OSLC PLM Workgroup

Sept 21st 2010

open-services.net

V0.1

Organisers today

- Workgroup lead: Rainer Ersch, Siemens
- Coordinator: Gray Bachelor, IBM

Today’s agenda

- Roll call and brief introductions - welcome new members
- Objective for today's meeting - Discuss progress with the investigation of defining a reference context for SE Scenario #1
- Overview and discussion on representation of context and implementation based upon STEP
- Discuss traceability scenarios within SE Scenario #1
- Overview and discussion representing STEP as resources
- Next steps for working with the OSLC SPECS
- Dates of next meetings - Oct 5th and 19th proposed
- AOB
- Summary and close

Today's objectives

1. To continue to discuss the product context and implementation based upon STEP
2. To agree an approach to define an initial resource definition for context and implementation

Summary of the approach

- Our scenario #1 provides the basis for exploring the coverage of the existing OSLC Specs
  - http://open-services.net/bin/view/Main/PlmSystemsEngineeringScenarioSystemsEngineerReactstoChangedRequirements
- We identified two actions as typical of the need to trace product and system context and implementation
  - a4 Locate requirements in change request context
  - a7 Locate Reusable Implementation to Satisfy Change ?
- These actions require that we identify means to represent
  - Requirements as configured text, documents and models
  - Context and implementation as configured structures, meta-data and models
  - Relationships between Requirements, Context and Implementations
- We propose initially to define a reference or boundary representation of product and/or system to use to evaluate the existing Specs (resources and services)
- There is not a single dominant representation of product and system structure to use as a reference
- We agreed to explore the Standard for the Exchange of Product model data (STEP)
  - Based upon ISO 10303 and is meant for product data exchange between tools
  - has a modular construction applied in multiple Application Protocols with significant industry support
  - has a proven and flexible core construct of Product, Product_version, Product_view_definition
- We agreed to explore and apply the SysML SUV example to support our investigation

Progress made

- Initial identification of relevant assets and information in the public domain
- Production of sample data from the SUV SysML example (Requirements diagram)
  - STEP representation (.stp file)
  - OWL representation (.owl file)
- Exploration of SUV Requirements representation in OWL
  - STEP file
  - ontoSTEP
  - Protege

Using the SUV example

- Requirements Diagram mark up
  - Mark up versions
- Derived Requirements mark up
  - Mark up versions
  - Identify not supported in Part 21
- Block Diagram
  - Mark up versions
  - Identify not supported in Part 21

