

# Requirements on (1) Structure (2) Properties, Values and Expressions

SysML v2 RFP Working Group

Hans Peter de Koning (European Space Agency)

# Structure Concepts Modeling Focus Team

- Manas Bajaj - Intercax
- Conrad Bock - NIST
- Roger Burkhart - John Deere
- Andy Ko – Phoenix Integration
- Hans-Peter de Koning - European Space Agency
- Sandy Friedenthal
- Chas Gayley
- Andrew Mullis - Lockheed Martin
- Marc Sarrel – NASA/JPL
- John Watson

Wiki page

[http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:structure\\_behavior\\_concepts\\_modeling\\_core\\_team\\_wiki\\_page](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:structure_behavior_concepts_modeling_core_team_wiki_page)

# Property & Expression Concepts Modeling Focus Team

- Manas Bajaj - InterCAX
- Conrad Bock - NIST
- Roger Burkhart - John Deere
- Hans Peter de Koning - European Space Agency
- Harald Eisenmann - Airbus Defence & Space
- Nerijus Jankevicius - No Magic
- Ilya Tolchinsky - Phoenix Integration

Wiki page

[http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:property\\_modeling\\_core\\_team](http://www.omgwiki.org/OMGSysML/doku.php?id=sysml-roadmap:property_modeling_core_team)

