Compliance and
Requirement Traceability for SysML v.1.0a

1. Introduction:

This document provides a formal statement of compliance and associated requirement traceability for the *SysML v. 1.0 alpha* specification, which the SysML Partners are submitting in response to the OMG *UML for Systems Engineering RFP* (OMG document ad/03-03-41).

Two tables are provided:

1. Statement of compliance to mandatory requirements
2. Statement of compliance to design goals (called "optional requirements" in the RFP document.)

Each table contains the following columns:

1. RFP requirement text
2. Attribute indicating whether the paragraph represents a requirement or optional requirement.
3. A statement of compliance with possible values
    a. Full – indicates that the proposal is fully compliant with the requirement
    b. Partial – indicates that the proposal is partially compliant with the requirement
    c. Non-compliant – indicates that the proposal is non-compliant with this requirement
4. A statement explaining how the solution satisfies the requirement
5. A reference to the chapter of the submission that addresses the requirement
6. A reference to the definition of the abstract and concrete syntax that addresses the requirement
7. A reference to the sample problem that illustrates the concepts

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6 Specific Requirements on Proposals						
6.2 Scope of Proposals Sought						
6.2.2 Definition of a UML Extension						
The customization of UML for SE must rely on extension mechanisms provided by UML and MOF. These include the various mechanisms that define a UML profile and optional use of additional extension mechanisms provided by the UML specification. A UML profile selects and constrains the use of existing UML modeling elements, optionally with new terminology and notations specific to that profile. Other extension mechanisms provided by the UML and MOF specifications may also be used, such as defining new types of UML modeling elements. A combination of these extension mechanisms may be used as well. For further information about the mechanisms that UML provides for its own extension, see the specifications referenced in Section 6.3 below.	True	Full	Strict profile approach taken. No new meta-classes.	Profiles and Model Libraries Language Architecture	Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	
6.5 Mandatory Requirements						
The "Resolution of RFP Requirements" section of any response to this RFP shall include a matrix indicating how the proposed solution satisfies each requirement in Sections 6.5 and 6.6, supplying the following information about each requirement:	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
a. Whether the proposed solution is a *full* or *partial* satisfaction of the requirement, or whether there is no solution provided.	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
b. Whether full or partial satisfaction of the requirement is accomplished using:	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
• a UML construct without modification	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
• a UML construct with modification	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
• use of an extension mechanism that defines a new UML modeling element	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
• other approach (with clarification)	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
c. Reference to the abstract and concrete syntax that satisfies the requirement	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
d. Reference to the part of the sample problem (see Section 6.7) that demonstrates how the proposed solution satisfies the requirement	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
e. Issues and comments	True	Full	This requirements traceability matrix satisfies this requirement.	App G: Requirement Traceablity Matrix		
6.5.1 Structure			Structure is described using SysML Block diagrams (Block Definition and Internal Block diagrams) and associated model elements.			
6.5.1.1 System hierarchy						
UML for SE shall provide the capability to model the hierarchical decomposition of a system into lower level logical or physical components, which include the following types:	True	Full	Hierarchical decomposition of blocks to any depth is possible. Blocks may contain blocks, which in turn may contain blocks. An external view of the block/part hierarchy may be depicted on Block definition diagrams. An internal view of the parts of a given block, including their interconnections and interfaces to the environment may be depicted on Internal Block diagram.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
a. Subsystem (logical or physical)	True	Full	May be represented by a block (definition), or part (usage in a given context).	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
b. Hardware (i.e. electrical, mechanical, optical)	True	Full	May be represented by a block (definition), or part (usage in a given context).	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
c. Software	True	Full	May be represented by a block (definition) which is based on UML structured class, or part (usage in a given context) based on UML 2.0 part. Seamless transition to software engineering.	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
d. Data	True	Full	May be represented by a block, attribute, part, data type or value type.	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
e. Manual procedure	True	Full	May be represented by a Block, part or activity.	Activities Blocks	Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-14 Activity Diagram: Control Power (page 1 of 4) Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
f. User/person	True	Full	May be represented by a block, part, or actor (external user of the system).	Use Cases Blocks	Table 14 Graphical Nodes for Use Cases Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-10 Use Case Diagram Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
g. Facility	True	Full	May be represented by a block or part.	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
h. Natural object	True	Full	May be represented by a block or part.	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
i. Node	True	Full	May be represented by a block or part.	Blocks	Table 3 Graphical Nodes for Blocks Table 4 Graphical Paths for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
6.5.1.2 Environment						
UML for SE shall provide the capability of modeling the environment as a collection of external systems and other elements of interest, which interact directly or indirectly with the system of interest.	True	Full	Depending on the level of detail required in the description of the external element of interest the environment may be represented as an actor or block (which may include internal structure (i.e. parts)). Interaction between the elements of the environment and the subject system may also be modeled. For example, if one consideres the Transmission of the sample problem as the "system" the Internal Block diagram of the Power Subsystem shows the Transmission in the context of other "external systems".	Use Cases Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 14 Graphical Nodes for Use Cases Table 15 Graphical Paths for Use Cases	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-10 Use Case Diagram
6.5.1.3 System interconnection						
UML for SE shall provide the capability to model the connections between systems.	True	Full	Connections between systems may be modeled using blocks and associations or parts and connectors. Ports and interfaces may be used for interface control.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem
6.5.1.3.1 Port						
UML for SE shall provide the capability to model the part of a system that provides access between the system's behaviors and properties, and its environment.	True	Full	Two types of port are defined in SysML: Service Ports used primarily for specifying digital command and telemetry interaction points ; and Flow Ports used primarily for specifying physical interaction points for the flow of material, energy or data.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-19 Block Definition Diagram: Properties of Transmission
6.5.1.3.2 System boundary						

