

# **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 - *Driveline Engineering Today*
- Project Summary - *Driveline System Engineering Using MBSE*
- Systems Engineering Concepts
- MBSE Concepts
- Driveline Definitions and Concepts
- Driveline Sizing - *Purpose, Inputs, Torque Transfer Physics*
- Driveline Model Structure - *Functional and Logical Decomposition*
- Requirements Management - *Import Process, Traceability, Test Cases*
- Parametric Relationships - *Constraint Modeling Applied to Sizing*
- Benefits of Applied MBSE
- Conclusion

# Introduction - *Driveline Engineering Today*

- 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
  - Requires constant updates and management
  - Running changes introduce potential problems
- Objective: Investigate the real world application of a SysML model in place of the document based approach to a mechanical driveline system



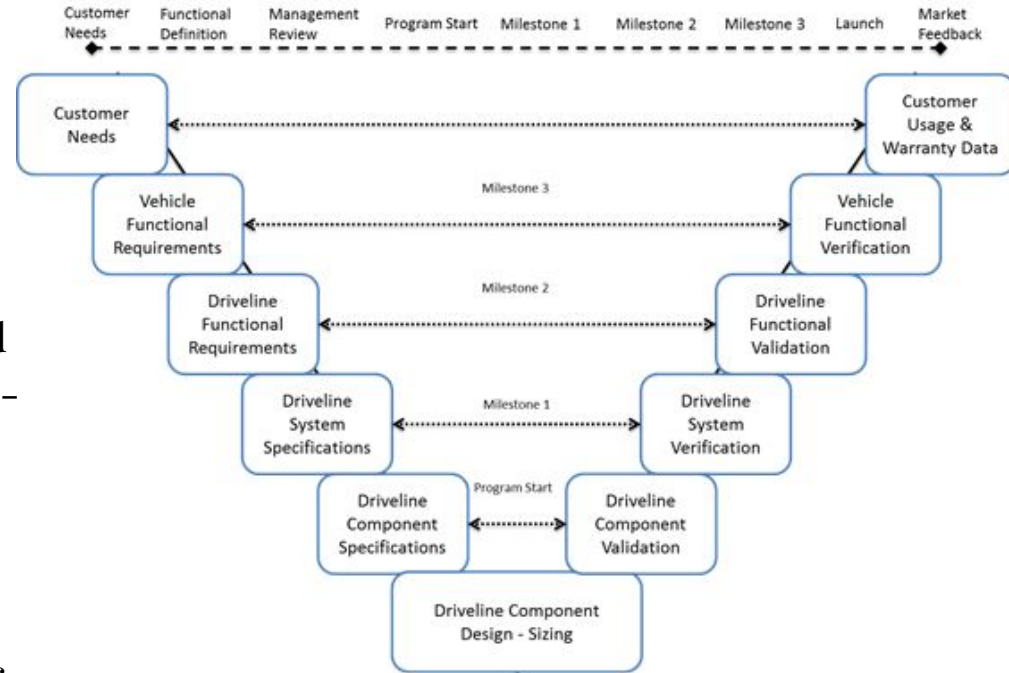
# Project Summary

- Investigated system engineering concepts
- Obtained and studied current driveline system methods and sizing tools
- Identified need for improving requirements traceability in driveline systems engineering
- Studied MBSE concepts
- 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



# System Engineering Concepts : *The System V-Model*

- The V-model is a graphical representation of the typical product engineering lifecycle
- Developed in the 1980s
- Left side of V deals with decomposition and definition
- Top level requirements are decomposed to the subsystem and component levels – becomes specifications at each level
- Right side deals with integration and reconstitution
- Designs are assembled, integrated and tested – each level of the V has a specific validation plan



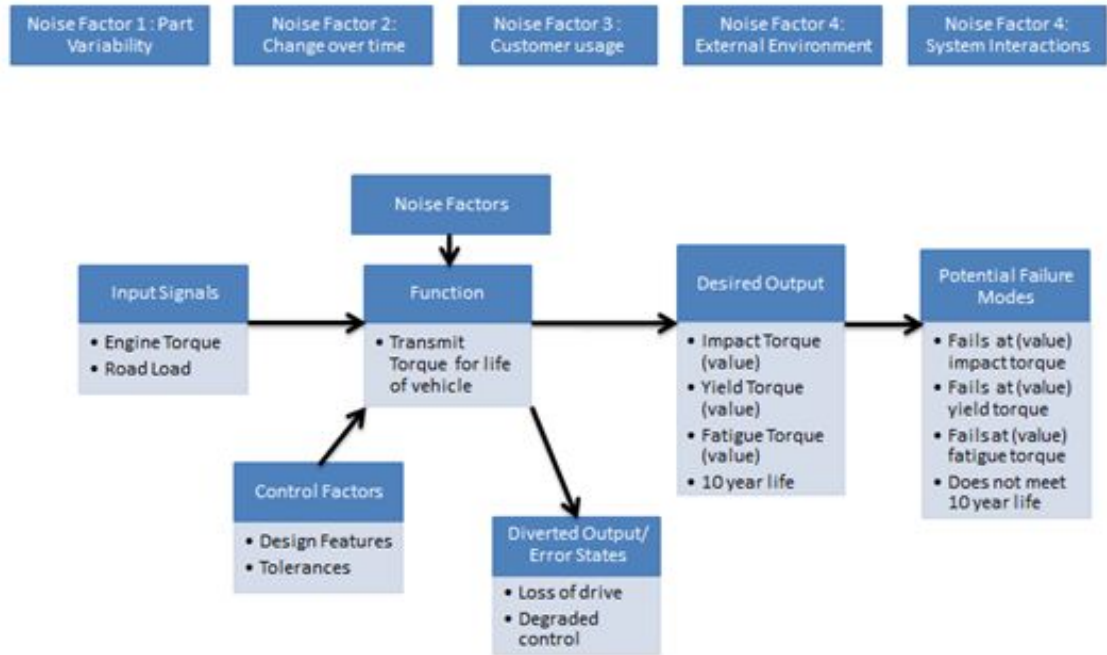
# System Engineering Concepts: *Boundary Definition*

- Context diagram shows system interactions with external environment
- Critical to define system boundary and interactions to reduce potential failures
- The driveline system and other interacting systems are represented as “black boxes”
- This approach delineates the definition of the *specification* (what the system is expected to do) from any particular *solution* (how the system could be implemented) (Pearce and Friedenthal 2013)



# System Engineering Concepts: *The P-Diagram*

- The system context can be further expanded and refined to include more detail to the black box approach in an illustration called the P-diagram
- The P-diagram includes detail on input signals, control factors, noise factors, outputs, and potential failure modes
- The P-diagram maintains a “solution neutral approach”



# Model Based System Engineering Concepts

- The function of systems engineering is “*to guide the engineering of complex systems*” (Seymour & Biemer 2011)
- Excessively complex *document-based* systems are difficult to manage
  - Large volume of documents requiring continuous management and updates
  - Requirements can be generated at any level with limited communication
  - Requirements are stored in multiple locations and databases
  - Changes are not rigorously cascaded up or down
  - No method to insure all requirements are satisfied or verified
- To resolve this, our team is proposing a *model based* approach to managing requirements
  - Requirements and specifications are decomposed using the logical structure of the systems engineering V-model
  - Requirements are stored and managed in a Model-Based Systems Engineering (MBSE) software package where they are tracked and verified through the project lifecycle

