

OMG Systems Modeling Language (OMG SysML™)

Hybrid SUV Non-Normative Example

Copyright © 2003-2006, American Systems Corporation
Copyright © 2003-2006, ARTiSAN Software Tools
Copyright © 2003-2006, BAE SYSTEMS
Copyright © 2003-2006, The Boeing Company
Copyright © 2003-2006, Ceira Technologies
Copyright © 2003-2006, Deere & Company
Copyright © 2003-2006, EADS Astrium GmbH
Copyright © 2003-2006, EmbeddedPlus Engineering
Copyright © 2003-2006, Eurostep Group AB
Copyright © 2003-2006, Gentleware AG
Copyright © 2003-2006, I-Logix, Inc.
Copyright © 2003-2006, International Business Machines
Copyright © 2003-2006, International Council on Systems Engineering
Copyright © 2003-2006, Israel Aircraft Industries
Copyright © 2003-2006, Lockheed Martin Corporation
Copyright © 2003-2006, Mentor Graphics
Copyright © 2003-2006, Motorola, Inc.
Copyright © 2003-2006, Northrop Grumman
Copyright © 1997-2010, Object Management Group
Copyright © 2003-2006, oose Innovative Informatik GmbH
Copyright © 2003-2006, PivotPoint Technology Corporation
Copyright © 2003-2006, Raytheon Company
Copyright © 2003-2006, Sparx Systems
Copyright © 2003-2006, Telelogic AB
Copyright © 2003-2006, THALES

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

The specification customizes the Unified Modeling Language (UML) specification of the Object Management Group (OMG) to address the requirements of Systems Engineering as specified in the UML for Systems Engineering RFP, OMG document number ad/2003-03-41. This document includes references to and excerpts from the UML 2 Superstructure Specification and UML 2 Infrastructure Specification with copyright holders and conditions as noted in those documents.

LICENSES

Redistribution and use of this specification, with or without modification, are permitted provided that the following conditions are met: (1) Redistributions of this specification must reproduce the above copyright notice, this list of conditions and disclaimers in the documentation and/or other materials provided with the distribution; (2) The Copyright Holders listed in the above copyright notice may not be used to endorse or promote products derived from this specification without specific prior written permission; (3) All modified versions of this specification must include a prominent notice stating how and when the specification was modified; and (4) No modifications to this OMG SysML™

specification may be published under or identified by that name, except for versions published by OMG and incorporating official changes made through the applicable procedures of OMG. OMG SysML™ is a trademark of OMG, and no unauthorized version or revision of the OMG SysML specification may use the trademark “OMG SysML” or claim any connection with or endorsement by OMG.

In accordance with the above copyright provisions, the companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute OMG SysML and to modify OMG SysML and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification. Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, nonsublicenseable, perpetual, worldwide license, to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of this document in your possession or control.

This document was derived from the non-normative Annex B of the "Systems Modeling Language (SysML) Specification, version 1.2, submitted to OMG in response to the "UML for Systems Engineering RFP" Review and editing in the OMG process produced the "OMG SysML Specification Final Adopted Specification." Subsequent changes to the specification are controlled through the OMG process as documented at the OMG Technology Document website - <http://www.omg.org/technology/documents/>.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED “AS IS” AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED

BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 140 Kendrick Street, Needham, MA 02494, U.S.A.

TRADEMARKS

MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™, Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™, IMM™, MOF™, OMG Interface Definition Language (OMG IDL)™, and OMG Systems Modeling Language (OMG SysML)™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with OMG SysML™. Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

HybridSUV

(informative)

1 Purpose

The purpose of this document is to illustrate how SysML can support the specification, analysis, and design of a system using some of the basic features of the language.

2 Scope

The scope of this example is to provide at least one diagram for each SysML diagram type. The intent is to select simplified fragments of the problem to illustrate how the diagrams can be applied, and also demonstrate some of the possible inter-relationships among the model elements in the different diagrams. The sample problem does not highlight all of the features of the language. The reader should refer to the individual chapters for more detailed features of the language. The diagrams selected for representing a particular aspect of the model, and the ordering of the diagrams are intended to be representative of applying a typical systems engineering process, but this will vary depending on the specific process and methodology that is used.

3 Problem Summary

The sample problem describes the use of SysML as it applies to the development of an automobile, in particular a Hybrid gas/electric powered Sport Utility Vehicle (SUV). This problem is interesting in that it has inherently conflicting requirements, viz. desire for fuel efficiency, but also desire for large cargo carrying capacity and off-road capability. Technical accuracy and the feasibility of the actual solution proposed were not high priorities. This sample problem focuses on design decisions surrounding the power subsystem of the hybrid SUV; the requirements, performance analyses, structure, and behavior.

This document is structured to show each diagram in the context of how it might be used on such an example problem. The first section shows SysML diagrams as they might be used to establish the system context; establishing system boundaries, and top level use cases. The next section is provided to show how SysML diagrams can be used to analyze top level system behavior, using sequence diagrams and state machine diagrams. The following section focuses on use of SysML diagrams for capturing and deriving requirements, using diagrams and tables. A section is provided to illustrate how SysML is used to depict system structure, including block hierarchy and part relationships. The relationship of various system parameters, performance constraints, analyses, and timing diagrams are illustrated in the next section. A section is then dedicated to illustrating definition and depiction of interfaces and flows in a structural context. The final section focuses on detailed behavior modeling, functional and flow allocation.

4 Diagrams

4.1 Package Overview (Structure of the Sample Model)

4.1.1 Package Diagram - Applying the SysML Profile

As shown in Figure 1, the HSUVModel is a package that represents the user model. The SysML Profile must be applied to this package in order to include stereotypes from the profile. The HSUVModel may also require model libraries, such as the SI Units Types model library. The model libraries must be imported into the user model as indicated.

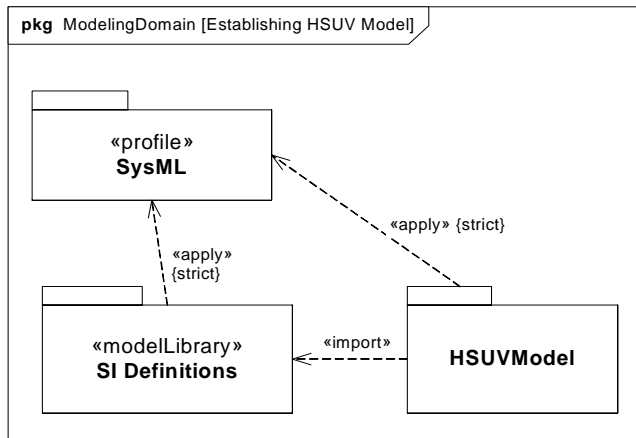


Figure 1 - Establishing the User Model by Importing and Applying SysML Profile & Model Library (Package Diagram)

Figure 2 details the specification of units and valueTypes employed in this sample problem.

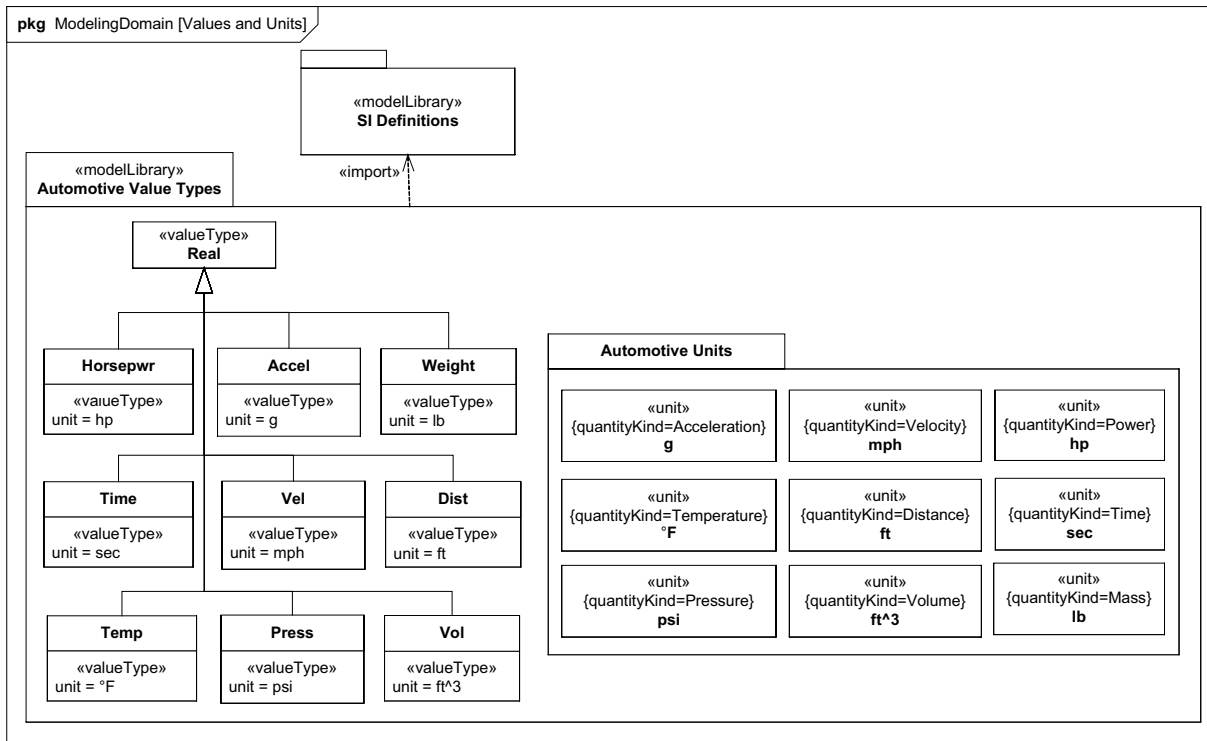


Figure 2 - Defining valueTypes and units to be Used in the Sample Problem

4.1.2 Package Diagram - Showing Package Structure of the Model

The package diagram (Figure 3) shows the structure of the model used to evaluate the sample problem. Model elements are contained in packages, and relationships between packages (or specific model elements) are shown on this diagram. The relationship between the views (OperationalView and PerformanceView) and the rest of the user model are explicitly expressed using the **«import»** relationship. Note that the **«view»** models contain no model elements of their own, and that changes to the model in other packages are automatically updated in the Operational and Performance Views.

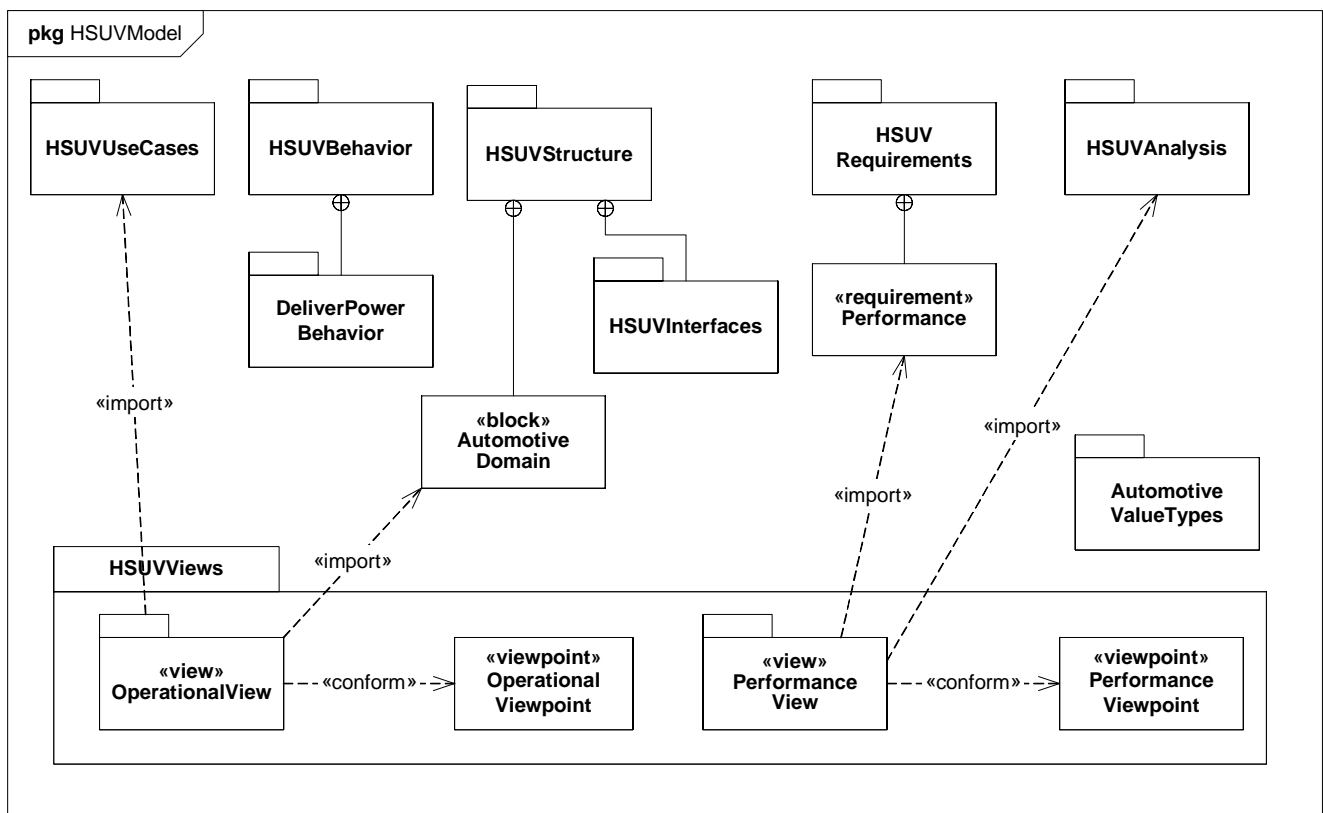


Figure 3 - Establishing Structure of the User Model 3 using Packages and Views (Package Diagram)

4.2 Setting the Context (Boundaries and Use Cases)

4.2.1 Internal Block Diagram - Setting Context

The term “context diagram,” in Figure 4, refers to a user-defined usage of an internal block diagram, which depicts some of the top-level entities in the overall enterprise and their relationships. The diagram usage enables the modeler or methodologist to specify a unique usage of a SysML diagram type using the extension mechanism. The entities are conceptual in nature during the initial phase of development, but will be refined as part of the development process. The `«system»` and `«external»` stereotypes are user defined, not specified in SysML, but help the modeler to identify the system of interest relative to its environment. Each model element depicted may include a graphical icon to help convey its intended meaning. The spatial relationship of the entities on the diagram sometimes conveys understanding as well, although this is not specifically captured in the semantics. Also, a background such as a map can be included to provide additional context. The associations among the classes may represent abstract conceptual relationships among the entities, which would be refined in subsequent diagrams. Note how the relationships in this diagram are also reflected in the Automotive Domain Model Block Definition Diagram, Figure 15.