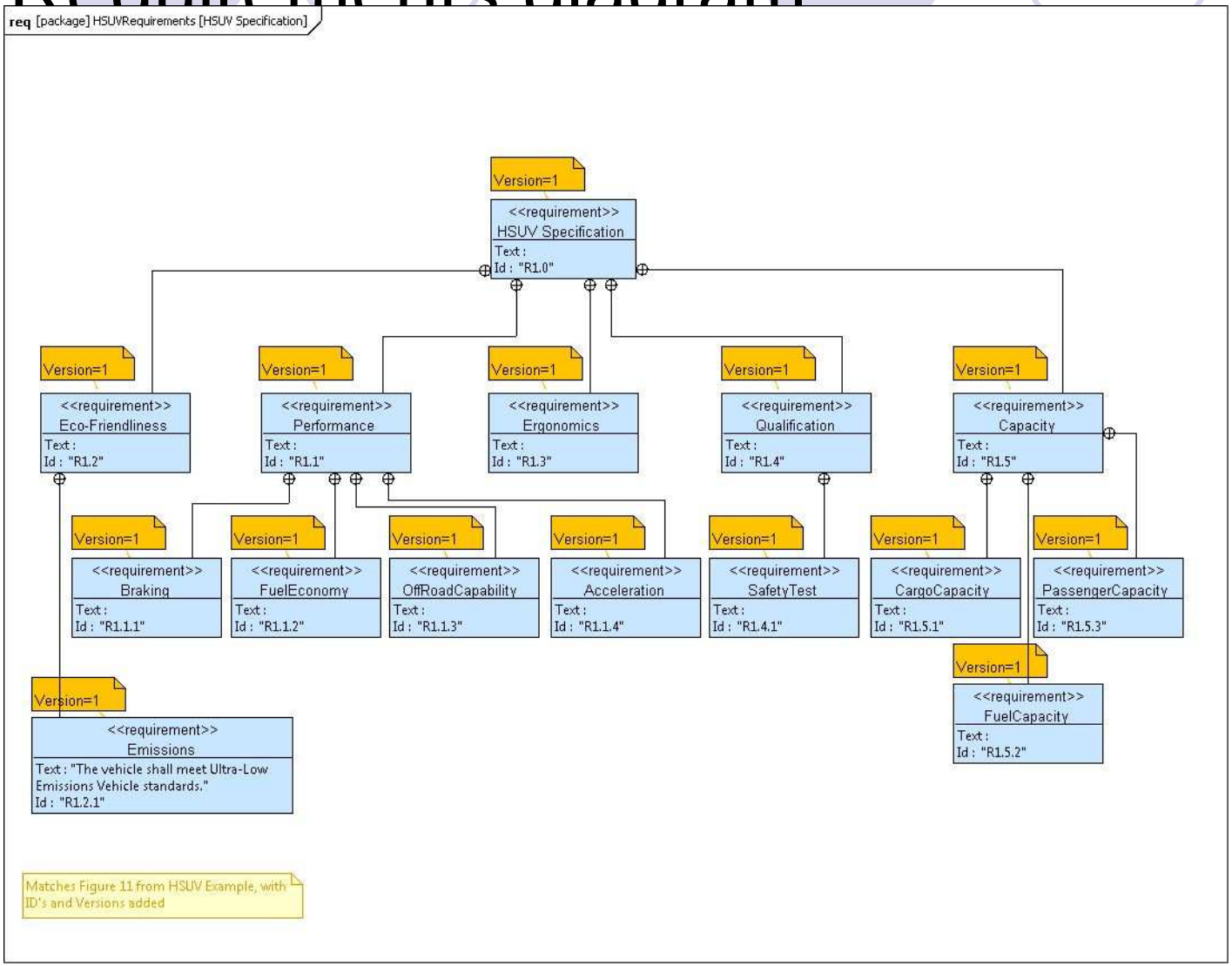
Progress made

Updated Hybrid SUV SysML assets

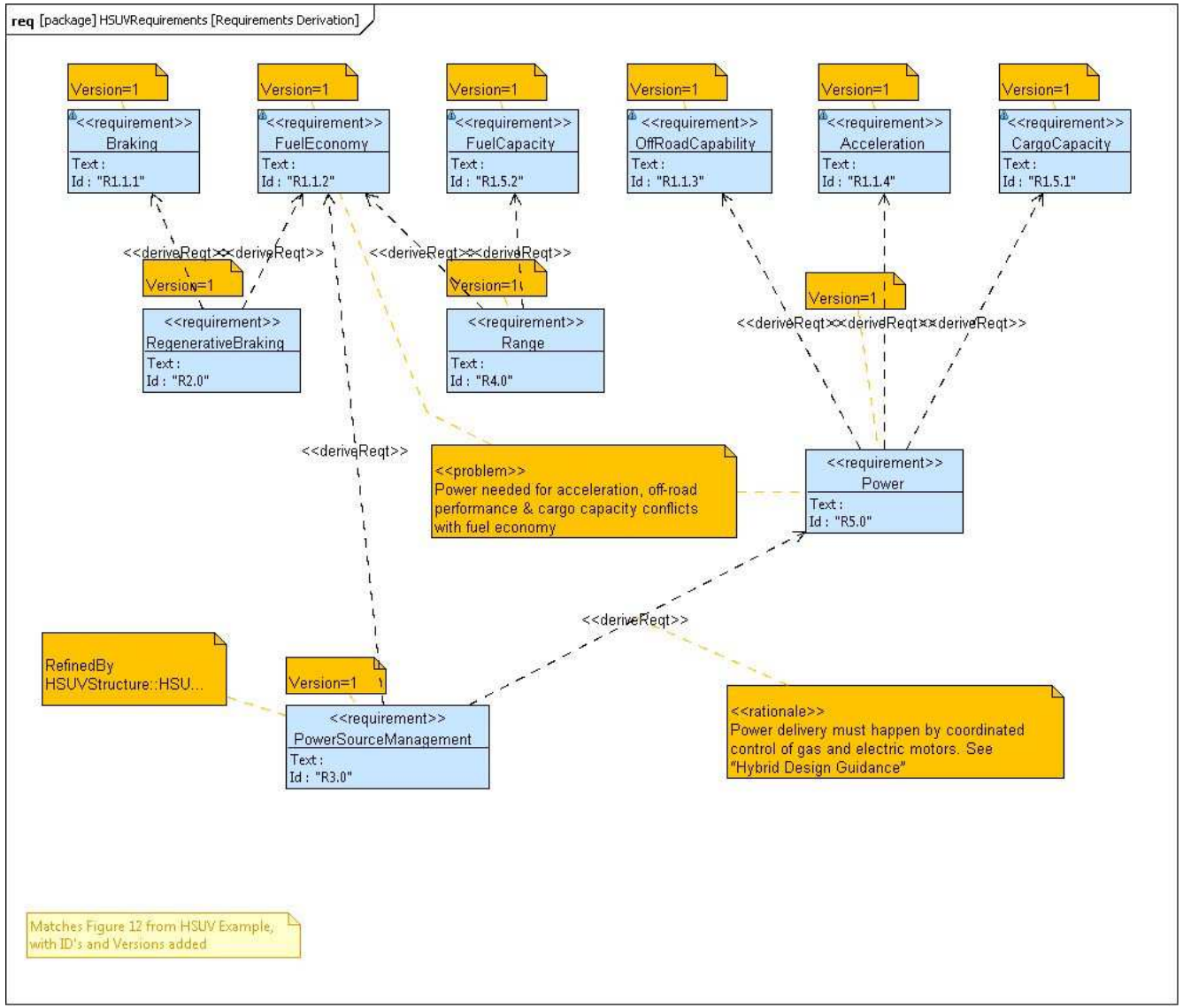
Annotated with product IDs and versions

- Requirements diagram
- Requirements derivation
- Block diagram
- Block definition diagram