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
UML for SE shall provide the capability to model the boundary of a system, which includes the set of all ports that connect the system to its environment.	True	Full	Blocks are encapsulated classifiers that may have specified interaction points (ports) and interface specifications. The "top-level" block in the block hierarchy can be used to model "the universe" with internal parts to model the subject system and elements of interest in the environment, which results in a clear specificaton of the system boundary and its external interfaces. In the Use Case diagram there is also a model element (Subject) that represents the boundary between the system and its environment, useful for identifying external entities of interest and corresponding external interfaces.	Blocks Use Cases	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 14 Graphical Nodes for Use Cases Table 15 Graphical Paths for Use Cases	Figure B-10 Use Case Diagram Figure B-19 Block Definition Diagram: Properties of Transmission
6.5.1.3.3 Connection						
UML for SE shall provide the capability to model the logical and physical connections between ports, and the associated interface information.	True	Full	The connection between ports is modeled using connectors. Connectors represent simple "lossless" connections between ports, in the sense that communication or flow across the connector is instantaneous and the characteristics of the item flowing are not modified in any way. If more elaborate models of the connection between ports is required a part can be used to model the communications/transmission channel/medium.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-23 Block Definition Diagram: Command and Telemetry Interfaces Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem
6.5.1.4 Deployment of components to nodes						
UML for SE shall provide the capability to model deployment of components to nodes as follows:	True	Full	Deployment of components to nodes may be modeled as a hierarchy of blocks (a block representing the node owns/hosts the block representing the component via composition) or via explicit allocation relationships.	Blocks Allocations	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
Software components, which execute on processing nodes	True	Full	Deployment of components to nodes may be modeled as a hierarchy of blocks (a block representing the node owns/hosts the block representing the software component via composition) or via explicit allocation relationships.	Blocks Allocations	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
Components, which are deployed to other types of components (i.e. nodes)	True	Full	Components deployed to other components may be modeled as a hierarchy of blocks (a block representing one component owns/hosts the block representing the other component via composition) or via explicit allocation relationships.	Blocks Allocations	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
Decomposition of nodes, such that a node may be decomposed into lower level nodes, which have components deployed to them	True	Full	Nodes may be decomposed into sub-nodes via block hierarchical decomposition (composition association). Deployment can then be represented as a further level of decomposition or via allocation relationship.	Blocks Allocations	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.1.5 Allocation of behavior to systems			Allocation relationships are defined to support explicit allocation of behavior to structure (functional allocation) or other allocations (requirements allocation to blocks, for example). Implicit allocation of behavior to structure can also be modeled using activity partitions, interaction lifelines or operations on a block.			
UML for SE shall provide the capability to model the allocation of behavior to systems and external systems as follows:	True	Full	See sub requirements.	Allocations	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission
a) Allocate function and states to systems	True	Full	Allocation relationships (explicit allocation), partitions, lifelines or operations may be used to model allocation of functions (activities) and states to systems.	Allocations	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission
b) Allocate inputs and outputs (including control inputs) to ports	True	Full	Allocation relationships may be used to allocate inputs/outputs to ports or to the attribute of their associated interfaces.	Allocations	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-21 Example tabular format of allocation traces Figure B-20 Block Definition Diagram: Allocation of Behavior to Transmission
6.5.2 Behavior			Behavior diagrams include activity, sequence, statemachine and use case diagrams.			
6.5.2.1 Functional Transformation of Inputs to Outputs			Activities (function based behavior) or statemachines (state based behavior) may be used to model the transformation of inputs to outputs.			
6.5.2.1.1 Input/Output			These can be modeled on activity diagrams as pins, parameters or object nodes .			

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
UML for SE shall provide the capability to model the inputs and outputs of a function with the following features:	True	Full	See sub requirements.	Blocks Activities	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
a) Inputs and outputs (I/O) are types of elements that can have properties	True	Full	Object nodes, pins, and parameters are used to model inputs and outputs. These elements are typed by blocks, datatypes or valuetypes that define the properties of the input/output item.	Blocks Activities	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4)
b) Inputs and outputs can be decomposed and specialized	True	Full	The classifier that types the inputs/outputs may be decomposed and specialize.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) Inputs and outputs are bound to ports	True	Full	Inputs/outputs are bound to ports via interfaces and/or allocation relationships.	Activities Allocations Blocks	Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem Figure B-24 Block Definition Diagram: Flow Specification Definitions
6.5.2.1.2 System store			A block may represent a system store in a block diagram, or an object node may be used to represent a store on activity diagrams.			
UML for SE shall provide the capability to model input/output elements that persist over time, with the following features:	True	Full	Object nodes may be used to represent depletable stores, datastore may be used to represent non-depletable stores.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
a) A store shall include non-depletable and depletable types.	True	Full	Object nodes may be used to represent depletable stores, datastore may be used to represent non-depletable stores. A block may be used to represent a store, for example the fuel tank or battery pack in the sample model.	Blocks Activities	Table 3 Graphical Nodes for Blocks Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-22 Internal Block Diagram: Internal structure of the Power Subsystem