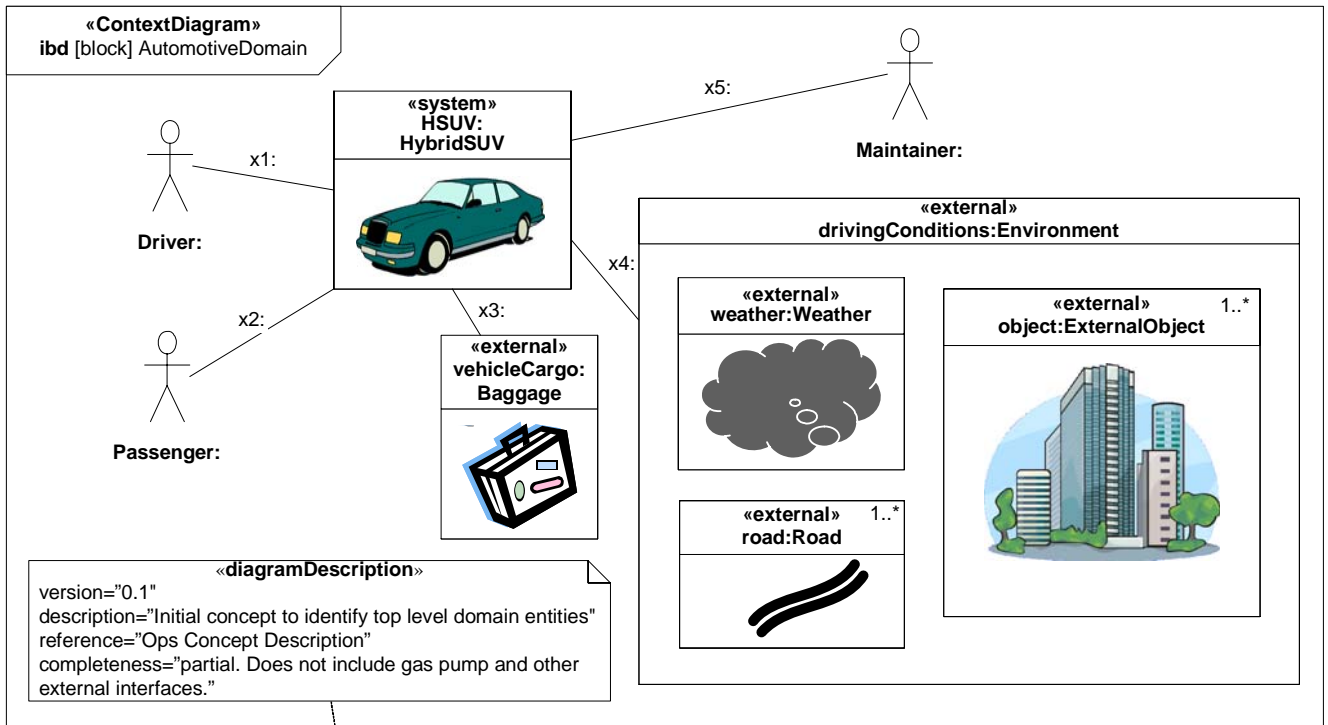


Figure 4 - Establishing the Context of the Hybrid SUV System using a User-Defined Context Diagram. (Internal Block Diagram) Completeness of Diagram Noted in Diagram Description

4.2.2 Use Case Diagram - Top Level Use Cases

The use case diagram for “Drive Vehicle” in Figure 5 depicts the drive vehicle usage of the vehicle system. The subject (HybridSUV) and the actors (Driver, Registered Owner, Maintainer, Insurance Company, DMV) interact to realize the use case.

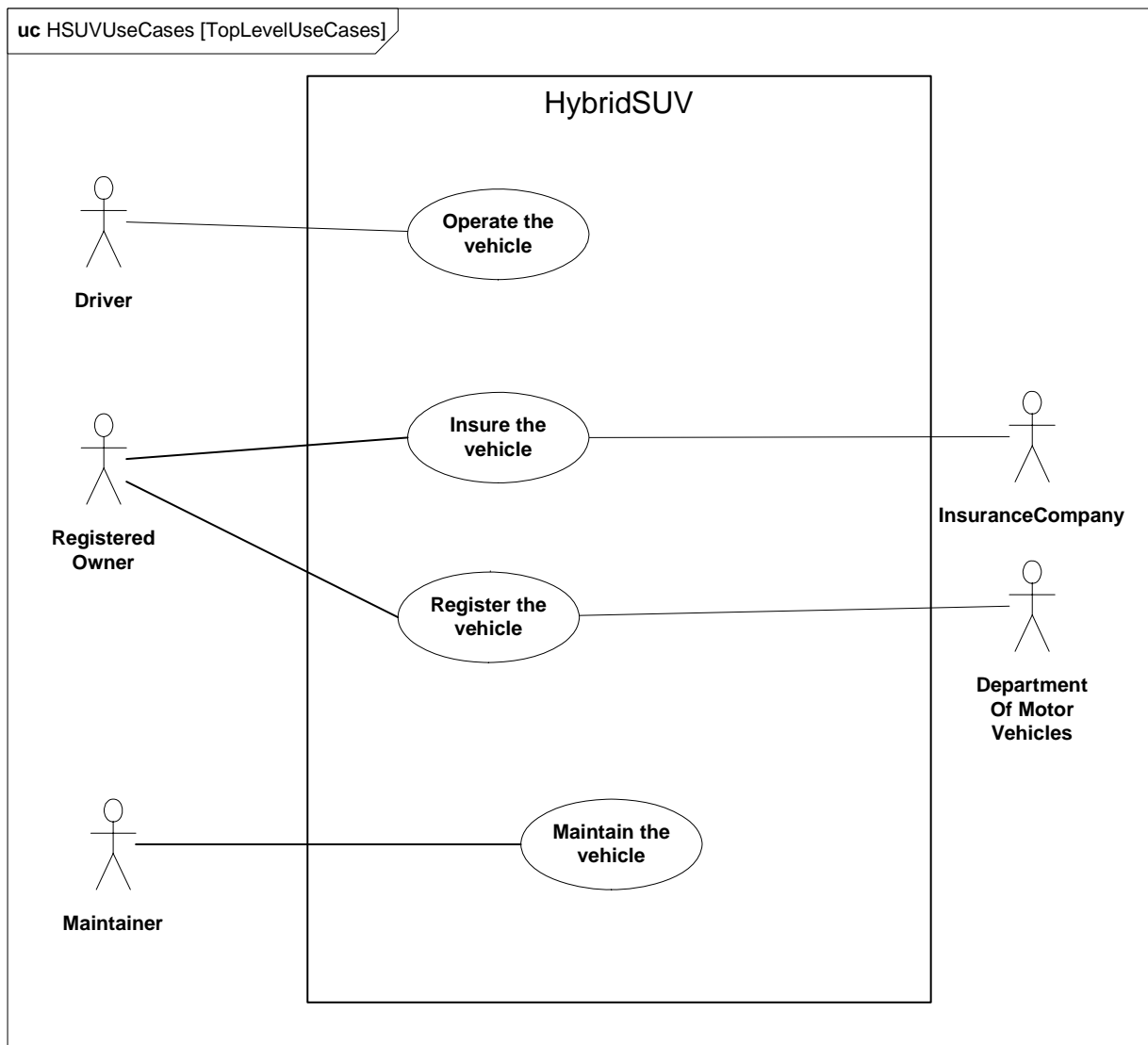


Figure 5 - Establishing Top Level Use Cases for the Hybrid SUV (Use Case Diagram)

4.2.3 Use Case Diagram - Operational Use Cases

Goal-level Use Cases associated with “Operate the Vehicle” are depicted in the following diagram. These use cases help flesh out the specific kind of goals associated with driving and parking the vehicle. Maintenance, registration, and insurance of the vehicle would be covered under a separate set of goal-oriented use cases.

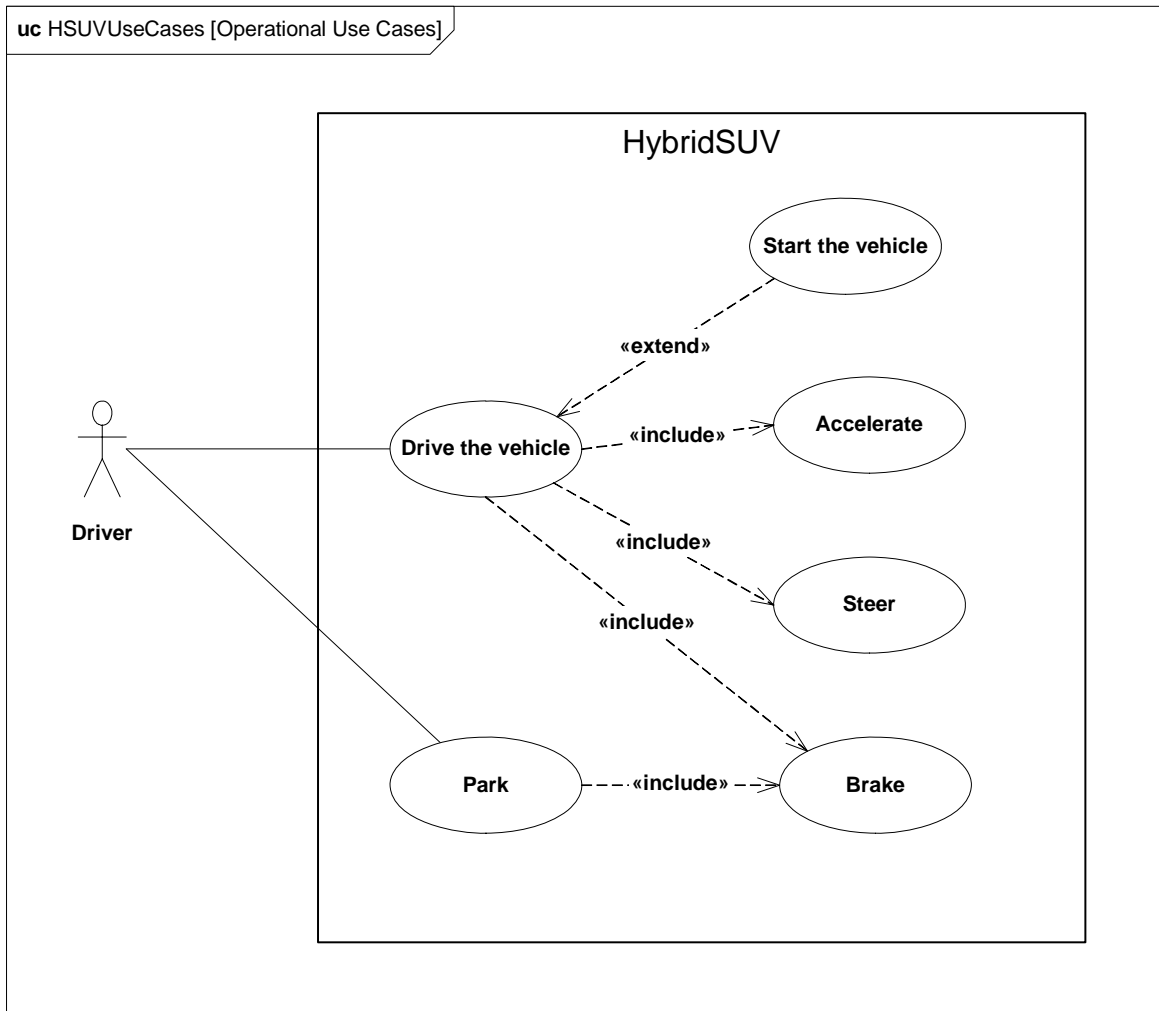


Figure 6 - Establishing Operational Use Cases for “Drive the Vehicle” (Use Case Diagram)

4.3 Elaborating Behavior (Sequence and State Machine Diagrams)

4.3.1 Sequence Diagram - Drive Black Box

Figure 7 shows the interactions between driver and vehicle that are necessary for the “Drive the Vehicle” Use Case. This diagram represents the “DriveBlackBox” interaction, with is owned by the AutomotiveDomain block. “BlackBox” for the purpose of this example, refers to how the subject system (HybridSUV block) interacts only with outside elements, without revealing any interior detail.

The conditions for each alternative in the alt controlSpeed section are expressed in OCL, and relate to the states of the HybridSUV block, as shown in Figure 8.

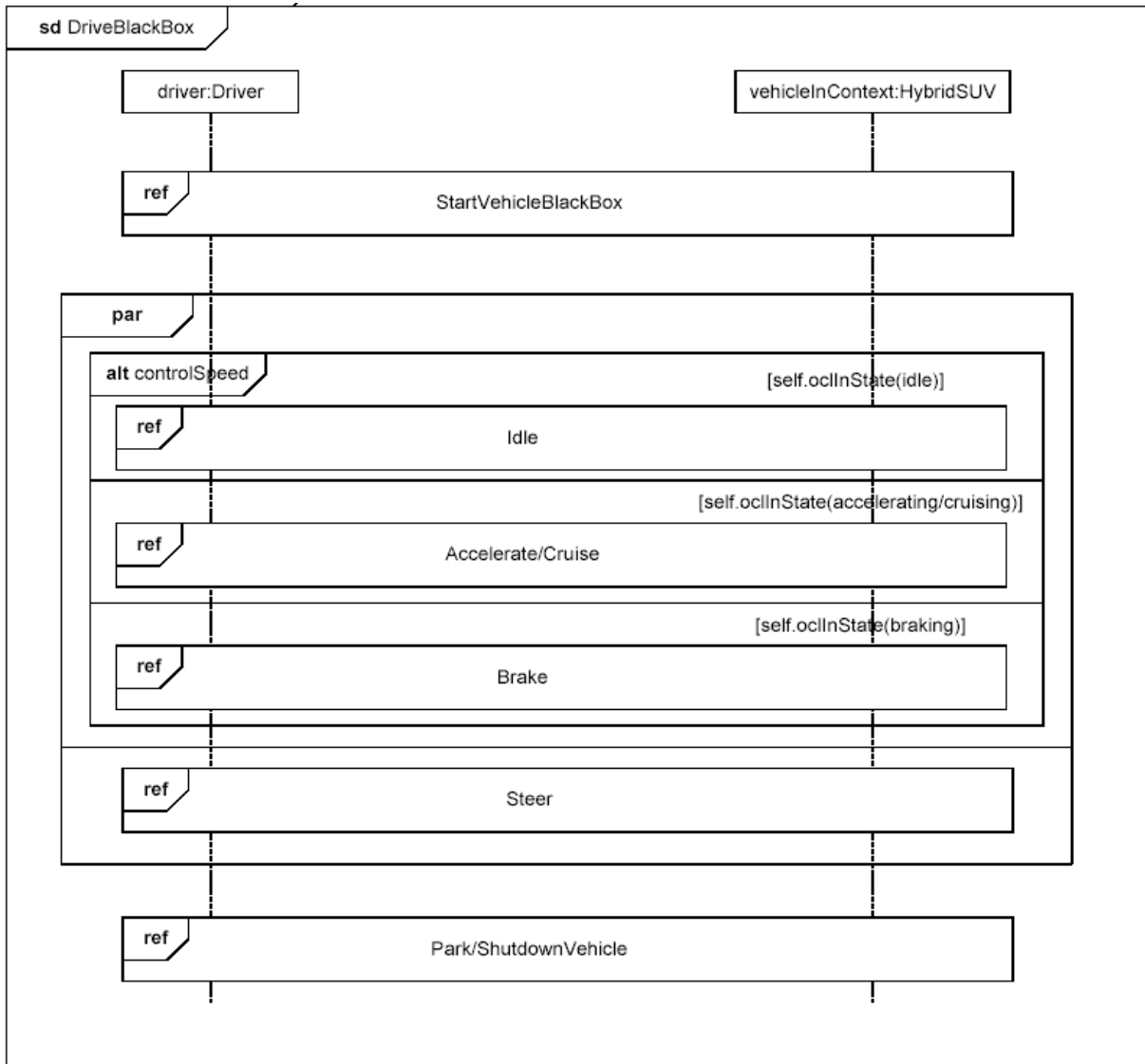


Figure 7 - Elaborating Black Box Behavior for the “Drive the Vehicle” Use Case (Sequence Diagram)

4.3.2 State Machine Diagram - HSUV Operational States

Figure 8 depicts the operational states of the HSUV block, via a State Machine named “HSUVOperationalStates.” Note that this state machine was developed in conjunction with the DriveBlackBox interaction in Figure 7. Also note that this state machine refines the requirement “PowerSourceManagment,” which will be elaborated in the requirements section of this sample problem. This diagram expresses only the nominal states. Exception states, like “acceleratorFailure,” are not expressed on this diagram.

