=TRUE, CH_AdaptiveDynamics=TRUE, ADAS_SurroundLidar_L2Plus=TRUE). When imported into Cameo, PLE tooling uses these fields to instantiate configuration models/variant realizations in the 150% architecture (Figure 10), enabling downstream design and verification.

Each configuration is assigned a persistent Decision ID and annotated with metadata (decision date, approving authority, active weight set, links to evaluation results) using Cameo elements (e.g., «configuration» blocks, «rationale» notes, tagged values). This provides traceability from realized variants back to MCDA scores and requirement pass/fail, aligning with Peterson's call to tightly couple systems engineering and decision analysis.[5]

5. DISCUSSION

5.1. Approach Benefits

The approach's primary benefit is re-uniting systems engineering and decision analysis in one workflow. Rather than using the SysML model as a black box feeding spreadsheets, the value model—criteria, normalization functions, and weighted objective—is implemented as constraint blocks and parametric diagrams in Cameo, making decision logic a first-class, traceable part of the model, consistent with Peterson's call to make system value explicit [5]. Because structure, behavior, requirements, and value live in the same model, architectural changes update both performance and value scores consistently, echoing model-driven decision support in energy-systems MBSE [6].

Second, MBPLE scales the method across a product line and lifecycle: feature models and a 150% architecture reuse shared assets (requirements, blocks, parametrics, value models) across Fleet, Family, and Performance segments. As Forlingieri et al. emphasize, feature-centered variability enables product-line success [7]; here, those features also become keys in the tradespace and BI data model.

Third, it closes the loop between Cameo rigor and Power BI communication: model exports populate BI fact tables, and BI decisions feedback as configuration selections and annotations, keeping the model as the "single source of truth." Finally, traceability improves versus document-based reviews: each tradespace point links to feature selections, requirement sets, and use cases, aligning with INCOSE Handbook expectations [8] and strengthening stakeholder alignment and defensibility.

5.2. Traditional Practice Comparison

Many organizations still perform early concept selection with Pugh matrices and spreadsheet scoring. Pugh's method compares options to a reference concept using plus/minus (and sometimes rough numeric) ratings [12][13], while more formal spreadsheets implement weighted sums informed by decision-analysis guidance such as Parnell's value-focused thinking and morphological analysis [14]. These tools work for small option sets, but they treat alternatives as table rows with no link to an architecture model; when powertrain, ADAS, or requirement thresholds change, updates are manual and synchronization is not assured.

Document-centric systems engineering has similar drawbacks: architecture options live in text, slide decks, or static diagrams; trade studies are embedded tables; and variant descriptions are copied across derivatives, making traceability questions costly to answer [8][9]. By contrast, the integrated MBSE–MBPLE–MCDA–Visualization approach addresses three shortcomings: (1) traceability, linking every evaluated configuration to feature selections, subsystem alternatives, requirements, and use cases in the SysML model, with the value model inside the model; (2) product-line configurability, using 150% architecture and feature models to evaluate multiple segments and future variants; and (3) scalability, where the Trade Study engine and Power BI handle many combinations via parametric evaluation and interactive tradespace dashboards—beyond what static Pugh tables can manage.

5.3. Practical Considerations

Deploying an integrated MBSE–MBPLE–MCDA–Visualization workflow is as much organizational change as technical integration, and early pilots surface recurring issues.

Skills gap: The method requires SysML/product-line capability alongside Power BI data-model and dashboard skills—often split across silos. MBPLE concepts (features, variability dimensions, configuration models) typically demand more training than document-centric SE [7]. Building robust BI datasets also requires knowledge beyond Excel (schemas, measures, governance). INCOSE emphasizes that shared models and data should be managed as organizational assets with clear ownership and lifecycle policies, not personal artifacts [8]; without deliberate role and skill investment, the approach risks remaining a one-off.