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) A store can be a type of input or output.	True	Full	The contents of an object node or datastore are typed elements that may be used as inputs of activities. Similarly activities may output items to an object node or datastore.	Blocks Activities	Table 3 Graphical Nodes for Blocks Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities	Figure B-24 Block Definition Diagram: Flow Specification Definitions
6.5.2.1.3 Function						
UML for SE shall provide the capability to model functions that transform inputs to outputs, and which include the following features:	True	Full	Activities may be used to model functions that transform inputs to outputs.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
a) Functions should include the definition of the function name and its associated inputs and outputs.	True	Full	Action nodes are named and have pins that represent input and output parameters.	Activities	Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) Functions should support the creation, destruction, monitoring, and/or modification of elements, as well as a null transformation.	True	Full	Actions are defined for creation, destruction, monitoring and modifying elements. A null transformation (i.e. an action/activity that simply outputs the input item) can be modeled.	Activities	Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
c) Function ports, including function input ports and function output ports, bind the inputs and outputs to the function.	True	Full	Activity pins and parameters bind inputs and outputs to the activity.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
d) The specification of how I/O is handled, including how inputs are queued, stored, discarded, or how they may interrupt an active function.	True	Full	Object nodes can be used as queues and one may specify the type of queuing that the node implements (LIFO or FIFO). SysML extensions (overwrite, nobuffer) control if inputs are stored or discarded. Further extensions in SysML are available to define disabling an active function (ControlValue), specify if an activity may accept inputs once it has begun execution (streaming). Interuptable regions are also available to interupt an active function.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
e) The specification of resources, which are generated, consumed, produced, and released when the function executes.	True		Activity pins and parameters specify the resource consumed or produced when the function executes. Behavior specified by the modeler can lock/release resources to avoid contention issues.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
f) An output from one function can be an input to one or more other functions.. An output from a function can also be an input to a different activation of the same function.	True	Full	Activity edges define the connecton between activity pins and/or object nodes. Fork is used to split an output to several inputs. It is possible to model the output of an activity "feeding back" to its input.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
g) Functions can be decomposed into lower level functions, and include the association between the function ports at different levels of decomposition.	True	Full	Activities and actions may be decomposed.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
h) A function may be interruptible or non-interruptible.	True	Full	An activity/action has control inputs that enable the execution of a function and a control value from a control operator that can enable or disable the execution. An execution can also be disabled (interupted) if it is enclosed in an interuptable region.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
i) Functions can be specified by mathematical expressions, which define the transformation of input to output values.	True	Full	Behaviors may be associated with action nodes in an activity to perform computations. The behavior may be implemented as a method or a statemachine.	State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
j) The function can represent discrete and/or continuous transformations. Continuous transformations can transform continuous time varying inputs into continuous time varying outputs.	True	Full	Stereotypes may be associated with inputs and outputs to specify whether the inputs are continuous or discrete and whether they are streaming (i.e. accept/produce inputs/outputs while executing.)	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
k) The number of replicated functions that can concurrently execute is specified.	True	Full	Fork is available to start concurrent execution of activities/actions. One can specify the number of concurent activities/actions.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4)
6.5.2.2 Function activation/deactivation						
6.5.2.2.1 Control input						

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
UML for SE shall provide the capability to model a control input, which activates/deactivates a function, based on the following:	True	Full	Control flows in activity diagrams provide a means to activate and deactivate activities/actions.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
a) Functions have control inputs, which are a special type of input. If multiple control inputs are included, a default control operator must be specified to describe the logic to activate/deactivate the function.	True	Full	Multiple control inputs are possible with the default logic that all inputs must be active to activate the activity/action. Control inputs may also be combined using a Join node. The default logic of a Join node is to perform a logical AND operation on the input control edges, however UML 2.0 states that a Join node may have an associate joinSpec giving the conditions under which the node emits a control token.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
b) A control input should be a discrete valued input (i.e. enable/disable). Note: Continuous valued control inputs may be modeled as regular inputs.	True	Full	Control inputs are discrete values that can enable/disable an activity/action.	Activities	Table 9 Other graphical Elements for Activites Table 8 Graphical Paths for Activities Table 7 Graphical Nodes for Activities	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) A function can only be activated when its control input is enabled.	True	Full	Activities/actions are activated when all control inputs are active (i.e. a token is available on all control inputs) and all required data is available.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
d) A function is deactivated when its control input is disabled.	True	Full	A control operator may produce a control input to disable an activity/action. An activity/action enclosed in an interuptable region will also be disabled.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
e) A function can be deactivated when it completes the transformation of its inputs, or when a timeout occurs.	True	Full	An activity with non-streaming inputs and outputs terminates when it completes the transformation to produce an output. A timer used in conjustion with a ControlValue or interuptable region to disable an activty/action when a timeout occurs.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
Note: An activation and deactivation event is generated as described under the requirements for events and conditions.						
f) A default capability for functions without control inputs represented, assumes that the control input is always enabled.	True	Full	Actions/activities without explicit control signals will begin execution when all required inputs are available which is equivalent to the assumption that the input is always enabled.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-15 Activity Diagram: Control Power (page 2 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
g) The control input is the output of a control operator as described below, or a discrete output from another function.	True	Full	SysML supports the control operator described below. Control flow may also be a discrete output from another activity/action or the output of a control operator.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
6.5.2.2.2 Control operator						
SE UML shall provide the capability to model control operators as specialized types of functions, with the following features:	True	Full	SysML provides the capability to model control operators.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
a) A control operator provides control logic, to transform input events and conditions as specified below to discrete outputs that control the activation and deactivation of functions.	True	Full	SysML includes a general control operator that can provide activation/deactivation controls to other activity/action based on user defined logic.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
b) A control operator has N inputs and M outputs. The control operator outputs are enabled based on the events and conditions on the control operator input and the specified control logic.	True	Full	Any number of inputs/outputs may be used with a control operator and the user may specify the logic for generating the control outputs.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
c) The following types of control operators shall be provided:	True	Full	Basic UML 2.0 provides all of the operators below.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
i) Selection	True	Full	A decision node with guards implements a selection.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
ii) Fork	True	Full	UML 2.0 Fork is used to split a input into multiple outputs.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
iii) Join	True	Full	UML 2.0 Join is used to syncronize multiple inputs and provide an output. (AND)	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
iv) Merge	True	Full	UML 2.0 merge is used to provide an output from multiple inputs (OR)	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
v) Loop and iteration (including the loop and iteration limit)	True	Full	A combination of Decision node with guards and Merge nodes can be used to implement iteration.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
vi) Control operator which controls the activation/deactivation of replicated functions	True	Full	SysML defines a ControlValue for activating/deactivating activities/actions.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
6.5.2.2.3 Events and conditions						