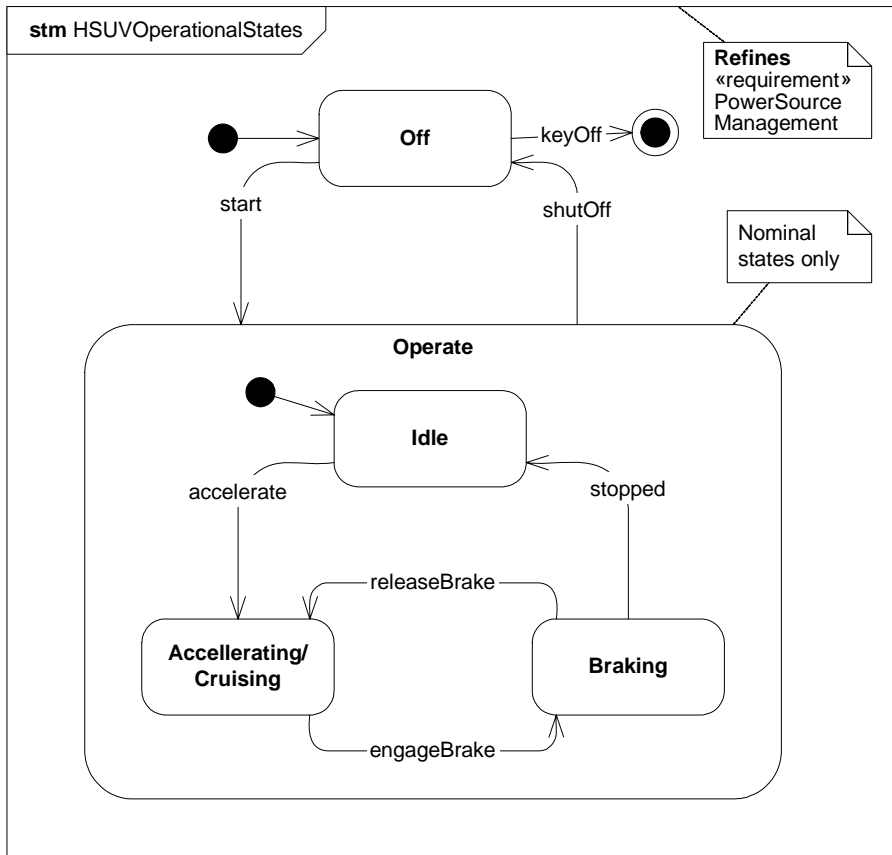


Figure 8 - Finite State Machine Associated with “Drive the Vehicle” (State Machine Diagram)

4.3.3 Sequence Diagram - Start Vehicle Black Box & White Box

Figure 9 shows a “black box” interaction, but references “StartVehicleWhiteBox” (Figure 10), which will decompose the lifelines within the context of the HybridSUV block.

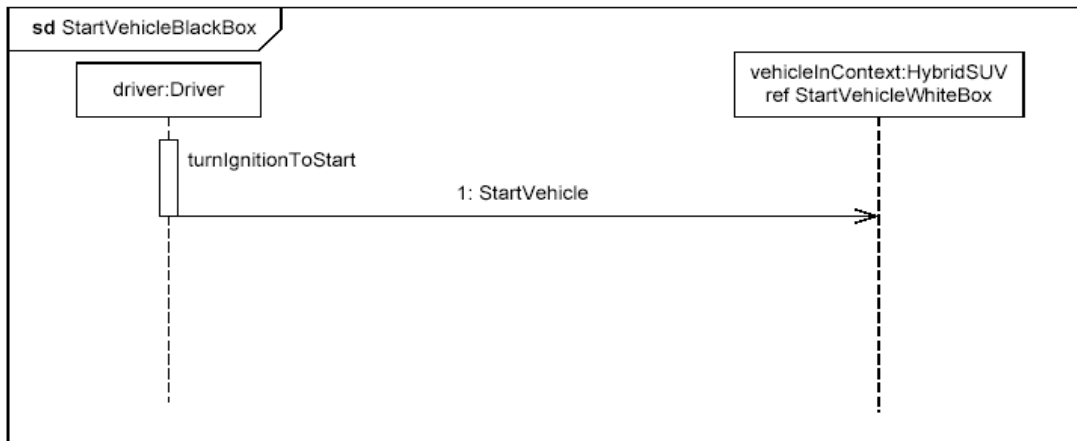


Figure 9 - Black Box Interaction for “StartVehicle,” referencing White Box Interaction (Sequence Diagram)

The lifelines on Figure 10 (“whitebox” sequence diagram) need to come from the Power System decomposition. This now begins to consider parts contained in the HybridSUV block.

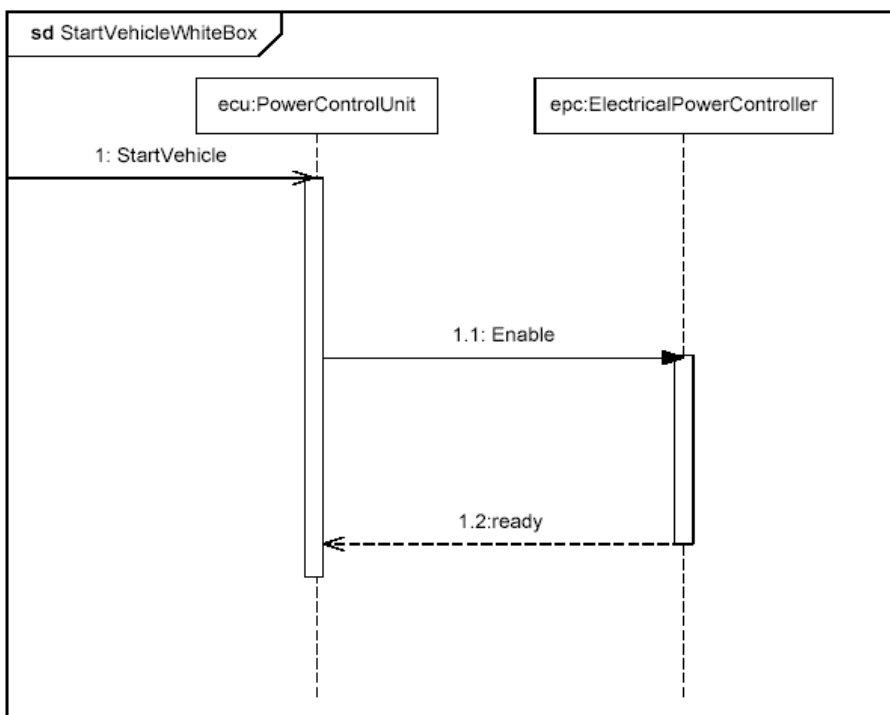


Figure 10 - White Box Interaction for “StartVehicle” (Sequence Diagram)

4.4 Establishing Requirements (Requirements Diagrams and Tables)

4.4.1 Requirement Diagram - HSUV Requirement Hierarchy

The vehicle system specification contains many text based requirements. A few requirements are highlighted in Figure 11, including the requirement for the vehicle to pass emissions standards, which is expanded for illustration purposes. The containment (cross hair) relationship, for purposes of this example, refers to the practice of decomposing a complex requirement into simpler, single requirements.

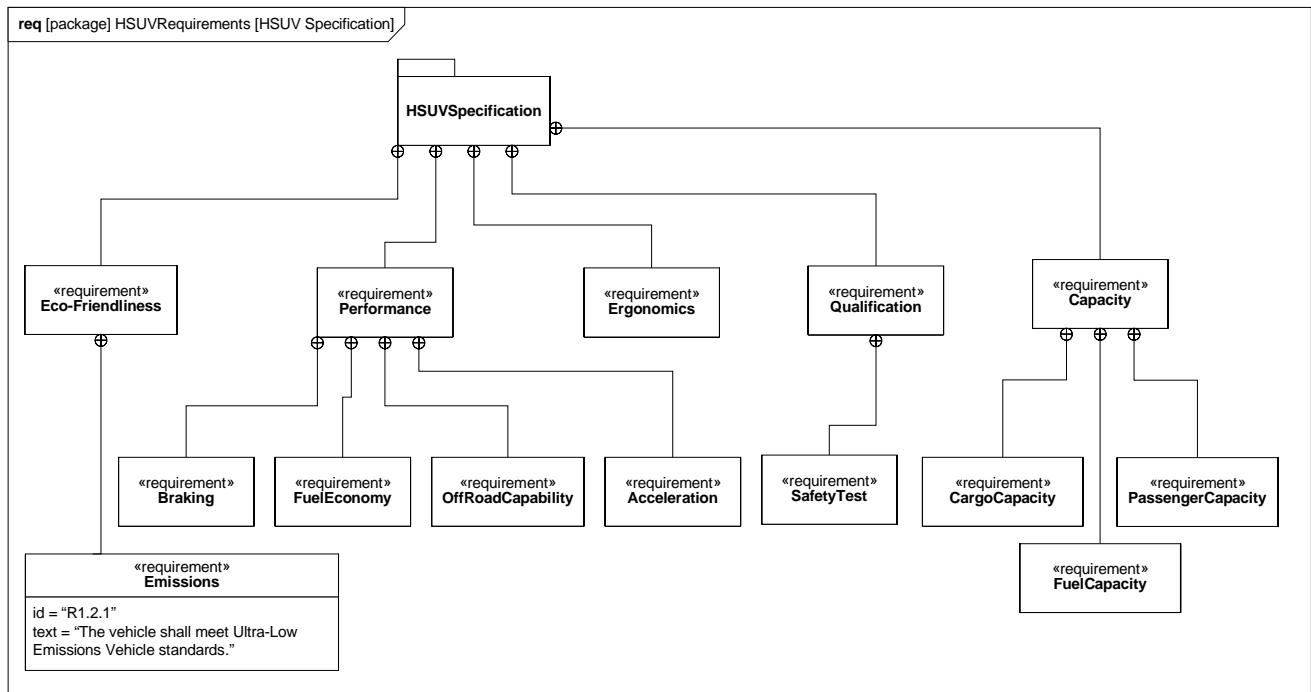


Figure 11 - Establishing HSUV Requirements Hierarchy (containment) - (Requirements Diagram)

4.4.2 Requirement Diagram - Derived Requirements

Figure 12 shows a set of requirements derived from the lowest tier requirements in the HSUV specification. Derived requirements, for the purpose of this example, express the concepts of requirements in the HSUVSpecification in a manner that specifically relates them to the HSUV system. Various other model elements may be necessary to help develop a derived requirement, and these model element may be related by a «refinedBy» relationship. Note how PowerSourceManagement is "RefinedBy" the HSUVOperationalStates model (Figure 8). Note also that rationale can be attached to the «deriveReq» relationship. In this case, rationale is provided by a referenced document "Hybrid Design Guidance."

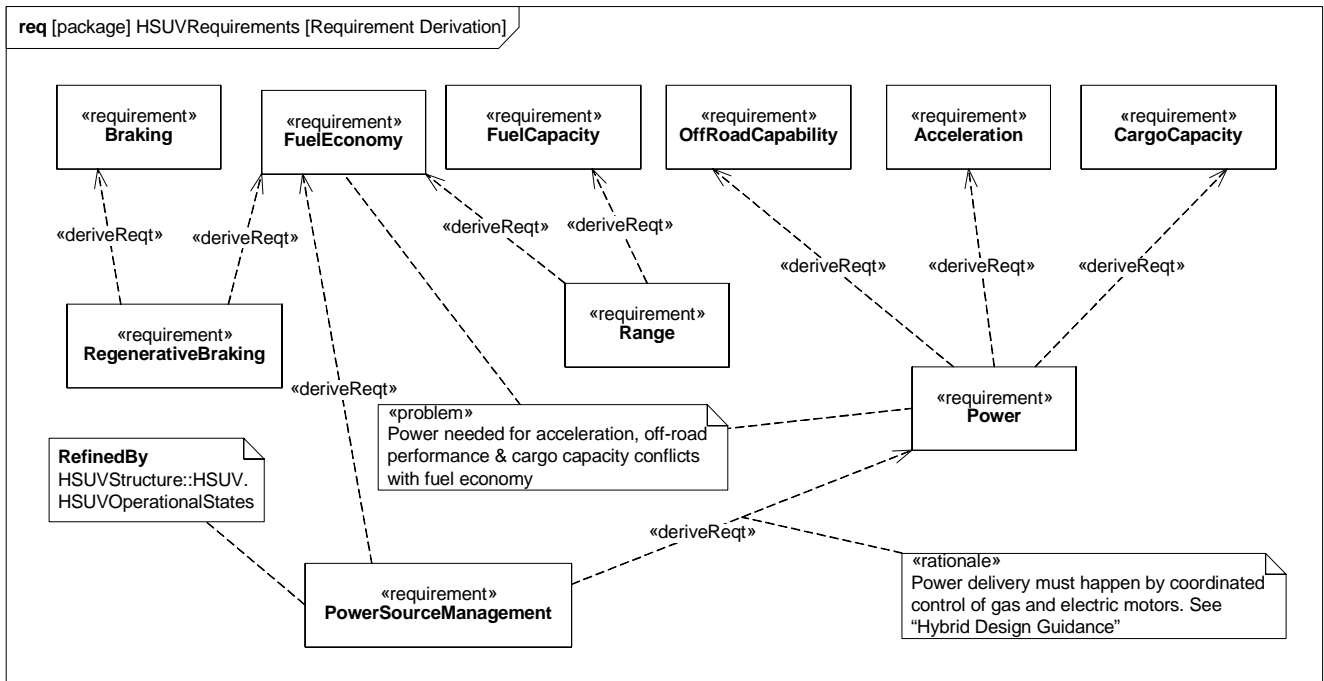


Figure 12 - Establishing Derived Requirements and Rationale from Lowest Tier of Requirements Hierarchy (Requirements Diagram)

4.4.3 Requirement Diagram - Acceleration Requirement Relationships

Figure 13 focuses on the Acceleration requirement, and relates it to other requirements and model elements. The “refine” relation, introduced in Figure 12, shows how the Acceleration requirement is refined by a similarly named use case. The Power requirement is satisfied by the PowerSubsystem, and a Max Acceleration test case verifies the Acceleration requirement.