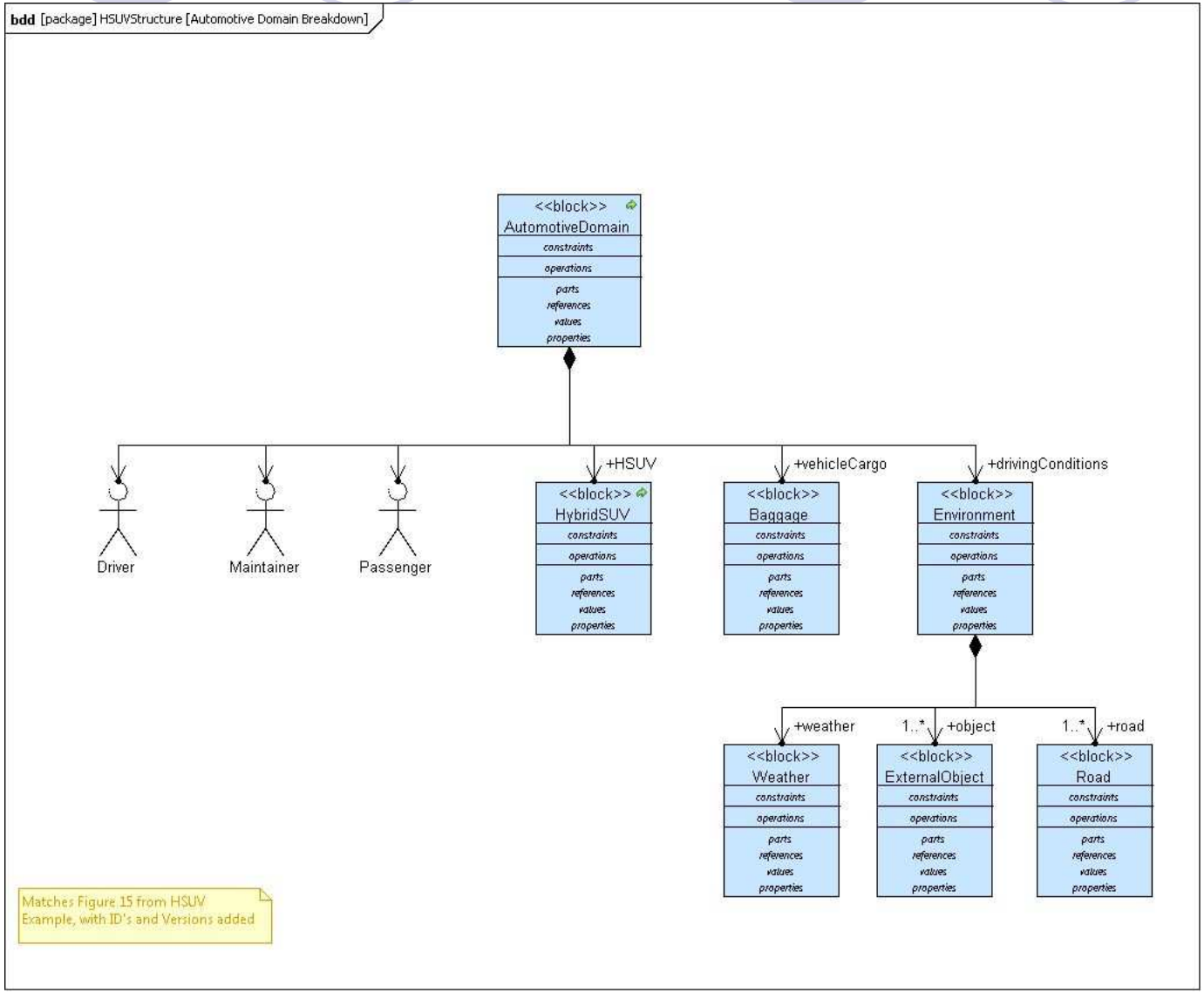
Updated SysML SUV Model - Requirements diagram



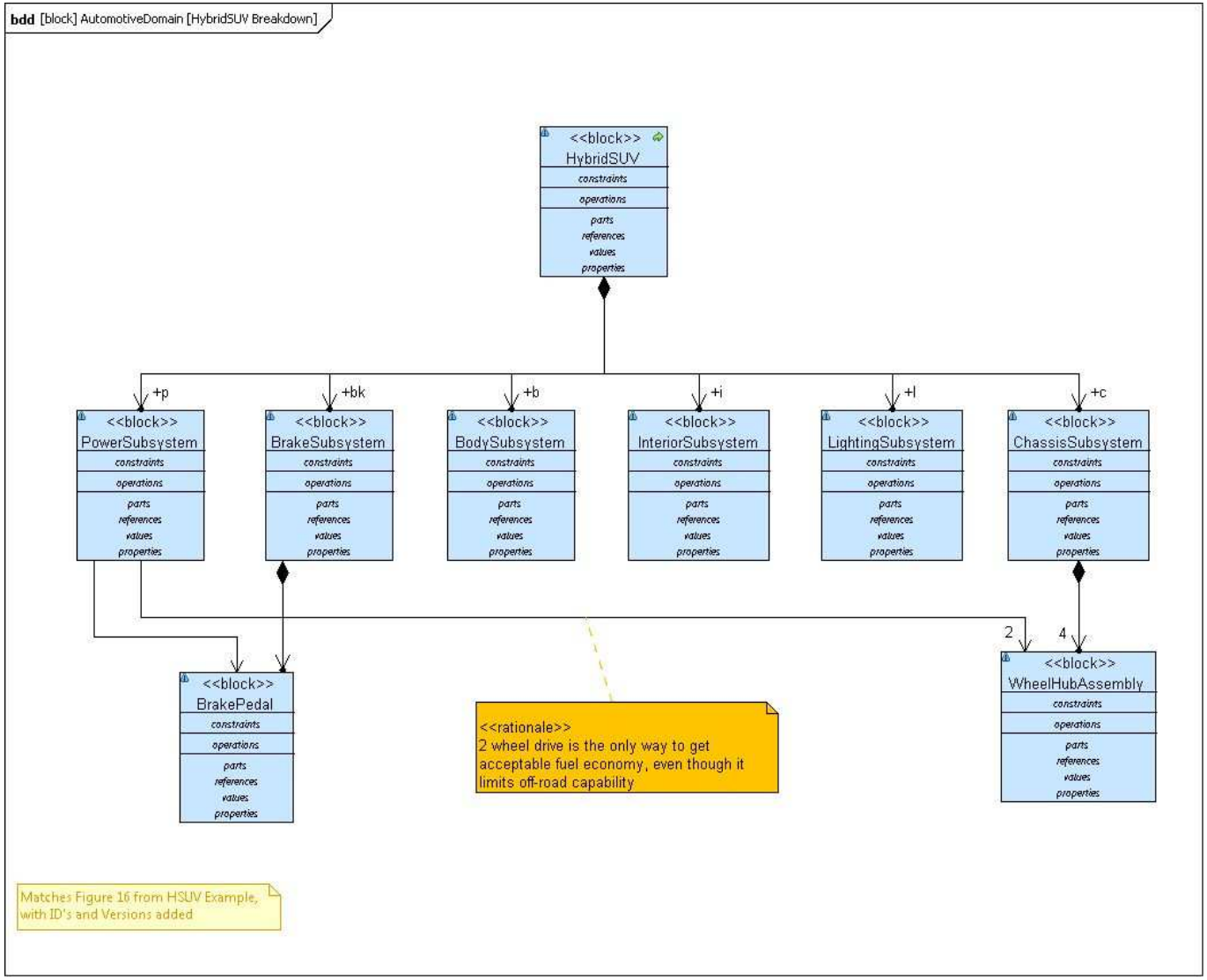
Updated SysML SUV Model - Derived Requirements