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
UML for SE shall provide the capability to model events and conditions as follows:	True	Full	See sub requirements.	Sequences State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
a) Events can be specified in terms of an expression with a discrete output. The event occurs at the time the expression evaluates true.	True	Full	A number of events can be modeled in UML including the change of a property value, the sending/receiving of a signal, the invocation of a method, etc. Actions are used to specify the logic/expression for occurance of an event. Event occurances can b e modeled on sequence diagrams (message), statemachines (trigger) and activities (control or object flow).	Sequences State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4) Figure B-13 Sequence Diagram: Accelerate Scenario allocated to components of Hybrid SUV Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) An activation/deactivation event occurs when a function is activated or deactivated respectively. A deactivation event can be used to support activation of another function, and therefore readily supports modeling a sequence of functions.	True	Full	Activity Edges (control flow or object flow) may be used on activity diagrams to specify the sequence of activation/deactivation of activities/actions. On statemachines a trigger may cause a state transition as well as another event that triggers other statemachine behaviors (in the same statemachine or in another). The sequence of events can be depicted on a sequence diagram.	Sequences State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-13 Sequence Diagram: Accelerate Scenario allocated to components of Hybrid SUV Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) Conditions can be specified in terms of an expression with a discrete output, which is true as long as the expression evaluates true, and is false otherwise.	True	Full	Sequence diagrams provide constructs for conditional execution of interaction fragments based on expressions with discrete output. Statemachines and Activities provide decision nodes and guards that permit the conditional execution of transitions and other activitied, respectively, based on expressions with discrete outputs.	Sequences State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-13 Sequence Diagram: Accelerate Scenario allocated to components of Hybrid SUV Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.2.3 Function-based behavior						
UML for SE shall provide the capability to model function-based behavior to include I/O transformations, control inputs, and control operators as described in 6.5.2.1 and 6.5.2.2.	True	Full	Activity diagrams provide the capability to model function based behavior.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)
The execution rules for function-based behavior shall be specified, including rules for how I/O is handled, and rules for function activation/deactivation.	True	Full	Semantics of activities and actions are specified.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	
6.5.2.4 State-based behavior			Statemachine diagrams provide the capability to model state based behavior with the specific modeling construct listed.			
UML for SE shall provide the capability to model state-based behavior of systems, to include:	True	Full	Statemachine diagrams provide the capability to model state based behavior with the specific modeling construct listed.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	
a) States (finite), which enable system behaviors	True	Full	SysML reuses UML 2.0 ability to model states.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) Simple states, which have no nested states	True	Full	Simple states are supported.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
c) Composite states, which can include nested concurrent or sequential states	True	Full	Composite states which include nested concurent or sequential states are supported.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
d) Transition between states, which are triggered by events and conditions	True	Full	Transitions between states are triggered by event occurances (for example signal reception) or conditions (for example time out or completion).	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
e) Internal transitions	True	Full	A transition between substates of a composite state is supported.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
f) Logical operators on transitions (i.e. control operators)	True	Full	Psuedo states such as fork, join, and choice nodes permit branching and looping. Guards on transition provide the ability to to apply logical decisions on transitions.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
g) Actions, which occur as part of the transition between states	True	Full	Actions may be associated with transitions executed when the trigger event occurs.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
h) Functions, which can be activated while in a state (entry, do-while, exit)	True	Full	Entry, do while and exit actions are supported.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
i) Exit logic, which specify the sequence for exiting a state(s)	True	Full	Exit actions and the specification of their sequencing are supported.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
j) Capability to interrupt the functions while in a state	True	Full	Certain events may be designated as exit events (leading to an exit pseudo-state) to interupt functions while in a state.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
k) Interactions between systems that may influence their state-based behavior.	True	Full	Statemachines in different system may communicate via signal send and receive actions.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
l) Specialized type of states to represent failure or exception states	True	Full	User defined states associated with exception events and exit points may be used to model failures and the associated actions.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
m) The execution rules for state-based behavior shall be specified, including rules for transitioning between states, entering a state, and exiting a state.	True	Full	Semantics of statemachine includes rules for transitions, entering and exiting states.	State Machines	Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	
6.5.2.4.1 Activation time						
UML for SE shall provide the capability to model the interval of time that a function or state is active and inactive.	True	Full	The interval of time that an activity or state is active can be modeled using durations and time expressions. Time constraints and measurement points may be specified on sequence diagrams.	Sequences	Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.3 Property						
6.5.3.1 Property type						
UML for SE shall provide the capability to model properties to include the following types:	True	Full	SysML provides the capability to model all of the listed property types.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
a) Integer	True	Full	Primitive type provided by by base UML 2.0	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
b) Boolean	True	Full	Primitive type provided by base UML 2.0	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) Enumerated	True	Full	Primitive type provided by base UML 2.0	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
d) String	True	Full	Primitive type provided by base UML 2.0	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
e) Real	True	Full	Pre-defined data type in SysML.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
f) Complex variable	True	Full	Pre-defined data type in SysML	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
g) Vector/tensor	True	Full	Supported by user defined data type composed of primitive types and pre-defined types listed in a-f.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
h) Compound structures of the properties listed above	True	Full	Supported by user defined data type composed of primitive types and pre-defined types listed in a-f.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-24 Block Definition Diagram: Flow Specification Definitions Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.3.2 Property value						
UML for SE shall provide the capability to associate property values to properties, including the following:	True	Full	SysML defines a ValueType which extends data type to include the properties listed below. The SysML model library provides a number of pre-defined data types representing physical quantities such as mass, and associated quantity, unit, and dimension.	Profiles and Model Libraries Blocks App D: Non-Normative Model Library	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	
a) Value (including a default value)	True	Full	ValueTypes extend data types which have a value that can be assigned a default.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-24 Block Definition Diagram: Flow Specification Definitions
b) Units (e.g., pounds, meters)	True	Full	ValueType provides the capability to define the unit. The SysML library provides a number of pre-defined value types representing physical quantities with unit and dimension.	Profiles and Model Libraries Blocks App D: Non-Normative Model Library	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-19 Block Definition Diagram: Properties of Transmission Figure B-24 Block Definition Diagram: Flow Specification Definitions
c) Probability distribution associated with the property value, including mean and variance	True	Full	SysML defines distributed quantity which is a data type with associated probability distribution.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
d) Source (e.g., calculated, measured)	True	Full	Comments can be associated with properties to capture the source.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	
e) Reference (i.e. links to related information about the property, such as tables of a coefficient versus temperature)	True	Full	Comments can be associated with properties to capture reference data.	Profiles and Model Libraries Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	
6.5.3.3 Property association						
UML for SE shall provide the capability for the following entities to have properties:	True	Full	See sub-requirements.		Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
a) An element (i.e. system, component, input/output)	True	Full	Blocks may may own or reference properties.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-19 Block Definition Diagram: Properties of Transmission
b) A function	True	Full	Activities may have properties and parameters.	Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites	Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4)