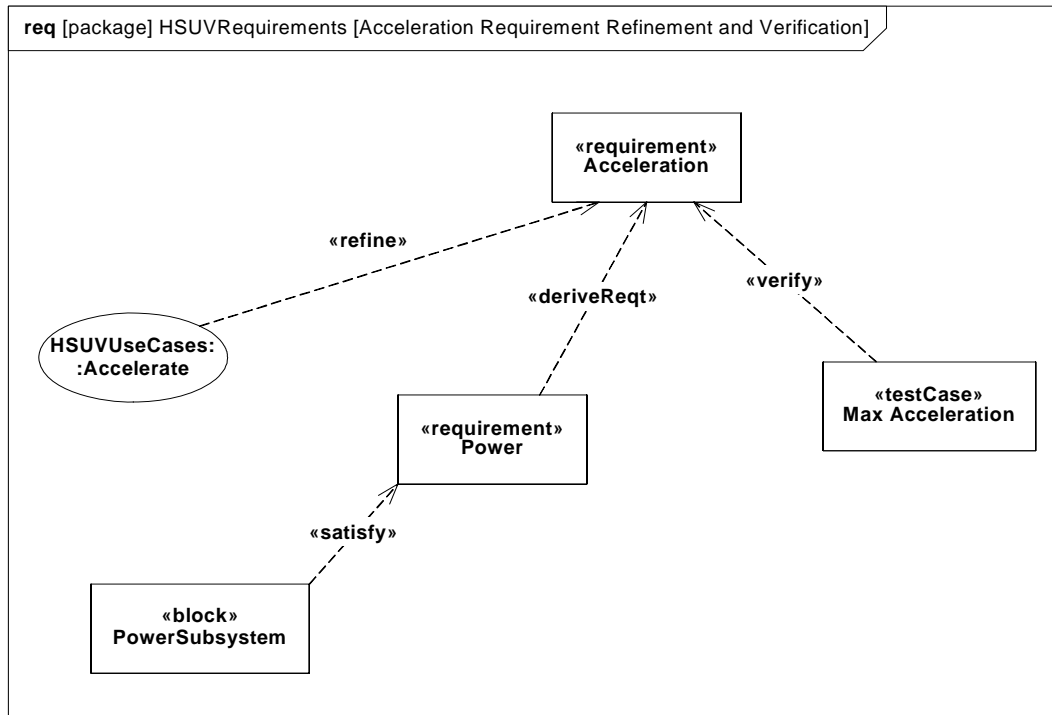


Figure 13 - Acceleration Requirement Relationships (Requirements Diagram)

4.4.4 Table - Requirements Table

Figure 14 contains two diagrams that show requirement containment (decomposition), and requirements derivation in tabular form. This is a more compact representation than the requirements diagrams shown previously.

table [requirement] Performance [Decomposition of Performance Requirement]		
id	name	text
2	Performance	The Hybrid SUV shall have the braking, acceleration, and off-road capability of a typical SUV, but have dramatically better fuel economy.
2.1	Braking	The Hybrid SUV shall have the braking capability of a typical SUV.
2.2	FuelEconomy	The Hybrid SUV shall have dramatically better fuel economy than a typical SUV.
2.3	OffRoadCapability	The Hybrid SUV shall have the off-road capability of a typical SUV.
2.4	Acceleration	The Hybrid SUV shall have the acceleration of a typical SUV.

table [requirement] Performance [Tree of Performance Requirements]							
id	name	relation	id	name	relation	id	name
2.1	Braking	deriveReqt	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReqt	d.1	RegenerativeBraking			
2.2	FuelEconomy	deriveReqt	d.2	Range			
4.2	FuelCapacity	deriveReqt	d.2	Range			
2.3	OffRoadCapability	deriveReqt	d.4	Power	deriveReqt	d.2	PowerSourceManagement
2.4	Acceleration	deriveReqt	d.4	Power	deriveReqt	d.2	PowerSourceManagement
4.1	CargoCapacity	deriveReqt	d.4	Power	deriveReqt	d.2	PowerSourceManagement

Figure 14 - Requirements Relationships Expressed in Tabular Format (Table)

4.5 Breaking Down the Pieces (Block Definition Diagrams, Internal Block Diagrams)

4.5.1 Block Definition Diagram - Automotive Domain

Figure 15 provides definition for the concepts previously shown in the context diagram. Note that the interactions DriveBlackBox and StartVehicleBlackBox (described in Section 4.3, “Elaborating Behavior (Sequence and State Machine Diagrams),” on page 7) are depicted as owned by the AutomotiveDomain block.

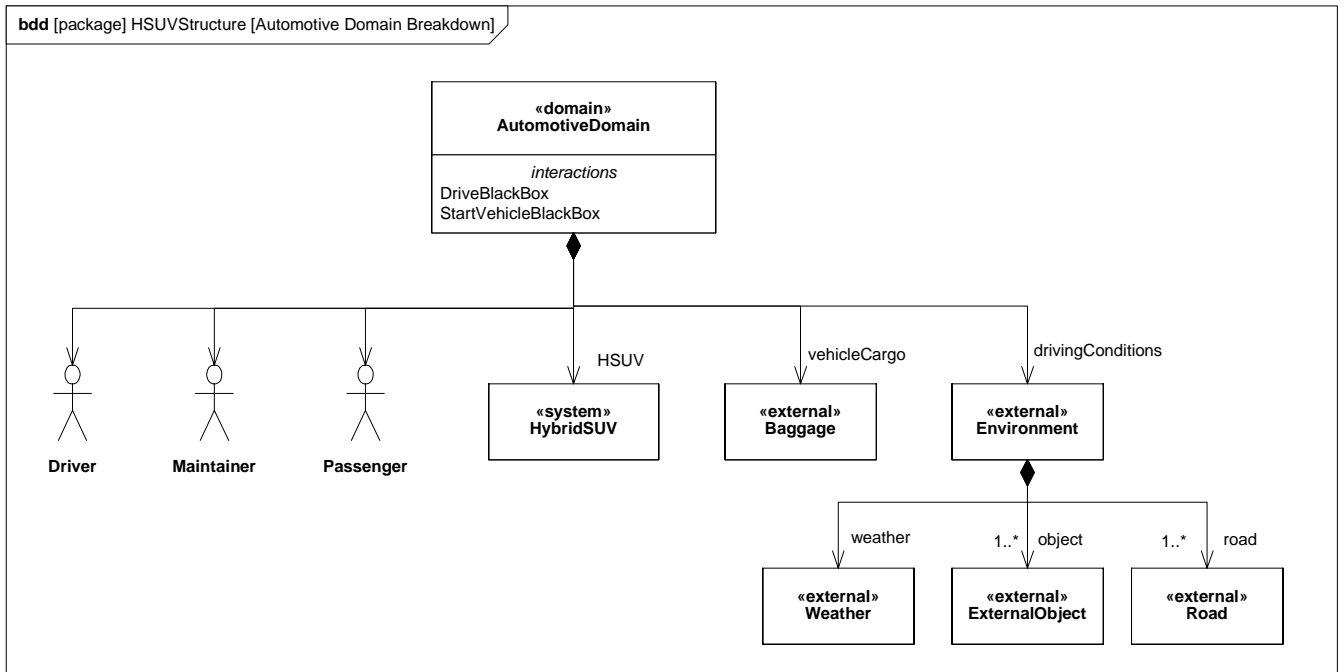


Figure 15 - Defining the Automotive Domain (compare with Figure 4) - (Block Definition Diagram)

4.5.2 Block Definition Diagram - Hybrid SUV

Figure 16 defines components of the HybridSUV block. Note that the BrakePedal and WheelHubAssembly are used by, but not contained in, the PowerSubsystem block.

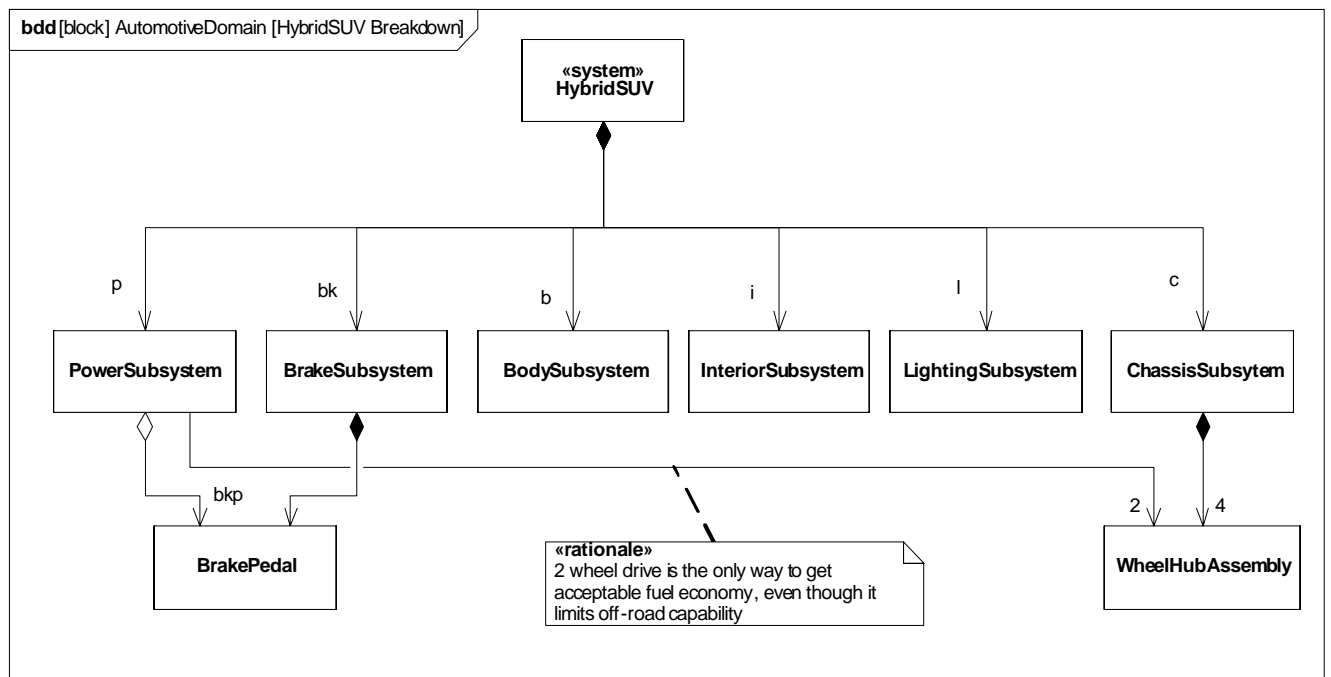


Figure 16 - Defining Structure of the Hybrid SUV System (Block Definition Diagram)

4.5.3 Internal Block Diagram - Hybrid SUV

Figure 17 shows how the top level model elements in the above diagram are connected together in the HybridSUV block.

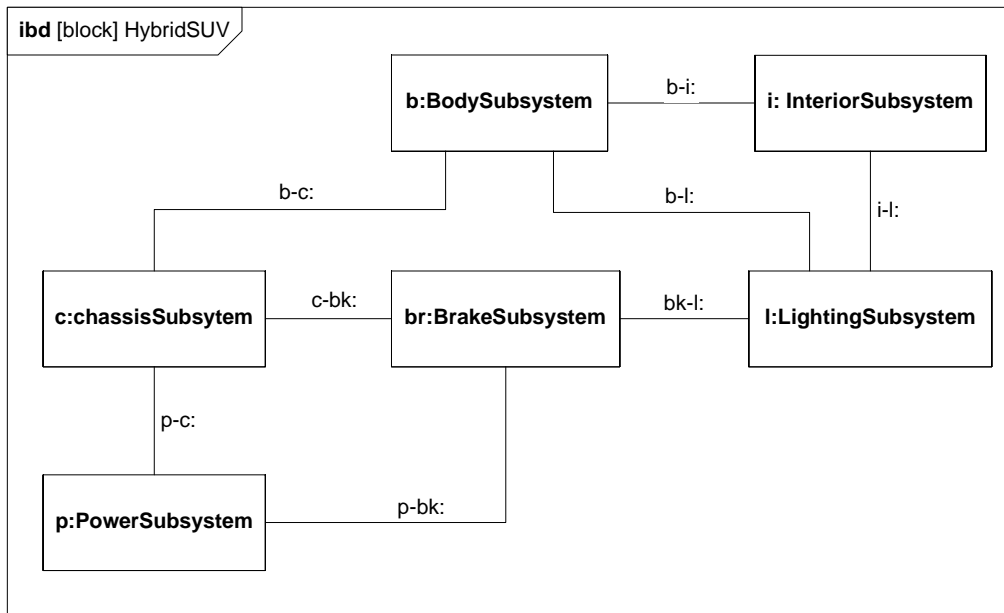


Figure 17 - Internal Structure of Hybrid SUV (Internal Block Diagram)

4.5.4 Block Definition Diagram - Power Subsystem

Figure 18 defines the next level of decomposition, namely the components of the PowerSubsystem block. Note how the use of white diamond (shared aggregation) on FrontWheel, BrakePedal, and others denotes the same “use-not-composition” kind of relationship previously shown in Figure 16.

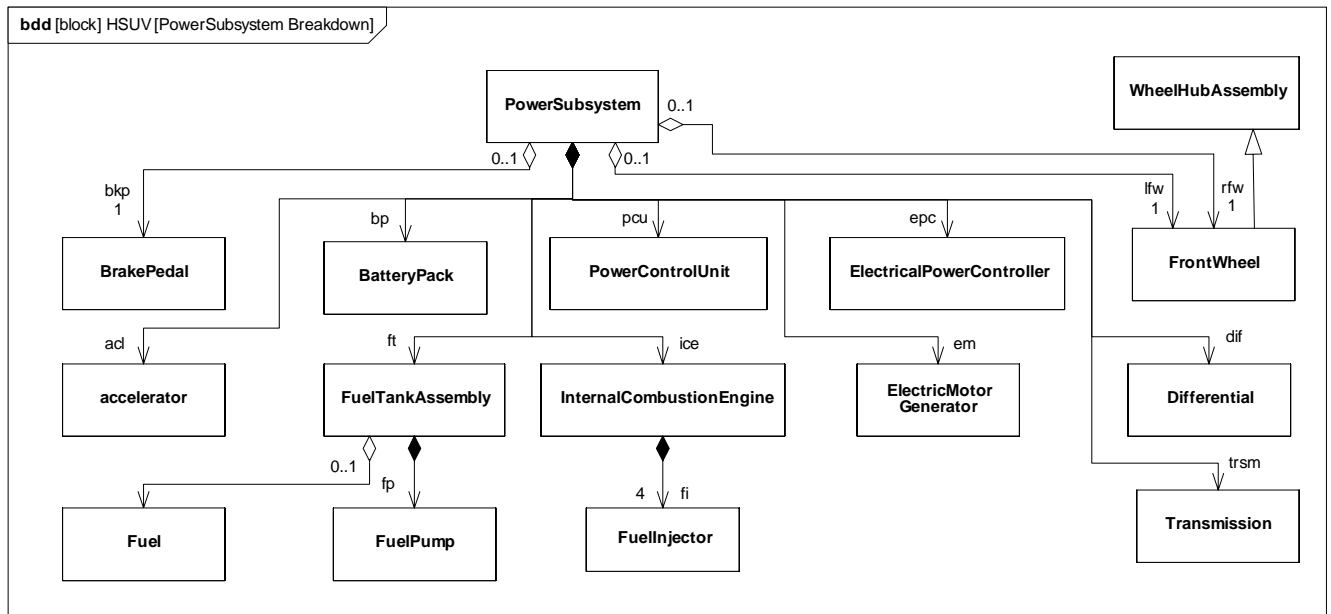


Figure 18 - Defining Structure of Power Subsystem (Block Definition Diagram)

4.5.5 Internal Block Diagram for the “Power Subsystem”

Figure 19 shows how the parts of the PowerSubsystem block, as defined in the diagram above, are used. It shows «connectors» between parts, «clientServerPorts», «flowPorts», «atomicFlowPorts», and «itemFlows». The dashed borders on FrontWheel and BrakePedal denote the “use-not-composition” relationship depicted elsewhere in Figure 16 and Figure 18. The dashed borders on Fuel denote a store, which keeps track of the amount and mass of fuel in the FuelTankAssy. This is also depicted in Figure 18.