Tool-integration friction: Coupling Cameo exports to Power BI can be derailed by inconsistent alternative/feature naming, changing CSV column headings, or segment-to-segment differences in requirement-flag labels. Standard export templates, stable naming conventions (configuration IDs, feature names, requirement identifiers), and a "data contract" defining required columns and change control help contain this.

Incremental rollout: A phased adoption—starting with a bounded pilot (e.g., ADAS + chassis for one segment, or Fleet vs Mainstream only)—supports learning and reduces resistance, aligning with Senge's "learning organization" framing [11]. Scale can then expand to powertrain, more segments, richer value models, and tighter decision traceability.

5.4. Limitations

Results come from a single notional automotive case (multi-segment platform with powertrain, chassis, ADAS variability). While the integrated MBSE/MBPLE + MCDA + BI visualization pattern may generalize to domains like aerospace, rail, or energy, the specific constructs, value functions, and dashboards require adaptation and validation. Evaluations are deterministic point estimates (cost, mass, braking capacity, detection probability, TRL), so rankings represent nominal preferences; uncertainty is only treated qualitatively, and future work should add risk/uncertainty propagation and probabilistic or fuzzy MCDA. MCDA quality also depends on criteria and weights; poorly defined or overlapping criteria and unstable weights can mislead [1][3].

6. CONCLUSIONS

6.1. Summary of Contributions

This paper presents a repeatable concept-selection workflow that integrates MBPLE, MCDA, and enterprise visualization. It draws on decision-analysis/MCDA theory [1][3], value-centered systems engineering and MBSE practice [5][6], and feature-based MBPLE principles [7]. By implementing value functions and objective weights as SysML constraint blocks tied to a 150% product-line architecture, the method makes the decision model explicit and traceable within the system model.

A notional automotive case demonstrates feasibility: a 150% vehicle architecture with multiple powertrain, chassis, and ADAS alternatives is modeled in Cameo; segment requirement sets (Fleet, Family, Performance) and use cases refine a baseline; and Cameo Trade Study evaluates the design space. Results are exported as a structured dataset and visualized in Power BI dashboards that reveal cost–performance–safety–experience trades and support cross-segment selection and reuse insights.

The paper also provides practical patterns for model structure (feature-oriented blocks, segment-aware requirement packages with presence conditions, normalized value and weighted objective constraints), standardized CSV export (star schema), and feedback of selected concepts into Cameo as configuration states—creating a coherent, model-based decision-support ecosystem.

6.2. Future Research Directions

Several extensions are needed for high-consequence programs. First, incorporate uncertainty and risk: the current framework uses deterministic point estimates for cost, mass, braking capacity, detection performance, and technology readiness. Future work should couple the SysML parametric network to Monte Carlo simulations and apply MCDA risk methods—fuzzy MCDM, Best–Worst Method (BWM), and enhanced FMEA—as surveyed by Ren et al. [3][4]. This would propagate attribute ranges and confidence through the value model, enabling probability-of-compliance, risk-adjusted scores, and robustness-oriented tradespace visualizations.

Second, validate generality in other domains. Clean-hydrogen and broader energy-systems work by Lawrence and Herber offers a strong testbed given their complexity and multi-criteria objectives [6], alongside rail, aviation, maritime, and cyber-physical infrastructure with different regulations and uncertainties.

Third, deepen the decision trace beyond selected configurations by capturing rationale, contested alternatives, and organizational learning—aligning with systems-thinking and learning-organization perspectives [10][11]—via explicit decision/rationale elements linked to the value model, tradespace views, and post-decision outcomes.

7. REFERENCES

[1.] Bouyssou, D., et al. Evaluation and Decision Models with Multiple Criteria: Stepping Stones for the Analyst. Springer, 2006.

[2.] Bouyssou, D., et al. Evaluation and Decision Models: A Critical Perspective. Kluwer, 2000.

[3.] Ren, J. (ed.). Multi-Criteria Decision Analysis for Risk Assessment and Management. Springer, 2021.