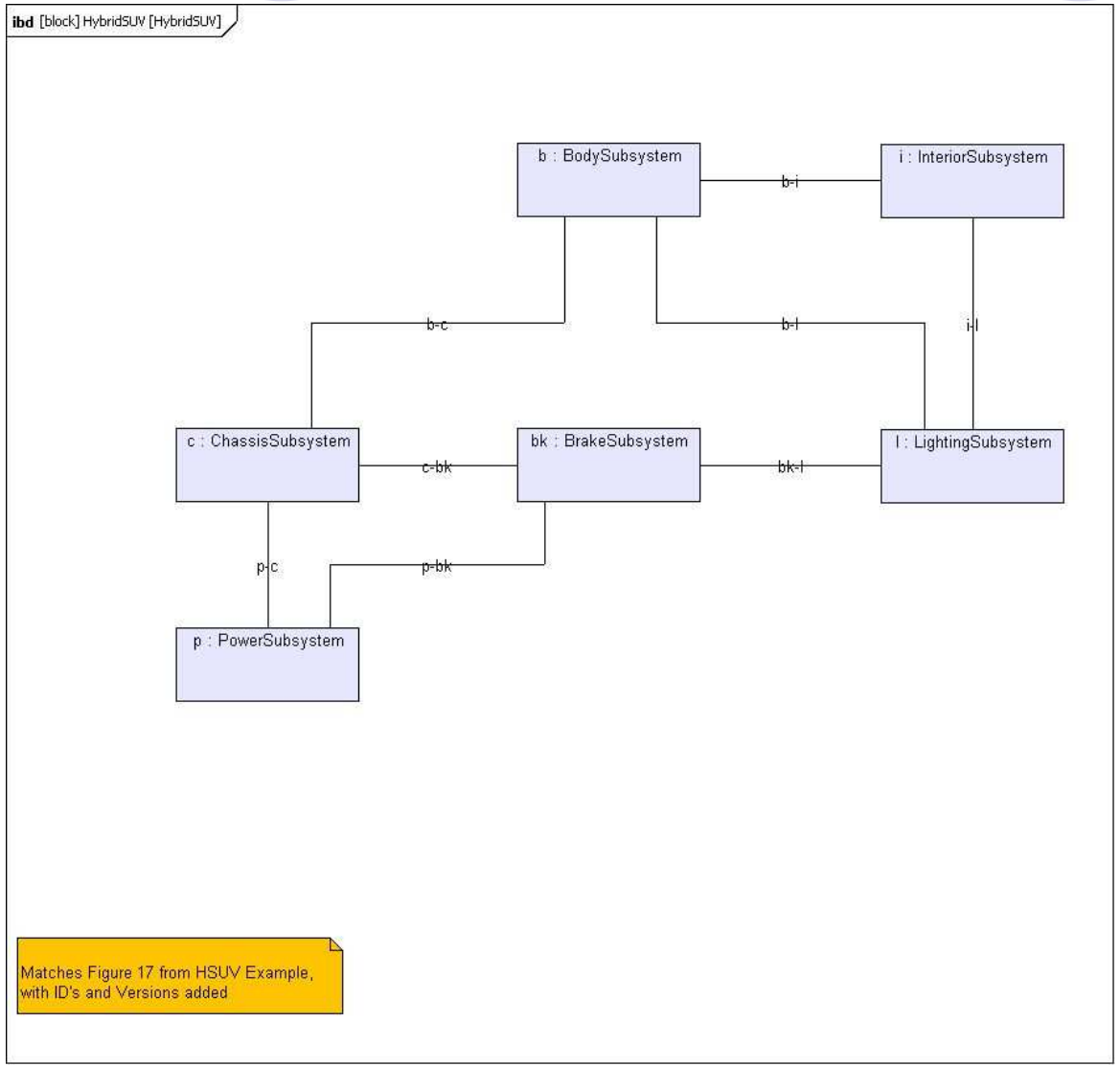
Updated SysML SUV Model - Block definition diagram