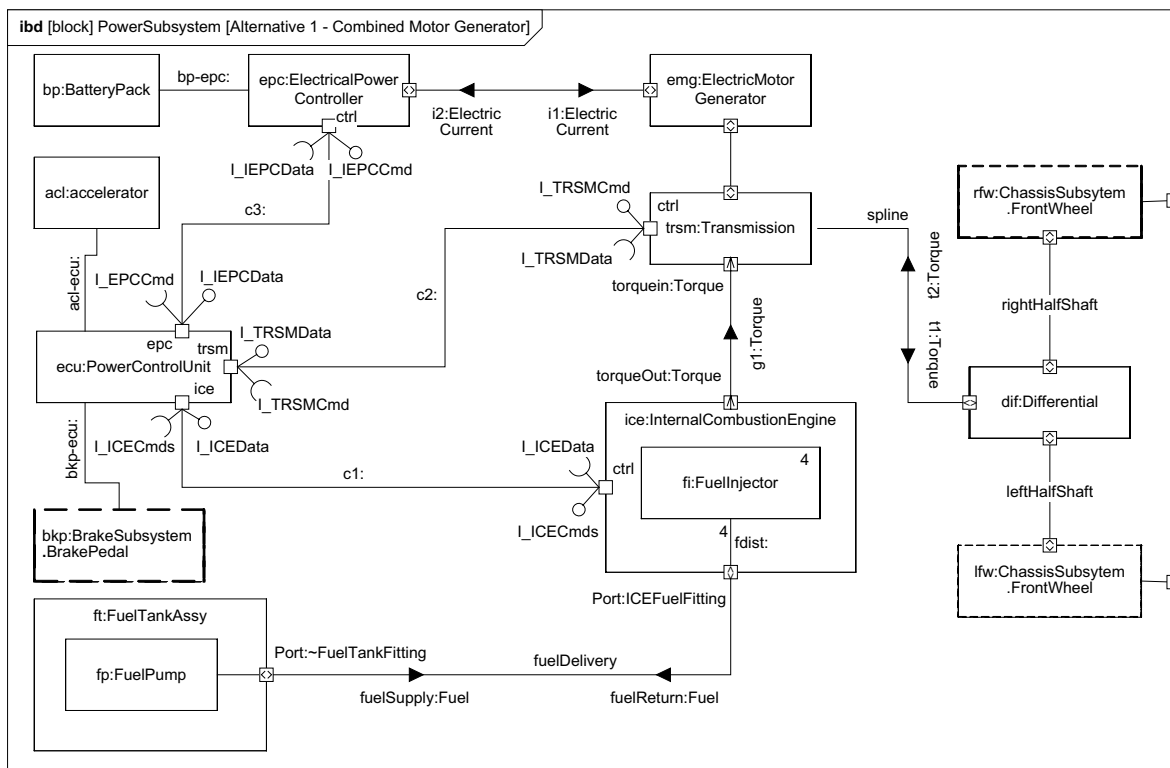


Figure 19 - Internal Structure of the Power Subsystem (Internal Block Diagram)

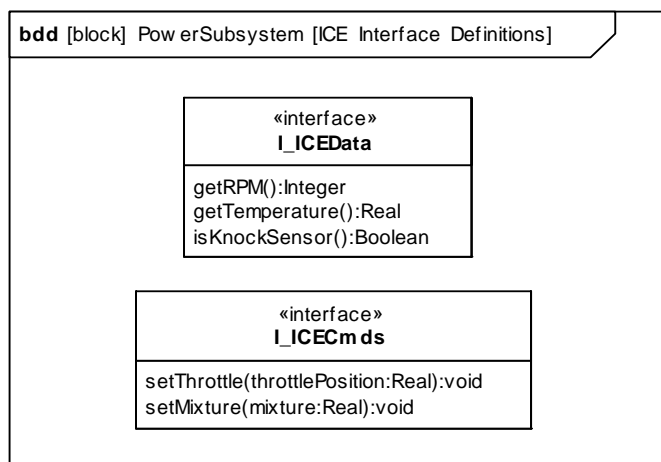


Figure 20 - Interfaces Typing StandardPorts Internal to the Power Subsystem (Block Definition Diagram)

Figure 20 provides definition of the interfaces applied to Standard Ports associated with connector c1 in Figure 19.

4.6 Defining Ports and Flows

4.6.1 Block Definition Diagram - ICE Interface

For purposes of example, the StandardPorts and related point-to-point connectors in Figure 19 are being refined into a common bus architecture. For this example, FlowPorts have been used to model the bus architecture. Figure 21 is an incomplete first step in the refinement of this bus architecture, as it begins to identify the flow specification for the InternalCombustionEngine, the Transmission, and the ElectricalPowerController..

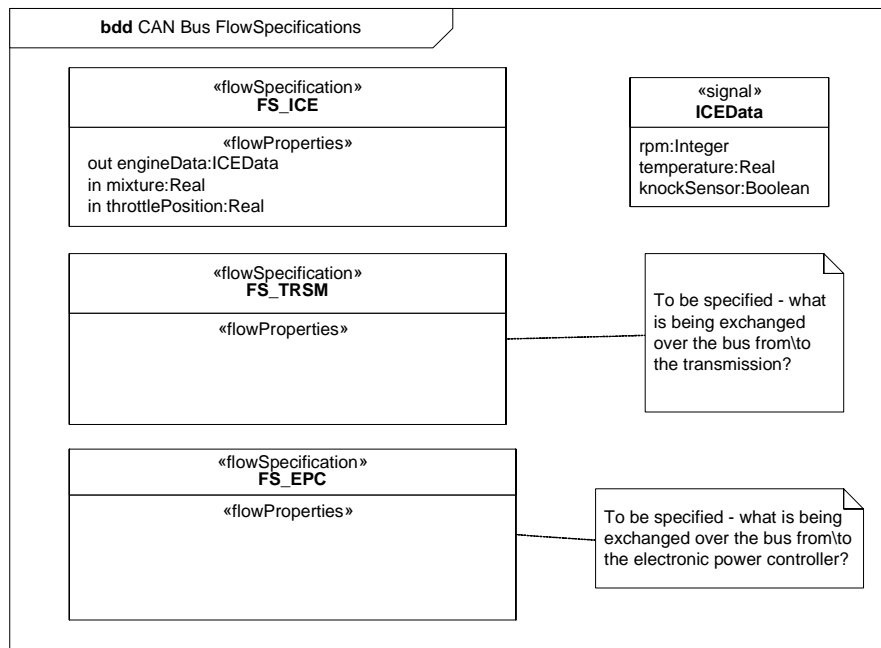


Figure 21 - Initially Defining Flow Specifications for the CAN Bus (Block Definition Diagram)

4.6.2 Internal Block Diagram - CANbus

Figure 22 continues the refinement of this Controller Area Network (CAN) bus architecture using FlowPorts. The explicit structural allocation between the original connectors of Figure 19 and this new bus architecture is shown in Figure 36.

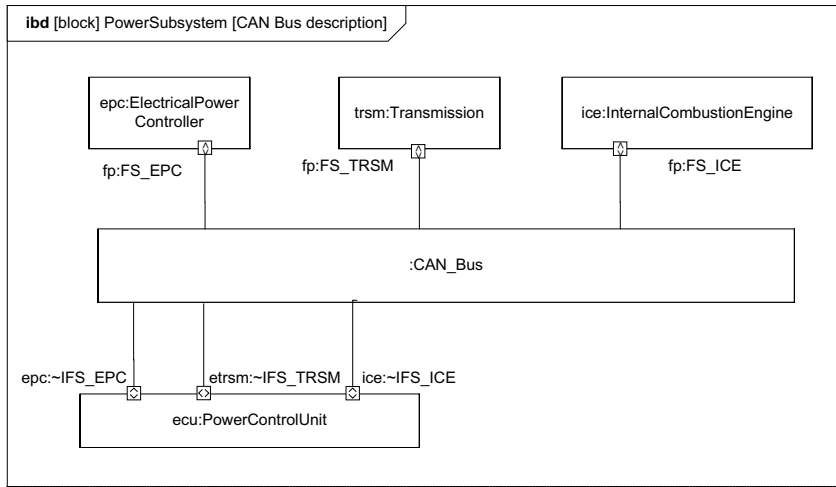


Figure 22 - Consolidating Interfaces into the CAN Bus. (Internal Block Diagram)

4.6.3 Block Definition Diagram - Fuel Flow Properties

The FlowPorts on the FuelTankAssembly and InternalCombustionEngine (as shown in Figure 19) are defined in Figure 23.

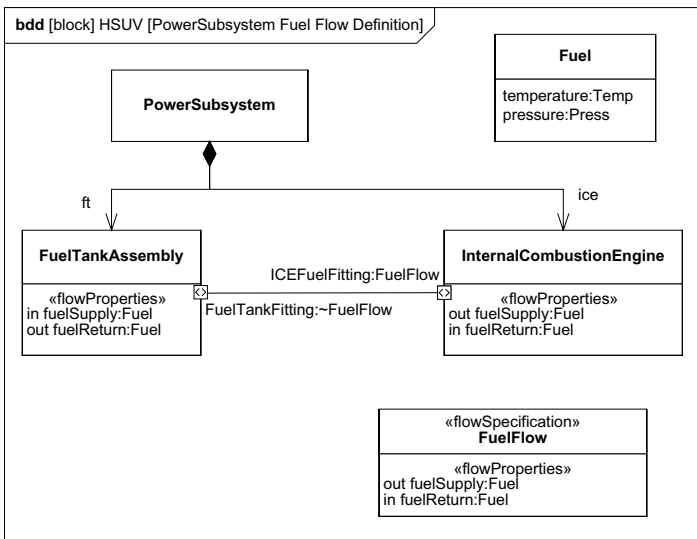


Figure 23 - Elaborating Definition of Fuel Flow. (Block Definition Diagram)

4.6.4 Parametric Diagram - Fuel Flow

Figure 24 is a parametric diagram showing how fuel flowrate is related to FuelDemand and FuelPressure value properties.

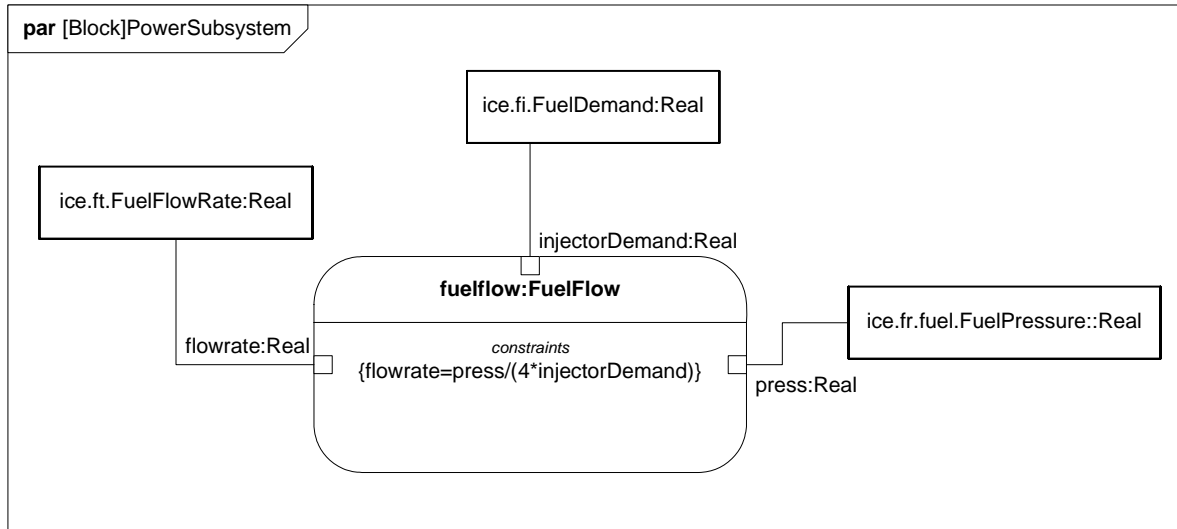


Figure 24 - Defining Fuel Flow Constraints (Parametric Diagram)

4.6.5 Internal Block Diagram - Fuel Distribution

Figure 25 shows how the connectors fuelDelivery and fdist on Figure 19 have been expanded to include design detail. The fuelDelivery connector is actually two connectors, one carrying fuelSupply and the other carrying fuelReturn. The fdist connector inside the InternalCombustionEngine block has been expanded into the fuel regulator and fuel rail parts. These more detailed design elements are related to the original connectors using the allocation relationship. The Fuel store represents a quantity of fuel in the FuelTankAssy, which is drawn by the FuelPump for use in the engine, and is refreshed, to some degree, by fuel returning to the FuelTankAssy via the FuelReturnLine.