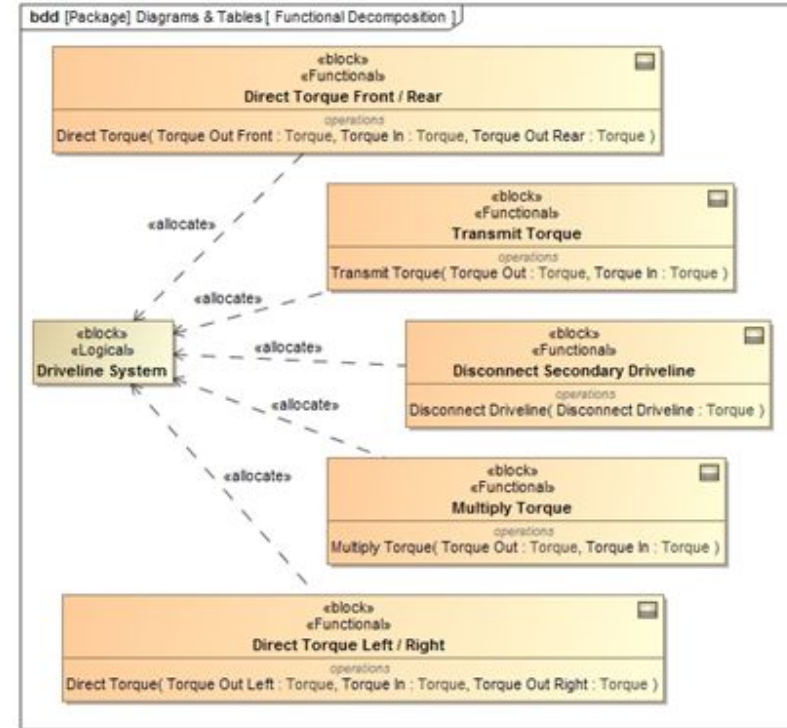


# Model Based System Engineering Concepts: *Language, Methods and Tools*

- MBSE and SysML were built directly upon the discipline of software engineering and architecture
- A single modeling language becomes the standardized medium for communications
  - A modeling method is “*a documented set of design tasks that a modeling team performs to create a system model*” (Delligatti 2003)
- The rules defined within the modeling language give unambiguous meaning to the model's elements and relationships
- The purpose defined will determine the level of detail required for defining the external environment and functional decomposition of the system
- Vehicle requirements and specifications were decomposed using the logical structure of the systems engineering V-model
- MagicDraw was chosen, but there are many commercial grade modeling tools available, in the same manner that there are many CAD or CAE tools

# Model Based System Engineering Concepts: *Functional Architecture*

- Functions define what actions / activities must be accomplished or completed to achieve a desired outcome
- In SysML language, we elected to represent functions as *operations*
- 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 sub-systems and components
- A block diagram (right) showing the functional decomposition of a driveline system, along with the sub-system through an *allocate* relationship
- *Allocate* relationship refers to the activity of associating a function with a structural element



# Model Based System Engineering Concepts: *System Requirements*

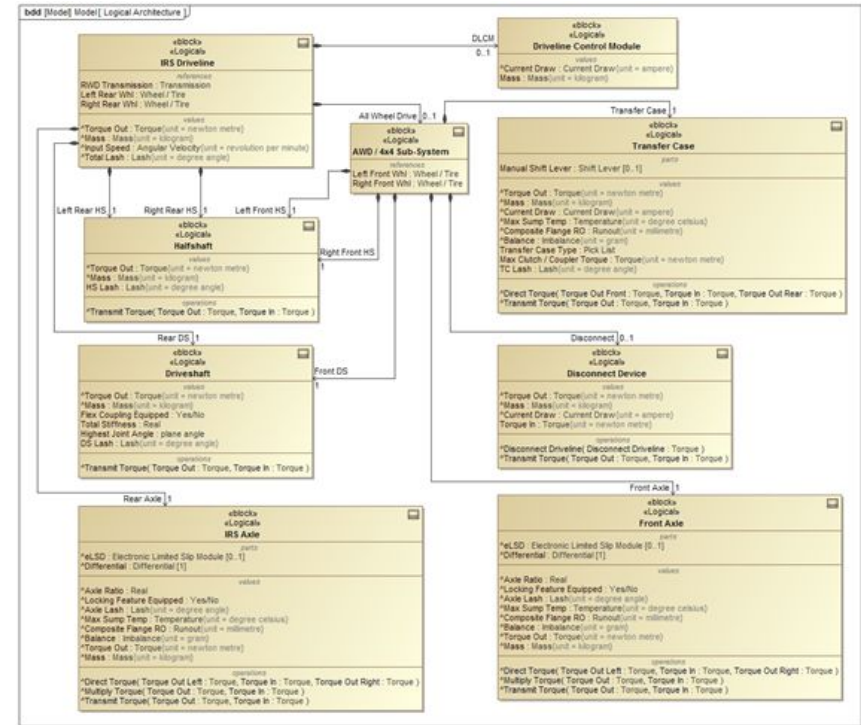
- Requirements define customer and stakeholder needs in technical terms
- In SysML, system requirement statements are defined as *objects*
- Each *object* contains the requirement text & a unique identifier
- The requirement *type* defines the features a requirement can be associated with
- *Generalizations* 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

Requirement Type	General Description	Example
Functional Requirement	Specifies a behavior of the system	Must transmit torque from transmission to wheels.
Performance Requirement	Specification, a quantifiable measurement of performance	Operational at vehicle top speed of 120 mph.
Interface Requirement	Specification for how system components connect	Must mount to transmission output flange PN FRZ102345.
Design Constraint	Design rule, or constraint on implementation	Threaded fasteners must use common metric threads and standard hex sizes.
Physical Requirement	Physical constraints on the system	Must fit within underbody package envelope.
Usability Requirement	Constraint on usage by physical actors	Must allow clearance for 95th% hand to access control lever.
Business Requirement	Constraint related to business processes	System must be back compatible with existing service axle lubricants.

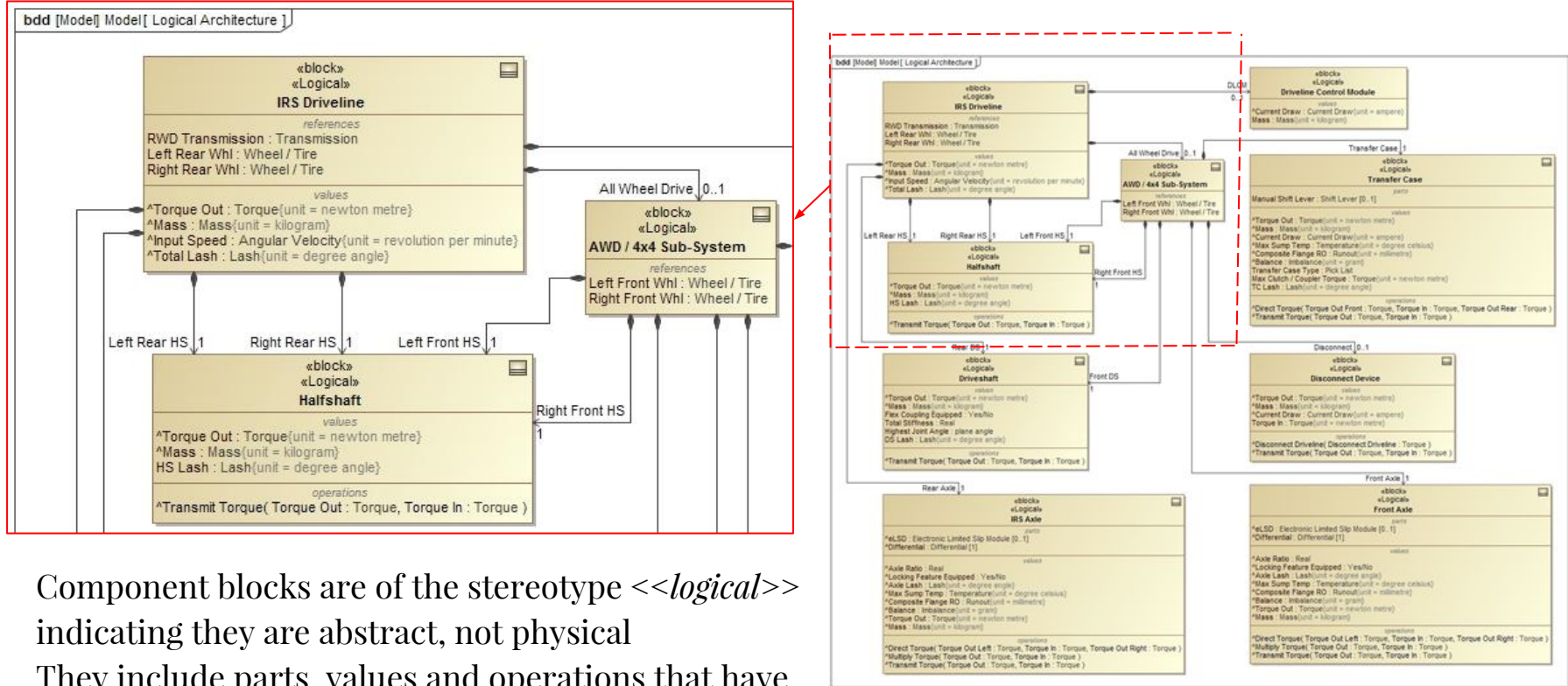
*Standard type requirements within SysML are used to provide rigor and clarity when defining the system*

