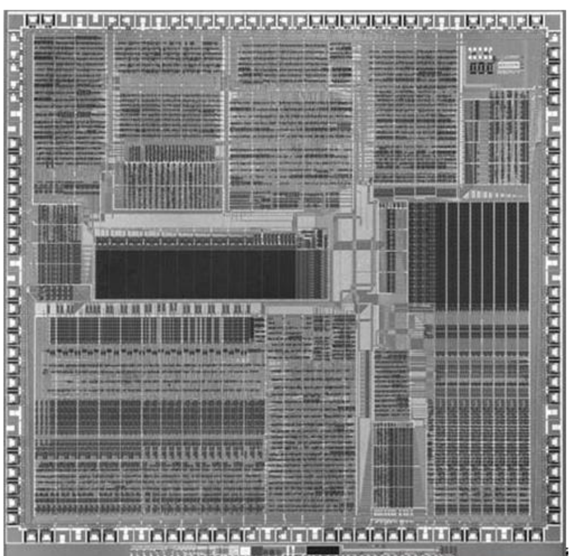
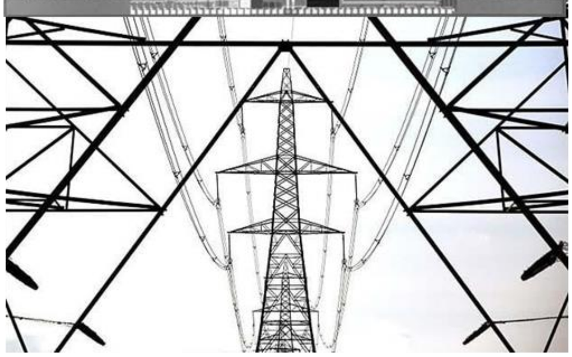
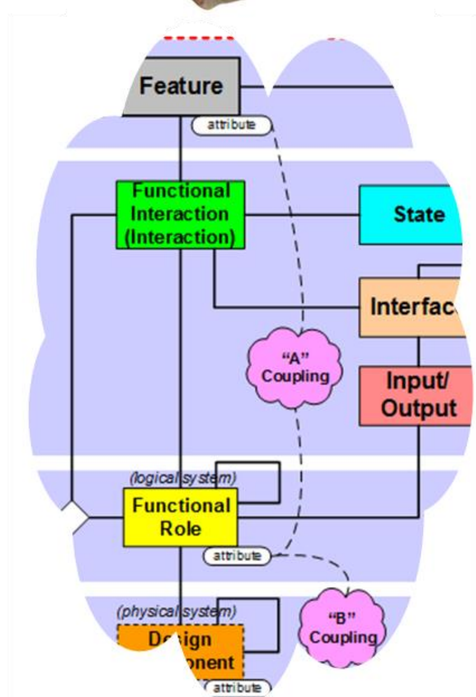
S*Models and the S*Metamodel: A Primer

INCOSE/OMG MBSE Patterns Working Group







ISBN: ____________________

Contents

Table of Figures

Document Change History

Document Version	Revision Date	By	Updated
1.1-1.2	01.01.2024 – 02.22.2024	Bill Schindel	Populate from existing materials and working group outline

1 Introduction

1.1 Document Purpose, Scope, Intended Readership

This publication is a light "primer" introduction to Systematica (S*) Models, the S*Metamodel, and their purposes and uses, in both Model-Based Systems Engineering (MBSE) and other model-based practices.

What follows does not assume the reader is already proficient with models, modeling, or systems engineering. Much of this content explains the "why" of S*Models, with only limited samplings of the "how". It may usefully be read by both newcomers to models as well as those who are more expert in the subject. References are provided for those interested in going further.

1.2 The INCOSE/OMG MBSE Patterns Working Group

This document is a publication of the INCOSE/OMG MBSE Patterns Working Group, a component of the joint INCOSE-OMG MBSE Initiative. INCOSE is the International Council on Systems Engineering (INCOSE) and OMG is the Object Management Group (OMG). Formed in 2013, this working group is concerned with advancing the practice, resources, and theory for creating and applying model-based patterns.

More information on this working group may be found on-line [1] and via the other References.

2 In a Nutshell: What Are S*Models? What Is the S*Metamodel? For What Purpose?

- Models are descriptions of Modeled Things, for use and interpretation by Model Interpreters (which may be humans, machines, or both). See Figure 1.

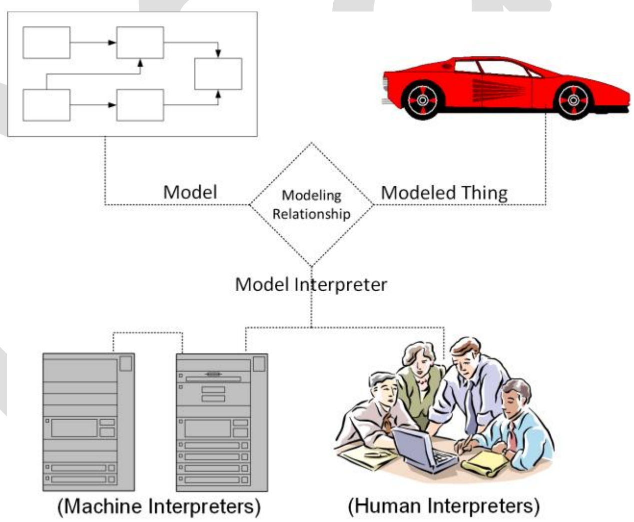


Figure 1: The Three-Way Setting for Models

- Modeled Things are portions of the natural and man-made world, for which Models encode what humans believe is (or may be or could eventually be) true about those real Modeled Things.
- Models describe the solar system, manufactured products, chemical elements and reactions, buildings and other civil structures, behavior of living things, aircraft flight, weather systems, commercial vehicles, defense systems, computer software, and diverse other systems.
- Models come in many forms, including drawings and diagrams, mathematical equations and

tables, executable computer programs or databases, and embodied physical replicas.

- Many models are created (authored) by human modelers, but machine learning algorithms also allow computing machines to create some models from observed data.
- Models provide essential information to decision-makers (humans or automated systems), in technical fields, business and finance, government, defense, and personal life. Accordingly, shared, reliable methods of establishing and communicating confidence (or lack of confidence) in what a model says are very important to model users.
- Models are the language of engineering and science. The discoveries and inventions they encode helped power the abrupt STEM-based acceleration in the human standard of living in only 300 years.
- During that revolution, modelers and model users (particularly scientists, mathematicians, engineers, and technologists) have been learning about the minimum conceptual content that a model must contain, answering the related question, *"What is the smallest effective model of a system, for purposes of science and engineering, over the life cycle of a system?"*
- An S*Model is any model, in any modeling language or in any modeling tooling, whose content can be shown to include that minimal content. (S* is short for "Systematica".)
- The S*Metamodel describes that minimal content, so that modelers and modeling tools can use it to generate or manage effective models, and share a common understanding of what an S*Model tells us—the model's "semantics" (meaning).
- In observed engineering widespread contemporary engineering practices, some models appear to lack some of that minimal content (model is too small for intended use) or contain redundant and conflicting content (model is too big). Both can create serious problems for model users.

3 Observed Phenomena and Physical Science Informed the S* Perspective

We call a model "valid" to the extent that we believe it accurately represents aspects of the Modeled Thing that it is claimed to describe. Two very different kinds of practice have come into play in determining the validity (accuracy of representation) of such models:

a. Agreement with Observation as the Arbiter of Validity: The successful rise of the physical sciences depended on testing proposed models against observations of real systems they claimed to model. Does a model accurately predict an observed phenomenon of the Modeled Thing? That standard enabled multiple parties to "test" models to see if they really described what was claimed, and was a major turning point in the physical sciences only 300 years ago—particularly for falsifying models that were "not right enough" [2,3].

b. Human-Based Authority and Consensus as the Arbiter of Validity: At the start of the scientific revolution, the above approach had to overcome resistance from human-based authority of other types. The Copernican Revolution [4] marked one such turning point. But in more recent times, other practices in making of formal consensus-based standards have likewise turned to human opinion about whether a model is agreeable to a community in authority. For example, that approach has been used with some success in the forming of standard models for information system databases and information exchange. [5].

These two approaches to validation need not necessarily be in conflict, as they may involve a consensus (per (b)) that real experiment and observation (per (a)) is in agreement with a model, or not—as in the case of peer review of scientific experiment-based publications. But they are also not necessarily in agreement. [6] Particularly in cases that have become disconnected from direct experimental observation of phenomena, this consensus-based approach can lead to confidence controversy that is difficult to resolve. [7].

Because of the spectacularly impactful effectiveness of models in the physical sciences, the S*Metamodel perspective on minimum model content is heavily influenced by the history of models in physical sciences of the last three centuries. That history is based upon models of observed phenomena, against which proposed models may be validated. Accordingly, we can ask what observable phenomena taught science about minimal necessary content of models.

Three types of STEM-based observed natural phenomena have informed the S*Metamodel and its use to describe S*Models [8, 9].

3.1 Phenomenon 1: The System Phenomenon

The traditional engineering disciplines have their technical bases and quantitative foundations in the hard sciences' descriptions of phenomena:

Engineering Discipline	Phenomena	Scientific Basis	Representative Scientific Laws
Mechanical Engineering	Mechanical Phenomena	Physics, Mechanics, Mathematics	Newton's Laws
Chemical Engineering	Chemical Phenomena	Chemistry, Mathematics. . . .	Periodic Table
Electrical Engineering	Electromagnetic Phenomena	Electromagnetic Theory	Maxwell's Equations
Civil Engineering	Structural Phenomena	Materials Science, . . .	Hooke's Law, etc.
Semiconductor Eng'g	Semiconductor Phenomena	Solid State Physics, . . .	Quantum Mechanics

$$\nabla \cdot \mathbf{D} = \rho$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$$

$$\frac{N_b}{N_a} = (\frac{g_b}{g_a})(e^{-(E_b - E_a)/kT})$$

$$H(t)|\psi(t)\rangle = i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle$$

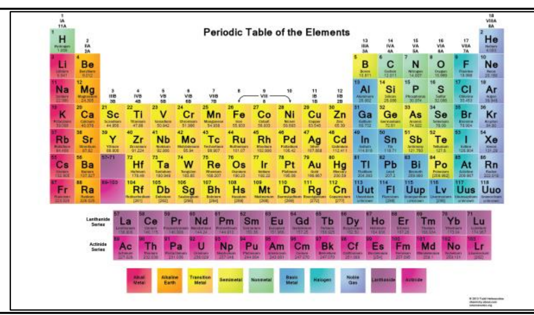


Figure 2: Engineering Disciplines Are Informed by Observable Phenomena of the Sciences

Specialists in individual engineering disciplines (ME, EE, CE, ChE—without them, we would be living as in 1500) sometimes argue that their fields are based on "real physical phenomena", "physical laws based in the hard sciences and first principles math", and the like, while sometimes claiming Systems Engineering lacks the equivalent phenomena-based theoretical foundation.

Instead, Systems Engineering is sometimes viewed as emphasizing process and procedure; use of critical thinking and good writing skills; organizing and accounting for information; integrating the work of the other engineering disciplines and stakeholder needs—but not based on an underlying "hard science" like other engineering disciplines. (However, this Primer argues otherwise.)

Each of the traditional physical sciences is based on a specific physical phenomenon (mechanical, electrical, chemical, etc.) and related mathematical formulation of physical laws and first principles. What is the equivalent "hard science" phenomenon for systems, where is its mathematics, and what are the impacts on future SE practice? Are there also "soft" aspects?