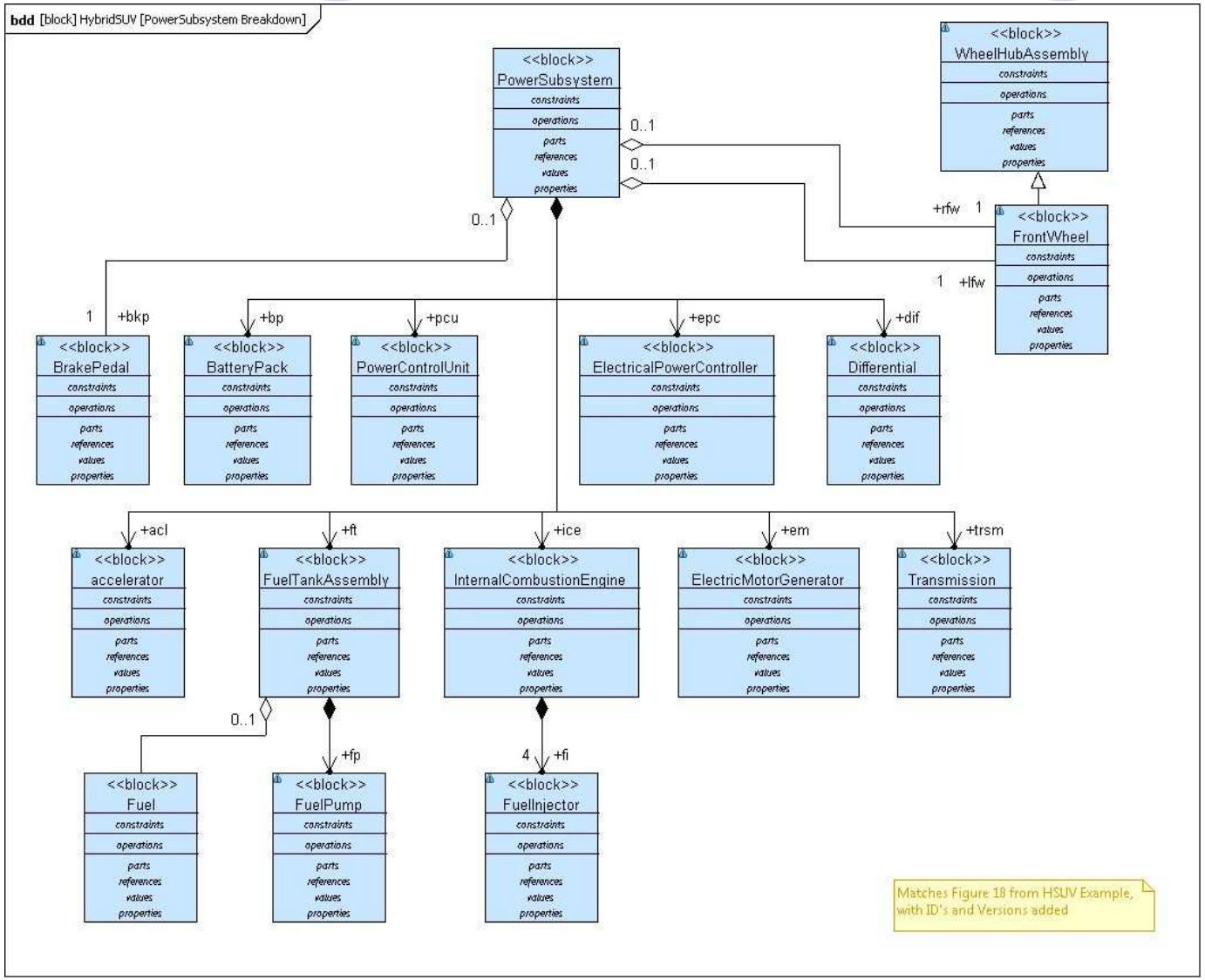
Updated SysML SUV Model - Block definition diagram



Updated SysML SUV Model - Internal Block diagram



Block definition diagram



Questions

- To allow a basic view of Product ….how to show an example of the representation of
  - Parts
  - Alternative part version and configurations
  - Associations and allocations
    - Requirements, Change Requests, Systems structures
- How much of the STEP representation is needed to identify OSLC needs, to
  - “support”, “adopt”
  - “know how to use” (actually discover, establish a representation and provide an interface and support use)

Next meeting

- Propose
  - Next working meetings
    - Oct 5th 11am Eastern proposed
    - Oct 19th 11am Eastern proposed

Any other business ?