# Model Based System Engineering Concepts: *Logical Architecture*

- 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*
- The logical architecture block definition diagram (right) of the IRS driveline system divides the top level system into appropriate sub-systems and components



# Model Based System Engineering Concepts: *Logical Architecture*



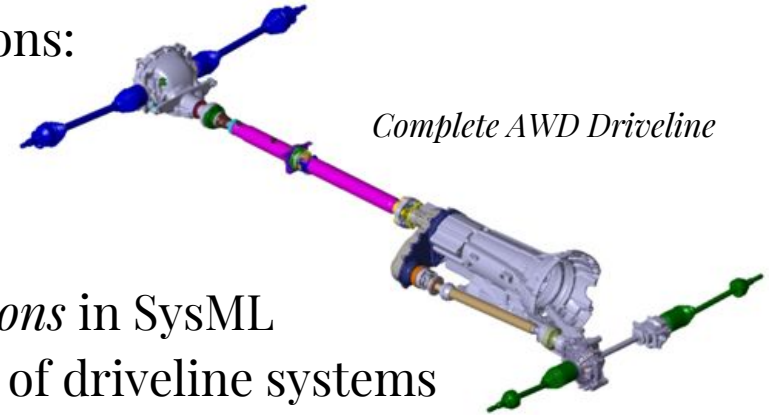
- Component blocks are of the stereotype <<logical>> indicating they are abstract, not physical
- They include parts, values and operations that have been defined or inherited from higher level element abstractions in the architecture

# Model Based System Engineering Concepts: *Physical Architecture*

- The physical architecture defines an actual instance of the real world product
- It represents an actual production part number used in the system BOM existing as CAD data
- It is the lowest level of abstraction in an MBSE implementation
- Actual part that can be visualized
- All of the component variable properties of the logical model become fixed physical properties in the physical model

# Driveline Definitions and Concepts: *Architecture*

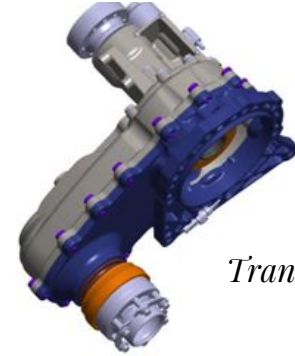
- An automobile driveline is the system of components linking the transmission output to the drive wheels
- The driveline's primary function is to transmit drive torque from the powertrain (engine and transmission) to the ground (wheels)
- Common driveline architecture configurations:
  - FWD (popular in most passenger cars)
  - RWD (sport and luxury cars)
  - AWD (for performance or upgrade)
  - 4x4 (common in trucks and SUV's)
- All above variants are treated as *generalizations* in SysML
- Generalizations represent specific subtypes of driveline systems
- Image shown (right) represents an AWD driveline architecture and the relationship and connections between various components



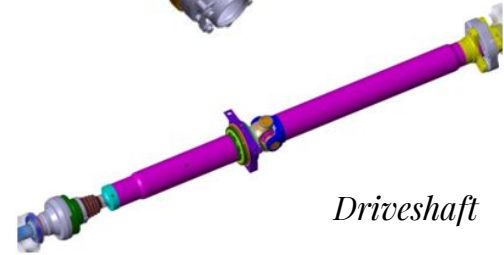


# Driveline Definitions and Concepts: *Components*

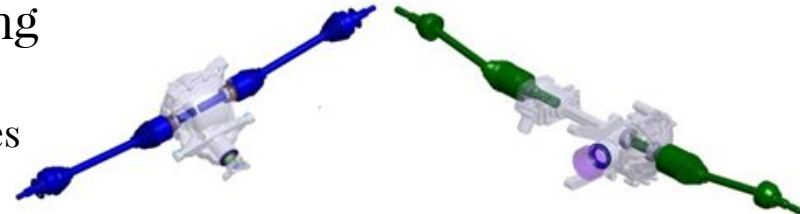
- *Transfer Case* – the primary function of a transfer case is to split the torque from the transmission and direct it to the front and rear wheels
  - Secondary function is to direct torque to the wheels with the most traction
- *Driveshaft* (front and rear) and *halfshaft* (left and right and front and rear) – primary function is to transmit torque and accommodate suspension and chassis flexing
  - Considered modular components
  - Manufactured as welded, tube shafts, swaged tubes
  - Typically designed as weakest component



*Transfer Case*



*Driveshaft*



*Halfshafts*

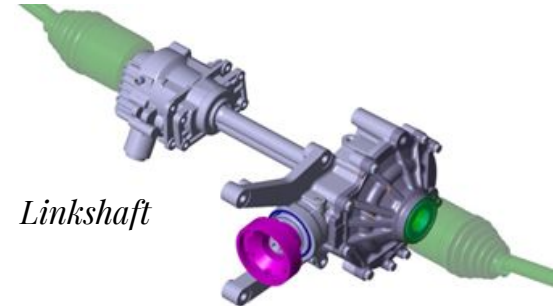


# Driveline Definitions and Concepts: *Components*

- *Axle* (front and rear) – function is to multiply driveshaft torque and direct it to the wheels
  - Torque multiplication is related to axle ratio
  - Both axles share same ratio
  - Front axle used in AWD and 4x4 driveline
- In FWD applications, a *PTU* replaces the function of the transfer case
  - In the case of our SysML model, the PTU is an instance of the generalization *Transfer Case*
- Final primary component in the AWD driveline system is the *disconnect device*
  - Purpose is to disconnect the secondary axle from the wheels
  - Linkshaft disconnect device shown (right)



*Axle - Rear*

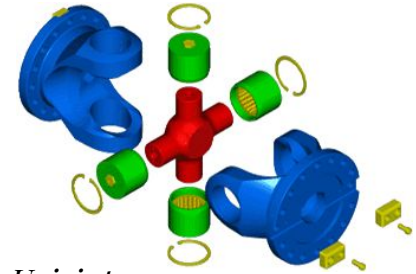


*Linkshaft*

*Axle - Front*

# Driveline Definitions and Concepts: *Joints*

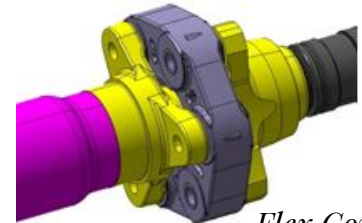
- Joint design is a primary consideration to help reduce NVH
- *U-joint* - commonly found in both front and rear driveshaft modular assemblies
  - Inherent design limitation - angular velocities of the components vary over a single rotation
- *CV-joint* - allows a shaft to transmit torque through a variable angle with a constant rotational speed
  - Eliminates the variation in angular velocity inherent in U-joints, which improves NVH, with a minimal increase in rotational friction
- *Flexible Coupler* - used to decouple NVH concerns between the transmission and driveshaft / axle by removing any rigid connection between them