[4.] Ivascu, L., & Artene, A. E. “Risk Assessment: Indicators and Organizational Models.” In [3].

[5.] Peterson, T., & Schindel, B. “Explicating System Value through First Principles: Re-Uniting Decision Analysis with Systems Engineering.” INCOSE INSIGHT, 2016.

[6.] Lawrence, S., & Herber, D. R. “A Model-Based Systems Engineering Approach for Effective Decision Support of Modern Energy Systems Depicted with Clean Hydrogen Production.” Systems, 12(8), 290, 2024.

[7.] Forlingieri, M., et al. Model-Based Product Line Engineering (MBPLE): The Feature-Based Path to Product Lines Success. Wiley, 2025.

[8.] INCOSE. Systems Engineering Handbook, 5th ed. Wiley, 2023.

[9.] Kossiakoff, A., et al. Systems Engineering: Principles and Practice, 3rd ed. Wiley, 2020.

[10.] Meadows, D. H. Thinking in Systems. Earthscan, 2009.

[11.] Senge, P. M. The Fifth Discipline. Doubleday, 2004.

[12.] Pugh, S. Total Design: Integrated Methods for Successful Product Engineering. Addison-Wesley, 1990.

[13.] Lønmo, L., & Muller, G. “Concept Selection – Applying Pugh Matrices in the Subsea Processing Domain.” INCOSE International Symposium, 2014.

[14.] Parnell, G. S., et al. Trade Study Analytics: Creating and Exploring the System Tradespace. Wiley, 2017.

8. CONTACT INFORMATION

Gary Kliczinski
Senior Principal Model-Based Systems Engineer
Strategic Technology Consulting
gary.kliczinski@arcfield.com
734-738-6800

Ryan Pykor
Senior Principal Model-Based Systems Engineer
Strategic Technology Consulting
ryan.pykor@arcfield.com
313-622-3055

9. ACRONYMS

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
BEV	Battery Electric Vehicle
BI	Business Intelligence
BOM	Bill of Materials
BDD	Block Definition Diagram
BWM	Best–Worst Method
CSV	Comma-Separated Values
ECU	Electronic Control Unit
FMEA	Failure Mode and Effects Analysis
HEV	Hybrid Electric Vehicle
HMI	Human–Machine Interface
ICE	Internal Combustion Engine
INCOSE	International Council on Systems Engineering
L2	SAE Level 2 (partial driving automation)
L2Plus	Level 2+ (enhanced Level 2 driver assistance)
LDW	Lane Departure Warning
MAVT	Multi-Attribute Value Theory
MBPLE	Model-Based Product Line Engineering
MBSE	Model-Based Systems Engineering
MCDA	Multi-Criteria Decision Analysis
MCDM	Multi-Criteria Decision Making
OEM	Original Equipment Manufacturer
PLE	Product Line Engineering
RPN	Risk Priority Number
SysML	Systems Modeling Language
TRL	Technology Readiness Level