In the perspective described here[1], System means a collection of interacting system components:

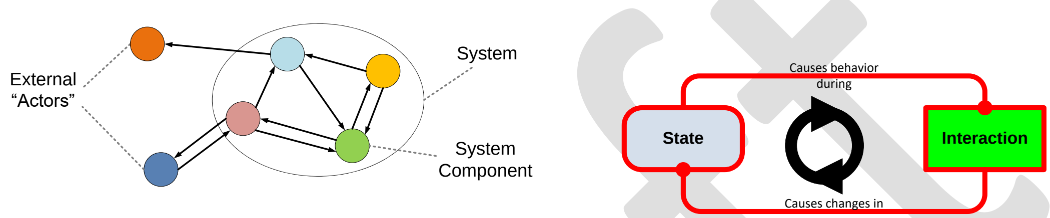


Figure 3: The System Perspective

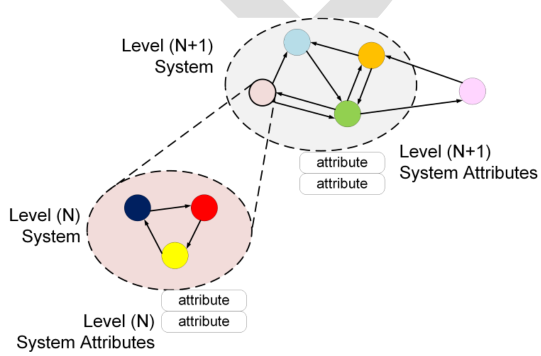
By interaction we mean exchange of energy, force, material, or information (these are input-outputs) by system components, through which one component impacts the state of another.

By state we mean a property of a component that impacts its input-output behavior during interactions. Note the circular cause-effect definition chain here. See Figure 3.

So, a component's "behavior model" describes input-output-state relationships during interaction—*there is no "naked behavior" in the absence of interaction*. Interactions are thus central to S*Models and the S*Metamodel.

The behavior of a system involves emergent states of the system as a whole, exhibited in its behavior during its own external interactions, resulting in observable holistic aspects.

Observable phenomena of the sciences in all instances occur in the presence of special cases of the (generalized) System Phenomenon: System behavior emerges from interaction of behaviors (phenomena themselves) of system components a level of decomposition lower. (Figure 4)



Inductive Ladder N to N+1: System level attributes (state variables, parameters) emerge in the characterization of system's behavior in its external interactions within the next level system. Each such emergent behavior is a case of the System Phenomenon, governed by Hamilton's Principle. Examples: Conductivity of elemental atoms; position and size of Solar System center of mass; diet of organism.

Figure 4: The System Phenomenon

[1] This definition of System is inspired by the success of 300 years of STEM using the mathematical foundations of Newton, Lagrange, Euler, Hamilton, and those who followed. There are other definitions of "system". For example [10, 11].

The resulting patterns of recurring larger-scale behavior become the basis for recognition, mathematical laws of motion or other hard science, heuristics, rules of thumb, intuition, prediction, or other exploitation of those regularities. Phenomena in the "softer" domains in all instances likewise occur in the presence of cases of the above System Phenomenon, even though the domain-specific phenomena, input-outputs, states, and behaviors are different.

All these patterns are recurrences, having both fixed and variable (configurable) aspects. The heart of physical science's life-changing 300-year success in prediction and explanation lies in recognition, representation, exploitation of recurring patterns. They are also at the heart of deep human intuition, expertise, and heuristics.

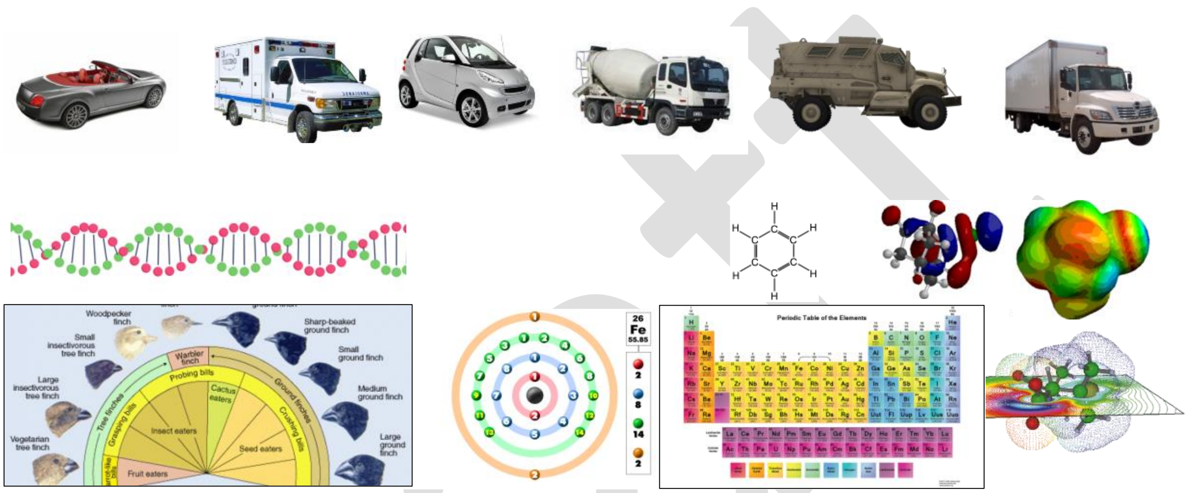


Figure 5: Recurrent Patterns: The Leverage of Scientific Laws, Rules of Thumb, Intuition

For each such emergent phenomenon pattern studied across the physical sciences, the emergent interaction-based behavior of the larger parent system was discovered by Lagrange, Euler, and Hamilton to be a stationary state space trajectory of the action integral—what came to be called Hamilton's Principle, expressed as equation. Extensions and alternatives to this formulation were developed by those who followed, for discrete systems, non-deterministic systems, and other cases. Variational and eventually Hamilton's generalizations became the theoretical foundations of each of the specialized phenomena of the various physical sciences (mechanics, electrical science, chemistry, quantum mechanics, etc.). On this common mathematical foundation across all of the sciences, Max Planck remarked that:

> *"It [science] has as its highest principle and most coveted aim the solution of the problem to condense all natural phenomena which have been observed and are still to be observed into one simple principle, that allows the computation of past and more especially of future processes from present ones. ...Amid the more or less general laws which mark the achievements of physical science during the course of the last centuries, the principle of least action is perhaps that which, as regards form and content, may claim to come nearest to that ideal final aim of theoretical research."* [12]

Each discipline's "fundamental" phenomena-based laws' mathematical expression (Newton, Maxwell, Schrodinger, et al) is derivable from the above formulation—as shown in many discipline-specific textbooks. So, instead of Systems Engineering lacking the kind of theoretical foundation the "hard sciences" bring to other engineering disciplines, it turns out that all those other engineering disciplines' foundations are themselves dependent upon the System Phenomenon and Hamilton's Principle mathematical expression of the inductive pattern from Level N to Level N+1. Many others followed with generalizations and extensions to other cases, including discrete and non-deterministic.

So, the underlying math and science of systems provides the theoretical foundation already used by all the hard sciences and their respective engineering disciplines. It is not Systems Engineering that lacks a foundation—instead, it has been providing the foundations claimed by each of the other disciplines! This also opens new perspective on how Systems Engineering relates to emerging future disciplines:

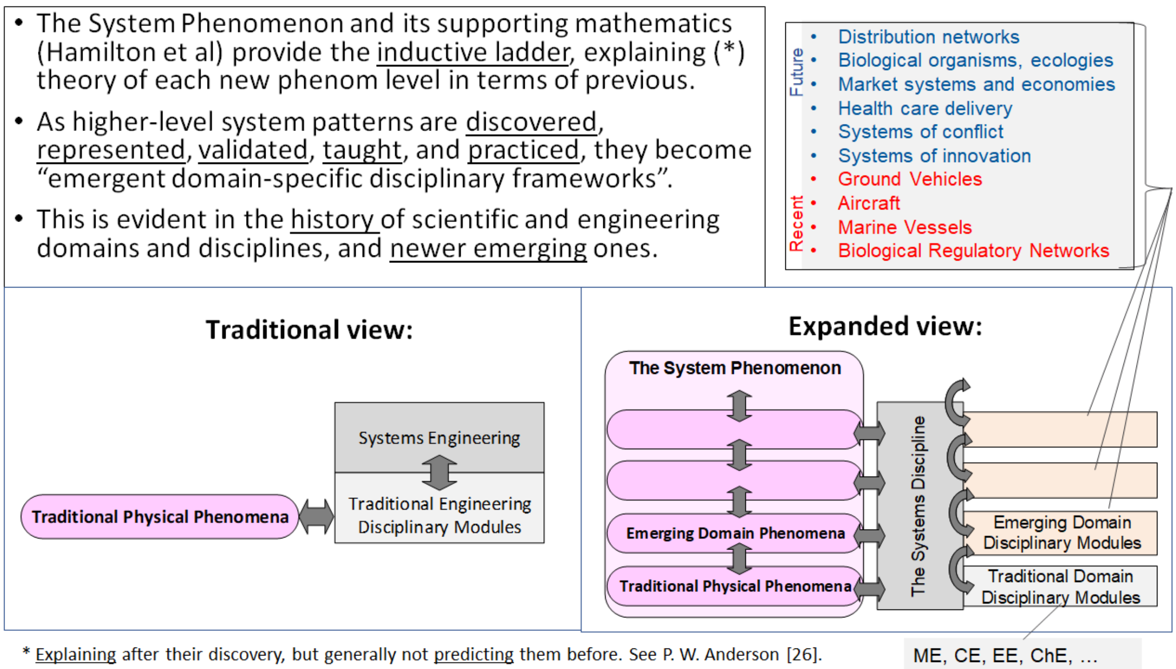


Figure 6: SE Foundations Support the Foundations of All the Engineering Disciplines

3.2 Phenomenon 2: The Value Selection Phenomenon

Engineers know that value is essential to their practice, but its "soft" or subjective nature seems challenging to connect to hard science and engineering phenomena. What is the bridge effectively connecting these, where is the related mathematics, and what are the impacts on future SE practice?

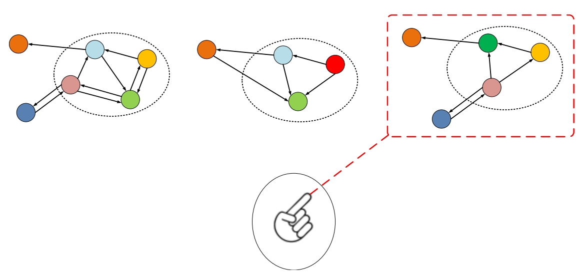


Figure 7: The Value Selection Phenomenon

System engineers currently learn to seek out and represent (may model in detail) stakeholder needs, measures of effectiveness, objective functions connected to derived requirements and technical performance, etc.--what value does your system contribute?

This nearly always includes "conflicting" dimensions of value, when "trade space" value dimensions appear to trade against each other—as in performance vs. cost. The resulting balancing act led to notions of Pareto Frontiers and other multi-variate forms, Arrow's Impossibility Theorem, and other formulations and insights.

For many systems, lack of good knowledge (by even the customer) about value has changed engineering into a discovery project, as in Agile Methods, Minimum Viable Products, Pivoting, Hypothesis Experiments, and similar approaches. This is also related to the Model Trust by Groups Phenomenon.

Meanwhile, what are the phenomena associated with value, what is the bridge between subjective value and objective science, where are the related mathematics and recurring patterns, and what are the impacts on future SE practice?

What follows is not the same as simply "modeling idealized value", which might seem natural but which has some shortcomings. What distinction are we making here? This is where the "objective science" comes in! We are interested in models that can be tested in experiments with real selection agents.

Systems engineering can catch up with what business has discovered and put into practice in recent years—driving discovery with real experiments that test the validity of hypothesized value, in a dynamic, pivoting enterprise. [13]

We are interested in what actual selection behavior tells us about value—not just what isolated offerings of opinion about value or statements of preference. What really gets observably selected? That is the distinction of the Value Selection Phenomenon. It is a real phenomenon that always occurs and can be observed. It also can be influenced by advertising, culture, context, and bias. It can also help us engage the "multi-variate value" challenge.