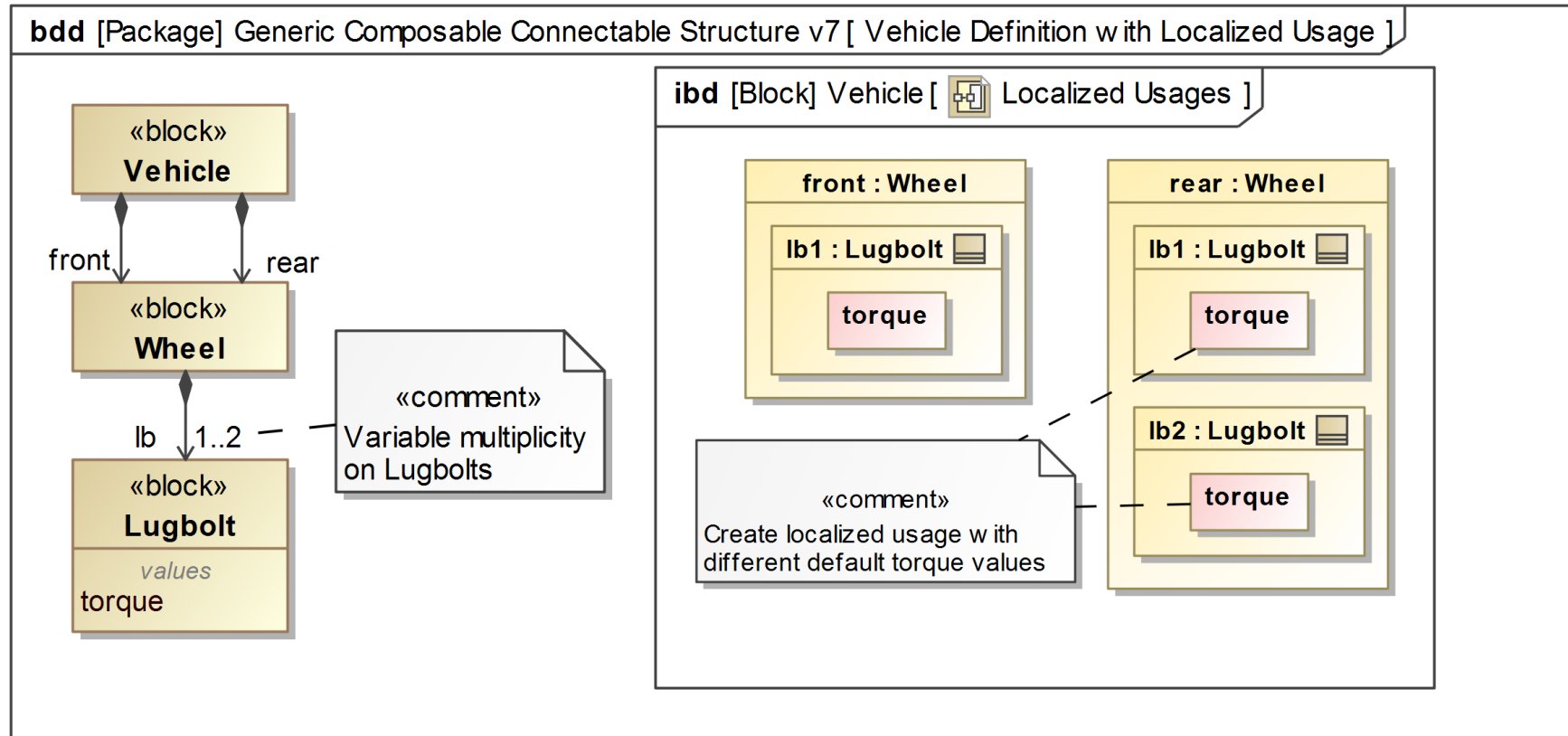
# Structure – Main Needs

- Overcome SysML v1 limitations
  - Different ways to represent composition structure and connectivity for blocks, activities, state machines, etc.
  - Deeply nested structure aspects: possible, but cumbersome and not complete
  - Local override / redefinition of features / properties: possible, but cumbersome
  - Clear distinction between models of as-designed vs as-built systems
  - Double bookkeeping for connector property and adjunct property
  - Better accommodation of variability concepts

(Full list on Structure Concepts wiki page)
- Ambition: A single generic pattern for hierarchical (de)composition that supports integrated typing and connectivity
  - Is basis for:  
blocks, parts, references, ports, interface connectors, (item) flows, flow properties, activities, control nodes, object nodes, actions, pins, ..., interactions, state machines, use cases, constraint blocks & properties, requirements, ...
  - Supports evolving model along the lifecycle

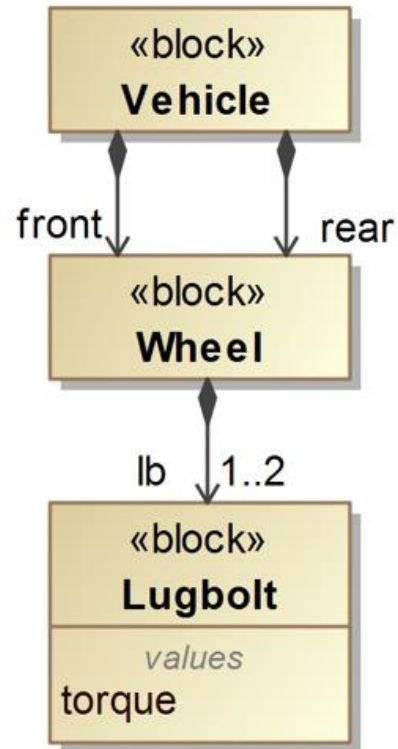
# Example Block / Part Structure to be Supported

- Transition to specific design configurations with localized value overrides



# Foundational Concepts: Definition & Usage

(SysML v1.5 concepts)

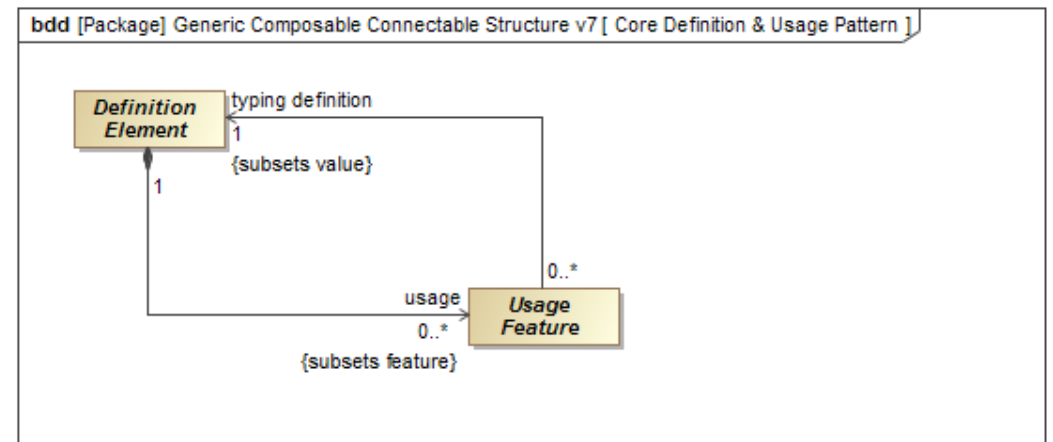
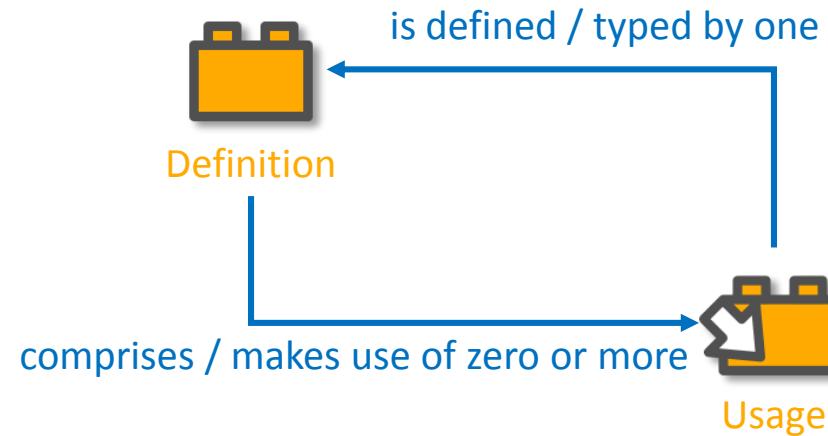