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) An event or condition	True	Partial	Events themselves, as defined in UML 2.0 do not have properties. However, the trigger (such as a signal) can have properties.	Sequences State Machines Activities	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines	Figure B-13 Sequence Diagram: Accelerate Scenario allocated to components of Hybrid SUV Figure B-17 Activity Diagram: Control Power (page 4 of 4) Figure B-16 Activity Diagram: Control Power (page 3 of 4) Figure B-15 Activity Diagram: Control Power (page 2 of 4) Figure B-14 Activity Diagram: Control Power (page 1 of 4) Figure B-26 Transition Centric State Machine Diagram: Transmission "Switch Gear" Behavior Figure B-25 State Centric State Machine Diagram: Transmission "Switch Gear" Behavior
d) Another property (via property relationships as referred to below under parametric model)	True	Full	Properties may have properties via the composition association (hierarchical properties). Properties may be related to other properties via parametric constraints.	Parametric Constraints Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	Figure B-18 Block Definition Diagram: Equipment Breakdown Structure Figure B-27 Mass Constraints

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.3.4 Time property						
UML for SE shall provide the capability to model time as a global property, which can be accessed by other properties (i.e. any property can be a function of time).	True	Full	Time can be modeled as a Real property. Time ultimately derives from clocks. Tools that support execution and simulation can use the host computer clock for real time execution or time can be simulated and constrained using UML Simple Time modeling elements. Time constraints can also be specified via parametric constraints.	Sequences Profiles and Model Libraries Parametric Constraints	Table 11 Graphical Paths for Sequences Table 10 Graphical Nodes for Sequences Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-7 Internal Block diagram of composite parametric constraint IntegrateAndNormalize Figure B-6 Block definition diagram for composite parametric constraint IntegrateAndNormalize
6.5.3.5 Parametric model						
UML for SE shall provide the capability to model the following:	True	Full	See sub-requirements.	Parametric Constraints	Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
a) Properties and their relationships, which represent an arbitrarily complex mathematical or logical expression or constraint, between properties	True	Full	Parametric constraints support the specification of relationships between properties.	Parametric Constraints	Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	Figure B-27 Mass Constraints Figure B-4 Useage of constraints to map KPPS to MoE for Acceleration MoE Figure B-7 Internal Block diagram of composite parametric constraint IntegrateAndNormalize Figure B-6 Block definition diagram for composite parametric constraint IntegrateAndNormalize Figure B-5 Constraint definitions for Newton's Law
b) The corresponding mathematical and logical expressions and constraints, which specify the allowable range of values for the properties	True	Full	Parametric constraint definitions include the parameters of the constraint and the constraint expression (mathematical, logical, or informal specifications). The parameters of the constraint are associated with properties of the system via binding dependencies.	Parametric Constraints	Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	Figure B-4 Useage of constraints to map KPPS to MoE for Acceleration MoE Figure B-7 Internal Block diagram of composite parametric constraint IntegrateAndNormalize Figure B-6 Block definition diagram for composite parametric constraint IntegrateAndNormalize Figure B-5 Constraint definitions for Newton's Law