Settings	Types of Selection	Selection Agents	
Consumer Market	Retail purchase selection	Individual Consumer; Overall Market	
Operational Use	Decision to use product A or use product B	User	
Military Conflict	Direct conflict outcome; threat assessment	Military Engagement	X
Product design	Design trades	Designer	
Commercial Market	Performance, cost, support	Buyer	
Biological Evolution	Natural selection	Environmental Competition	X
Product Planning	Opportunity selection	Product Manager	
Market Launch	Optimize choice across alternatives	Review Board	
Securities Investing	What to buy, what to sell, acceptable price	Individual Investor; Overall Market	
College-Student "Matching Market"	Selection of individuals, selection of class profile, selection of school	Admissions Committee; Student & Family	
Life choices	Ethical, moral, religious, curiosities, interests	Individual	
Democratic election	Voting	Voters; Voting Blocks	
Business	Risk Management, Decision Theory	Risk Manager, Decision Maker	

Not all selection is by human agents

Figure 8: Different Types of Selection--Not Always by Humans

Even if value (human-based or not) seems subjective, expression of value in the real world is always via selection, which is an objectively measurable interaction-based instance of the System Phenomenon:

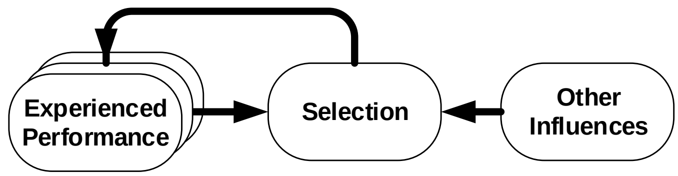


Figure 9: Selection versus Performance--Not the Same

Value refers to Interactions of two very different types:

1. Performance Interactions (real or planned, present, past, future) embody and deliver Value from Performers (this is currently more familiar to systems engineers):
   - Example: The "ride" a passenger experiences, over a bumpy road in a vehicle. See Figure *10*.
   - An actually experienced, simulated, imagined, or promised performance interaction.
   - This might seem like what we'd want to model (and we should), but there is more than this.
2. Selection Interactions (human or otherwise) express the comparative Values of a Selection Agent, human or otherwise (familiar to consumer marketers, behavioral economics specialists, web-based experimentalists, big data specialists):
   - Example: The selection of a vehicle to buy, from among competing alternatives. See Figure *10*.
   - This is what we advocate also be modeled. It might seem it ought to produce the same result, but there is more to it. For example, what is the effect of advertising, or reference networks?

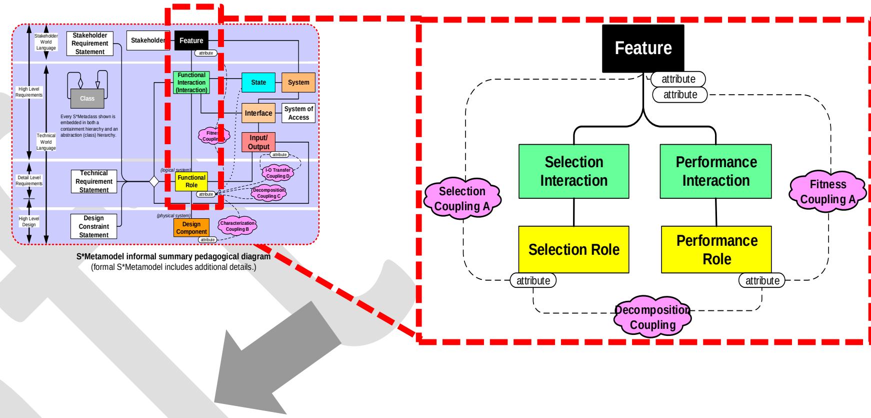
Here we are emphasizing selection outcome as the ultimate expression of value. Performance Interactions remain essential to representing the possible choices. Selection Interactions typically choose from across multiple dimensions all at once, in the real world.

Value is not solely inherent to a subject system's performance. A performing system, moved from one country-culture-application-market segment to another, with no technical changes, could offer the very same technical performance (assuming the application/operating environment remained the same otherwise), but be valued differently by the new and different stakeholders in that different culture—as their Selection behavior will ultimately express.

The Selection Phenomenon is what we want to understand to quantify relative value, always expressed as selection--influenced in part by the Performance Interaction, but also by the nature and behavior of the Selection Agent/agency and environment, which may be impacted by past experience, learning and habituation, advertising and promotion, trends and fashion, peer groups, etc. Much innovation has been occurring in those other spaces—such as choice and distribution through on-line and other non-traditional systems.

The Features portion of the S*Metamodel expresses selectable aspects of a system's performance, which are the basis of the selection interactions. These Features are only partly a function of the performance of the system of interest—they are also a description of the selection agency, whether it is based in human judgment or survival in combat and the environment.

See also Section 4:



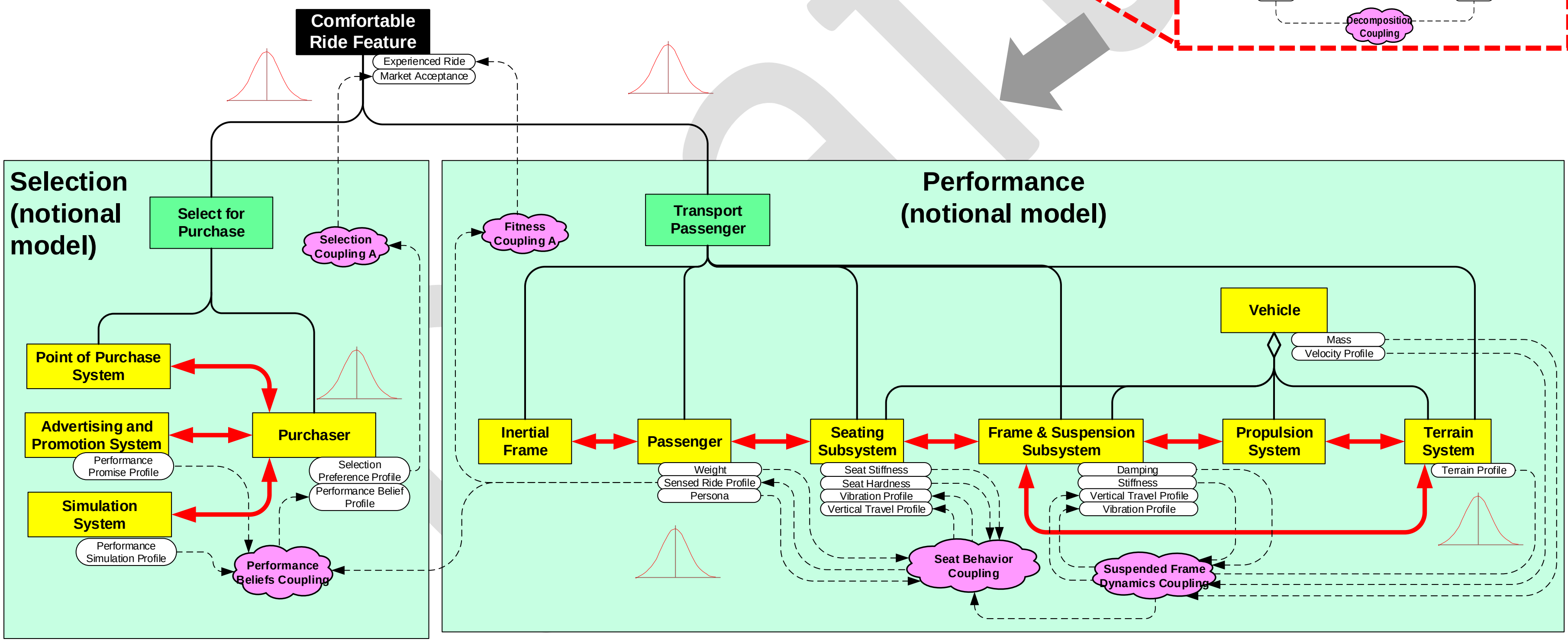


Figure 10: Example: Selecting Vehicle "Ride"

We note that <u>human subjectivity</u> appears in two separate ways within the framework of Figure 10:

1. A human may be a part of the Performance Interaction, and form sensory and mental perceptions about what performance is occurring—not its value. (e.g., Passenger in Figure *10* example.)
2. A human may be the Selection Agent in the Selection Interaction, acting on acquired beliefs about relative value. (e.g., Purchaser in Figure *10* example.)

A further insight: Note that neither of these two parties is the <u>Modeler</u>: The role of the Modeler is to discover, express, and validate models of both the Performance and Selection aspects of the systems at hand--whether those humans are flying aircraft or choosing products.

This clearly involves modeling of human behaviors. That should hardly be a surprise, after decades of impactful modeling, Nobel prize recognition, and now on-line machine learning and millions of confirming experiments, about the value-based behavior of <u>humans making choices</u> [13, 14, 15, 16]:

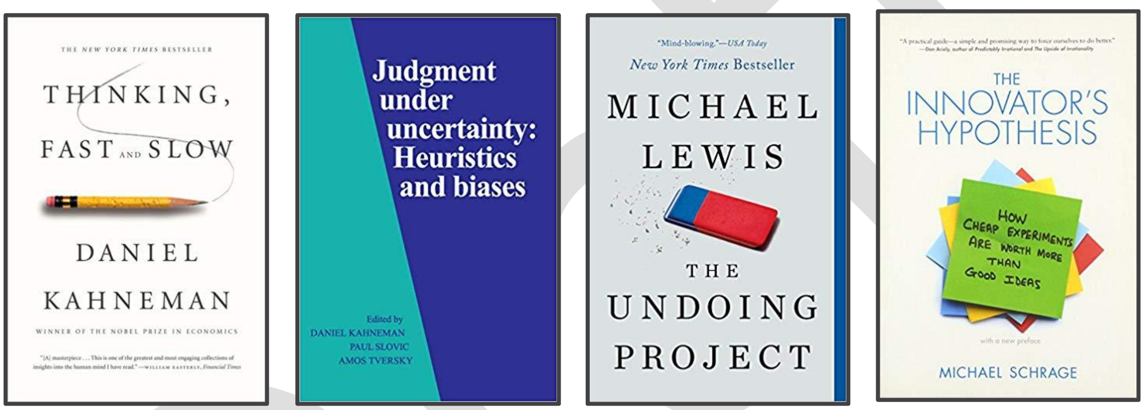


Figure 11: There Has Been Major Research and Practice Advance Concerning Human Choice

3.3 Phenomenon 3: The Model Trust by Groups Phenomenon

The physical sciences accelerated progress in the last three centuries, as they demonstrated means for not just the discovery and representation of Nature's patterns, but also the managed awarding of graduated shared trust in them by their users (Figure 1). What is the scientific basis of such group learning, how is it related to machine learning, and how does it impact future practices?

Phenomena 1 and 2 above are about motivation for the content of S*Models and the S*Metamodel, both discussed in the next section. Phenomenon 3 is about motivation for the content of learned S*Patterns and the Innovation Ecosystem Pattern--including how confidence in models is managed and represented. Those subjects are described in [27].

4 What Is an S*Model? What Is the S*Metamodel?

What is the smallest model of a system, having sufficient content for the purposes of engineering or science, over the life cycle of the system? This question has both practical and theoretical significance [17]:

- Practically, we don't want a model to be missing information that will be needed. But we also don't want a model that is too big--cumbersome to understand and maintain. An internally redundant model can be self-contradictory and harder to maintain.
- Theoretically, scientists have sought the smallest model sufficient to describe the behavior of a system. That is, they have favored the simpler of two explanations of a system—a criterion called "Occam's Razor" [18]. Further theoretical significance is that the size of a system's smallest model is one of the measures of the complexity of that system. [19]

An <u>S*Model</u> is any model, in any modeling language or in any modeling tooling, whose content is made up of that minimal content, consisting of a targeted set of formal model concepts that were selected from the successful history of the system models of STEM. "S*" is short for "Systematica". That minimal content framework is formally described by the S*Metamodel. [20]

A <u>metamodel</u> is a model describing other models, so that those other models are similar enough to each other that they may be described by a single such metamodel. A metamodel provides the "rules of the game" for creating or interpreting the meaning of the models it describes. It describes the language or semantics of models conforming to it. Metamodels are also used to formalize automated tool-based languages. Examples of metamodels include the metamodels for the software modeling language UML or the systems modeling language SysML. [21].

The formal definition of the S*Metamodel is provided by [20], which is a detailed description of over 100 pages. Most of the time, an intuitive, less complete reference is useful, such as provided by the diagram of **Error! Reference source not found.**. That "intuitive pedagogical" diagram depicts a subset of some of the prominent classes of information entities ("metaclasses") and approximately how they are related to each other ("metarelationships").