Two usages of Wheel, named front, rear (part properties of Vehicle)

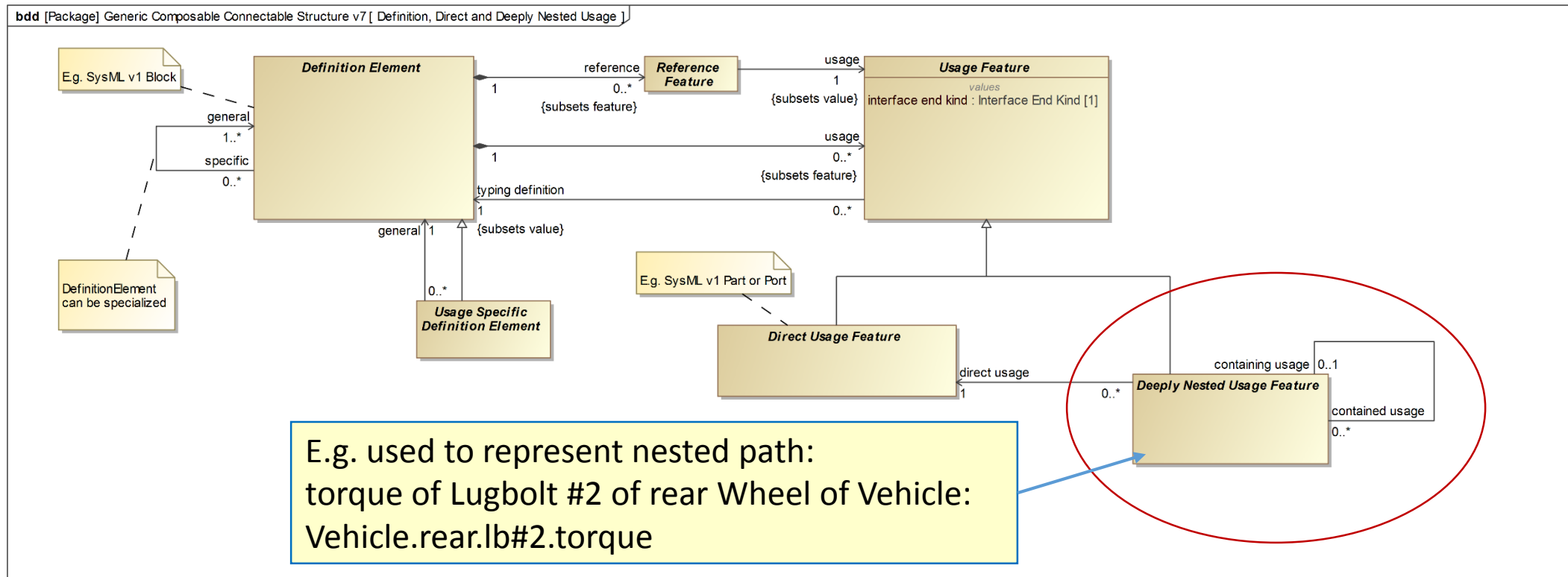
One or two usages of Lugbolt, named lb (part property of Wheel)

Need for index notation to distinguish?  
lb[1], lb[2]? lb\_1, lb\_2? lb#1, lb#2?

Can specify default value for torque at Definition level, and override value at direct or deeply nested usage level



# Adding Unambiguous Deeply Nested Usage in SysML v2



## Disclaimer:

Any SysML v2 RFP Concept Model should be regarded as just ideas on possible solution directions.

By no means these concepts are prescriptive toward future submitters, but rather used at the RFP preparation stage to validate whether the RFP requirements make sense and can be satisfied in principle.