Summary

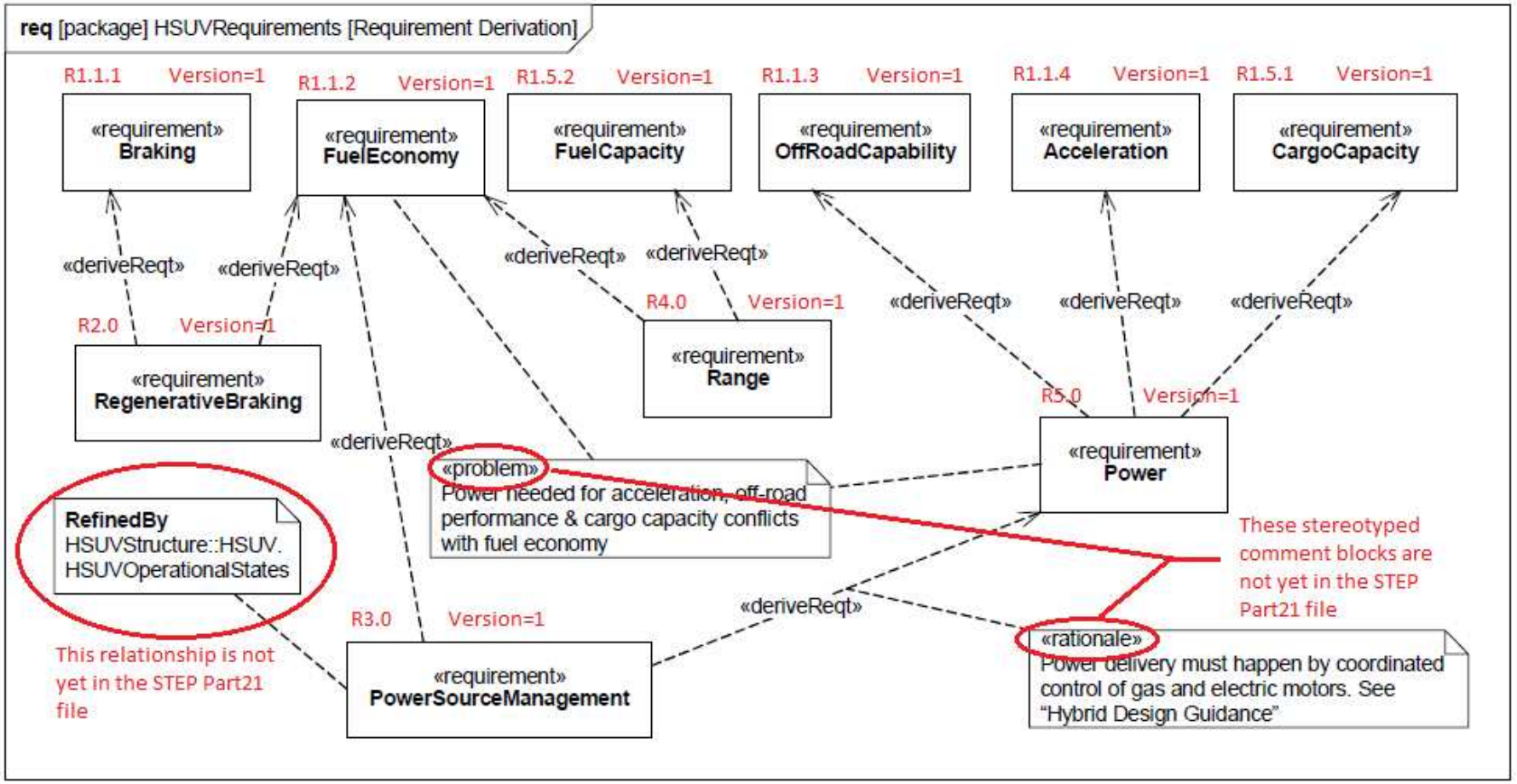
Thank you

rainer.ersch@siemens.com

gray_bachelor@uk.ibm.com

Backup

derived_requirements.png