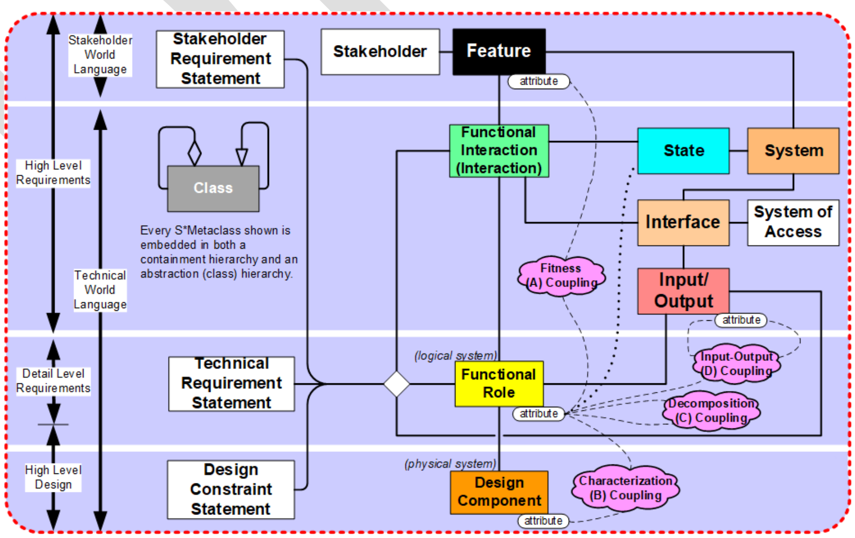


Figure 12: Informal Representation of Core of S*Metamodel

(Note: This is an informal "pedagogical" summary. For the formal S*Metamodel, refer to [20].)

The S*Metamodel is not a replacement for commercial modeling tools or modeling languages. Instead, it has been “mapped” into such languages and tools, to establish a minimal consistent and portable core. S*Models are therefore commonly constructed and used within various commercial off the shelf (COTS) modeling toolsets and modeling languages, such as SysML tools and language. (Refer to Section 5.)

4.1 Sample of Selected S*Model Classes (Metaclasses)

The following summarizes some of the S*Metamodel classes (metaclasses) that are components of S*Models.

4.1.1 System

A System is a collection of interacting components. By “interact” we mean the components exchange input-outputs (typically energy, force, material, or information) that change the state of the components. The components transform inputs into outputs, depending upon the state of the components. A component can itself be a System, called a sub-system.

4.1.2 Interaction (Functional Interaction)

This section is motivated in part by Phenomenon 1 of Section 3.1.

A Functional Interaction is an interaction of two or more System Components, whose behavior is described by their Functional Roles. Interaction means the exchange of Input-Outputs (typically force, energy, material, or information) whereby one System Component affects the State (see State) of another System Component. Interactions are the observable phenomena-grounded basis of the theoretical foundations of the physical sciences and engineering disciplines. All behavior occurs in the context of interactions. The behavior of each interacting component is determined by its state, and that state can in turn be changed by the interactions.

Model authors are interested in discovering and including all the interactions relevant to their models. The S*Metamodel makes this completeness goal less difficult by providing three independent conceptual pathways to check, for what should be the same set of interactions (see Figure 12):

- Value: The Feature-to-Interaction relationship trace helps to verify that all the stakeholder-valued Features have representation in Interaction space.
- External Actors: The Interaction-to-Interface relationship trace helps to verify that all the external actor interfaces have representation of the Interactions associated with them.
- Time/Mode/Situation: The Interaction-to-State relationship trace helps to helps to verify that all the “situations”, modes, use cases, or periods of time that the modeled system will encounter have representation of all the Interactions associated with them.

This discovery power is magnified by realizing that every Interaction should appear in all three of the above sets. This completeness of Interactions has dramatic consequence in Section 4.1.6.

4.1.3 Feature (Stakeholder Feature)

This section is motivated in part by Phenomenon 2 of Section 3.2.

A Stakeholder Feature is a collection of Functional Interactions having stakeholder value implications. Features are used to summarize product functionality in terms of value, service, or capability recognized by customers or other stakeholders. Economics, quality, performance, risk, or other measures of effectiveness are often associated with Features. The total Feature set of a system of interest establishes the “trade space” in which various issues are traded off against or compared to each other (a form of selection itself), as to the relative stakeholder appeal, score, or likelihood of subsequent selection by other agencies. In addition to the value-laden concepts, the same Features also represent risk—all risk is risk to Features (see Feature Impact). For system families, product line engineering (PLE), and configurable platforms or S*Patterns, Features are the primary point at which stakeholder configuration choices are expressed, thereafter driving all other points of variation within a S*Model of such a system pattern. This is directly related to the formal standard on Feature-Based Product Line Engineering (PLE) [28].

4.1.4 Functional Role (Role)

A Functional Role is the behavior displayed by one of the interacting entities during a Functional Interaction. (A component "plays a role in the interaction" according to the Functional Role(s) allocated to it.) Because it is entirely described as behavior, a Functional Role is a Logical System. A Functional Role may eventually be allocated to a Design Component to perform that behavior, but the Functional Role is viewed as meaningful whether or not so allocated.

The behavior demonstrated by a Functional Role is its input-to-output transformation behavior, modulated by its attribute values. Its behavior also includes changes to its (state variable) attribute values.

4.1.5 Design Component

A Design Component is a System defined based upon its identity or composition (but not its behavior). Emphasizing identity, Design Components are sometimes given proper names, such as names of commercial products, materials, chemical elements or compounds, part numbers, corporate systems, people, organizations, buildings, etc. Design Components fulfill the Functional Roles (Logical Systems) allocated to them by the allocation of Roles to a Design Component.

4.1.6 Requirement Statement

A Requirement Statement is a description of behavior, in prose, mathematical, or other form, relating a System's Inputs, Outputs, and Attributes, against which an engineered System may be verified. More generally, a Requirement Statement may be viewed as a non-linear transfer function [22], describing the relationship of Outputs to Inputs, parameterized by Attribute values.

Requirement Statements participate in three-way relationships with Interactions and Roles. By this we mean that the Requirement Statement describes the Functional Role's behavior during the Interaction. Requirement Statements are seen as allocated to Functional Roles (whose behavior they describe), which are in turn allocated to Design Components.

Model authors are very interested to discover and include in their models all the relevant Requirements Statements that are relevant. The S*Metamodel makes this completeness goal less difficult. That is because of recognizing that finding all the Requirements Statements is directly related to finding all the Interactions. Section 4.1.2 summarized how intermediate completeness goal is accomplished.

4.1.7 Attribute

A modeled Attribute is a modeled property or characteristic of any of the metaclasses, which can take on different attribute values to further describe (parameterize) the various instances of those classes and how they vary.

An attribute may belong to (describe) any metaclass, including another Attribute. Particularly prominent cases include:

- Feature Attributes: Attributes of Features parameterize, quantify, or modulate stakeholder-perceived value (positive or negative), financial performance, risk, capability, configuration options, or other variable aspects of Features as seen by stakeholders. Such attributes may have continuous, discrete, quantitative or otherwise enumerated values. Feature Attributes include what system engineers call “Measures of Effectiveness” (MOEs).
- Role Attributes: Attributes of Roles quantify or modulate the technical descriptions of behavior provided by Roles. They include dynamical state variables, relatively more fixed configuration parameters (e.g., capacities), behavior profiles, quantitative numerical or qualitative kinds of values.
- Design Component Attributes: Attributes of Design Components describe identity. They include part or assembly numbers, names or identifiers,
- Input-Output Attributes: Attributes of Input-Outputs parameterize Input-Outputs, typically describing IO quantity, strength, or other characteristics of an Input-Output.

4.1.8 Attribute Coupling

An Attribute Coupling is a relationship between two or more Attributes that defines or constrains the value relationship between the Attributes. An Attribute Coupling is sometimes called a Parametric Coupling. Prominent cases of Attribute Couplings are shown in Figure 13:

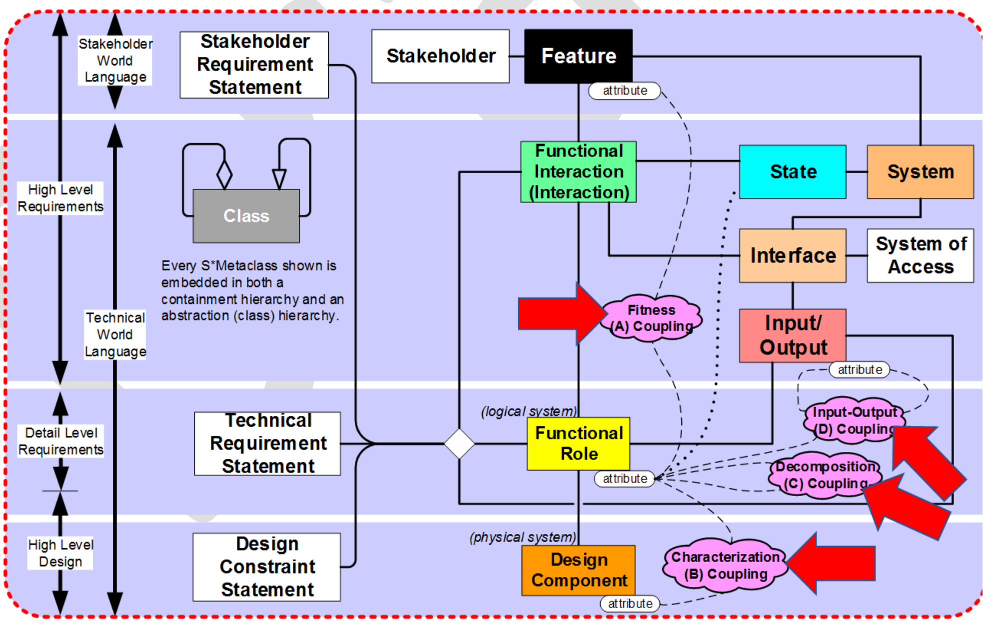


Figure 13: Prominent Types of Attribute Couplings

- Fitness Couplings: These describe the quantitative or other relationships between Feature Attributes and Role Attributes. These characterize perceived value or capability variables as a function of technical behavior variables.
- Decomposition Couplings: These describe the quantitative or other relationships between Roles Attributes at one level of decomposition and Role Attributes at another level of

decomposition. They describe how emergent higher-level system attributes are dependent upon lower-level subsystem or component attributes.

- Characterization Couplings: These describe how Role Attributes (which describe behavior) vary as a function of Design Component Attributes (which describe identity of Design Components. In effect, Characterization Couplings are equivalent to "commercial data sheets" or catalogs, showing curves, tables, or formulae for performance across different Design Component allocations.
- IO Couplings: These describe how Attributes of an Input-Output are related to Attributes of other Input-Outputs.

4.1.9 State

A State is the value of a state variable describing some changing or changeable condition, characteristic, or parameter of a system. Some state variables can take on a continuum of values, and others are constrained to a finite list of possible values. In the latter case, a finite state model enumerates those States. In the finite state case, each State persists for a period of time. In all cases, the state of a System helps determine future behavior in which Functional Interactions are to be performed, entered, and exited based upon events. The finite States of an environmental System of a subject system are use cases for the subject system. During a use case, the subject system is required or expected to perform certain functions, interacting with the environmental system.

4.1.10 Input-Output (IO)

An Input/Output is that which is exchanged between interacting systems. Particular Input-Outputs of interest are forces, energy, material, or information.

4.1.11 Interface

An Interface is an association of a System (which owns, provides, displays, or exposes the Interface), one or more Input/Outputs (which flow through the Interface), one or more Functional Interactions (which describe behavior at the Interface), and a System of Access (SOA), which is the medium enabling or mediating the interaction between systems or transporting their exchanged Input-Outputs.

4.2 Additional Parts of the S*Metamodel

There are other types of S*Metaclasses, as well as S*Metarelationships, not shown in the informal diagram of **Error! Reference source not found.**, that are parts of the S*Metamodel. The more complete a nd more formal reference is [20]. Among these are failure and risk analysis classes, meta-relationships between the meta-classes, and S*Pattern configuration rules.

4.3 A Simple Example S*Model in OMG SysML

Sample views from a simple S*Model in SysML form are shown in Figure 14.