# Definition vs Configuration vs Individual Models

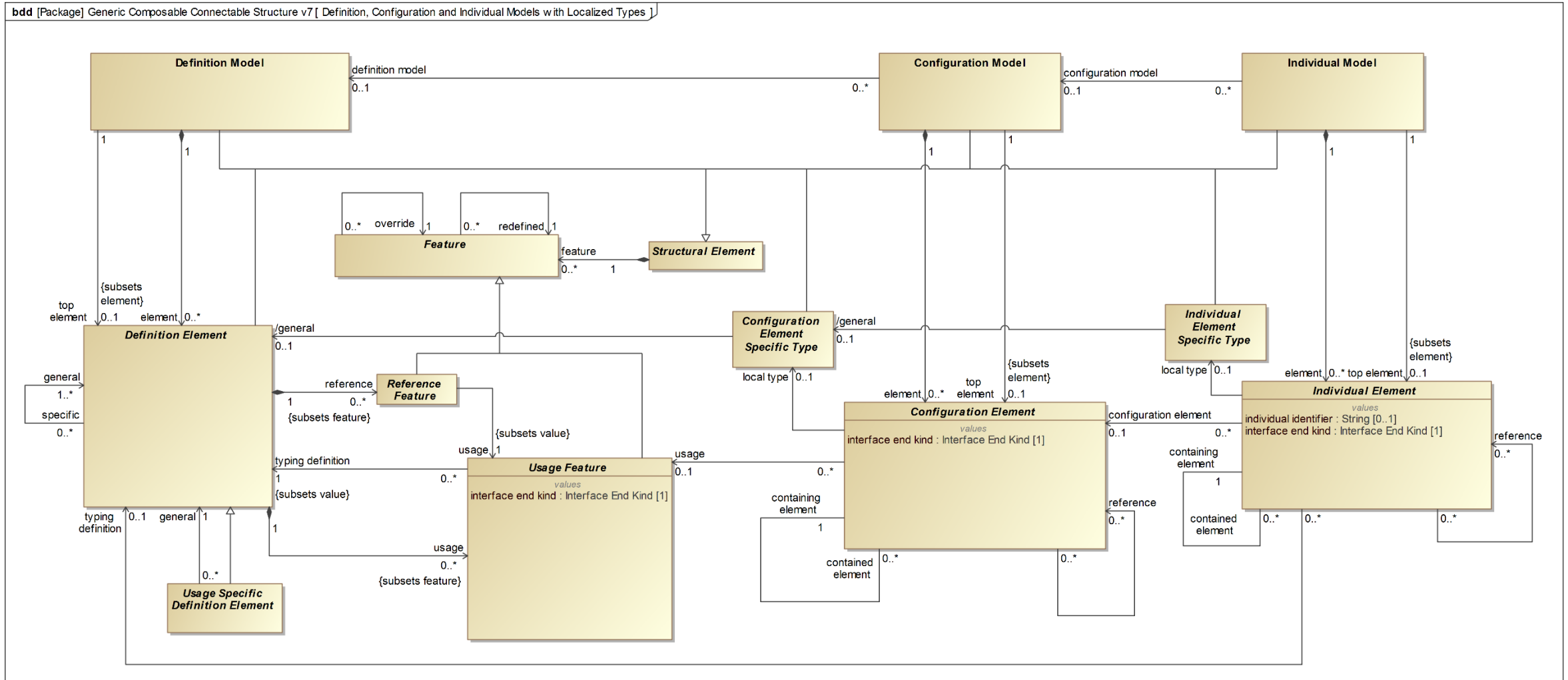
## ... from more abstract to more concrete

- The Definition – Usage pattern that defines the system-of-interest with variation points (e.g. multiplicity, specialization and value overrides) with **deeply nested usage support**  
→ proposed name “**Definition Model**”
- However, for analysis, verification and build we need the **explicit, fully expanded model of a single resolved variant**  
→ proposed name “**Configuration Model**”
  - A default Configuration Model can be auto-generated from a Definition Model after choices have been made for all variation points
- In addition we need a model of the (potential) real-world thing  
→ proposed name “**Individual Model**”
  - Representing e.g. serial-numbered items
  - Also sometimes referred to as the “Digital Twin”