structure_bdd.png

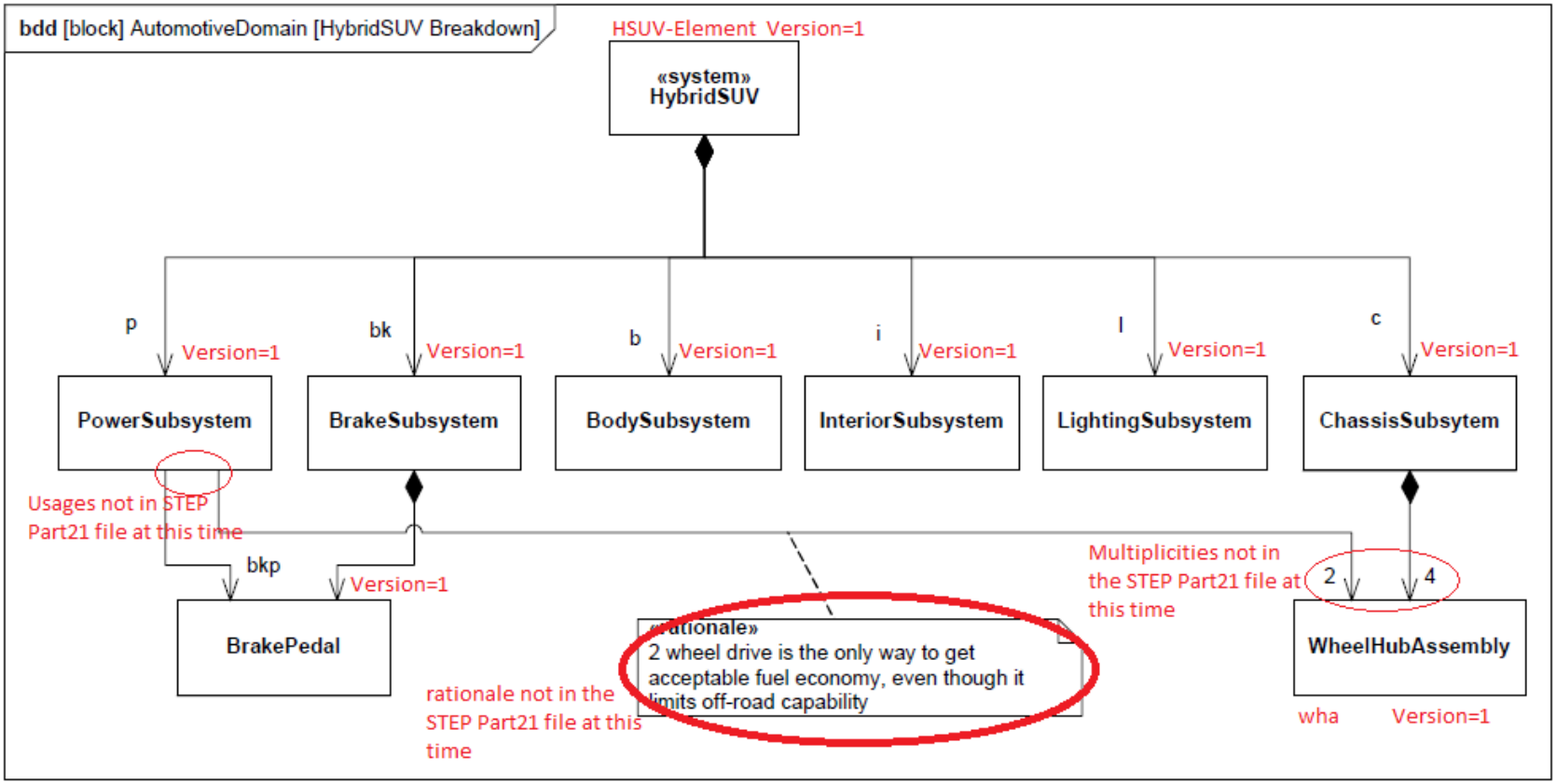
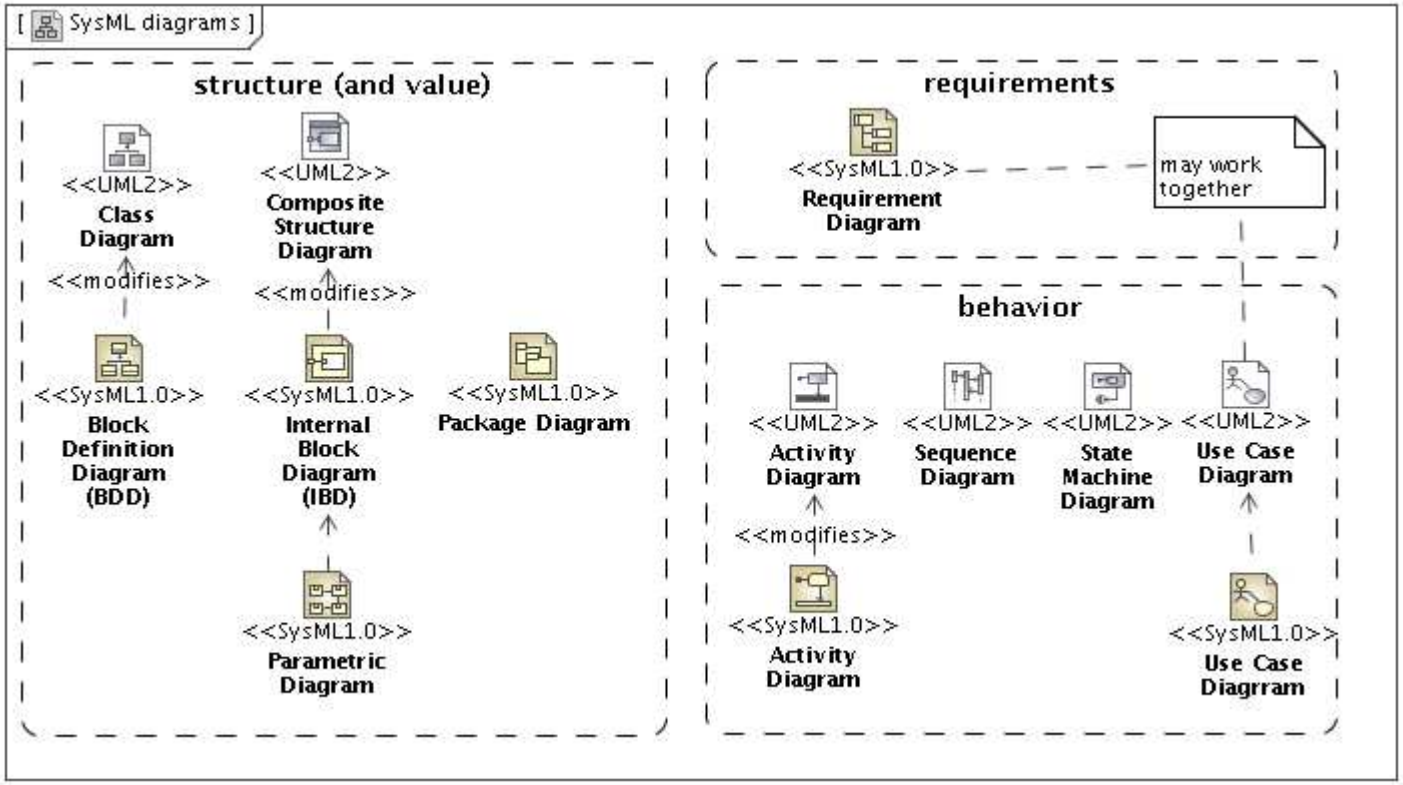
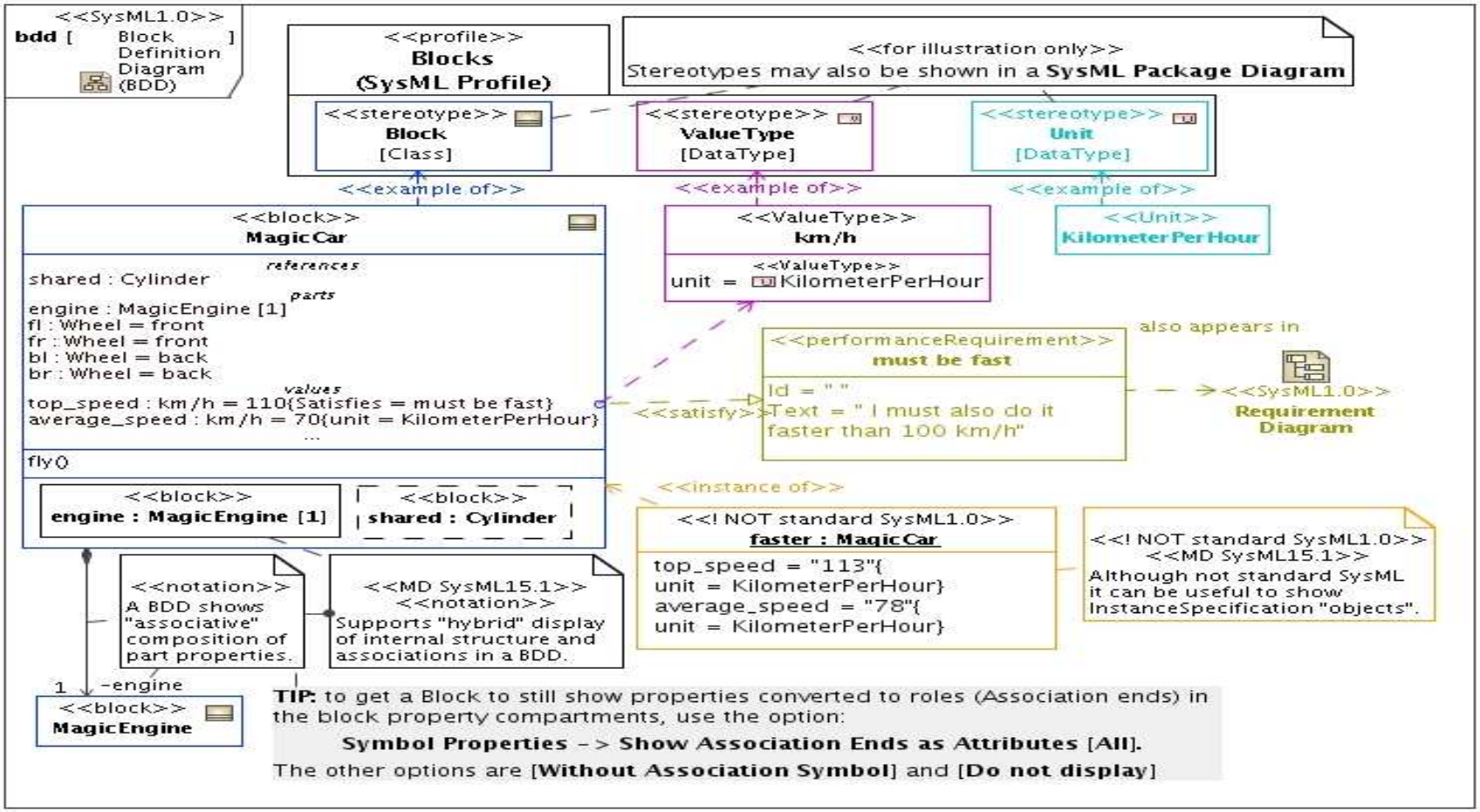