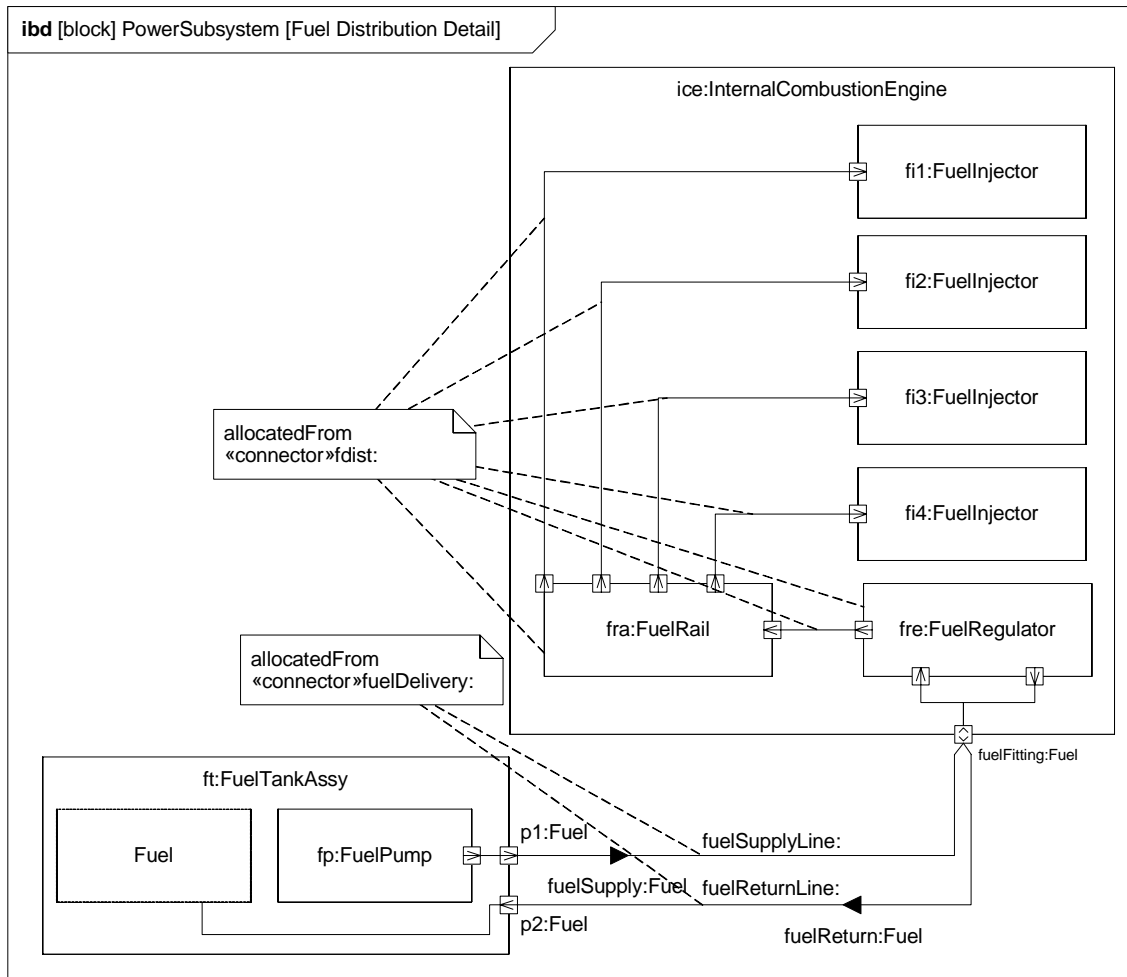


Figure 25 - Detailed Internal Structure of Fuel Delivery Subsystem (Internal Block Diagram)

4.7 Analyze Performance (Constraint Diagrams, Timing Diagrams, Views)

4.7.1 Block Definition Diagram - Analysis Context

Figure 26 defines the various model elements that will be used to conduct analysis in this example. It depicts each of the constraint blocks/equations that will be used for the analysis, and key relationships between them.

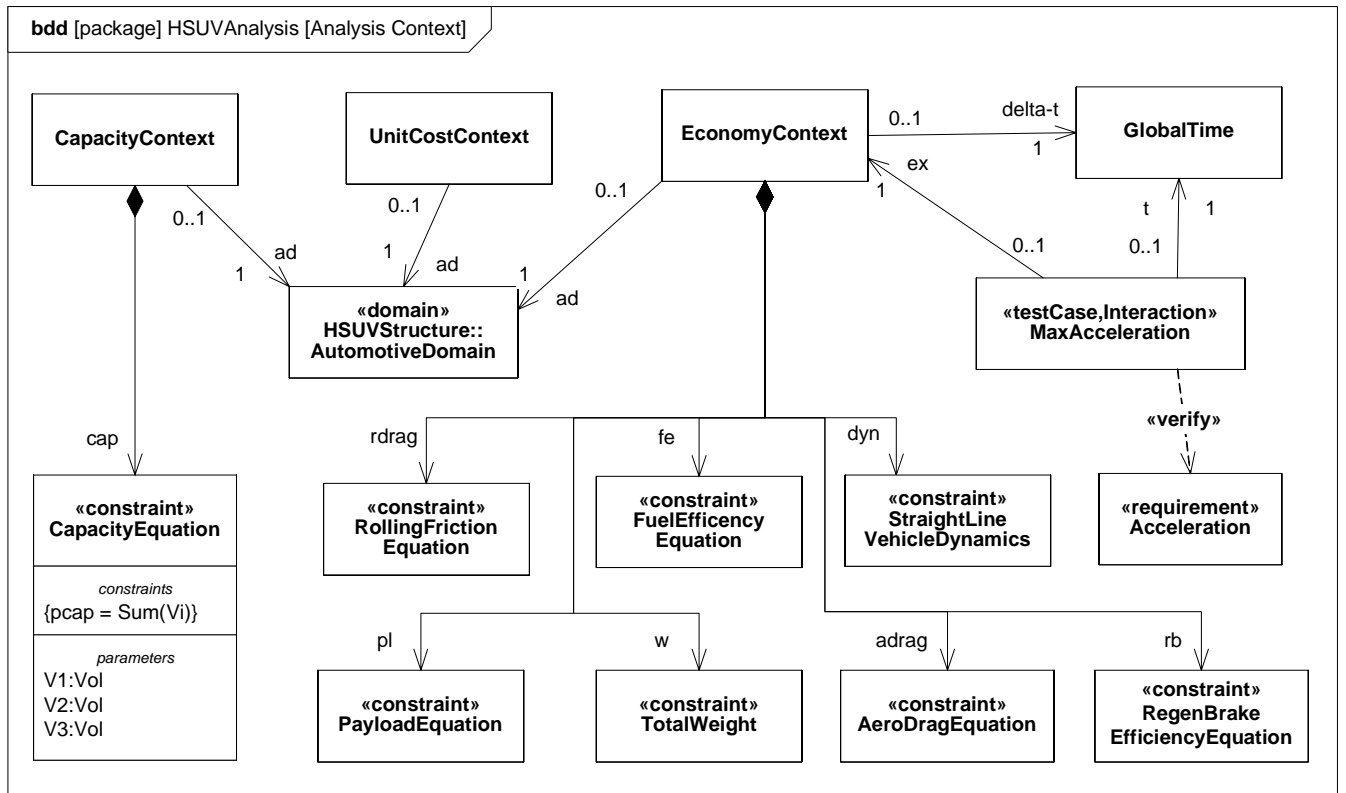


Figure 26 - Defining Analyses for Hybrid SUV Engineering Development (Block Definition Diagram)

4.7.2 Package Diagram - Performance View Definition

Figure 27 shows the user-defined Performance Viewpoint, and the elements that populate the HSUV specific PerformanceView. The PerformanceView itself may contain of a number of diagrams depicting the elements it contains.

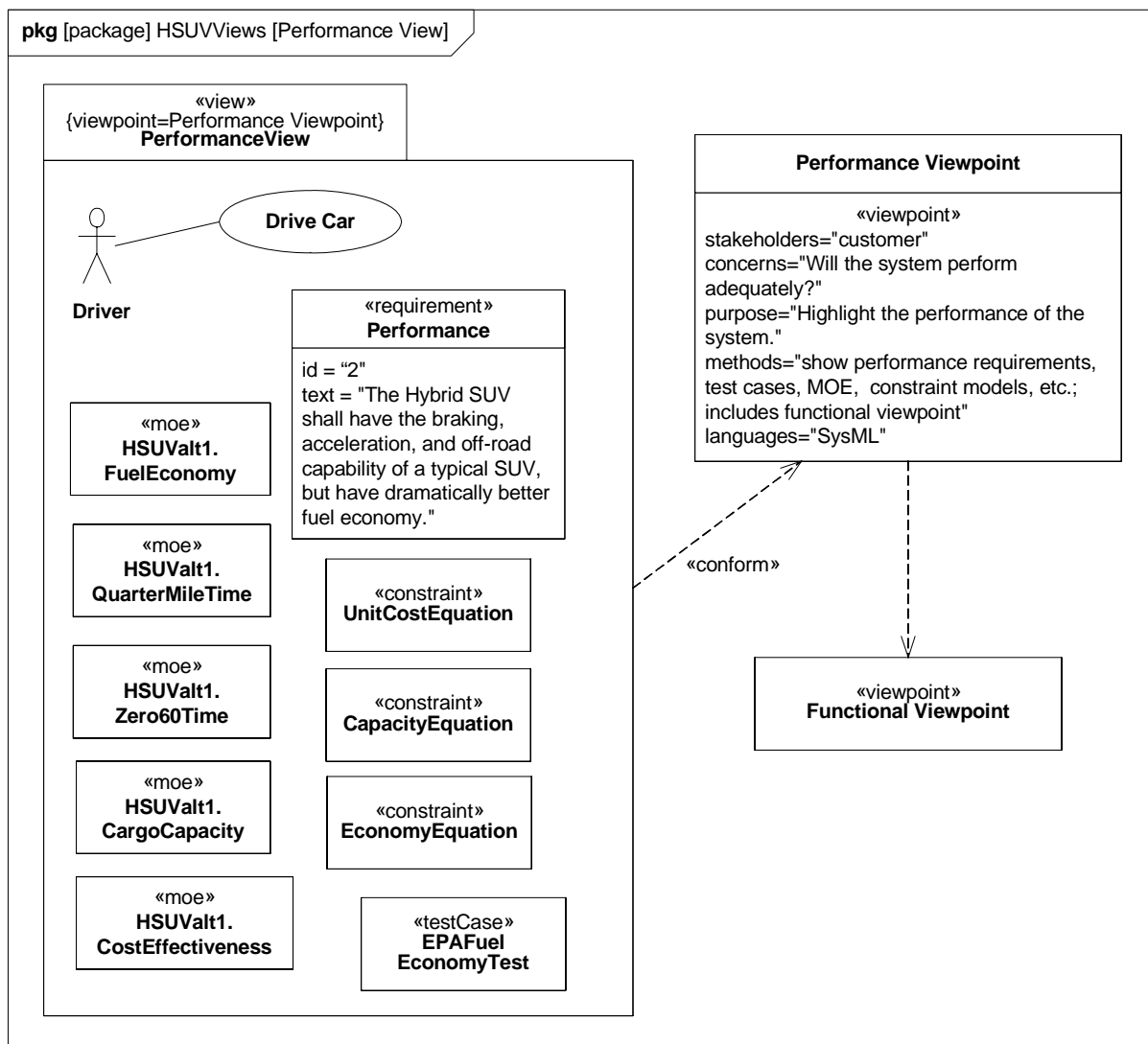


Figure 27 - Establishing a Performance View of the User Model (Package Diagram)

4.7.3 Parametric Diagram - Measures of Effectiveness

Measure of Effectiveness is a user defined stereotype. Figure 28 shows how the overall cost effectiveness of the HSUV will be evaluated. It shows the particular measures of effectiveness for one particular alternative for the HSUV design, and can be reused to evaluate other alternatives.

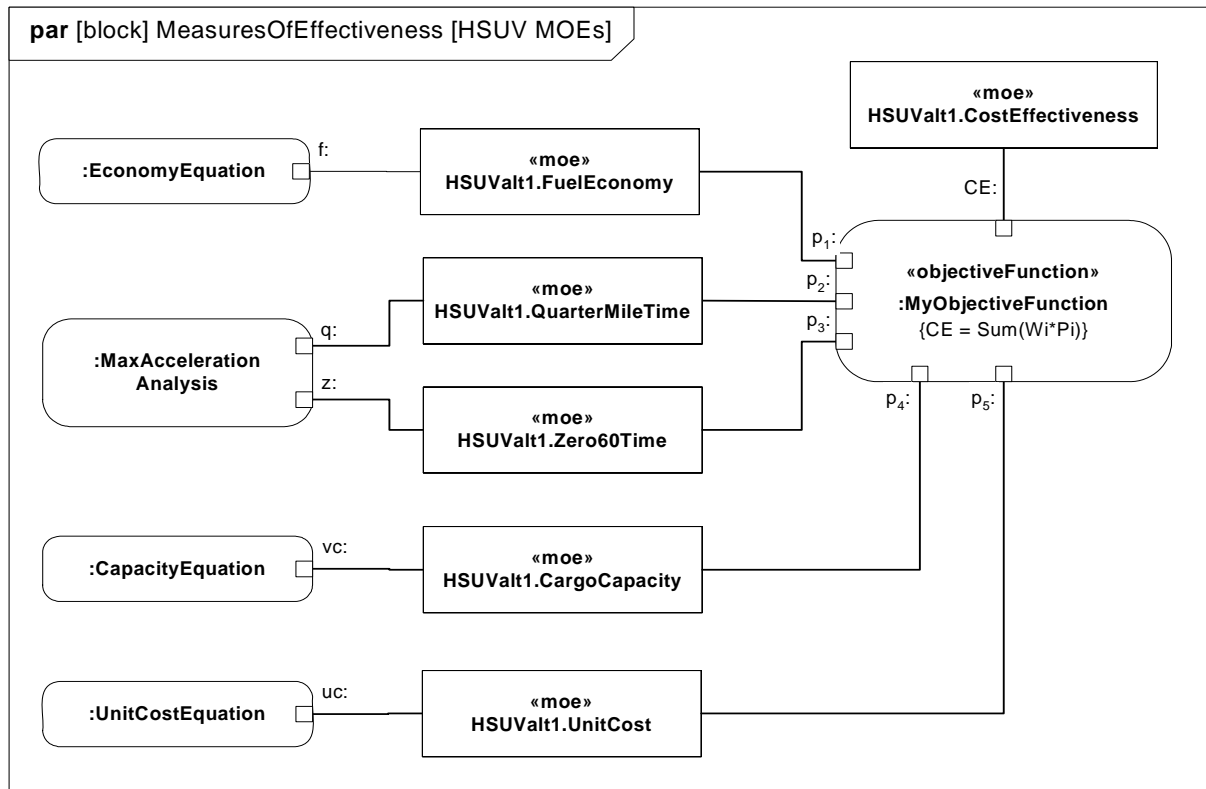


Figure 28 - Defining Measures of Effectiveness and Key Relationships (Parametric Diagram)

4.7.4 Parametric Diagram - Economy

Since overall fuel economy is a key requirement on the HSUV design, this example applies significant detail in assessing it. Figure 29 shows the constraint blocks and properties necessary to evaluate fuel economy.