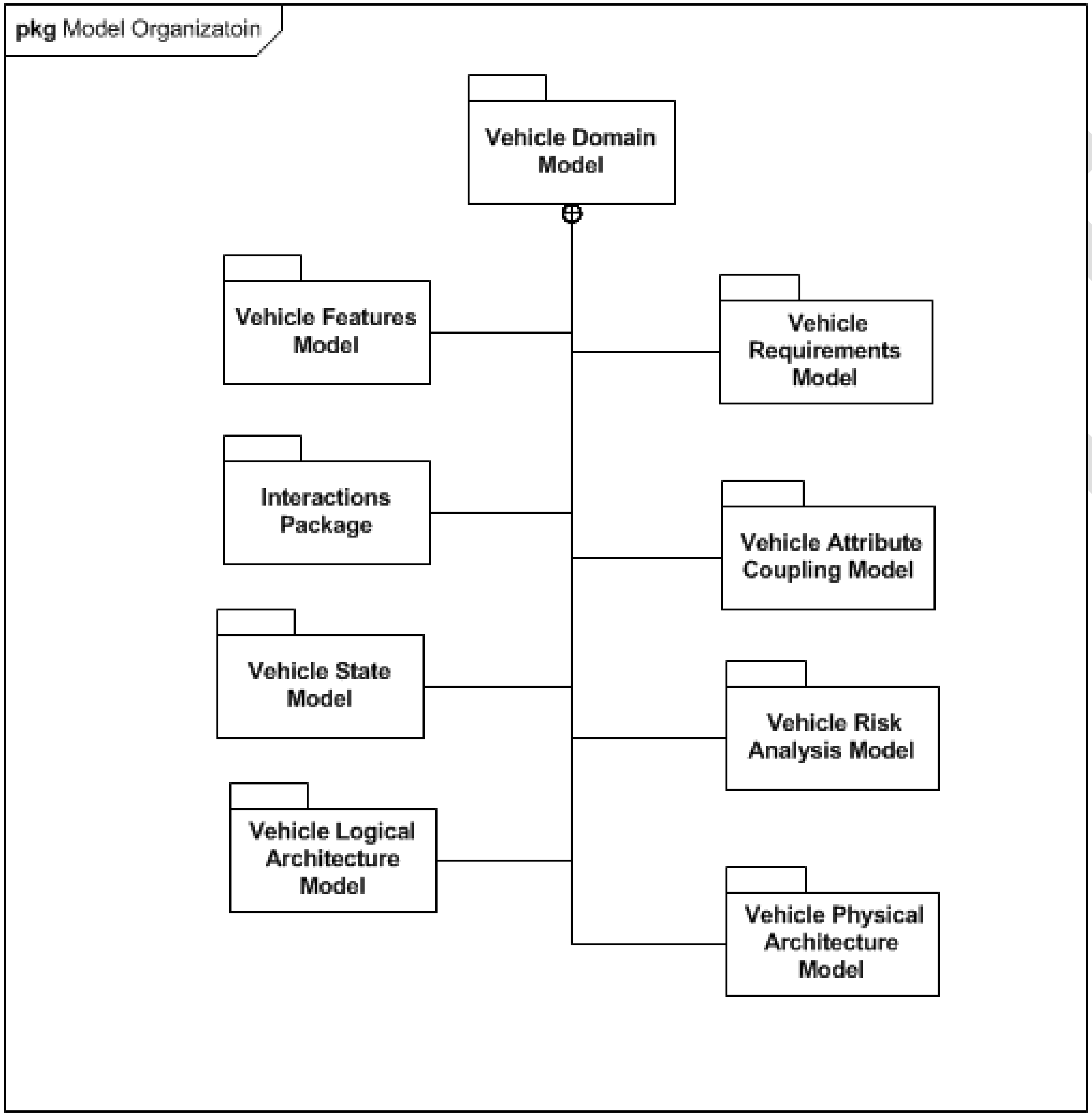
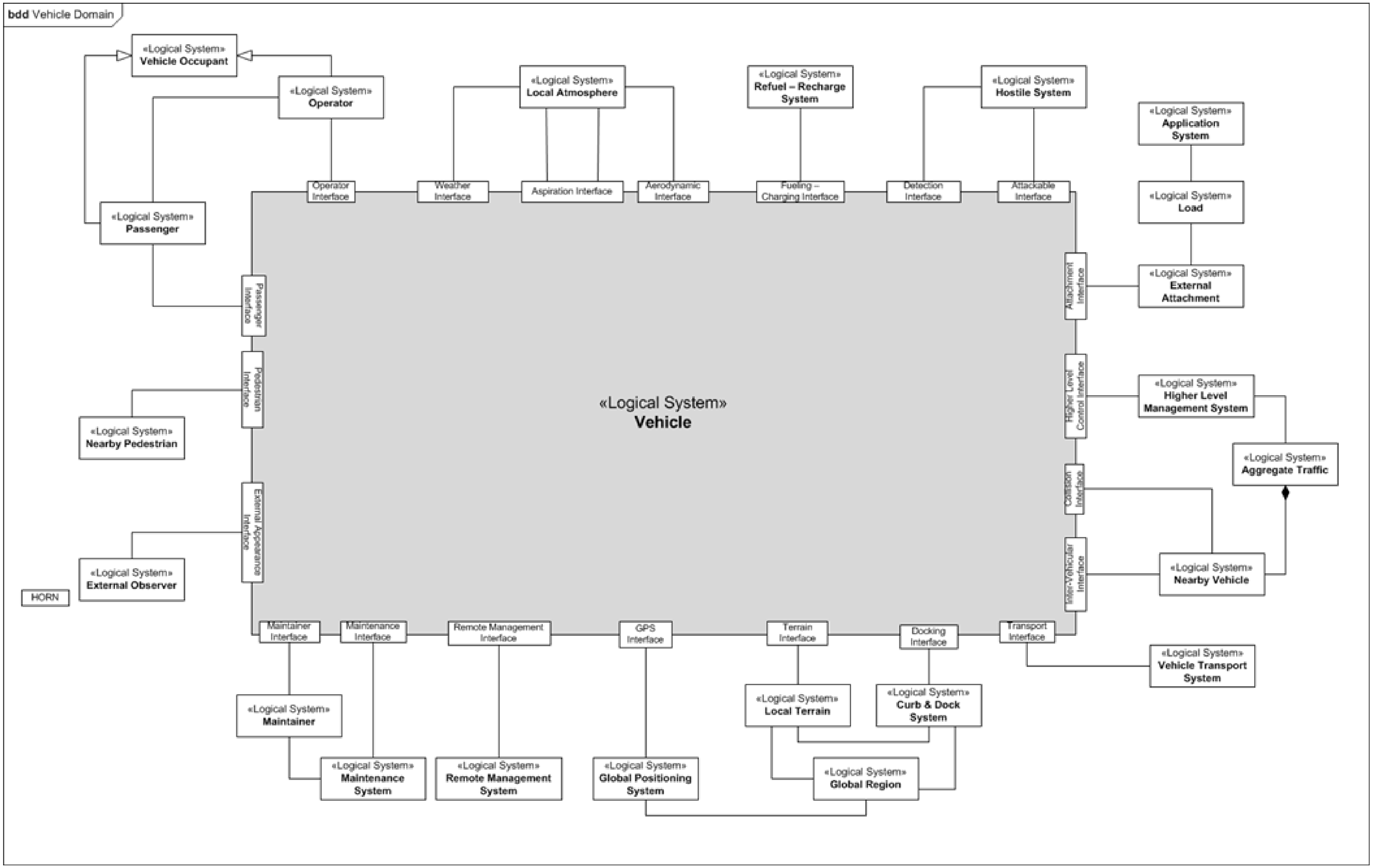
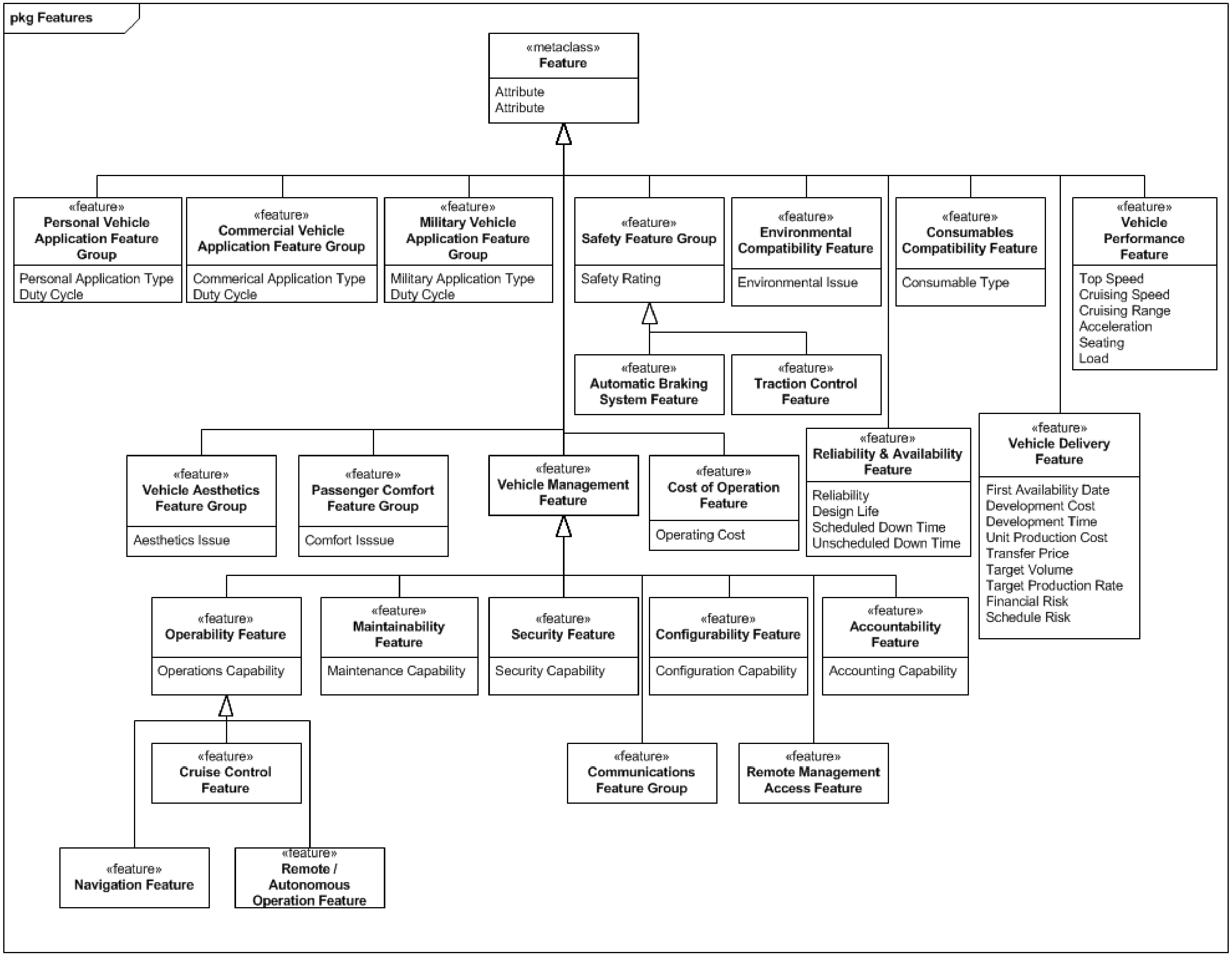