Figure 16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)


[SysML diagrams]
structure (and value)
<<UML2>>
Class Diagram
<<UML2>>
Composite Structure Diagram
<<modifies>>
<<modifies>>
<<SysML1.0>>
Block Definition Diagram (BDD)
<<SysML1.0>>
Internal Block Diagram (IBD)
<<SysML1.0>>
Package Diagram
<<SysML1.0>>
Parametric Diagram
requirements
<<SysML1.0>>
Requirement Diagram
may work together
behavior
<<UML2>>
Activity Diagram
<<UML2>>
Sequence Diagram
<<UML2>>
State Machine Diagram
<<UML2>>
Use Case Diagram
<<modifies>>
<<SysML1.0>>
Activity Diagram
<<SysML1.0>>
Use Case Diagrram

BDD example


<<SysML1.0>>
bdd [Block Definition Diagram (BDD)]
<<profile>>
Blocks
(SysML Profile)
<<for illustration only>>
Stereotypes may also be shown in a SysML Package Diagram
<<stereotype>>
Block
[Class]
<<stereotype>>
ValueType
[DataType]
<<stereotype>>
Unit
[DataType]
<<example of>>
<<example of>>
<<example of>>
<<block>>
MagicCar
references
shared : Cylinder
parts
engine : MagicEngine [1]
fl : Wheel = front
fr : Wheel = front
bl : Wheel = back
br : Wheel = back
values
top_speed : km/h = 110{Satisfies = must be fast}
average_speed : km/h = 70{unit = KilometerPerHour}
...
fly()
<<ValueType>>
km/h
<<ValueType>>
unit = KilometerPerHour
<<Unit>>
KilometerPerHour
<<satisfy>>
<<performanceRequirement>>
must be fast
Id = " "
Text = " I must also do it faster than 100 km/h"
also appears in
<<SysML1.0>>
Requirement Diagram
<<instance of>>
<<block>>
engine : MagicEngine [1]
<<block>>
shared : Cylinder
<<! NOT standard SysML1.0>>
faster : MagicCar
top_speed = "113"{
unit = KilometerPerHour}
average_speed = "78"{
unit = KilometerPerHour}
<<! NOT standard SysML1.0>>
<<MD SysML15.1>>
Although not standard SysML it can be useful to show InstanceSpecification "objects".
<<notation>>
A BDD shows "associative" composition of part properties.
<<MD SysML15.1>>
<<notation>>
Supports "hybrid" display of internal structure and associations in a BDD.
1
-engine
<<block>>
MagicEngine
TIP: to get a Block to still show properties converted to roles (Association ends) in the block property compartments, use the option:
Symbol Properties -> Show Association Ends as Attributes [All].
The other options are [Without Association Symbol] and [Do not display]