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
c) A reference to the language used to state the expressions and constraints	True	Full	A comment referencing he language for intepreting the constraint expression may be included as part of the contraint definition.	Parametric Constraints	Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	
6.5.3.6 Probe						
UML for SE shall provide the capability to model a probe, which is an element that monitors the values associated with one or more parameters (i.e. properties).	True	Full	The user model may contain blocks and associated parts that monitor specific values. SysML tools that support model execution typically provide the ability to "watch" any property in the model, simplifying the user model in that explicit probes are not needed. Parametric constraints can also be associated with properties to specify constraints.	Parametric Constraints	Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	
6.5.4 Requirement						
6.5.4.1 Requirement specification						
UML for SE shall provide the capability to model requirements associated with the desired capabilities, properties, behavior, and/or structure of a system, including the following types of requirements:	True	Full	SysML provides a Requirement model element that includes properties for the id, text, verification method, risk, and requirement type. Via user defined enumeration literals all of the listed requirement type can be supported. The default requirement types are: Functional, Performance and Interface.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
a) Operational	True	Full	Supported via user customization of requirement kind literals.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
b) Functional	True	Full	Default requirement kind literal.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-1 Requirement Diagram: Top-Level User Requirements
c) Interface (inputs and outputs, ports, etc.)	True	Full	Default requirement kind literal.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-1 Requirement Diagram: Top-Level User Requirements

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
d) Performance	True	Full	Default requirement kind literal.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-1 Requirement Diagram: Top-Level User Requirements
e) Activation/deactivation	True	Full	Supported via user customization of requirement kind literals.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
f) Storage	True	Full	Supported via user customization of requirement kind literals.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
g) Physical	True	Full	Supported via user customization of requirement kind literals.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
h) Design constraint or resource constraint	True	Full	Supported via user customization of requirement kind literals. Can also be specified via parametric constraints.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
i) Specialized (i.e. safety, reliability, maintainability, usability, security, cost, other life cycle requirements, etc.)	True	Full	Supported via user customization of requirement kind literals.	Requirements App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
j) Measure of effectiveness (MOE)	True	Full	Supported via user customization of requirement kind literals. SysML defines an <<effectiveness>> model element with properties score, weight, ID, text, and optimization direction to support trade studies.	Requirements App C: Non-Normative Extensions App D: Non-Normative Model Library	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-2 Measures of Effectiveness for the Hybrid SUV

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.4.2 Requirement properties						
UML for SE shall provide the capability to associate properties to a requirement.	True	Full	Default properties of requirements include id, text, requirement kind, verification method, and risk. Other properties could be added via user extensions as described in Profiles and Model Libraries Chapter.	Requirements Profiles and Model Libraries	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	Figure B-1 Requirement Diagram: Top-Level User Requirements
6.5.4.3 Requirement relationships						
UML for SE shall provide the capability to associate a requirement to one or more model elements, which include associations between:	True	Full	SysML defines a number of standard relationships between requirements and between requirements and other model elements. These are all forms of traceability (based on trace dependency). Standard relationships include: satisfy, verify, derive, allocate. Furthermore, requirements can be decomposed into sub-requirements.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-30 Sample Traceability Matrix Figure B-29 Requirement Diagram: Requirement Traceability
a) Derived requirements and their source requirements (trace)	True	Full	SysML provides standard derive trace relationship.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-8 Requirement Diagram: Requirements Derivation
b) Requirements and the model elements that realize and/or implement the requirements	True	Full	SysML provides standard allocate trace relationship and satisfy trace relationship.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-28 Requirement Diagram: Requirement Satisfaction
c) Requirements and goals of a system by hierarchical decomposition into lower level requirements and sub-goals	True	Full	SysML support the hierarchical decomposition of requirements via composition association.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-1 Requirement Diagram: Top-Level User Requirements
6.5.4.4 Problem						
UML for SE shall provide the capability to model a deficiency, limitation, or failure of one or more model elements to satisfy a requirement or need, or other undesired outcome.	True	Full	SysML provides a Problem model element that can capture issues, deficiencies, limitations or failures of one or more model elements to satisfy a requirement.	Auxiliary Constructs	Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.4.5 Problem association						
UML for SE shall provide the capability to associate a problem with one or more model elements.	True	Full	Problem model elements may be associated with one or more model elements.			
6.5.4.6 Problem cause						
UML for SE shall provide the capability to model a relationship between a problem and its source problems (i.e. cause).	True	Partial	The problem model element may capture descriptions of the root cause.			
6.5.5 Verification						
6.5.5.1 Verification Process						
UML for SE shall provide the capability to model the verification of a system, which is a process used to demonstrate the following:	True	Full	SysML suppports various aspects of verification planning and execution. This begins with the ability to capture verification method as part of each requirements properties. The verification relationship between a requirement and a test case provides traceability to the specification of the verification proceedure. SysML testcases are based upon the UML Test Profile. Finally, SysML tools that support model execution can excute the model as an early risk reduction/verification that the system performs as expected.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-9 Requirement Diagram: Requirement Verification Figure B-1 Requirement Diagram: Top-Level User Requirements
a) The system requirements have been properly allocated to the system components, such that the system requirements are satisfied if the components satisfy their requirements.	True	Full	Traceability analysis of allocation, derive and satisfy relationships (coverage, gap, impact analysis) and associated Rationale can be performed.	Requirements Allocations Auxiliary Constructs	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	Figure B-30 Sample Traceability Matrix Figure B-29 Requirement Diagram: Requirement Traceability