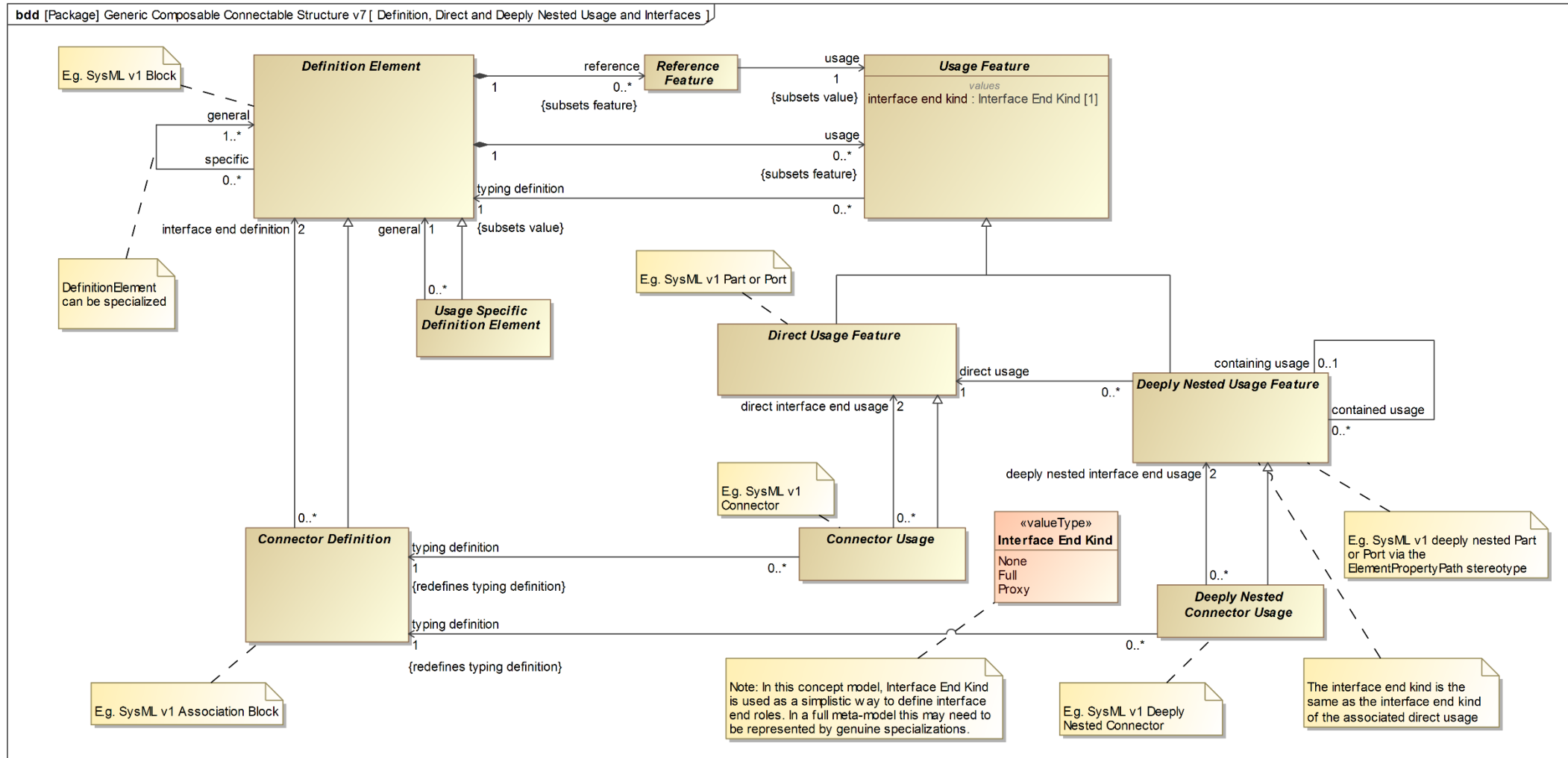




# Unification of Generic Structural Element with Features and the three kinds of Model ... and Localized Types

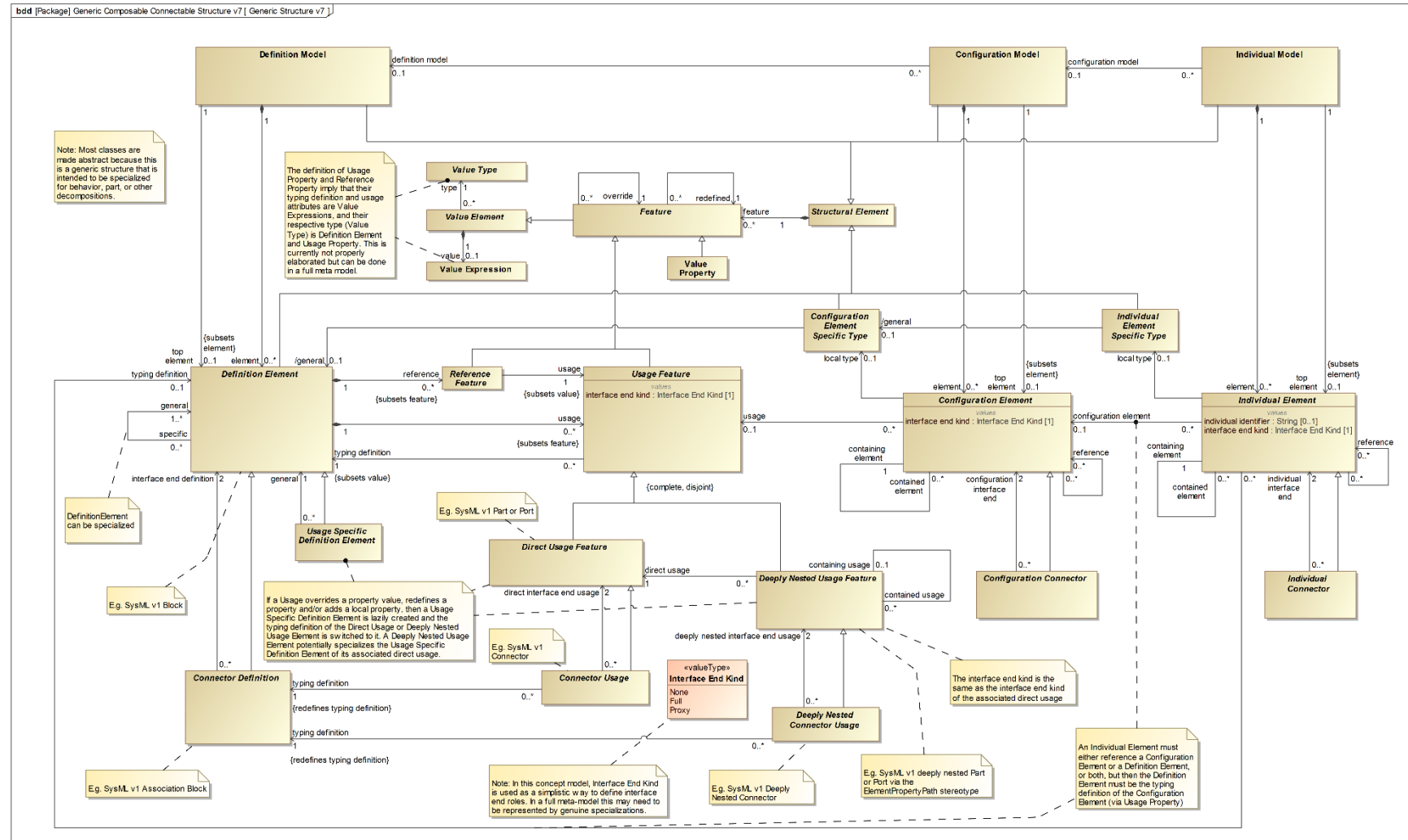


# ... and basic support for Interfaces as well



# Putting it All Together ... Integrated Concept Model for Generic Composable Connectable Structure

- Definition Element can be specialized into: block, activity, ...
- Feature can be specialized into: value property, part, reference, I/F end, I/F connector, action, ...
- Integrated Interface End and Connector to represent any connections
- Representation of modular definition with variability, explicit single variant, individual product
- Localized subtyping at any level



# Structure Requirements

## – details on wiki page

A	B	C	D	E	F	G	H	I	J
Id	Name	Text	Supporting Info	Req Group Owner	Req Owner	Req Status	SysML 1.x	SysML 1.x Construct	SysML 1.x Issue
STR-1	Structure Requirements Group	Generic composable, deeply nested, connectable structure with and without variants.	Refer to Vehicle Definition Example for illustration of specific localized type.	.Structure		Proposed			
STR-1.1	Modular Unit of Structure	SysML v2 shall include a capability to represent a modular unit of structure - called a Definition Element - that defines its characteristics through value properties, interface ends (ports) and connectors, constraints, and behavioral features.			.Structure	Proposed	Partial	Block, Activity, ConstraintBlock, ...	
STR-1.2	Usage Feature	SysML v2 shall include a capability to represent a usage of any Definition Element - called a Usage Feature - by another Definition Element in order to support the definition of modular, deeply nested hierarchical composition structures.			.Structure	Proposed	Partial	Part property, Port property, CallAction, ConstraintProperty, ElementPropertyPath, NestedConnectorEnd, ...	Largely limited to part and port properties. Complicated and cumbersome representation of deeply nested structure.