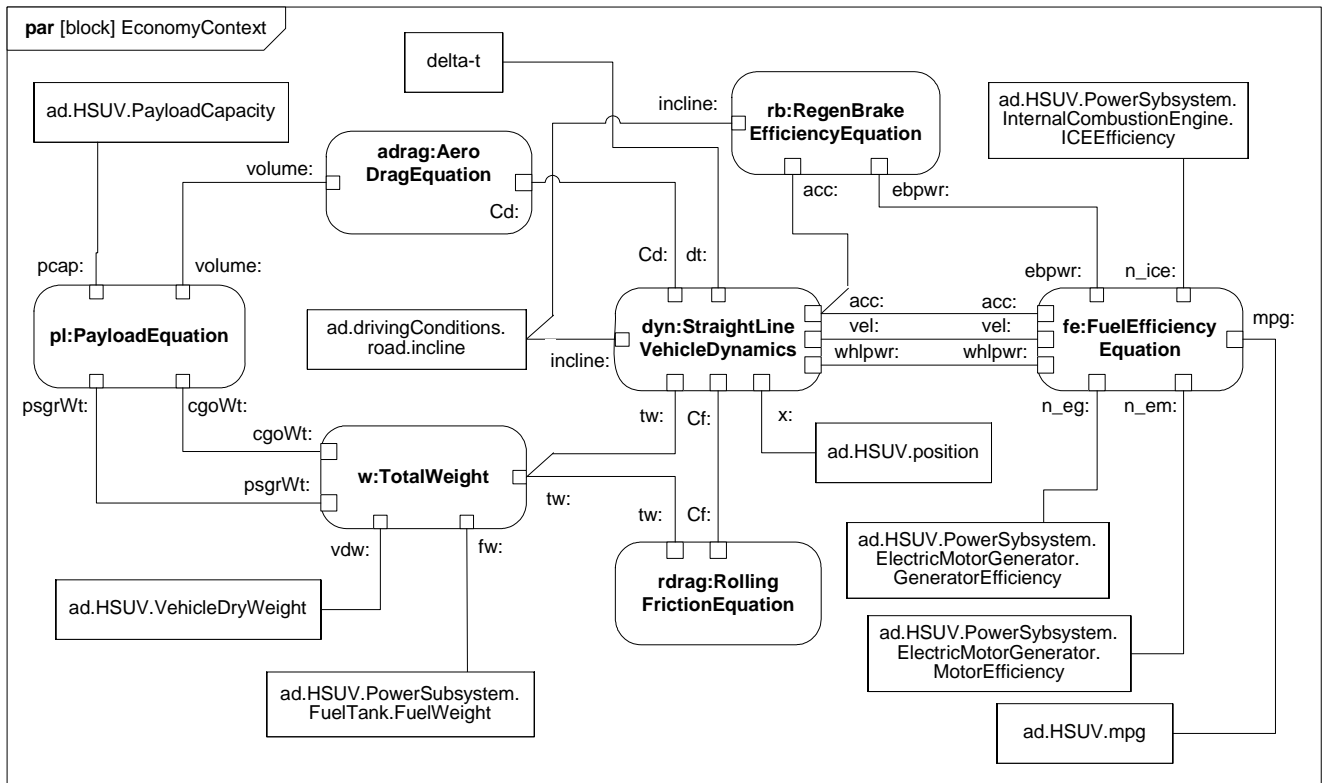


Figure 29 - Establishing Mathematical Relationships for Fuel Economy Calculations (Parametric Diagram)

4.7.5 Parametric Diagram - Dynamics

The **StraightLineVehicleDynamics** constraint block from Figure 29 has been expanded in Figure 30. **ConstraintNotes** are used, which identify each constraint using curly brackets { }. In addition, **Rationale** has been used to explain the meaning of each constraint maintained.

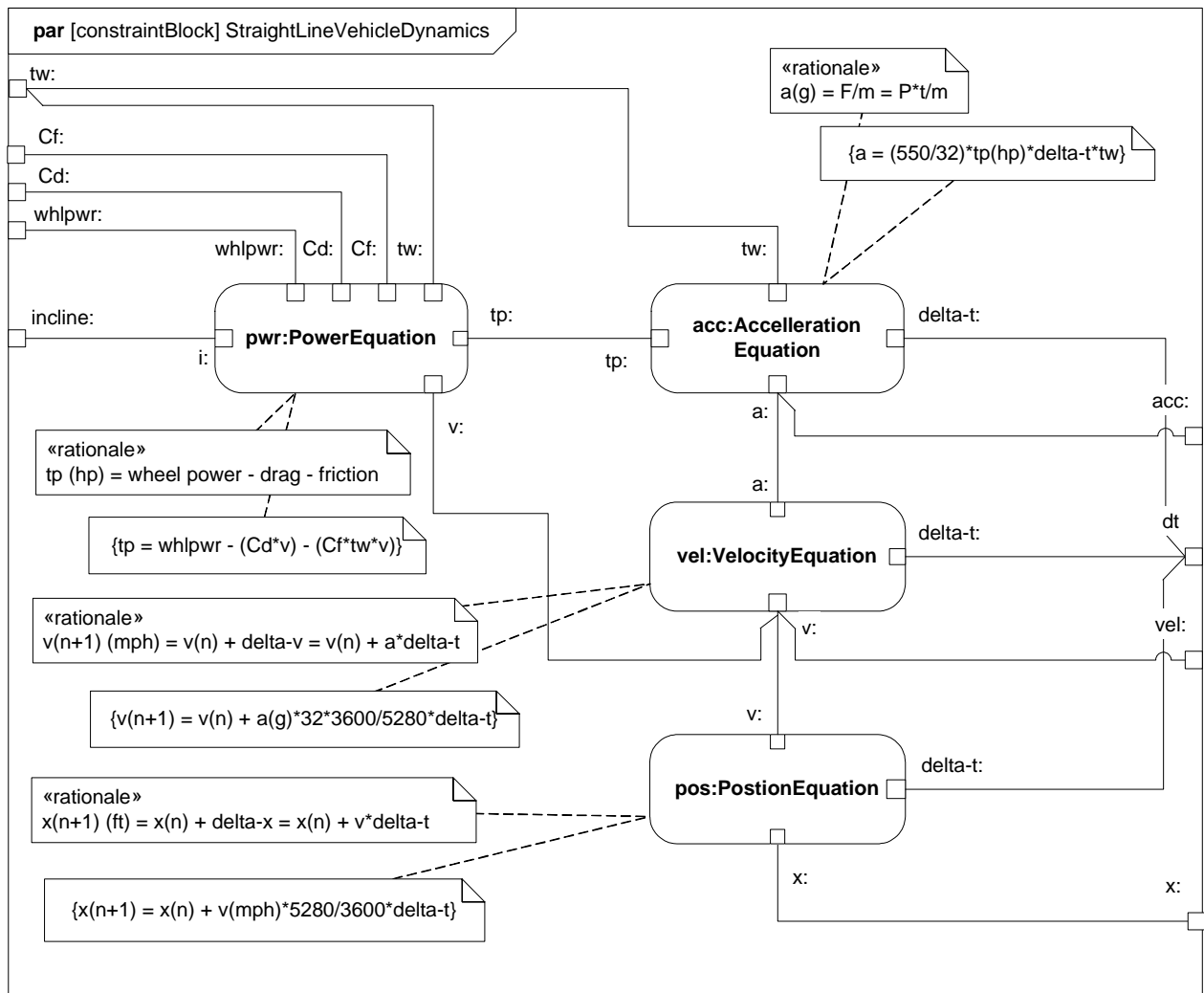


Figure 30 - Straight Line Vehicle Dynamics Mathematical Model (Parametric Diagram)

The constraints and parameters in Figure 30 are detailed in Figure 31 in Block Definition Diagram format.

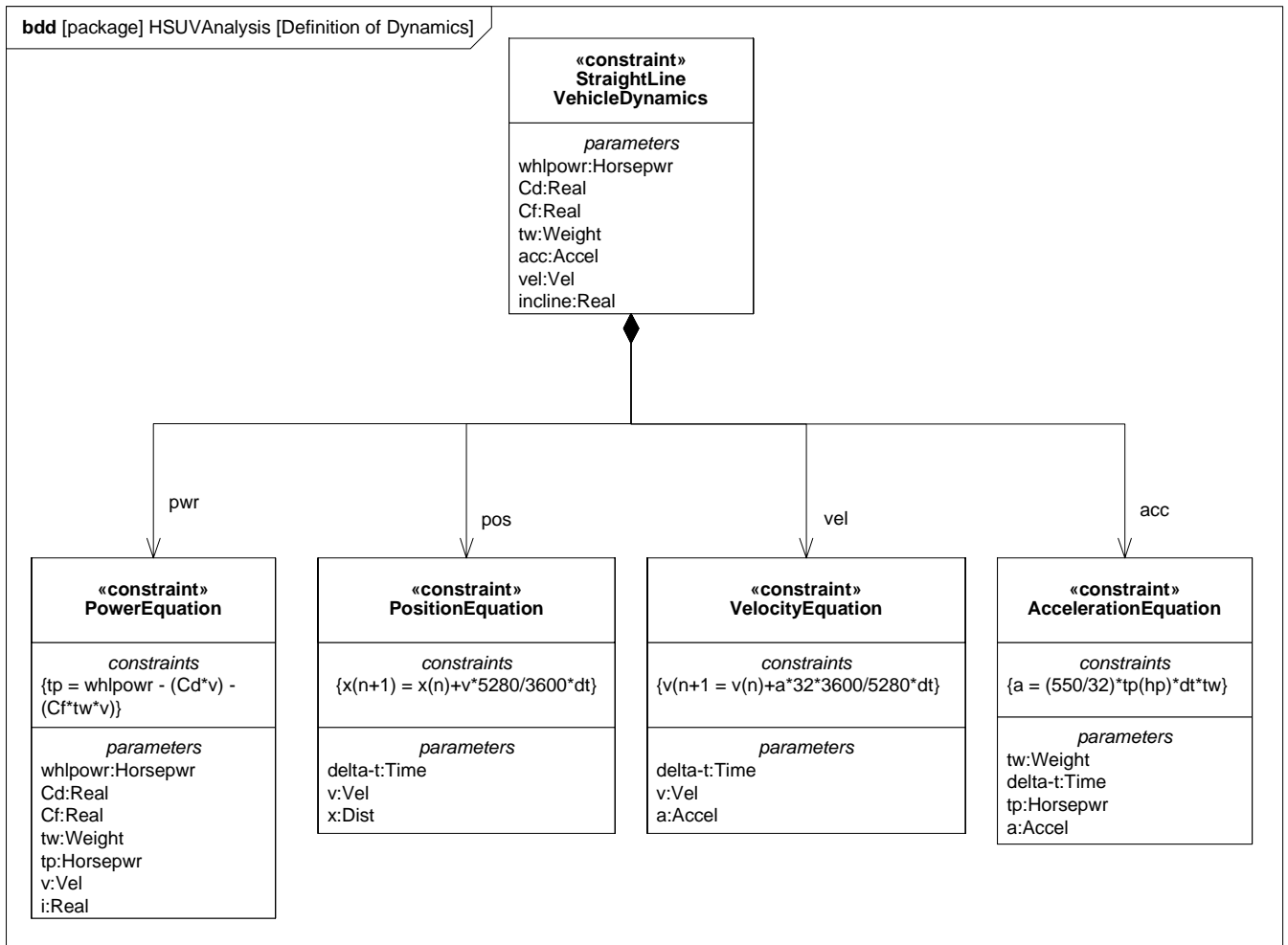


Figure 31 - Defining Straight-Line Vehicle Dynamics Mathematical Constraints (Block Definition Diagram)

Note the use of valueTypes originally defined in Figure 2.

4.7.6 (Non-Normative) Timing Diagram - 100hp Acceleration

Timing diagrams, while included in UML 2, are not directly supported by SysML. For illustration purposes, however, the interaction shown in Figure 32 was generated based on the constraints and parameters of the *StraightLineVehicleDynamics* constraintBlock, as described in the Figure 30. It assumes a constant 100hp at the drive wheels, 4000lb gross vehicle weight, and constant values for *Cd* and *Cf*.

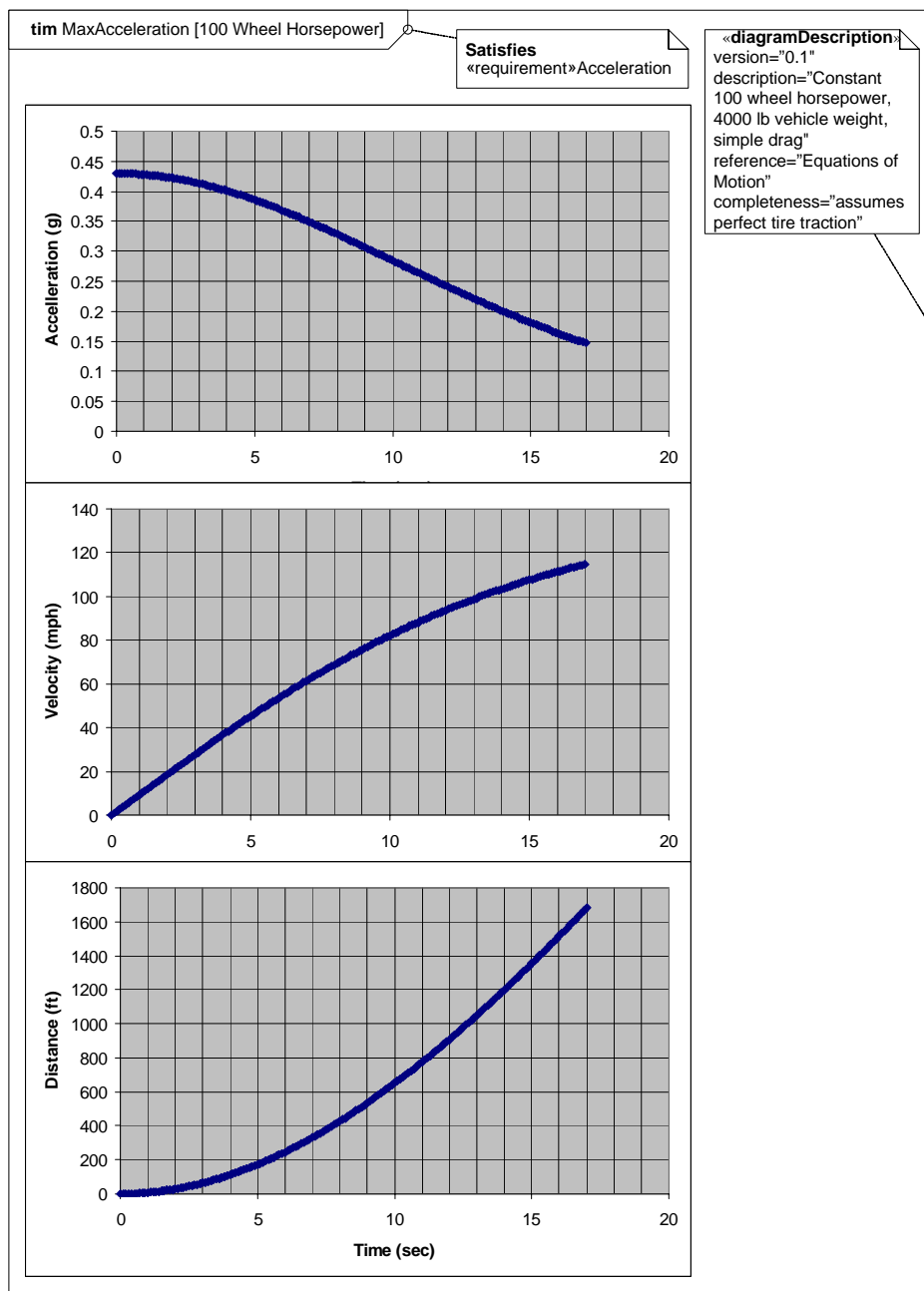


Figure 32 - Results of Maximum Acceleration Analysis (Timing Diagram)

4.8.2 Block Definition Diagram - Acceleration

Figure 34 defines a decomposition of the activities and objectFlows from the activity diagram in Figure 33.

