Application of Model Based System Engineering (MBSE) Principles to an Automotive Driveline Sub-System Architecture

Presented By: Robert Kraus, George Papaioannou and Arun Sivan
Discussion Agenda

- Introduction & Project Summary
- Driveline Definitions and Concepts
- Systems Engineering Concepts
- MBSE Concepts
- Driveline Model Structure – *Functional and Logical Decomposition*
- Requirements & Test Case Management
- Requirements Management: Satisfy Relationships
- Parametric Relationships – *Constraint Modeling Applied to Sizing*
- Benefits of Applied MBSE
Introduction & Project Summary

Current State: Today’s automotive driveline system engineering process is “document based”

- Complex system requirements and specifications are communicated through large amounts of electronic data
- Often leads to incomplete or conflicting requirements
- Inefficient, redundant, error prone
- Running changes introduce potential problems

Project Summary:

- Obtained and deconstructed existing driveline system methods and sizing tools
- Identified need for improved requirements traceability in driveline systems engineering
- Created detailed driveline system model to apply the concepts of MBSE using SysML
- Added parametric constraints for sizing calculations
- Delivered functional MBSE model as proof of concept
Driveline Definitions and Concepts

Architecture:

- A driveline system links the powertrain output to the drive wheels
- Primary function is to transmit drive torque from the powertrain to the ground (wheels)
- Driveline subtypes such as FWD, RWD, AWD are treated as generalizations in SysML

Components:

- Driveshaft / half shaft - transmits torque to front/rear or left/right
- Axle - multiplies driveshaft torque and directs to wheels
- Additional - Transfer case, PTU, disconnect device, U-joints, CV-joints, flex coupler

Sizing:

- Design optimization of each component, system and subsystem is the primary objective
- A sizing tool converts input data into torque outputs for all vehicle variations and uses industry standard equations and some correction factors.
System Engineering Concepts

● V-Model:
  ○ Top level requirements are decomposed to the subsystem and component levels, each with a specific validation plan flowing down the left side of the V and back up the right side.

● Context Diagram:
  ○ Represents system interactions to an external environment
  ○ Interacting systems are defined as “black boxes”

● P-Diagram:
  ○ Expands and refines context for more detailed black boxes
  ○ Includes detail on input signals, control factors, noise factors, outputs, and potential failure modes

● MBSE
  ○ Modeling Language (SysML, UML etc)
  ○ Modeling Method
  ○ Modeling Tool (Magicdraw, IBM Rational Rhapsody etc)
● Requirements define customer and stakeholder needs in technical terms
● In SysML, system requirement statements are defined as objects
● Each object contains the requirement text & a unique identifier
● The requirement type defines the features a requirement can be associated with
● Generalizations manage and allocate requirements through inheritance relationships
● Requirements must be verified by test cases
● Test cases are checkpoints, such as design reviews or physical tests
Model Based System Engineering Concepts: *Functional & Logical Architecture*

- Functions define what actions / activities must be accomplished or completed to achieve a desired outcome.
- An *operation* is a property of a *block*.
- A *block* is an abstract representation of any part of a system, like physical hardware or a signal.
- Functions are linked through logical relationships to the various subsystems and components.
- The *logical architecture* describes how a system will be implemented.
- It abstractly defines a technical solution based on the system’s required sub-systems, components and their relationships.
- A logical architecture should only be created after the system’s functions and requirements are clearly defined.
- It does not define any *particular* system implementation, but rather the general guidelines so as to remain *solution-neutral*. 
Methods of Modeling: *Functional Decomposition*

- Five basic operations of the driveline systems were identified from the P-diagram
  - The system needs to
    - Transmit torque
    - Direct torque left / right
    - Direct torque fore / aft
    - Multiply torque
    - Disconnect the secondary driveline
  - Each function is associated, or mapped, to at least one logical block
  - The function, or operation, is then inherited through generalizations
Methods of Modeling: *Functional Decomposition*

- Transmission of propulsion force from the powertrain to the wheels is the most basic, and most obvious, driveline function.
- At the functional level of abstraction the degree or amount of torque transferred is not necessarily relevant – all that matters is that all drivelines transmit torque.
- In our model the “transmit torque” function is inherited by all mechanical sub-systems and components.
- It is important to note that no one component alone delivers or satisfies the transmit torque function – it is an emergent function of the total system.
- The next function is to direct torque left/right which is the operation or behavior of a axle/transaxle differential logical block.
- The next function is to direct torque fore–aft which is the behavior of a transfer case and is only present in AWD or 4X4 driveline configuration.
- In a driveline system model, the multiply torque function is a behavior of the axle only.
- The last function is the disconnect function which is the behavior of a disconnect device.
Methods of Modeling: Logical Decomposition