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) The implemented/realized system satisfies its requirements.	True	Partial	Model review or model execution may be used to check that the system as modeled performs as expected /desired. Executable test cases may be used to test the model or the implemented system.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-30 Sample Traceability Matrix Figure B-29 Requirement Diagram: Requirement Traceability
c) The requirements have been specified correctly to satisfy the higher-level needs (i.e. validation).	True	Full	Requirements traceability analysis, using satisfy, derive, allocate relationships and associated rationale may be used to assess coverage/completeness of requirements derivation and allocation.	Requirements Allocations Auxiliary Constructs	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	Figure B-30 Sample Traceability Matrix Figure B-29 Requirement Diagram: Requirement Traceability

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.5.2 Test case						
UML for SE shall provide the capability to model the input stimulus, expected output, and associated test criteria that verify that the system satisfies its requirements or needs.	True	Full	SysML defines a test case, based upon the UML Test Profile to model the stimulous, expected output, associated test criteria. All of the modeling elements and capabilities of SysML can be applied to modeling the test system as well as the system under test. In this case the system modeled is the test system.	Activities State Machines Sequences Requirements	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-9 Requirement Diagram: Requirement Verification
6.5.5.3 Verification result						
UML for SE shall provide the capability to specify the outcome from executing one or more test cases or test runs.	True	Full	SysML Test case provides a verdict to capture outcome of test runs.	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	
6.5.5.4 Requirement verification						
UML for SE shall provide the capability to model the comparison between a requirement and the verification results.	True	Full	The expected result of verification and verdict may captured in the test case associated with each requirement (via the verify relationship).	Requirements	Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.5.5 Verification procedure						
UML for SE shall provide the capability to model the functions needed to support execution of a test case or test run.	True	Full	An activity, statemachine or sequence diagram may be stereotyped as a test case to specify the verification procedure.	State Machines Sequences Activities Requirements	Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements	Figure B-9 Requirement Diagram: Requirement Verification

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.5.6 Verification system						

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
UML for SE shall provide the capability to model the system that implements the verification procedures.	True	Full	SysML may be used to model any system, including the system under test and the test system. All modeling elements and concepts applied to modeling the system under test are transferable to modeling the test system.	Profiles and Model Libraries Allocations Requirements Use Cases State Machines Sequences Activities Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 7 Graphical Nodes for Activities Table 8 Graphical Paths for Activities Table 9 Other graphical Elements for Activites Table 10 Graphical Nodes for Sequences Table 11 Graphical Paths for Sequences Table 12 Graphical Nodes for State Machines Table 13 Graphical Paths for State Machines Table 14 Graphical Nodes for Use Cases Table 15 Graphical Paths for Use Cases Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.5.6 Other						
6.5.6.1 General relationships						
UML for SE shall provide the capability to model the following relationships among one or more model elements, to support modeling abstractions, elaborations, and refinements:	True	Partial	Relationships and corresponding semantics defined in UML 2.0 and re-used in SysML support all of the listed relationship types.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
a) Association	True	Full	The association relationship.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
b) Collections (i.e. packages) to support organization of model elements	True	Full	Packages provide a means of structuring a model.	Auxiliary Constructs	Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	
c) Decomposition, including differentiating a partial versus a complete set of components	True	Full	SysML supports decomposition via the composition relationship. On its own the semantics of composition do not support the specification of complete vs. partial decomposition, however a comment associated with the relationship can be used to specify whether the decomposition is a partition or not.	Blocks Auxiliary Constructs	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	Figure B-18 Block Definition Diagram: Equipment Breakdown Structure

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
d) Dependency	True	Full	Dependencies and various extensions are supported.	Auxiliary Constructs Requirements Allocations	Table 16 Graphical Nodes for Allocations Table 17 Graphical Nodes for Allocations Table 18 Graphical Nodes for Requirements Table 19 Graphical Paths for Requirements Table 22 Graphical Nodes for Auxiliary Constructs Table 23 Graphical Paths for Auxiliary Constructs	
e) Generalization/specialization, including taxonomies of categories	True	Full	Generalization is supported.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
f) Instantiation	True	Partial	Parts represent proto-typical instances. With the exception of the case where the multiplicity of the part is greater than 1, the mapping from part to instance is 1 to 1.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
6.5.6.2 Model views						
UML for SE shall provide the capability to specify a model view as a subset of model elements and associated relationships that are of use to the modeler for a particular purpose and context. The model view shall include a description of its purpose and its context.	True	Full	Model views and viewpoints are supported.	Model Management	Table 20 Graphical Nodes for Model Management Table 21 Graphical Paths for Model Management	
6.5.6.3 Diagram types						
UML for SE shall provide the following diagram types:	True	Partial	See sub-requirements	App A: Diagrams		