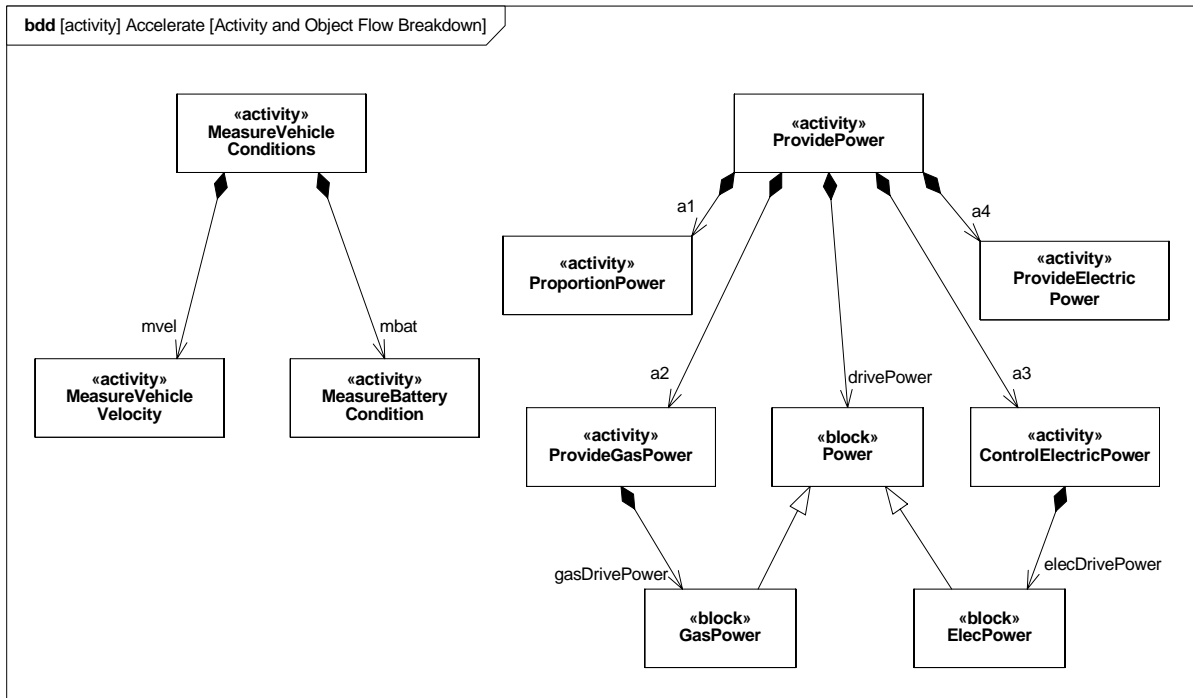


Figure 34 - Decomposition of “Accelerate” Function (Block Definition diagram)

4.8.3 Activity Diagram (EFFBD) - Acceleration (detail)

Figure 35 shows the ProvidePower activity, which includes Actions invoking the decomposed Activities and ObjectNodes from Figure 34. It also uses AllocateActivityPartitions and an allocation callout to explicitly allocate activities and an object flow to parts in the PowerSubsystem block.

Note that the incoming and outgoing object flows for the ProvidePower activity have been decomposed. This was done to distinguish the flow of electrically generated mechanical power and gas generated mechanical power, and to provide further insight into the specific vehicle conditions being monitored.

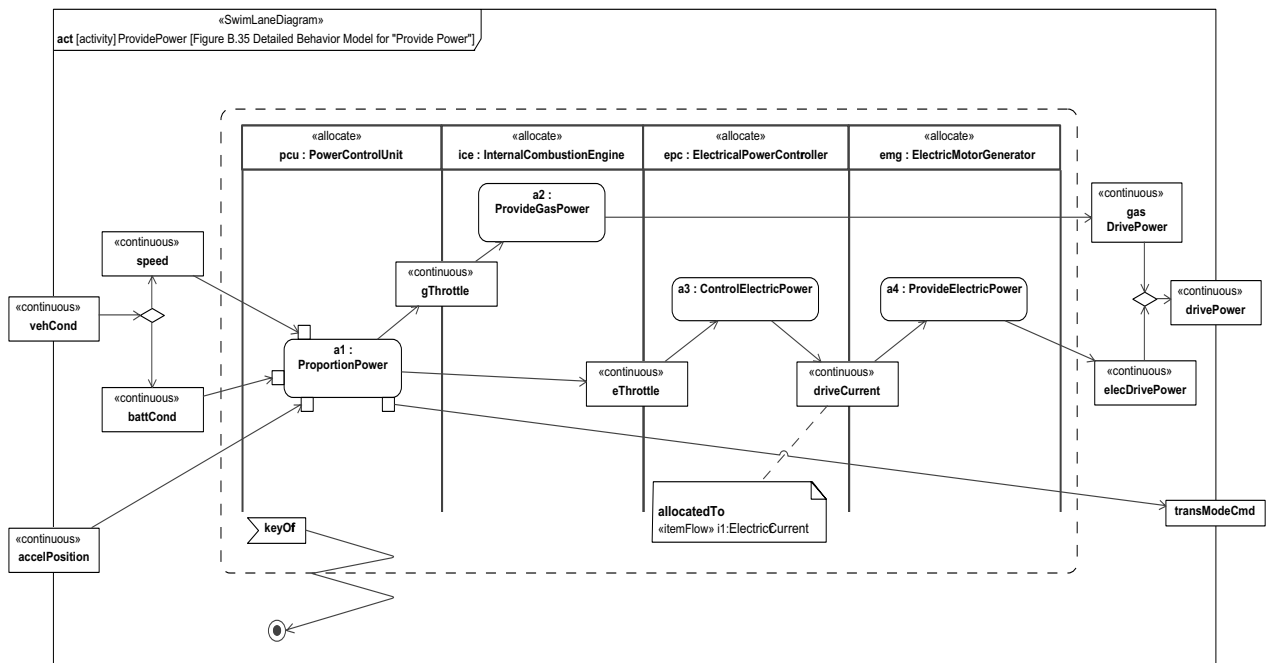


Figure 35 - Detailed Behavior Model for "Provide Power" (Activity Diagram)
 Note hierarchical consistency with Figure 33.

4.8.4 Internal Block Diagram - Power Subsystem Behavioral and Flow Allocation

Figure 36 depicts a subset of the PowerSubsystem, specifically showing the allocation relationships generated in Figure 35.

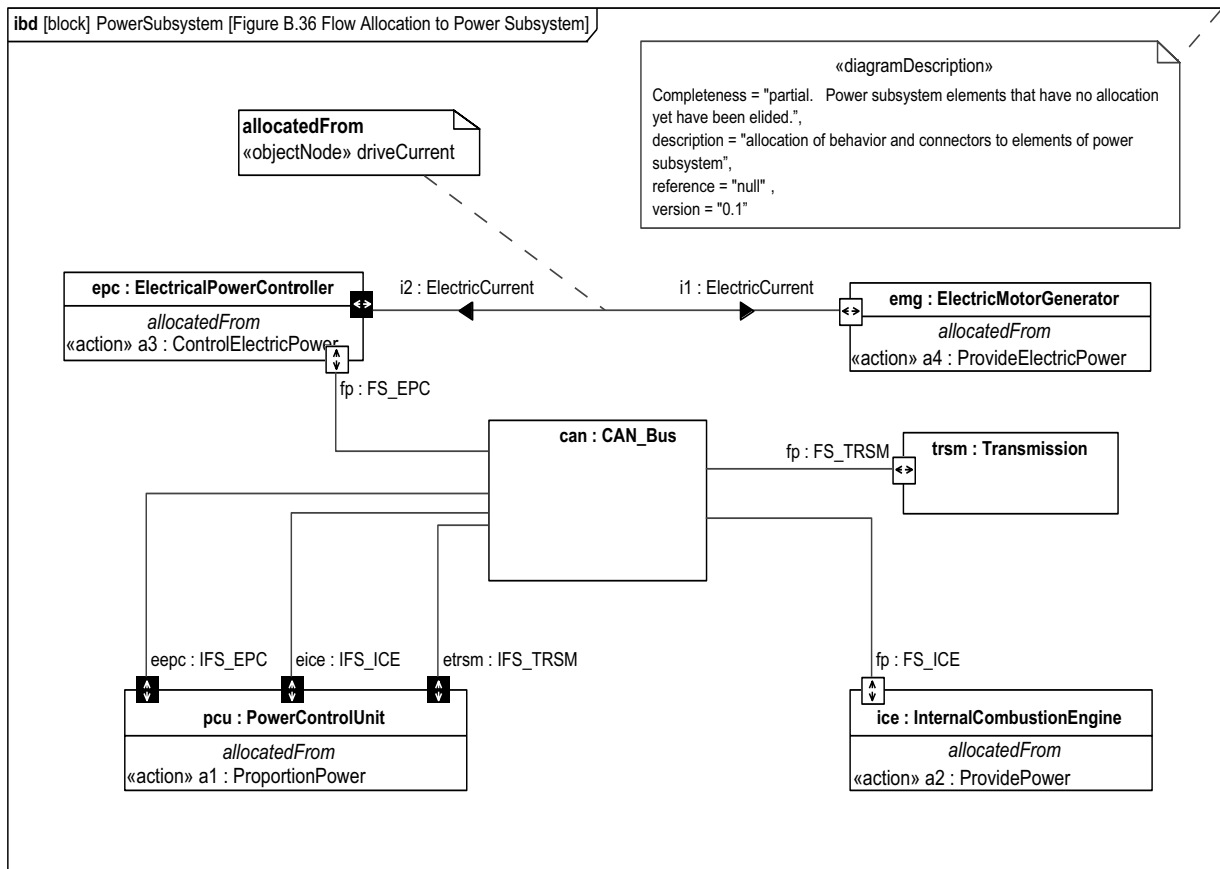


Figure 36 - Flow Allocation to Power Subsystem (Internal Block Diagram)

4.8.5 Table - Acceleration Allocation

Figure 37 shows the same allocation relationships shown in Figure 36, but in a more compact tabular representation.

bdd [package] HSUV Behavior [Figure B.37 Tabular Representation of Allocation from "Accelerate" Behavior Model to Power Subsystem]

Type	Name	End	Relation	End	Type	Name
action	a1 : ProportionPower	from	allocate	to	part	ecu : PowerControlUnit
action	a2 : ProvideGasPower	from	allocate	to	part	ice : InternalCombustionEngine
action	a3 : ControlElectricPower	from	allocate	to	part	epc : ElectricPowerController
action	a4 : ProvideElectricPower	from	allocate	to	part	emg : ElectricMotorGenerator
objectFlow	o6	from	allocate	to	connector	epc-emg.1

Figure 37 - Tabular Representation of Allocation from "Accelerate" Behavior Model to Power Subsystem (Table)

4.8.6 Internal Block Diagram: Property Specific Values - EPA Fuel Economy Test

Figure 38 shows a particular Hybrid SUV (VIN number) satisfying the EPA fuel economy test. Serial numbers of specific relevant parts are indicated.

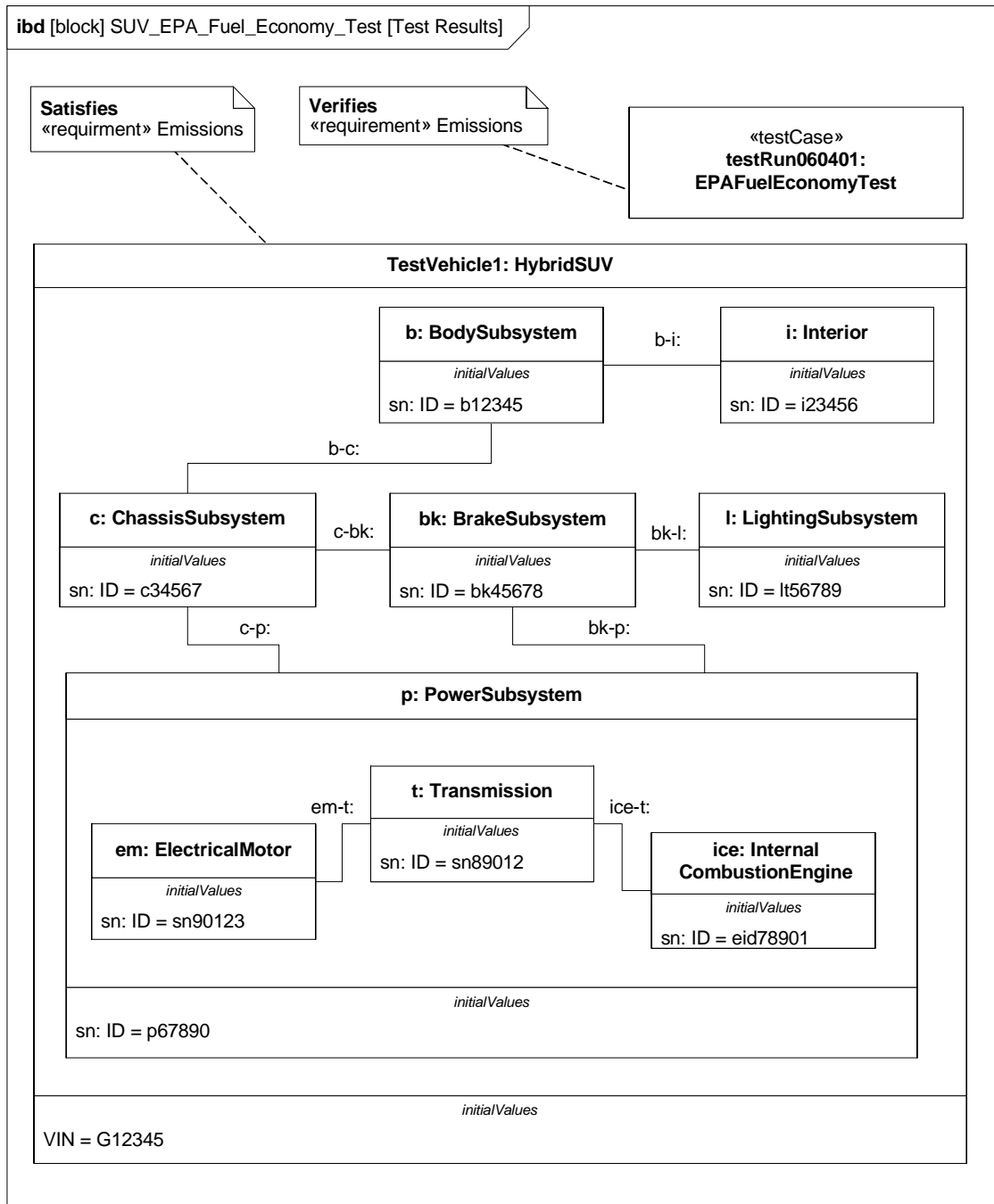


Figure 38 - Special Case of Internal Block Diagram Showing Reference to Specific Properties (serial numbers)