- Logical blocks are constructed after the functional blocks are defined, requiring some experience and engineering judgement.
- A good logical decomposition should remain untethered to any specific design concept, since it is unlikely that the first design concept will be the best design concept.
- The driveline logical diagram provides the structure required to capture the vehicle inputs required to support parametric equations for calculating driveline impact torques, which is the basis for sizing.
Requirements & Test Case Management

- Import the requirements using the import feature in MagicDraw into a requirements package.
- Existing test methods are imported using a method similar to that of importing requirements.
- Verify matrices are created and each requirement is verified by one or more test methods.
**Requirement Management: Satisfy Relationships**

- Requirements are *satisfied* by physical elements.
- In MBSE this relationship is created at the logical level of abstraction, prior to any physical parts being designed.
- The satisfy matrix indicates that each requirement is mapped to at least one logical block through a satisfy relationship that indicates that the logical block is required to deliver or meet the requirement.
- The table shows all the logical elements in the driveline model that satisfy requirements, and the requirements are separated into the seven different SysML requirement types.

I think I am done for now. Will revise a bit more after Bob and Mike review.
Parametric Relationships

- In systems engineering, design involves making decisions between solution alternatives
- General process is:
  - Generate ideas
  - Evaluate alternatives - engineering analysis
  - Decide between alternatives - interpret results
- SysML provides a language to express and perform mathematical system analysis through parametric diagrams
- Parametric diagrams show mathematical relationships between the blocks of the system model.
- They act as constraints on the system design.
Parametric Relationships: Sizing Inputs

- Parametric input parameters are obtained from industry standard equations for sizing (sizing tool).
- Mapping of parametric inputs as value properties associated with logical blocks based on the property’s logical ownership are shown here.
- The impact torque sizing inputs are the value properties associated with the appropriate logical blocks.
- At logical architecture levels, these value properties are to be defined variables that are not associated with any specific instance.
- Specific instances can be created when required.

<table>
<thead>
<tr>
<th>Parametric Input</th>
<th>Logical Block Ownership</th>
<th>Value Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle Assumptions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Onboard Configuration (Type)</td>
<td>Vehicle Inputs</td>
<td>Pick List</td>
</tr>
<tr>
<td>Brake Traction Control Equipped</td>
<td>Vehicle Inputs</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Max GVWR</td>
<td>Vehicle Inputs</td>
<td>Kg</td>
</tr>
<tr>
<td>Max Front GVR</td>
<td>Vehicle Inputs</td>
<td>Kg</td>
</tr>
<tr>
<td>Max Rear GVR</td>
<td>Vehicle Inputs</td>
<td>Kg</td>
</tr>
<tr>
<td>GVWR</td>
<td>Vehicle Inputs</td>
<td>Kg</td>
</tr>
<tr>
<td>Usability, Road Load Data Set (FLD)</td>
<td>Vehicle Inputs</td>
<td>Pick List</td>
</tr>
<tr>
<td>Engine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine Torque in Let Gear</td>
<td>engine</td>
<td>N·m</td>
</tr>
<tr>
<td>Max Engine Torque Available</td>
<td>engine</td>
<td>N·m</td>
</tr>
<tr>
<td>Engine Torque in Rev. Gear</td>
<td>Engine</td>
<td>N·m</td>
</tr>
<tr>
<td>Engine Moment of Inertia</td>
<td>engine</td>
<td>M·m²</td>
</tr>
<tr>
<td>Transmission</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transmission Type</td>
<td>Transmission</td>
<td>Pick List</td>
</tr>
<tr>
<td>1st Gear Torque Transmission</td>
<td>Transmission</td>
<td>N·m</td>
</tr>
<tr>
<td>2nd Gear Torque Transmission</td>
<td>Transmission</td>
<td>N·m</td>
</tr>
<tr>
<td>Transmission Efficiency</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Torque Converter Ratio @ Stall</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Transmission Efficiency</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Transmission Efficiency</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Engine Torque at 0.5 Speed Ratio</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Transaxle (R/R)</td>
<td>Transmission</td>
<td>M/D</td>
</tr>
<tr>
<td>Physical Moments of Inertia - Manual Trans</td>
<td>Transmission</td>
<td>M/D</td>
</tr>
<tr>
<td>Clutch Plate Torque Limiter - Manual Trans</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Number of Forward Gears</td>
<td>Transmission</td>
<td>Integer</td>
</tr>
<tr>
<td>1st Gear Ratio</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>2nd Gear Ratio</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Reverse Gear Ratio</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Reverse Gear Ratio</td>
<td>Transmission</td>
<td>Real</td>
</tr>
<tr>
<td>Drive axle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anti Transfer Case Lock - Anti Roll Demand</td>
<td>Transmission</td>
<td>Pick List</td>
</tr>
<tr>
<td>Anti Transfer Case Lock</td>
<td>Transmission</td>
<td>Pick List</td>
</tr>
<tr>
<td>Anti Transfer Case Lock</td>
<td>Transmission</td>
<td>Pick List</td>
</tr>
<tr>
<td>Torsion D Shaft Stiffness</td>
<td>Drive Shaft</td>
<td>N/mm/ rad</td>
</tr>
<tr>
<td>Torsion D Shaft Stiffness</td>
<td>Drive Shaft</td>
<td>N/mm/ rad</td>
</tr>
<tr>
<td>Highest Joint Angle @ Design</td>
<td>Drive Shaft</td>
<td>Degrees</td>
</tr>
<tr>
<td>Wheel / Tire</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Traction &amp; Tires</td>
<td>Wheels / Tires</td>
<td>Yes / No</td>
</tr>
<tr>
<td>Wheel / Tire</td>
<td>Wheels / Tires</td>
<td>Pick List</td>
</tr>
<tr>
<td>Tire Size</td>
<td>Wheels / Tires</td>
<td>Pick List</td>
</tr>
<tr>
<td>Max Tire SL</td>
<td>Wheels / Tires</td>
<td>Pick List</td>
</tr>
<tr>
<td>Tire Coefficient of Static Friction</td>
<td>Wheels / Tires</td>
<td>Real</td>
</tr>
</tbody>
</table>