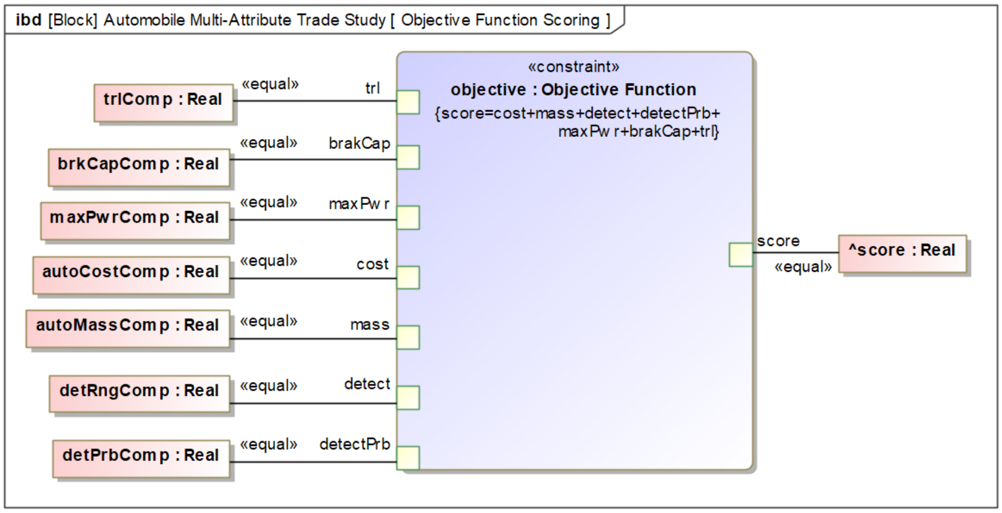
10. APPENDIX

#	Id	Name	Text	Refined By	Satisfied By
1	F-1	F-1 Fleet Auto Cost Limit	When MarketSegment = Fleet, the Automobile cost shall not exceed 14500 (USD).	Automobile Cost Maintain Cruise & Energy Efficiency	/totalCost : Real
2	F-2	F-2 Fleet Auto Mass Limit	When MarketSegment = Fleet, the Automobile mass shall not exceed 750 (kg).	Automobile Mass Perform Highway Acceleration & Overtake Support Payload & Occupant Transport	/totalMass : Real
3	F-3	F-3 Technology Readiness	The ADAS Technology Readiness level shall be greater than or equal to 10.	Technolog Readiness Provide Lane-Keeping / Lane-Centering Support Provide Forward Collision Warning & AEB Provide Driver Information & HMI Feedback Provide Adaptive Cruise Control Perform Emergency Braking	TRL : Real
4	F-4	F-4 Fleet Powertrain Power Limit	When MarketSegment = Fleet, the Maximum Power delivered by the Powertrain shall be at least 125 (kW).	Powertrain Power Max Provide Normal Propulsion Support Payload & Occupant Transport	maxPwr : Real
5	F-5	F-5 Fleet ADAS Range Limit	When MarketSegment = Fleet, the ADAS Detection Range shall be at least 100 (m).	ADAS Detection Range Perform Emergency Braking Provide Forward Collision Warning & AEB Provide Adaptive Cruise Control Provide Lane-Keeping / Lane-Centering Support Provide Driver Information & HMI Feedback	detectRng : Real
6	F-6	F-6 Fleet ADAS Detection Probability Limit	When MarketSegment = Fleet, the ADAS Detection probability shall be at least 92 (%).	ADAS Detection Probability Perform Emergency Braking Provide Forward Collision Warning & AEB Provide Adaptive Cruise Control Provide Lane-Keeping / Lane-Centering Support Provide Driver Information & HMI Feedback	detectPrb : Real
7	F-7	F-7 Fleet Chassis Braking Capacity	When MarketSegment = Fleet, the braking capacity energy delivered by the chassis shall be at least 190 (kW).	Braking Capacity Perform Repeated High-Energy Braking Provide Adaptive Cruise Control Support Payload & Occupant Transport Provide Driver Information & HMI Feedback	brakingCap : Real