*U-joint*



*CV-joint*



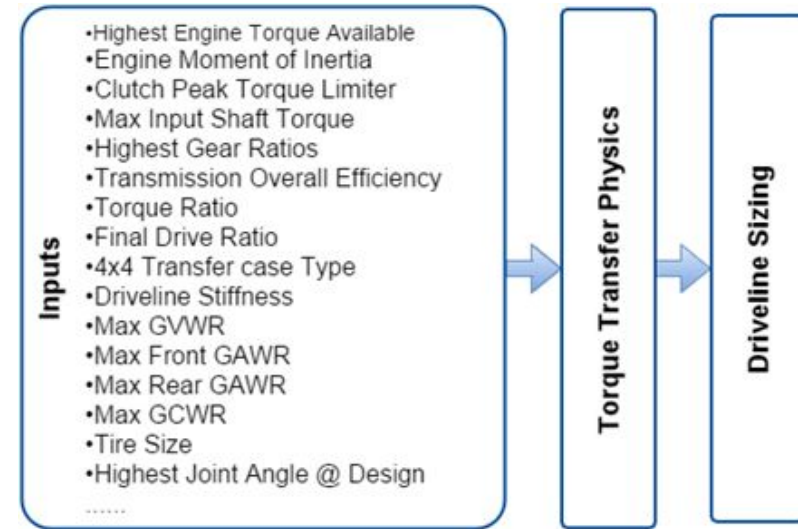
*Flex Coupler*

# Driveline Sizing: *Purpose*

- Design optimization of each component, system and sub-system is the primary objective of any driveline sizing activity
- A sizing tool takes the inputs from the vehicle program team and converts them into three basic outputs:
  - Impact torque
  - Fatigue torque
  - Yield torque
- The driveline must be sized to include all powertrain configurations for
  - Multiple engine sizes, outputs, type (diesel, hybrid, etc.)
  - Manual or automatic transmission
  - Drivetrain configurations (RWD, FWD, AWD)
  - Customer usage profiles (towing, final gear ratio, wheel / tire sizing)
  - Vehicle configurations (multiple wheelbases, track widths, cab height, suspension displ.)
- A useful driveline sizing tool must tabulate output for each required instance

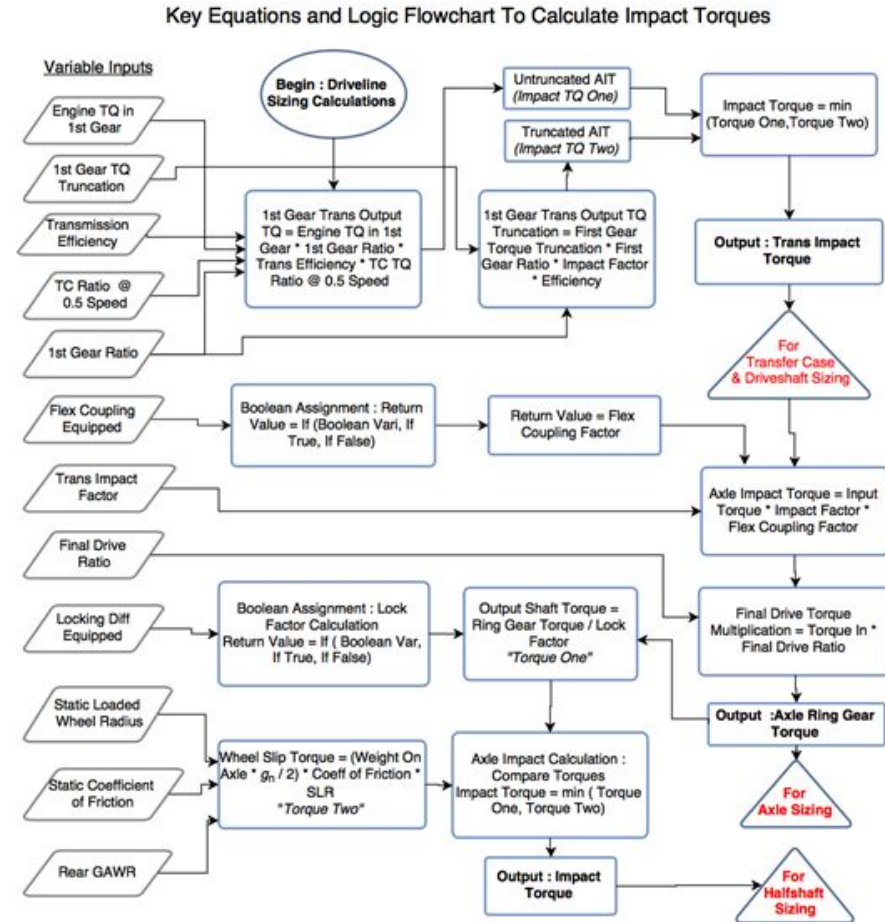
# Driveline Sizing: *Methodology*

- Sizing of driveline components is the alignment of physical design attributes to the demands of rotational inertia and transmitted torque to meet customer expectations.
- Driveline engineering traditionally relied on input from upstream suppliers to size each sub-system in series.
- Ford Motor Company and Dana Corporation developed in-house Microsoft Excel based driveline sizing calculation tools to organize, integrate and automate the required inputs, torque transfer calculations and outputs.
- A basic context diagram for driveline sizing and data flow through system is shown (right).



# Driveline Sizing: *Overview*

- Flowchart overview of the key inputs and logical calculations needed to generate the driveline impact torques (right)
  - Input data on the *left* side flows right
  - Physics calculations and logic decisions in the *middle*
  - Three critical torque outputs on *right* side are used to size specific sub-system components
    - Transmission torque: used for transfer case & driveshaft sizing
    - Axle ring gear torque: used for axle sizing
    - Axle output torque: used for halfshaft sizing



# Driveline Sizing: *Torque Transfer Physics*

- Driveline sizing begins with an understanding of the physics behind torque transfer through the system and its effect on the driveline components
- Engine torque passes through the transmission (clutch or converter) to the driveline system to the wheels, limited by wheel slip, which represents *max* torque transmitted
- Torque can be defined as *impact* or *continuous* through vehicle life
  - *Impact* torque is the maximum seen by the driveline and is related to inertia loading
  - *Continuous* torque is also known as fatigue torque and represents the average torque transmitted through the driveline during normal operation
- Calculations for driveline *impact* torques are:
  - Power Pack Torque (Tpp) = Engine Torque \* 1st Gear Ratio \* Transfer Efficiency \* Impact Factor
  - Truncated Torque (TT) = 1st Gear Torque Limit \* 1st Gear Ratio \* Impact Factor \* Efficiency Factor
  - Axle Ring Gear Torque (Trg) = Input Torque \* Final Drive Ratio \* Impact Factor \* Flex Coupling Factor
  - Wheel Slip Torque (Tws) = (Gross Vehicle Weight / 2) \* Gravity Force \* Surface Friction \* Tire Radius / (Final Drive Ratio \* Mechanical Efficiency)
- Similar calculations are used to determine *fatigue* torque and *yield* torque

# Driveline Sizing: *Required Inputs and Data*

- Minimum input data required to complete sizing is listed in the table (right)
- Determining and communicating the correct input data needed for the driveline sizing process is critical to ensure useful output
- Large amounts of supplemental data (acquired through testing, benchmarking, supplier input) is accumulated in a database and utilized for final component design optimization
- Component engineers are responsible for sizing and delivery of individual driveline components