STR-1.3	Generic Hierarchical Structure	SysML v2 shall include a capability to represent hierarchical composition structure in a generic such that it can be re-used (i.e. specialized) for any set of concepts that have hierarcharchical composition traits.			.Structure	Proposed	No		Different hierarchical composition patterns are used for structure, behavior, constraints, ...
STR-1.4	Usage Reference	SysML v2 shall include a capability to represent a reference from a Definition Element to any usage of another Definition Element in the subtree of the referencing element.			.Structure	Proposed	Yes	reference property	
STR-1.5	Multiplicity of Usage	SysML v2 shall include a capability to define the multiplicity of any particular Usage Feature or Usage Reference, either as a single value or a value range.			.Structure	Proposed	Yes	Multiplicity on properties	
STR-1.6	Definition Element Specialization	SysML v2 shall include a capability to represent a specialization from a more general Definition Element into a more specific Definition Element, where the more specific element inherits all features of the more general element.			.Structure	Proposed	Yes	All classifiers: Block, Activity, ConstraintBlock, ...	
STR-1.7	Unambiguous Deeply Nested Structure	SysML v2 shall support a capability to represent and unambiguous identify deeply nested Usage Features in a way that is fully integrated with direct (one level deep) Usage Features.	Deeply nested Usage Features may be lazily instantiated, i.e. only when needed for specific localized typing or interface representation.		.Structure	Proposed	Partial	ElementPropertyPath, NestedConnectorEnd	
STR-1.8	Structure With Variation	SysML v2 shall include a capability to represent multiple possible variants of a system-of-interest through a single			.Structure	Proposed	Partial	Multiplicity of properties, specialization of classifiers	