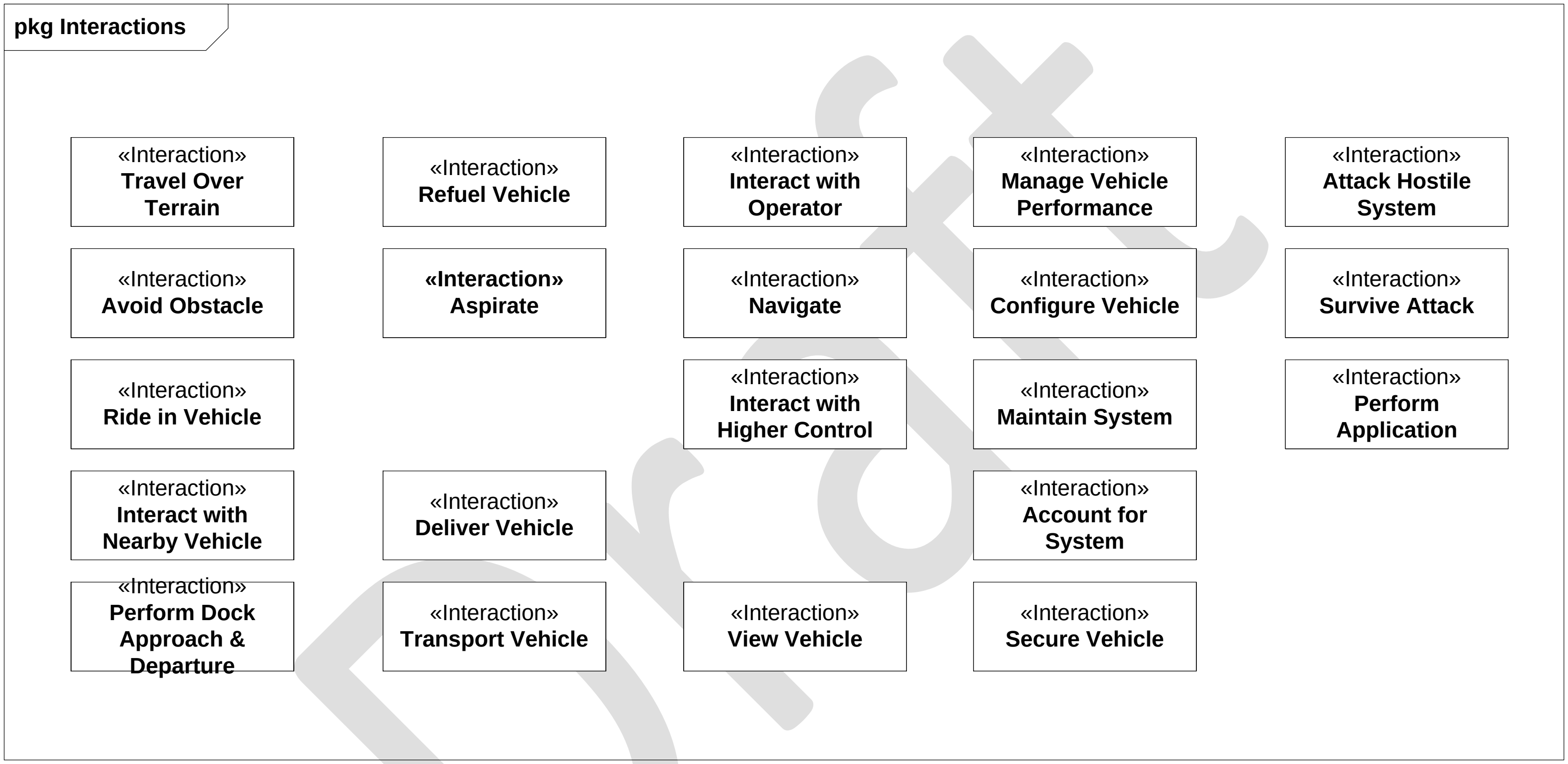
Figure 14: Views from a Sample S*Model in OMG SysML

From [23].

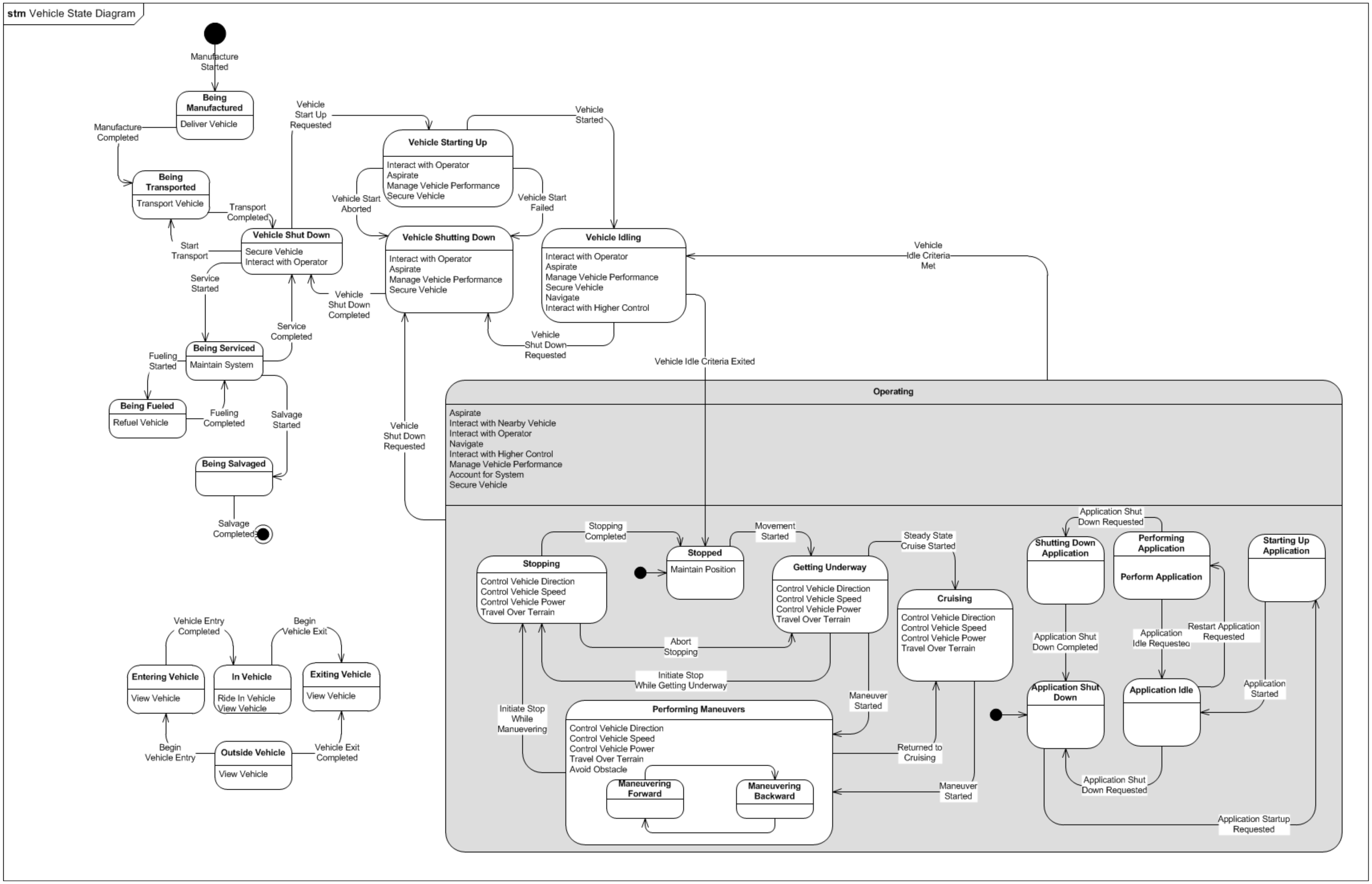

bdd Vehicle Domain
«Logical System» Vehicle Occupant
«Logical System» Operator
«Logical System» Passenger
«Logical System» Local Atmosphere
«Logical System» Refuel – Recharge System
«Logical System» Hostile System
«Logical System» Application System
«Logical System» Load
«Logical System» External Attachment
«Logical System» Higher Level Management System
«Logical System» Aggregate Traffic
«Logical System» Nearby Vehicle
«Logical System» Vehicle Transport System
«Logical System» Nearby Pedestrian
«Logical System» External Observer
HORN
«Logical System» Vehicle
Operator Interface
Weather Interface
Aspiration Interface
Aerodynamic Interface
Fueling – Charging Interface
Detection Interface
Attackable Interface
Passenger Interface
Pedestrian Interface
External Appearance Interface
Attachment Interface
Higher Level Control Interface
Collision Interface
Inter-Vehicular Interface
Maintainer Interface
Maintenance Interface
Remote Management Interface
GPS Interface
Terrain Interface
Docking Interface
Transport Interface
«Logical System» Maintainer
«Logical System» Maintenance System
«Logical System» Remote Management System
«Logical System» Global Positioning System
«Logical System» Local Terrain
«Logical System» Curb & Dock System
«Logical System» Global Region

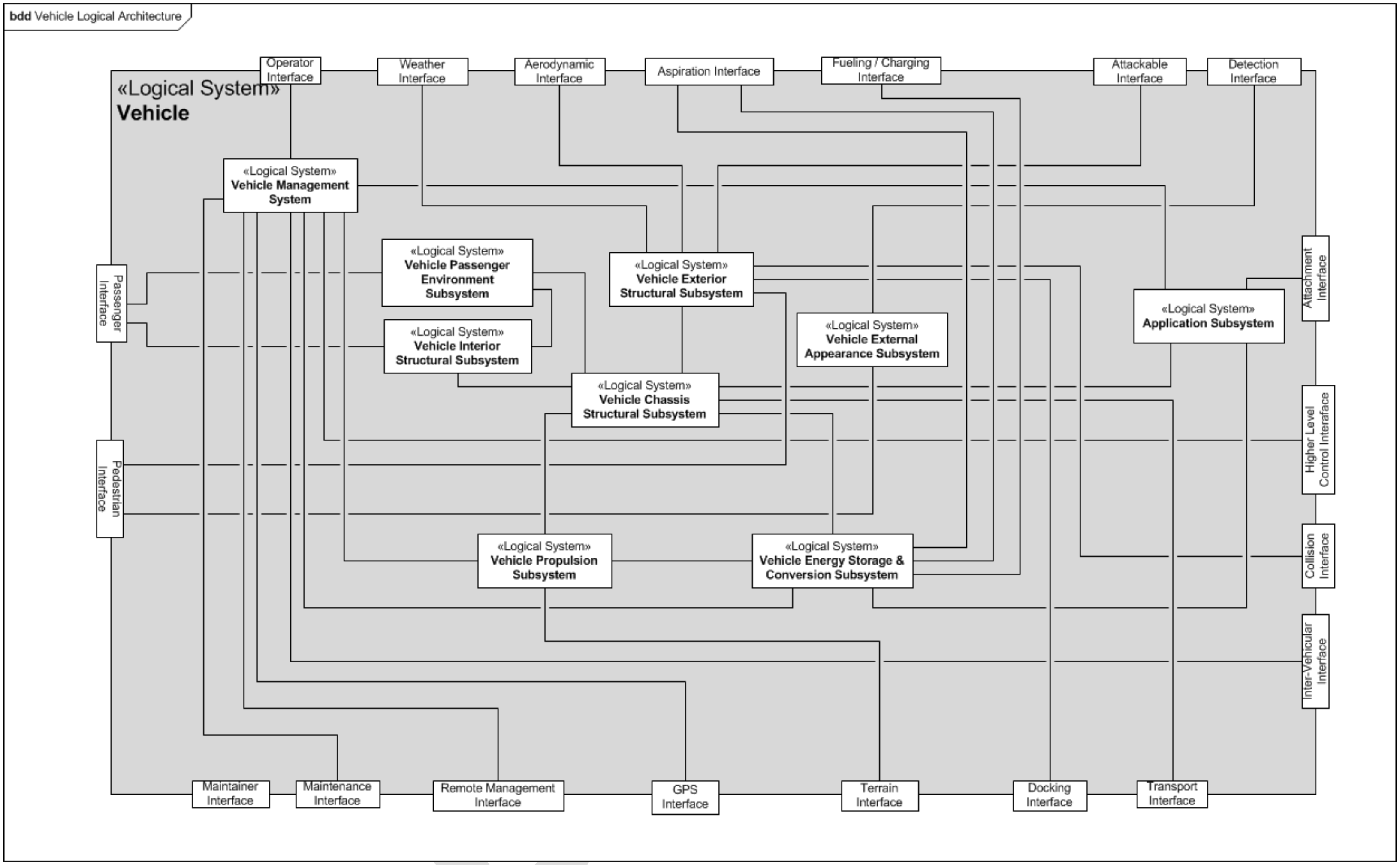


pkg Features
«metaclass» Feature
Attribute
Attribute
«feature» Personal Vehicle Application Feature Group
Personal Application Type
Duty Cycle
«feature» Commercial Vehicle Application Feature Group
Commerical Application Type
Duty Cycle
«feature» Military Vehicle Application Feature Group
Military Application Type
Duty Cycle
«feature» Safety Feature Group
Safety Rating
«feature» Environmental Compatibility Feature
Environmental Issue
«feature» Consumables Compatibility Feature
Consumable Type
«feature» Vehicle Performance Feature
Top Speed
Cruising Speed
Cruising Range
Acceleration
Seating
Load
«feature» Automatic Braking System Feature
«feature» Traction Control Feature
«feature» Reliability & Availability Feature
Reliability
Design Life
Scheduled Down Time
Unscheduled Down Time
«feature» Vehicle Delivery Feature
First Availability Date
Development Cost
Development Time
Unit Production Cost
Transfer Price
Target Volume
Target Production Rate
Financial Risk
Schedule Risk
«feature» Vehicle Aesthetics Feature Group
Aesthetics Issue
«feature» Passenger Comfort Feature Group
Comfort Isssue
«feature» Vehicle Management Feature
«feature» Cost of Operation Feature
Operating Cost
«feature» Operability Feature
Operations Capability
«feature» Maintainability Feature
Maintenance Capability
«feature» Security Feature
Security Capability
«feature» Configurability Feature
Configuration Capability
«feature» Accountability Feature
Accounting Capability
«feature» Cruise Control Feature
«feature» Communications Feature Group
«feature» Remote Management Access Feature
«feature» Navigation Feature
«feature» Remote / Autonomous Operation Feature