Sizing Input	Typical	Notes
<b>Vehicle Assumptions</b>		
Driveline Configuration (Type)	4WD/4x4	What style of driveline is required?
Brake Traction Control Equipped	Yes	Brakes affect axle loads and tire slip.
Max GVWR	6000	How much can the vehicle weight?
Max Front GAWR	2800	How much load is on the front axle?
Max Rear GAWR	3200	How much load is on the rear axle?
Max GCWR	11000	How much can the vehicle tow?
Durability Road Load Data Set (RLD)	RLD	What is the durability requirement?
<b>Engine</b>		
Engine Torque in 1st Gear	280	Drag start torque.
Highest Engine Torque Available	400	Total max output torque.
Engine Torque in Rev. Gear	280	Max torque in reverse.
Engine Moment of Inertia	7.7	Inertia is related to impact load.
<b>Transmission</b>		
Transmission Type	Auto	Auto or manual transmission?
1st Gear Torque Truncation	350	Max torque out of transmission.
2nd Gear Torque Truncation	400	Max torque out of transmission.
Transmission Efficiency	0.93	Transmission losses.
Torque Converter Ratio @ Stall	2	Ratio determines start performance.
Torque Converter Ratio @ 0.5 Speed Ratio	1.6	Ratio determines start performance.
Transaxle FDR	3.7	Final drive ratio for FWD.
Flywheel Moment of Inertia - Manual Trans	6.8	Inertia is related to impact load.
Clutch Peak Torque Limiter - Manual Trans	100	Max torque out of transmission.
Number of Forward Gears	10	How many gears?
1st Gear Ratio	4.2	Torque multiplication.
2nd Gear Ratio	2.4	Torque multiplication.
Reverse Gear Ratio	3.4	Torque multiplication.
<b>Driveline</b>		
4x4 Transfer case type - Active/On Demand	Active	Determines torque bleed front / rear.
4x4 Transfer case Max Coupler Torque	1250	Max torque to front.
Flex Coupling Equipped	Yes	Attenuates impact.
Total Driveshaft Stiffness	600	Attenuates impact.
Highest Joint Angle @ Design	5	Joint sizing input.
Rear Axle Ratio	3.7	Final drive ratio for RWD.
Locking Differential Equipped	Yes	Increases halfshaft torque.
<b>Wheels / Tires</b>		
High Rotational Inertia Tire Equipped	Yes	Increases wheel slip impact.
Tire Size	265/55R18	Tire slip.
Max Tire SLR	13.75	Tire slip.
Tire Coefficient of Static Friction	0.92	Tire slip.



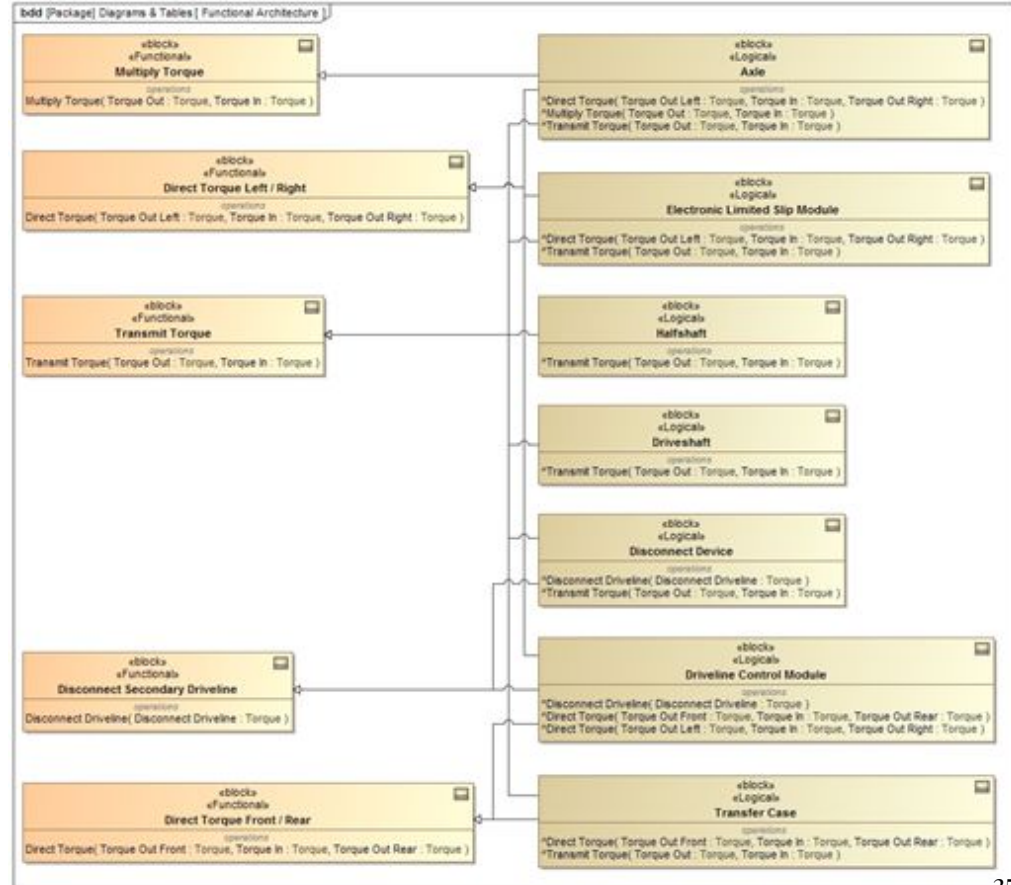
# Driveline Model Structure

- *“Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.”* (Operations 2007)
- According to the 2007 report of the International Council on Systems Engineering (INCOSE), model based systems engineering was very likely to replace the document-centric approach practiced by most systems engineers. By 2011 SysML was used by 20% of aircraft and defense companies and 7% of automotive manufacturers (Paredis and Davies 2011)