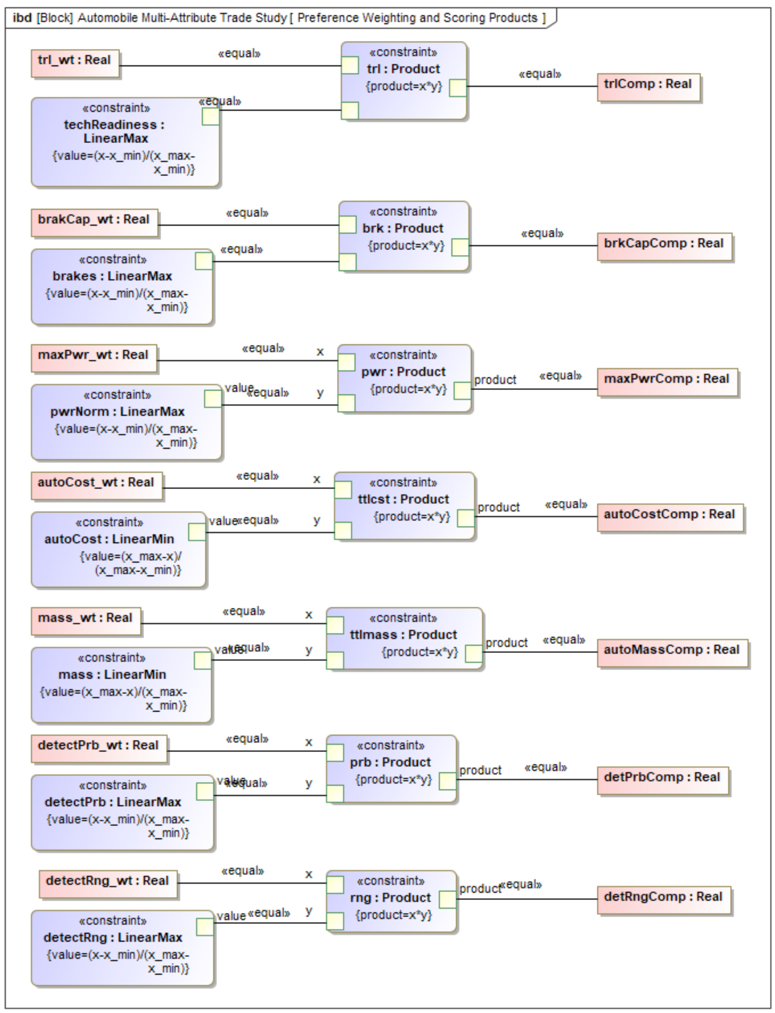
Value-Fleet Requirements

ADAS	Chassis	Powertrain	perfAutoCost	perfAutoMass	perfBrakeCap	perfDetPrb	perfDetRng	perfMaxPwr	perfPtCost	perfTRL
CamOnly_LDW	AdaptiveDynamics	HEV_Full	pass	pass	pass	fail	fail	pass	pass	pass
CamRadar_EnhancedACC	AdaptiveDynamics	HEV_Full	pass	pass	pass	fail	pass	pass	pass	pass
SurroundLidar_L2Plus	AdaptiveDynamics	HEV_Full	pass	pass	pass	pass	pass	pass	pass	pass
CamRadar_BasicACC	AdaptiveDynamics	HEV_Full	pass	pass	pass	fail	fail	pass	pass	pass
CamRadarLidar_L2	AdaptiveDynamics	HEV_Full	pass	pass	pass	pass	pass	pass	pass	pass
360Radar_Cam	AdaptiveDynamics	HEV_Full	pass	pass	pass	pass	pass	pass	pass	pass
CamOnly_LDW	AdaptiveDynamics	BEV_Performance	pass	pass	pass	fail	fail	pass	pass	pass
CamRadar_EnhancedACC	AdaptiveDynamics	BEV_Performance	pass	pass	pass	fail	pass	pass	pass	pass
SurroundLidar_L2Plus	AdaptiveDynamics	BEV_Performance	pass	pass	pass	pass	pass	pass	pass	pass
CamRadar_BasicACC	AdaptiveDynamics	BEV_Performance	pass	pass	pass	fail	fail	pass	pass	pass
CamRadarLidar_L2	AdaptiveDynamics	BEV_Performance	pass	pass	pass	pass	pass	pass	pass	pass
360Radar_Cam	AdaptiveDynamics	BEV_Performance	pass	pass	pass	pass	pass	pass	pass	pass

Performance Requirements Analysis Results



Objective Function Scoring Diagram



Composite Preference Weighting and Scoring Diagram

Alternative Comparison