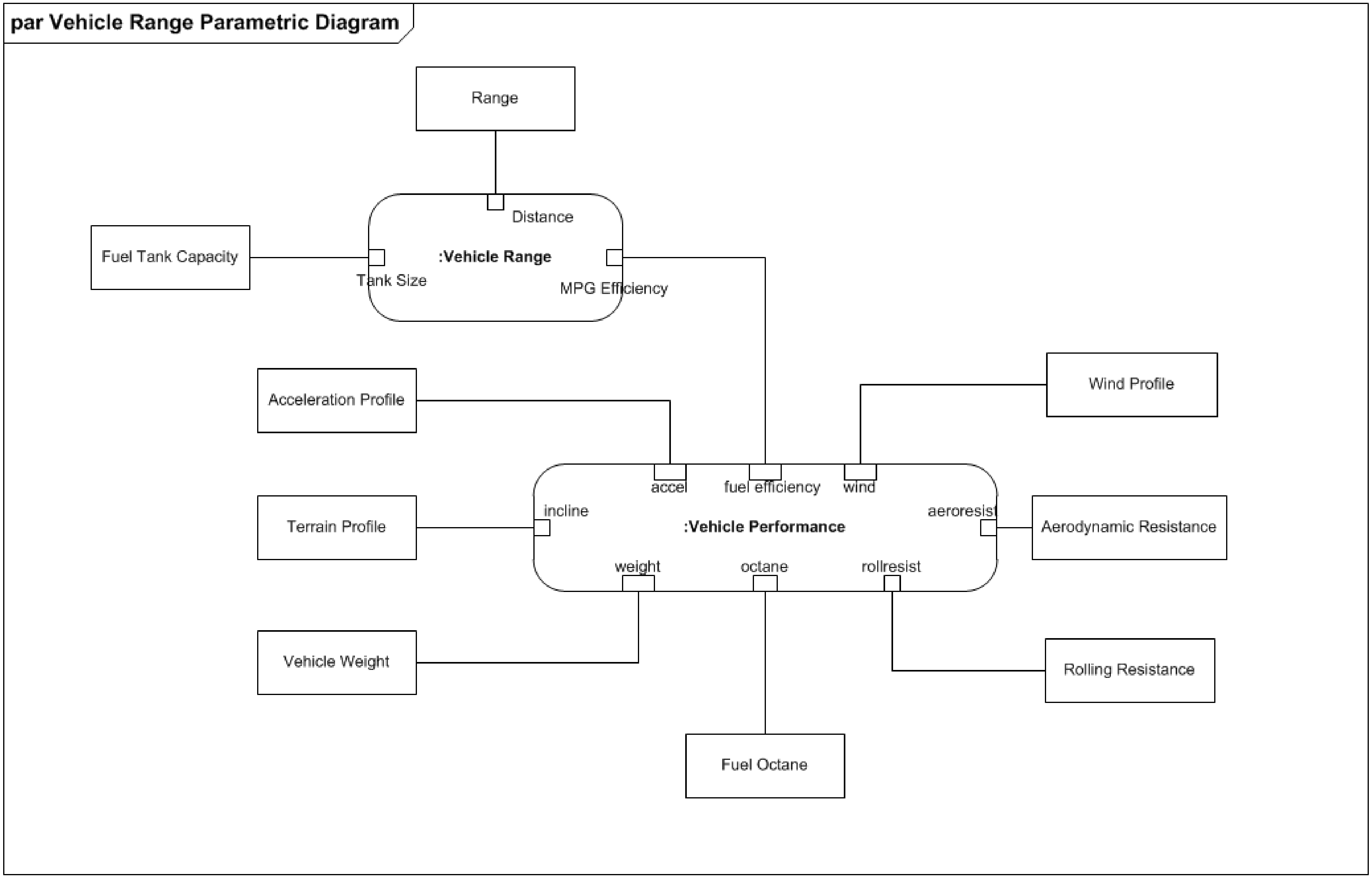

pkg Interactions
«Interaction» Travel Over Terrain
«Interaction» Refuel Vehicle
«Interaction» Interact with Operator
«Interaction» Manage Vehicle Performance
«Interaction» Attack Hostile System
«Interaction» Avoid Obstacle
«Interaction» Aspirate
«Interaction» Navigate
«Interaction» Configure Vehicle
«Interaction» Survive Attack
«Interaction» Ride in Vehicle
«Interaction» Interact with Higher Control
«Interaction» Maintain System
«Interaction» Perform Application
«Interaction» Interact with Nearby Vehicle
«Interaction» Deliver Vehicle
«Interaction» Account for System
«Interaction» Perform Dock Approach & Departure
«Interaction» Transport Vehicle
«Interaction» View Vehicle
«Interaction» Secure Vehicle

1			Actors																					
2	Interaction Name	Interaction Definition	Vehicle	Operator	Passenger	Vehicle Occupant	Nearby Pedestrian	External Observer	Maintainer	Maintenance System	Local Atmosphere	Refuel System	Hostile System	External Attachment	Load	Application System	Higher Level Management	Nearby Vehicle	Vehicle Transport	Curb & Dock System	Local Terrain	Global Region	Remote Management System	Global Positioning System
3	Account for System	The interaction of the vehicle with its external managers, in which it accounts for vehicle utilization.	x	x					x	x							x						x	
4	Aspirate	The interaction of the vehicle with the Local Atmosphere, through which air is taken into the vehicle for operational purposes, and gaseous emissions are expelled into the atmosphere.	x								x													
5	Attack Hostile System	The interaction of the vehicle with an external hostile system, during which the vehicle projects an attack onto the hostile system's condition.	x										x											
6	Avoid Obstacle	The interaction of the vehicle with an external object, during which the vehicle minimizes contact with or proximity to the object.	x				x																	
7	Configure	The interaction of the vehicle with people or systems that manage its arrangement or configuration for intended use.	x						x	x														
8	Deliver Vehicle	The interaction of the vehicle with the process of its delivery, including manufacture, distribution, and development. This includes delivery of each configured version and update of the vehicle product line or family.																						
9	Interact with Higher Control	The interaction of the vehicle with an external higher level management system, along with the vehicle operator, through which the vehicle is fit into larger objectives.	x														x							
10	Interact with Nearby Vehicle	The intearction of the vehicle with another vehicle, in which information is exchanged to identify one vehicle to another.																						
11	Interact with Operator	The interaction of the vehicle with its operator.																						
12	Maintain System	The interaction of the vehicle with a maintainer and/or maintenance system, through which faults in the vehicle are prevented or corrected, so that the intended qualified operating state of the vehicle is maintained.	x						x	x														
13	Manage Vehicle Performance	The interaction of the vehicle with its operator and/or external management system, through which the performance of the vehicle is managed to achieve its operational purpose and objectives.	x	x																				
14	Navigate	The interaction of the vehicle with the Global Positioning System, by which the Vehicle tracks is position on the Earth.	x																					x
15	Perform Application	The interaction of the vehicle with an external Application System, through which the vehicle performs a specialized application.	x													x								
16	Perform Dock Approach & Departure	The interaction of the vehicle with an external docking system, through which the vehicle arrives at, aligns with, or departs from a loading / unloading dock.	x																	x				
17	Refuel Vehicle	The interaction of the vehicle with a fueling system and its operator, through which fuel is added to the vehicle.	x									x												
18	Ride In Vehicle	The interaction of the vehicle with its occupant(s) during, before, or after travel by the vehicle.	x	x	x	x																		
19	Secure Vehicle	The interaction of the vehicle with external actors that may or may not have privileges to access or make use of the resources of the vehicle, or with actors managing that vehicle security.	x	x																				
20	Survive Attack	The interaction of the vehicle with an external hostile system, during which the vehicle protects its occupants and minimizes damage to itself.	x										x											
21	Transport	The interaction of the vehicle with a Vehicle Transport System, through which the Vehicle is transported to an intended destination.	x																x					
22	Travel Over Terrain	The interaction of the vehicle with the terrain over which it travels, by means of which the vehicle moves over the terrain.	x																		x			
23	View Vehicle	The interaction of the vehicle with an external viewer, during which the viewer observes the vehicle.	x					x																