# 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

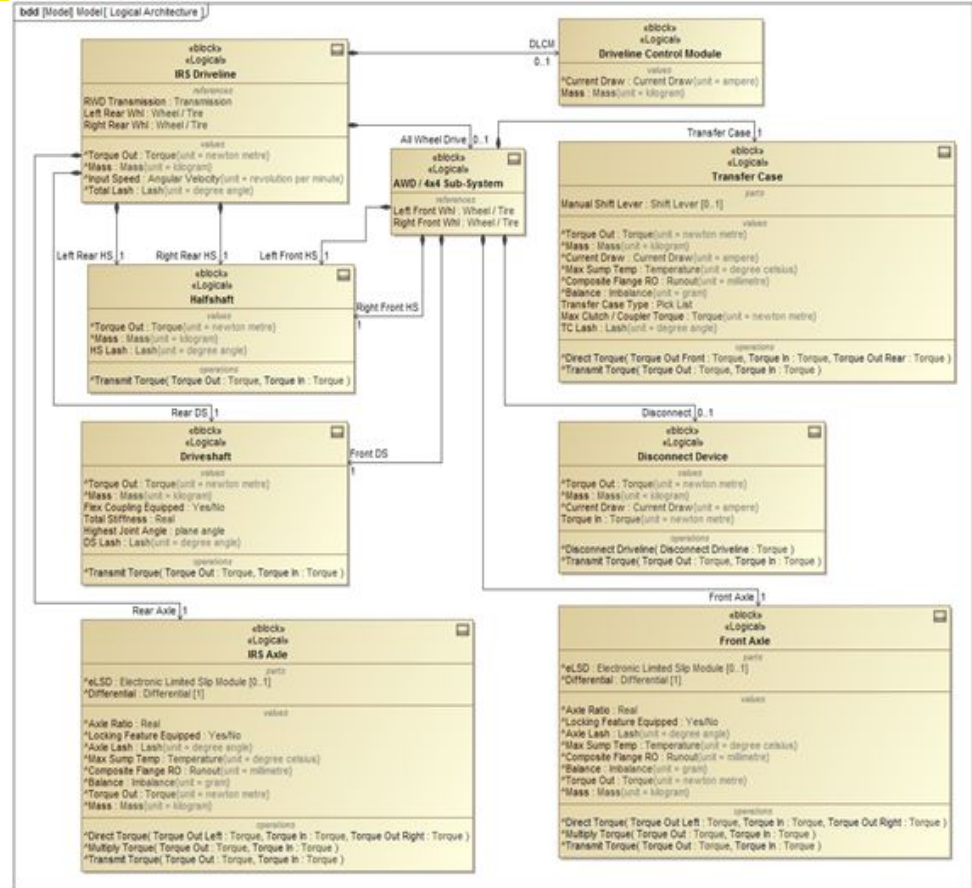


# Methods of Modeling: *Functional Decomposition*

- 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

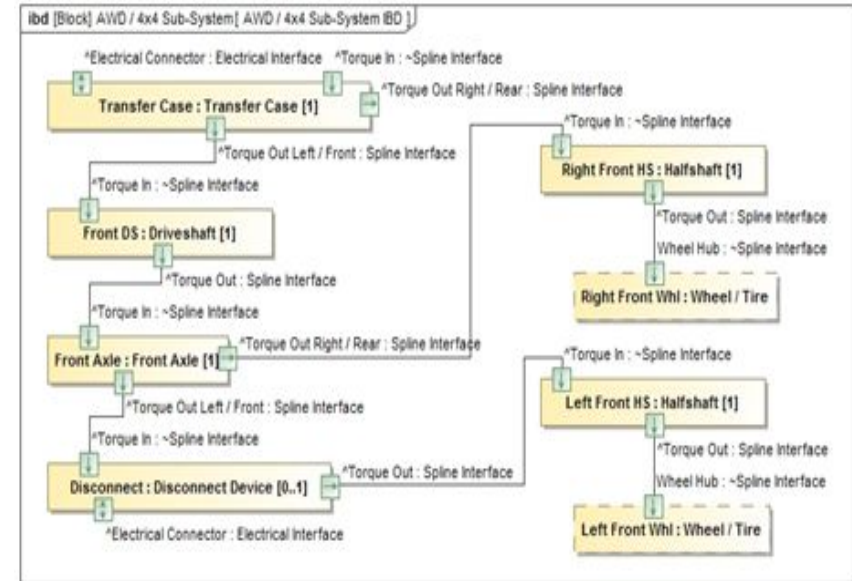
# Methods of Modeling: *Logical Decomposition*

- 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



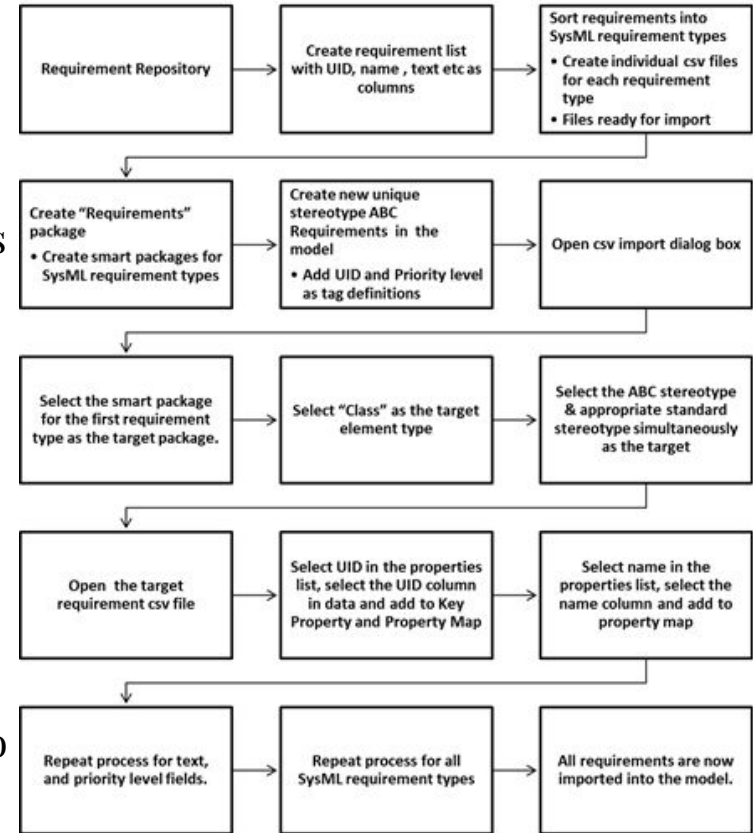
# Methods of Modeling: *Creation of internal relationships*

- Internal relationships are created in SysML through the use of internal block diagrams (IBD)
- An IBD captures the internal structure of a block in terms of properties and connections among properties
- A block includes properties, so that its values, parts, and references to other blocks can be specified
- An IBD created for a block (as a model inner element) will only display the inner elements of the block, such as *parts*, *ports* and *connectors*
- The outer boundary of the diagram is essentially an abstract representation of the outer boundary of the assembly



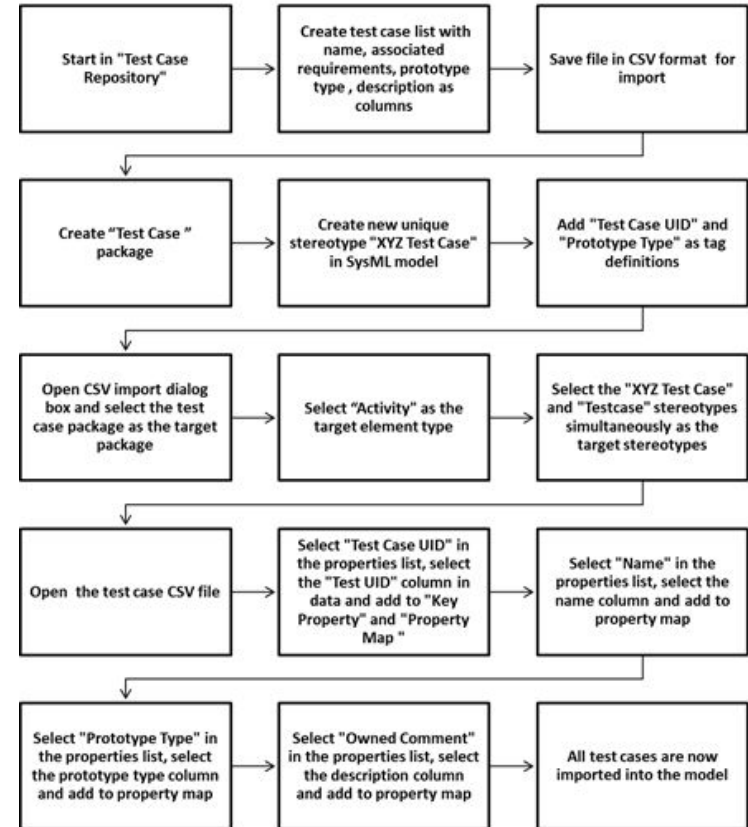
# Requirement Management: *Import Requirements*

- Over 500 requirement documents originally existed in the requirement repositories
- Engineer creates requirement list in .csv format with details such as UID, name, and text
- Sort requirements into SysML requirement types based on the relevant text description
- A new unique stereotype containing the Ford UID and priority levels as fields is required for the import process
- Import the requirements using the csv import feature in MagicDraw into a requirements package using “class” as the target element type
- Duplicates are removed in this process
- Total number of unique requirements reduced to just over 300



# Requirement Management: *Import/Create Test Cases*

- All requirements must have at least one associated test case, but many have more than one, leading to duplication in the document based system
- Create test case list from requirements list with name, test ID, prototype type, and description as the columns
- Create new stereotype with test case UID and prototype type as tag definitions
- Import the test cases into a test case package with “activity” as the target element type
- In total, 182 test cases were imported into the model



# Requirement Management: *Verification*

- The verification matrix defines the *verify relationships* from each test case to the appropriate requirements
- Over five hundred verify relationships were made in the process of verifying requirements
- Verify relationships were made with appropriate test case(s) verifying individual requirements
- Some requirements were verified by multiple test cases and some test cases verified multiple requirements, as is evident from the table