# Structure Requirements – Improvements w.r.t. SysML v1

Id	Name	SysML 1.x	SysML 1.x Construct	SysML 1.x Issue
STR-1.1	Modular Unit of Structure	Partial	Block, Activity, ConstraintBlock, ...	
STR-1.2	Usage Feature	Partial	Part property, Port property, CallAction, ConstraintProperty, ElementPropertyPath, NestedConnectorEnd, ...	Largely limited to part and port properties. Complicated and cumbersome representation of deeply nested structure.
STR-1.3	Generic Hierarchical Structure	No		Different hierarchical composition patterns are used for structure, behavior, constraints, ...
STR-1.7	Unambiguous Deeply Nested Structure	Partial	ElementPropertyPath, NestedConnectorEnd	
STR-1.8	Structure With Variation	Partial	Multiplicity of properties, specialization of classifiers	
STR-1.9	Structure Resolved to Single Variant	No		
STR-1.10	Structure of an Individual	No		
STR-1.11	Usage Specific Localized Type	Partial	PropertySpecificType	Possible, but complicated and cumbersome.
STR-1.12	Interface Ends	Partial	Port, Pin, ...	However not unified between structure and behavior. Deeply nested ends cumbersome to use.
STR-1.13	Interface Connectors	Partial	Connector	However not unified between structure and behavior. Deeply nested connectors cumbersome to use.
STR-1.14	Interface Typing	Partial	AssociationBlock	
STR-1.15	Logical Expressions for Variation Points	No		

# Properties, Values and Expressions

- Overcome SysML v1 limitations
  - No variable length value collections: sequence, set, ordered set, bag
  - Too simplistic support for measurement scales other than ratio scale (in QUDV)
  - The inability to easily restrict a valid range of values for a value property
  - Complex meta-model (including QUDV) leads to implementation inefficiencies for tool vendors and usability issues for end-users
  - A (numerical) value type defines (and fixes) a selected measurement unit. Rather for a given value property the actual measurement scale should be selectable from a set of permissible measurement scales.
- SysML v1 has limited support for:
  - Compound value properties: e.g. computer data record, vector, matrix, nth-order tensor, array, quaternion, ...
  - Automated scale conversion of numerical property values to support model integration from multiple sources
  - Tabular (discretely sampled) data like time series, frequency spectra, temperature (pressure, enthalpy, ...) dependent material properties, etc.
  - Uncertainties and probability distributions / density functions