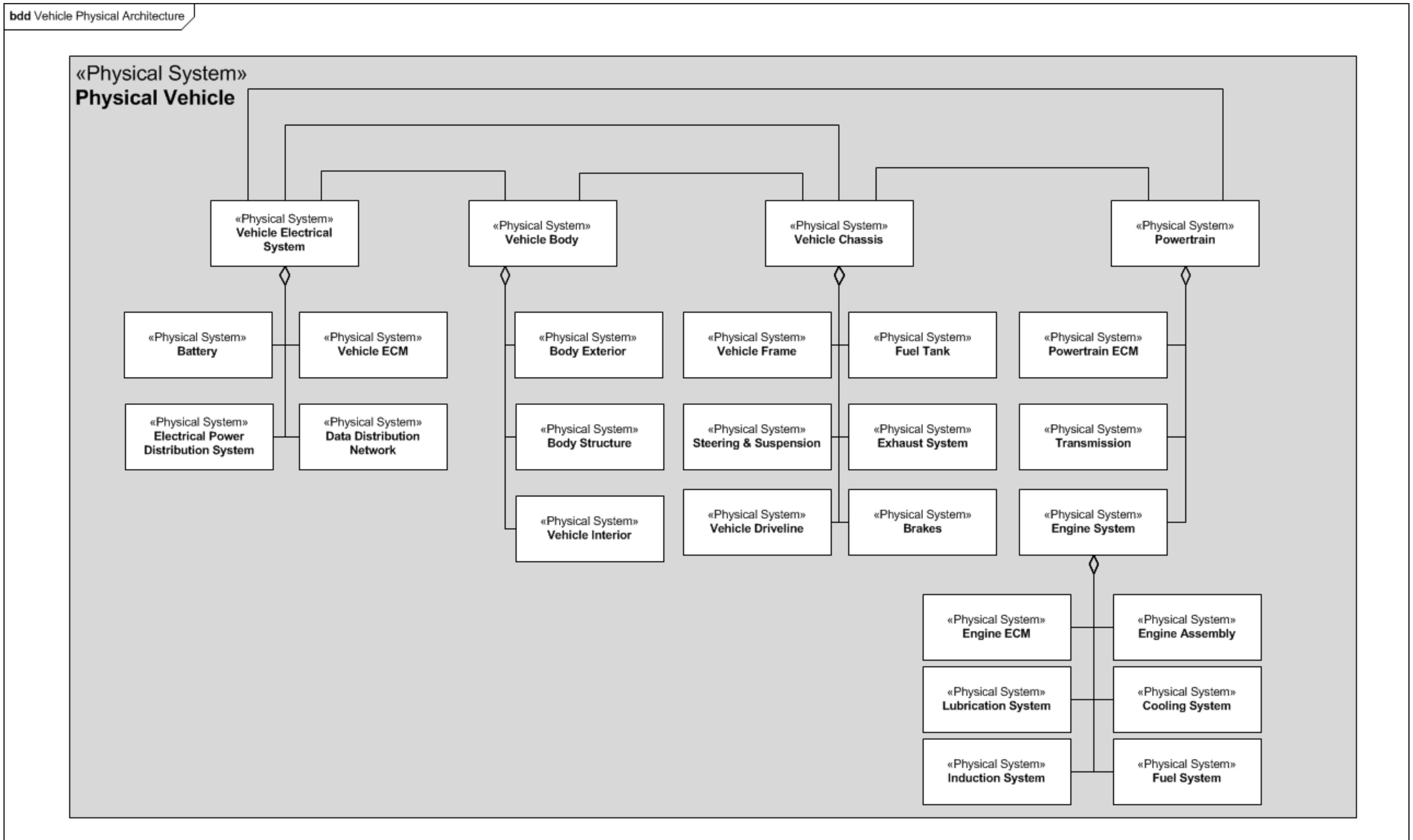

stm Vehicle State Diagram
Manufacture Started
Being Manufactured
Deliver Vehicle
Manufacture Completed
Being Transported
Transport Vehicle
Transport Completed
Start Transport
Vehicle Shut Down
Secure Vehicle
Interact with Operator
Service Started
Service Completed
Being Serviced
Maintain System
Fueling Started
Being Fueled
Refuel Vehicle
Fueling Completed
Salvage Started
Being Salvaged
Salvage Completed
Vehicle Start Up Requested
Vehicle Starting Up
Interact with Operator
Aspirate
Manage Vehicle Performance
Secure Vehicle
Vehicle Start Aborted
Vehicle Start Failed
Vehicle Started
Vehicle Shutting Down
Interact with Operator
Aspirate
Manage Vehicle Performance
Secure Vehicle
Vehicle Shut Down Completed
Vehicle Idling
Interact with Operator
Aspirate
Manage Vehicle Performance
Secure Vehicle
Navigate
Interact with Higher Control
Vehicle Shut Down Requested
Vehicle Idle Criteria Met
Vehicle Idle Criteria Exited
Vehicle Shut Down Requested
Operating
Aspirate
Interact with Nearby Vehicle
Interact with Operator
Navigate
Interact with Higher Control
Manage Vehicle Performance
Account for System
Secure Vehicle
Stopping Completed
Movement Started
Steady State Cruise Started
Stopping
Control Vehicle Direction
Control Vehicle Speed
Control Vehicle Power
Travel Over Terrain
Stopped
Maintain Position
Getting Underway
Control Vehicle Direction
Control Vehicle Speed
Control Vehicle Power
Travel Over Terrain
Cruising
Control Vehicle Direction
Control Vehicle Speed
Control Vehicle Power
Travel Over Terrain
Abort Stopping
Initiate Stop While Getting Underway
Initiate Stop While Manuevering
Maneuver Started
Returned to Cruising
Maneuver Started
Performing Maneuvers
Control Vehicle Direction
Control Vehicle Speed
Control Vehicle Power
Travel Over Terrain
Avoid Obstacle
Maneuvering Forward
Maneuvering Backward
Application Shut Down Requested
Shutting Down Application
Performing Application
Perform Application
Starting Up Application
Application Shut Down Completed
Application Idle Requested
Restart Application Requested
Application Started
Application Shut Down
Application Idle
Application Shut Down Requested
Application Startup Requested
Vehicle Entry Completed
Begin Vehicle Exit
Entering Vehicle
View Vehicle
In Vehicle
Ride In Vehicle
View Vehicle
Exiting Vehicle
View Vehicle
Begin Vehicle Entry
Outside Vehicle
View Vehicle
Vehicle Exit Completed


bdd Vehicle Logical Architecture
«Logical System»
Vehicle
Operator Interface
Weather Interface
Aerodynamic Interface
Aspiration Interface
Fueling / Charging Interface
Attackable Interface
Detection Interface
«Logical System» Vehicle Management System
«Logical System» Vehicle Passenger Environment Subsystem
«Logical System» Vehicle Interior Structural Subsystem
«Logical System» Vehicle Exterior Structural Subsystem
«Logical System» Vehicle External Appearance Subsystem
«Logical System» Vehicle Chassis Structural Subsystem
«Logical System» Application Subsystem
«Logical System» Vehicle Propulsion Subsystem
«Logical System» Vehicle Energy Storage & Conversion Subsystem
Passenger Interface
Pedestrian Interface
Attachment Interface
Higher Level Control Interface
Collision Interface
Inter-Vehicular Interface
Maintainer Interface
Maintenance Interface
Remote Management Interface
GPS Interface
Terrain Interface
Docking Interface
Transport Interface

Features	Interaction	Interaction PK Value	Functional Role	Req ID	Requirement
Accountability Feature[Operating Hours Accounting]	Account for System	Operating Hours Accounting	Vehicle	VEH-1002	The system shall record and make available for display the accumulated hours of vehicle operation.
Accountability Feature[Vehicle Mileage Accounting]	Account for System	Vehicle Mileage Accounting	Vehicle	VEH-1147	The system shall record and make available for display the accumulated distance since vehicle manufacture.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1132	The vehicle shall travel under the control of its operator, as to vehicle speed, acceleration, direction, and power.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1133	The vehicle shall be capable of sustained cruising speed of 80 miles per hour over Class 7C terrain.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1134	The vehicle shall be capable of accelerating from standing start to 60 miles per hour in not more than 12 seconds.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1135	The vehicle, loaded with its passenger and other load maximum, shall be capable of stopping from a speed of 60 miles per hour in 200 feet on dry pavement.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1136	The vehicle shall be capable of operating 5,000 miles between oil changes
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1137	The vehicle shall be capable of operating 50,000 miles between tire changes.
Automatic Braking System Feature[], Cost of Operation Feature[],	Travel Over Terrain		Vehicle	VEH-1138	The vehicle shall be capable of operating 25,000 miles between air filter changes.


par Vehicle Range Parametric Diagram
Range
Distance
Fuel Tank Capacity
:Vehicle Range
Tank Size
MPG Efficiency
Wind Profile
Acceleration Profile
accel
fuel efficiency
wind
incline
aeroresist
Terrain Profile
:Vehicle Performance
Aerodynamic Resistance
weight
octane
rollresist
Vehicle Weight
Rolling Resistance
Fuel Octane

Feature	Effect (Failure Impact)	Severity	Functional Failure (Counter Requirement)	Component	Failure Mode	Probability	Mitigation (Control)
Navigation Feature [GPS-based Location Sensing]	No Confidence in Displayed Position	Serious (4)	The system displays a location that is not accurate to 10 feet.	Vehicle ECM	Erratic ECM	0.0015	Nav Backup Mode: External Nav Module
Navigation Feature [GPS-based Location Sensing]	False Confidence in High Error Displayed Position	Critical (5)	The system displays a location confidence indicator that is not correct.	Vehicle ECM	Erratic ECM	0.0015	None
Navigation Feature [GPS-based Location Sensing]	No Displayed Location	Serious (4)	The system does not display the graphic map presentation.	Panel Display	Fractured Display	0.0003	Nav Backup Mode: External Nav Module



Acknowledgement: Influenced by related physical architecture work of John Thomas

5 Tooling and Language Mapping

S*Models and the S*Metamodel are neutral as to specific modeling languages and tools, and meant to be used with those tools and languages by means of "mappings" of the S*Metamodel into various languages and tools. An example of such a mapping into an OMG SysML tool is [24]. Following such a mapping and preparation of a tool-specific profile, S*Models may be created and managed in any such COTS tool.

5.1 A Starter Kit for S*Modelers

For those with access to Dassault Cameo Systems Modeler configured for OMG SysML modeling, the following resources can be used as a "starter kit" to establish an environment for creating S*Models:

Resource	S*Models Starter Kit [25]	S*Patterns Starter Kit
1. S*Models and S*Metamodel Primer	X	X
2. S*Patterns Primer		X
3. S*Patterns Configuration Wizard		X
4. S*Patterns Configuration Wizard Guide		X
5. S*Metamodel Document	X	X
6. Mapping Document to SysML in CSM	X	X
7. S*Profile and Project Template for CSM	X	X
8. S*Pattern Management Guide for CSM	X	X
9. Loadable Example S*Model: International Power Converter	X	X

6 References

1. INCOSE/OMG MBSE Patterns Working Group web site -- https://www.omgwiki.org/MBSE/doku.php?id=mbse:patterns:patterns
2. Tinniswood, A., *The Royal Society & The Invention of Modern Science*, NY, Basic Books, 2019.
3. Bowler, P. and Morus, I., *Making Modern Science: A Historical Survey*, Second Edition, Chicago, U. of Chicago Press, 2020.
4. Sobel, D., *A More Perfect Heaven: How Copernicus Revolutionized the Cosmos*, Bloomsbury Publishing Plc, London, 2011.
5. ISO 10303-1:1994 "Industrial Automation Systems and Integration -- Product Data Representation and Exchange -- Part 1: Overview and Fundamental Principles", International Standards Organization, 1994.
6. Schindel, W., "Got Phenomena? Science-Based Disciplines for Emerging Systems Challenges", in *Proc. Of INCOSE 2016 International Symposium*, Edinburg, UK, 2016.
7. Ioannidis, J., "Why Most Published Research Findings Are False", PLoS Medicine, 2005., - https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124
8. Schindel, W., "SE Foundation Elements: Discussion Inputs to INCOSE Vision 2035 Theoretical Foundations Section",V2.3.2a., INCOSE Patterns Working Group, 2020. Download from— https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:science_math_foundations_for_systems_and_systems_engineering--1_hr_awareness_v2.3.2a.pdf
9. Friedenthal et al (2021*) INCOSE Systems Engineering Vision 2035: Engineering Solutions for a Better World*. International Council on Systems Engineering, San Diego, CA.

10. Sillitto, H., Martin, J., McKinney, D., Griego, R., Dori, D., Krob, D., Godfrey, P., Arnold, E., Jackson, S., "Systems Engineering and System Definitions", San Diego, INCOSE, 2019. Download from -- https://www.incose.org/docs/default-source/default-document-library/final_-se-definition.pdf
11. Walden, D., et al, eds. (2023) *INCOSE Systems Engineering Handbook*, Fifth Edition, International Council on Systems Engineering, San Diego, CA, 2023.
12. Kline, M., *Mathematics and the Physical World*. Ch. 25: From Calculus to Cosmic Planning, pp. 441-442. 1959.
13. Schrage, M., The Innovator's Hypothesis: How Cheap Experiments Are Worth More Than Good Ideas, Cambridge, MA: MIT Press, 2014.
14. Kahneman, D., *Thinking, Fast and Slow*, Farrar, Straus and Giroux; 1st edition, 2013.
15. Kahneman, D., Slovic, P., Tversky, A., eds, *Judgement Under Uncertainty: Heuristics and Biases,* Cambridge U. Press, 1982.
16. Lewis, M., *The Undoing Project: A Friendship that Changed Our Minds*, W. W. Norton & Co. 2016.
17. Schindel, W., "What Is the Smallest Model of a System?", in Proc of 2011 INCOSE International Symposium, 2011. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:what_is_the_smallest_model_of_a_system_v1.4.4.pdf
18. Bargagli, F., Cevolani, G., Gnecco, G., "Simple Models in Complex Worlds: Occam's Razor and Statistical Learning Theory", Minds and Machines (2022) 32:13–42. https://doi.org/10.1007/s11023-022-09592-z
19. Li, Ming, Vitany, Paul. 1997. An introduction to Kolmogorov complexity and its applications. Second edition. Springer.
20. "Systematica Metamodel: Metamodel Version 8.0", April 7, 2022. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_6_metamodel_v8.0.34.pdf
21. "Metamodeling", Wikipedia, https://en.wikipedia.org/wiki/Metamodeling
22. Schindel, W., "Requirements Statements Are Transfer Functions: An Insight from Model-Based Systems Engineering", in Proc of INCOSE 2005 International Symposium, Rochester, NY. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:incose_2005.pdf
23. Schindel, W., and Peterson, T., "Introduction to Pattern-Based Systems Engineering (PBSE): Leveraging MBSE Techniques", INCOSE Great Lakes Regional Conference, 2016. Download from - https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_tutorial_glrc_2016_v1.7.4.pdf
24. S*Metamodel Mapping for MagicDraw/Cameo Systems Modeler Version 19 download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_mapping_for_magicdraw_csm_v1.11.4.pdf
25. "S*Models and the S*Metamodel--A Starter Kit", INCOSE MBSE Patterns Working Group, 2024.
26. Anderson, P., "More Is Different: Broken Symmetry and the Nature of the Hierarchical Structure of Science", *Science* 04 Aug 1972: Vol. 177, Issue 4047, pp. 393-396 DOI: 10.1126/science .
27. "Model-Based S*Patterns: A Primer", INCOSE MBSE Patterns Working Group, 2024.
28. "ISO 26580 Software and Systems Engineering: Methods and Tools for the Feature-Based Approach to Software and Systems Product Line Engineering", International Standards Organization.