Requirement Type	Number of Unique Requirements	Total Number of Test Cases (For Each Reqnt. Type)
Functional Requirements	0	0
Design Constraint Requirements	284	471
Performance Requirements	15	36
Physical Requirements	4	5
Interface Requirements	0	0
Business Requirements	9	11
Usability Requirements	14	18



# Requirement Management: *Satisfy Relationships*

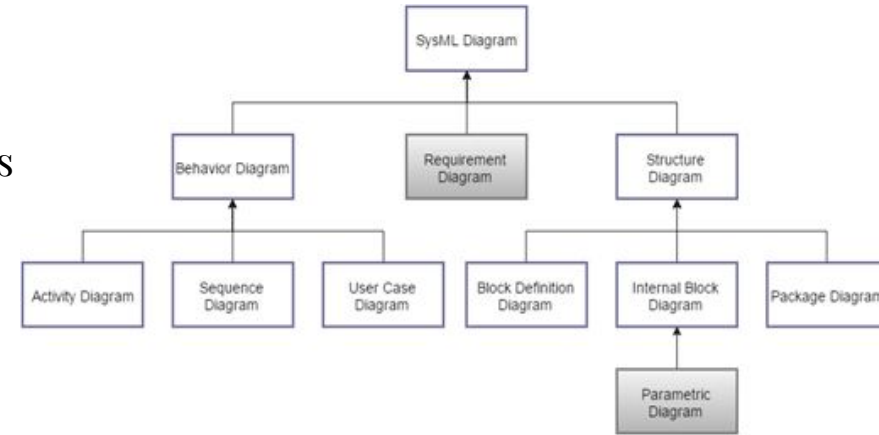
- Requirements are *satisfied* by physical elements
- In MBSE this relationship is created at the logical level of abstraction, prior to any physical parts being designed
- The satisfy matrix indicates that each requirement is mapped to at least one logical block through a satisfy relationship that indicates 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

Logical Elements	No. of Functional Reqmt. Satisfied	No. of Des Constraint Reqmt. Satisfied	No. of Perf. Reqmt. Satisfied	No. of Physical Reqmt. Satisfied	No. of Interface Reqmt. Satisfied	No. of Business Reqmt. Satisfied	No. of Usability Reqmt. Satisfied
AWD / 4x4 Sub-System		16		1		2	
AWD Sub-System		4		1		2	
Axle		54				2	2
Axle Lash/Value Properties: Lash			2				
Axle Ratio: Real							
Axle / T-Case Inheritance							
Balance/Value Properties: Imbalance			2				
Composite Flange RO/Value Properties: Runout			2				
Max Sump Temp/Value Properties: Temperature			4				
Beam Axle		3		1			
Controls Inheritance Block		1					
Coupling		3					
Disconnect Device		4					
Driveline Control Module		1					
Driveline System		14					1
Total Lash/Value Properties: Lash			1				
Driveshaft		109				1	5
DS Lash/Value Properties: Lash			1				
Electronic Limited Slip		1					
Electronic Locking		1					
Front Axle		11				1	2
Halfshaft		21					1
HS Lash/Value Properties: Lash			1				
IRS Driveline		1				1	
Mechanical Limited Slip		2					
PTU		17		1		1	
PTU Lash/Value Properties: Lash			1				
Shift Lever		5					2
Transfer Case		20		1		1	1
TC Lash/Value Properties: Lash			1				
Transmission		7					
Wheel / Tire		2					



# 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.



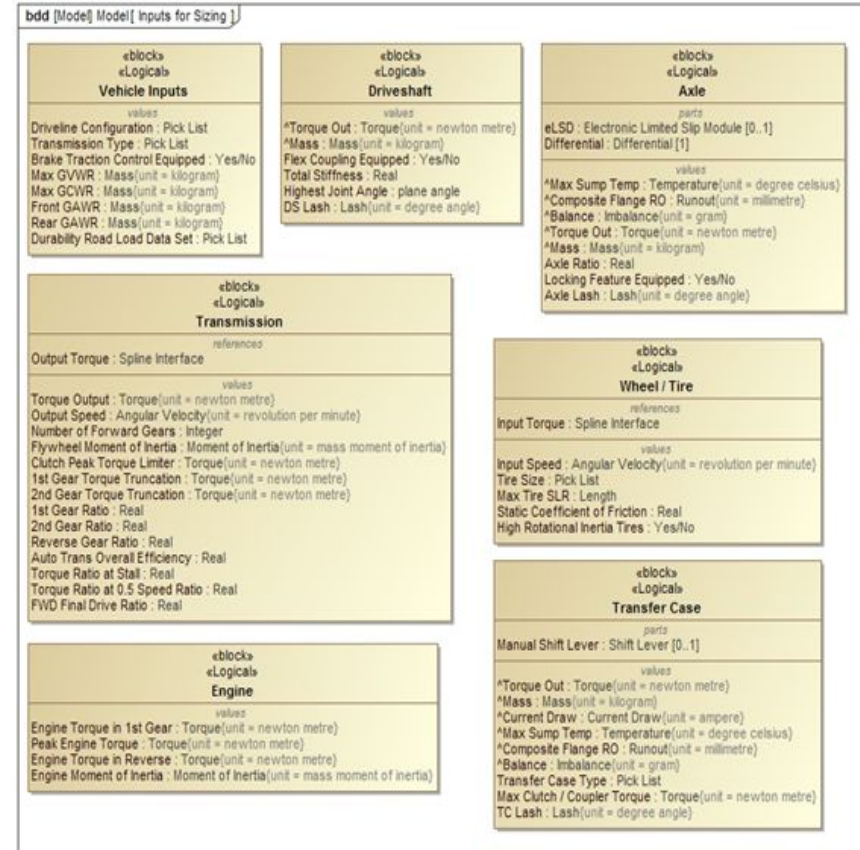
# Parametric Relationships: *Sizing Inputs*

- Existing driveline sizing tools are currently used to obtain impact torque , fatigue torque and yield torque (sizing torques)
- SysML model performs the same basic functions if properly constructed
- Important to define sizing inputs as value properties in the model
- These value properties must be associated and linked to the correct logical constructions at the sub-system and sub-component level
- The inputs must be associated or related to logical constructions within the SysML model

Parametric Input	Logical Block Ownership	Value Type
<b>Vehicle Assumptions</b>		
Driveline Configuration (Type)	Vehicle Inputs	Pick List
Brake Traction Control Equipped	Vehicle Inputs	Yes / No
Max GVWR	Vehicle Inputs	Kg
Max Front GAWR	Vehicle Inputs	Kg
Max Rear GAWR	Vehicle Inputs	Kg
Max GCWR	Vehicle Inputs	Kg
Durability Road Load Data Set (RLD)	Vehicle Inputs	Pick List
<b>Engine</b>		
Engine Torque in 1st Gear	Engine	N-m
Highest Engine Torque Available	Engine	N-m
Engine Torque in Rev. Gear	Engine	N-m
Engine Moment of Inertia	Engine	MMOI
<b>Transmission</b>		
Transmission Type	Transmission	Pick List
1st Gear Torque Truncation	Transmission	N-m
2nd Gear Torque Truncation	Transmission	N-m
Transmission Efficiency	Transmission	Real #
Torque Converter Ratio @ Stall	Transmission	Real #
Torque Converter Ratio @ 0.5 Speed Ratio	Transmission	Real #
Transaxle FDR	Transmission	Real #
Flywheel Moment of Inertia - Manual Trans	Transmission	MMOI
Clutch Peak Torque Limiter - Manual Trans	Transmission	Real #
Number of Forward Gears	Transmission	Integer
1st Gear Ratio	Transmission	Real #
2nd Gear Ratio	Transmission	Real #
Reverse Gear Ratio	Transmission	Real #
<b>Driveline</b>		
4x4 Transfer case type – Active/On Demand	Transfer Case	Pick List
4x4 Transfer case Max Coupler Torque	Transfer Case	N-m
Flex Coupling Equipped	Driveshaft	Yes / No
Total Driveshaft Stiffness	Driveshaft	N-m/rad
Highest Joint Angle @ Design	Driveshaft	Degrees
Rear Axle Ratio	Axle	Real #
Locking Differential Equipped	Axle	Yes / No
<b>Wheels / Tires</b>		
High Rotational Inertia Tire Equipped	Wheel/Tire	Yes / No
Tire Size	Wheel/Tire	Pick List
Max Tire SLR	Wheel/Tire	Pick List
Tire Coefficient of Static Friction	Wheel/Tire	Real #