# Properties, Values & Expressions Requirements

## – details on wiki page

Id	Name	Text	Supporting Info	Req Group Owner	Req Owner	Req Status	SysML 1.x	SysML 1.x Construct	SysML 1.x Issue
PVE-1	Properties, Values and Expressions Group	Unified support to represent numerical and non-numerical variables , properties, constants, value expressions, including variable size collections, compound value types, measurement units and scales.		.Properties		Proposed			
PVE-1.1	Unified representation of values	SysML v2 shall include a capability to represent the common aspects of a value property of a model element, a constant, a variable in an expression as well as a formal parameter and the return type of an operation in a unified way. The set of such model concepts is called Value Element.	In order to facilitate further specification a thi...		.Properties	Proposed	Partial	ValueType, QuantityKind, Unit, Annex E.5 QUDV	
PVE-1.2	Value Type	SysML v2 shall include a capability to represent a named definition of the essential semantics and structure of the set of possible values of an observable characteristic, without the value itself. Such a definition is called a Value Type.			.Properties	Proposed	Partial		
PVE-1.3	Value Expression	SysML v2 shall include a capability to represent a value expression that can be evaluated to yield a value, where the expression language is defined in a platform independent formalism, and one or more mappings to a platform specific concrete syntax and grammar.	It is probably wise and practical to select an existing modeling language with strong expression capabilities like Modelica verbatim or as guidance, rather than to develop a full expression language from scratch.		.Properties	Proposed	Partial	Opaque or OCL value expression	
PVE-1.4	Unification of Expression and Constraint Definition	SysML v2 shall include a capability to represent a constraint through definition of an equality or inequality where the left and right hand sides are expressed in the same expression language as the Value Expression language.			.Properties	Proposed	Partial	Opaque or OCL value expression	
PVE-1.5	Intended Use of Value	SysML v2 shall include a capability to distinguish between the intended use of a value, as a minimum between default value, boundary condition value and initial value.			.Properties	Proposed	No		
PVE-1.6	System of Quantities	SysML v2 shall include a capability to represent a named system of quantities and physical dimensions that supports definition of numerical Value Types in accordance with formal or de facto standards.			.Properties	Proposed	Yes	Annex E.5 QUDV	
PVE-1.7	System of Units and Scales	SysML v2 shall include a capability to represent a named system of measurement units and scales to define the precise semantics of numerical Value Types.			.Properties	Proposed	Partial	Annex E.5 QUDV	Incomplete support for measurement scales. Cumbersome definition and use of QUDV model libraries.
PVE-1.8	Range Restriction Numerical Values	SysML v2 shall include a capability to represent a value range restriction for any numerical Value Type.	This can potentially be combined with lower and or upper bounds on an associated measurement scale.		.Properties	Proposed	No		
PVE-1.9	Mapping to SysML v1 QUDV	SysML v2 shall include a capability to represent a mappable superset of all SysML v1 concepts concerning valueType, Unit, QuantityKind and QUDV.			.Properties	Proposed	Yes	Annex E.5 QUDV	Incomplete support for measurement scales. Cumbersome definition and use of QUDV model libraries.
PVE-1.10	Automated quantity value conversion	SysML v2 shall include a capability to represent all information necessary to perform automated conversion of the value of a quantity expressed on one measurement scale to the value expressed on another, compatible measurement scale in order to support model integration from multiple sources.			.Properties	Proposed	Partial	Annex E.5 QUDV	Possible for ratio scales. Incomplete w.r.t. all possible kinds of measurement scale.
PVE-1.11	Computer data types	SysML v2 shall include a capability to represent the following computer data types as a minimum: signed and unsigned integer, signed and unsigned double precision real, string, boolean, enumeration type, ISO 8601 date and time, selected universally unique identifier (UUID).			.Properties	Proposed	Partial	Primitive ValueType	ISO 8601 date and time missing
PVE-1.12	Variable Length	SysML v2 shall include a capability to represent variable length value collections where			.Properties	Proposed	No		



# Properties, Values & Expressions Requirements – Improvements w.r.t. SysML v1

Id	Name	SysML 1.x	SysML 1.x Construct	SysML 1.x Issue
PVE-1.1	Unified representation of values	Partial	ValueType, QuantityKind, Unit, Annex E.5 QUDV	
PVE-1.2	Value Type	Partial		
PVE-1.3	Value Expression	Partial	Opaque or OCL value expression	
PVE-1.4	Unification of Expression and Constraint Definition	Partial	Opaque or OCL value expression	
PVE-1.5	Intended Use of Value	No		
PVE-1.7	System of Units and Scales	Partial	Annex E.5 QUDV	Incomplete support for measurement scales. Cumbersome definition and use of QUDV model libraries.
PVE-1.8	Range Restriction Numerical Values	No		
PVE-1.10	Automated quantity value conversion	Partial	Annex E.5 QUDV	Possible for ratio scales. Incomplete w.r.t. all possible kinds of measurement scale.
PVE-1.11	Computer data types	Partial	Primitive ValueType	ISO 8601 date and time missing
PVE-1.12	Variable Length Collection Value Type	No		
PVE-1.13	Compound Value Type	Partial		
PVE-1.14	Discretely Sampled Function Value Type	No		
PVE-1.15	Discretely Sampled Function Interpolation	No		
PVE-1.16	Probabilistic Value Distributions	Partial	Annex E.7 Distribution Extensions	

# Examples: Discretely Sampled Function Value Type

scalar value type for  
mission\_elapsed\_time

compound value types for  
location and velocity

ascent profile						
domainValueType	rangeValueType					
mission_elapsed_time	location			velocity		
	location.x	location.y	location.z	velocity.x	velocity.y	velocity.z
[s]	[km]	[km]	[km]	[m/s]	[m/s]	[m/s]
0	0	0	0	0	0	0
30	0.1	0.2	1	1	1	30
59	0.2	0.3	3	1.4	2.1	111
93	...	...	...	...	...	...
...	...	...	...	...	...	...

# Properties, Values & Expressions

– Full concept model available on wiki page

- Currently being integrated into SECM on Team Cloud Server