14
Parametric Relationships: **Constraint modeling**

- Parametric constraints specify equivalence relationships between logical blocks
- Defined in a similar manner to IBDs, but they use internal relationships with constraint parameters instead of part parameters
- Restricted to connecting only through binding connectors, typically with a parametric constraint at one end of the connection
- The key element is the constraint block, which is used to constrain the properties of one or more other blocks
- Constraint blocks consist of constraint expressions \( \tau = F \ast d \) and constraint parameters (such as \( \tau, F \) and \( d \))
Parametric Relationships: *Driveline Sizing*

- **Goal**: Show proof of concept for the calculation of sizing torques
- A total of eight different constraint blocks used to model impact torque, consisting of torque flow equations, logic flow and Boolean assignments
- Instances need to be created with specific inputs to complete parametric analysis for driveline sizing
- Parametric calculations for impact torque were completed for a proposed vehicle program, and compared to the values obtained from existing analytical tools
- Successful proof of concept, however, requires more time and work to reliably replace existing analytical tool
Benefits of Applied MBSE

- Object oriented modeling can be equally applicable to a fully mechanical system
- MBSE improves engineering productivity and efficiency
- System models are more flexible, consistent and scalable across all sub-systems
- Streamlined communication of requirements by making all key input and output parameters available to all model users
- Better traceability and linkages between requirements and their methods of verification
- The reduction of requirement redundancies and automatic validation of test case verification could result in the elimination of entire tracking departments
- The ability to continuously update and manage component design inputs through parametric relationships with vehicle level inputs
- Every individual with access to the model can not only see and verify his or her subsystem, but also view all of the interactions of their subsystem with other parts of the entire system minimizing cross functional issues and miscommunication
Benefits of Applied MBSE

- We emphasized careful categorization of existing requirements and during the import process eliminated redundancies, reduced 500+ to ~300
- Through our integration efforts into SysML categories, we discovered a clear need and benefit of improved elicitation and partitioning of existing requirements
- Requirements and test cases can be added to the model fairly easily, and can be easily linked with the entire driveline system.
- A new requirement with the document based approach will require a lot of cross referencing with other requirements, and redundancies and total misses are quite possible
Benefits of Applied MBSE: *Parametric Input Cascade and Control*

- Through parametric relationships, top level assumption changes are immediately cascaded down and can be verified against existing component variable properties
  - For example, if engine torque in first gear goes up, it will immediately be calculated into transmission output torque and compared against the axle maximum input torque limit. These are SysML numbers calculated, used as an alert, but details need to also exist to define past systems level. Not too deep and complex. You lose sight of big picture.
  - Changes in tire properties can be linked to and compared against halfshaft joint design limits automatically
- If the input assumptions exceed design limits the model shows an output error alerting engineering to act