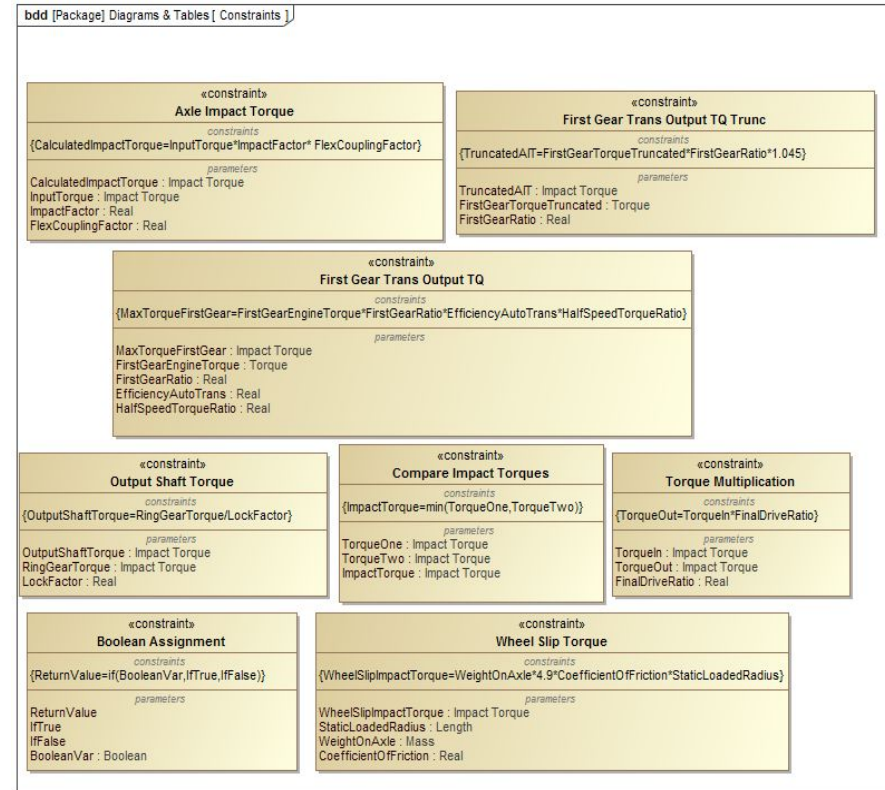
# Parametric Relationships: *Sizing Inputs*

- 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 if and when required



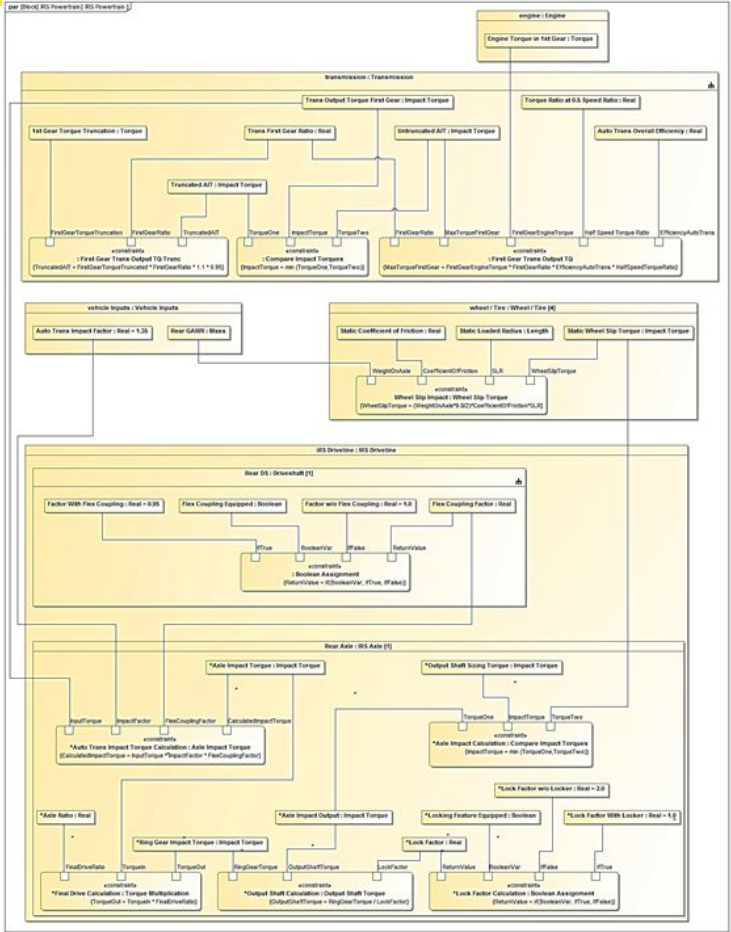
# Parametric Relationships: *Constraint modeling*

- 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  $\{\tau = F * d\}$  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 was developed for and has traditionally been applied to software applications, but this project demonstrates that it can be equally applicable to a fully mechanical system
- Improvements in at least three areas compared to document based approach:
  - Reduction in requirement redundancy and consistency across all levels of abstraction
  - Streamlined communication of requirements by making all key input and output parameters available to all model users
  - The ability to continuously update and manage component design inputs through parametric relationships with vehicle level inputs
- The driveline system model was completed in just nine weeks by three powertrain engineers with almost no previous SysML or MagicDraw modeling experience
- The MagicDraw application is reasonably intuitive and the driveline system model's complexity and scope increased as the team's familiarity and skill level grew

# Benefits of Applied MBSE: *Improved Communication*

- Implementation of MBSE will improve engineering productivity by linking requirements across all sub-systems and improving tracking and communication of requirement changes
- The reduction of requirement redundancies and automatic validation of test case verification could result in the elimination of entire tracking departments
- The system model is more flexible and scalable than the document based approach
- 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
- Sizing tools will require extensive tear up or new ones are made when new technologies are introduced leading to confusion and miscommunication in the document based approach, whereas with the system model, it could be as easy as a couple of new logical blocks and constraint blocks



# Benefits of Applied MBSE: *Management of Requirements*

- MBSE supports requirement management through reduced redundancy, better traceability and linkages between requirements and their methods of verification
- Requirements are a key element in the success of a system and its development and are considered by some authors to be “the cornerstone of systems engineering” (Salado and Nilchiani 2014)
- We emphasized careful categorization of existing requirements and during the import process eliminated redundancies. Through our integration efforts into SysML categories, we discovered a clear need and benefit of improved elicitation and partitioning of existing requirements
- Reduced the 500+ requirement documents to 300+ requirements in the model
- 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
  - Changes in tire properties can be linked to and compared against halfshaft joint design limits automatically
- If the input assumptions exceed design limits the SysML model will immediately throw an error code to alert the system and component engineers that their attention is required
- All design data can be stored and managed in one place at an abstract level

# Benefits of Applied MBSE: *Conclusion*

- First known application of MBSE at Ford Motor Company
- Proof of concept successfully completed with verified results
- Significant tool for automotive system design that offers many benefits, especially in communication through a common language
- Requirement traceability is vastly improved
- Faster delivery time along with more robust results and less cost
- Implementation of MBSE may be similar to CAD application growth in the 1980's
- Emerging market that is likely to grow