Table 1: Statement of Compliance to Mandatory Requirements						
OMG UML for Systems Engineering RFP	**Req?**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
Standard UML diagrams with applicable extensions for UML for SE	True	Partial	The following UML 2.0 diagrams are supported: Class (Block Definition), Composite Structure (Internal Block), Sequence, Activity, and Use Case. In some cases these are extended (ex. Activity) in others they are restricted (Composite Structure and Class) to meet the needs of SE in accordance with the requirements of the RFP and the design approach to minimize the number of diagrams and model elements that must be learned. SysML does not specify Component, Deployment, Package, Object, Communications, Interaction Overview and Timing diagrams, however it is compatible with these diagrams.	App A: Diagrams		
Other diagram types as needed to support the requirements of this RFP, which include system context, parametric models, requirements relationships, causal analysis, verification models, and decision trees	True	Full	SysML adds the following diagrams to UML: Block Definitions (Class), Internal Block (Composite Structure), Parametric, and Requirement diagrams.	App A: Diagrams		
6.5.6.4 System role						
UML for SE shall have the capability to model a system role as a subset of the behavior, properties, and structure of the system, in support of specific interactions.	True	Full	Any Block may be used in a specific role/context via Internal Block diagrams which specify the context and interfaces of the role. SysML also supports collaborations, extended as Parametric Constraint definitions, that may be used to bind a system into a specific context.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
6.7 Issues to be discussed						
6.7.1 Sample Problem Description						
Submissions shall include models of one or more sample problems to demonstrate how their customization of UML for SE addresses the requirements of this RFP. The submitter may select one or more sample problems of their choosing, or apply their proposed solution to the sample problem descriptions included on the RFP page at http://syseng.omg.org/UML_for_SE_RFP.htm. The compliance matrix referred to in Section 6.5, must include a reference to the portion of the sample problem, which demonstrates how each requirement is being addressed.	True	Full		App B: Sample Problem		

Table 2: Statement of Compliance to Design Goals (Optional Requirements) of RFP						
OMG UML for Systems Engineering RFP	**Optional Req**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
4 Instructions for Submitters						
4.5 Responding to RFP items						
4.5.1 Complete proposals						
Submitters are highly encouraged to propose solutions to any optional requirements enumerated in Chapter 6.	True	Partial	A number of design goals ("optional requirements") have been addressed in V.1.0a. Others will be incorporated in future revisions of the language based on user feedback.			
6 Specific Requirements on Proposals						
6.6 Optional Requirements						
6.6.1 Topology						
UML for SE may provide the capability to model a network of nodes connected by arcs, which include the following features:	True	Partial	SysML provided the capability to model a network of nodes (parts, actions, states) connected by arcs (connectors, activity edges, transitions).			
a) Properties can be associated with an arc or node.	True	Non-Compliant	Properties of arcs would be modeled using Association classes in UML. SysML has purposely omitted association classes as it was felt that the cost/benefit of including this did not justify it. These could be included in a future version if desired.			

Table 2: Statement of Compliance to Design Goals (Optional Requirements) of RFP						
OMG UML for Systems Engineering RFP	**Optional Req**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
b) Model elements can be associated with an arc or node.	True	Partial	Some model elements (ex. rationale, problem) can be associated with arcs. Nodes may be associate with other nodes via connectors, associations, activity edges, and transitions.	Auxiliary Constructs Requirements Allocations Use Cases State Machines Sequences Activities Parametric Constraints Blocks		
6.6.2 Documentation						
UML for SE may provide the capability to represent a document with the following features:	True	Partial	In UML 2.0 a document is a stereotype of artifact.			
The document has attributes that can be used to capture information about the document.	True	Full	A UML <<document>> can have properties.			
The document can be related to one or more model elements.	True	Full	A document can be related to other model elements via trace dependencies.	Allocations Requirements	Table 17 Graphical Paths for Allocations Table 16 Graphical Nodes for Allocations Table 19 Graphical Paths for Requirements Table 18 Graphical Nodes for Requirements	
The document can be represented in text that can be specified in terms of the information contained in the related model elements.	True	Non-Compliant	Don't understand the requirement.			

Table 2: Statement of Compliance to Design Goals (Optional Requirements) of RFP						
OMG UML for Systems Engineering RFP	**Optional Req**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.6.3 Trade-off studies and analysis						
UML for SE may provide the capability to support trade-off studies and analysis with the following features:	True	Full				
a) Describe alternative models that may use common modeling elements.	True	Full	Views and packages may be used to represent alternative "models". Model libraries may be defined by the user and a standard library is provied.	Model Management Profiles and Model Libraries App D: Non-Normative Model Library	Table 20 Graphical Nodes for Model Management Table 21 Graphical Paths for Model Management Table 24 Graphical Nodes for Profiles Table 25 Graphical Paths for Profiles	
b) Describe criteria for evaluating the alternatives, along with their weighting.	True	Full	SysML defines a non-normative extension for Measures of Effectiveness that define the criteria for evaluating alternatives.	App C: Non-Normative Extensions	Table C-2 Graphical Nodes for non-normative extensions to Requirements	Figure B-2 Measures of Effectiveness for the Hybrid SUV
c) Model the effectiveness measures and corresponding optimization function, to assess how well the alternatives satisfy the criteria and weighting.	True	Full	The use of Measures of Effectiveness in conjunction with parametric constraints provides the capability to model the effectiveness measures, optimization function, nomalization (or value) function and weights in support of trade studied.	App C: Non-Normative Extensions Parametric Constraints	Table C-2 Graphical Nodes for non-normative extensions to Requirements Table 5 Graphical Nodes for Parametric Constraints Table 6 Graphical Paths for Parametric Constraints	Figure B-7 Internal Block diagram of composite parametric constraint IntegrateAndNormalize Figure B-6 Block definition diagram for composite parametric constraint IntegrateAndNormalize Figure B-5 Constraint definitions for Newton's Law Figure B-4 Useage of constraints to map KPPS to MoE for Acceleration MoE Figure B-3 Key Performance Parameters for Hybrid SUV

Table 2: Statement of Compliance to Design Goals (Optional Requirements) of RFP						
OMG UML for Systems Engineering RFP	**Optional Req**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.6.4 Spatial representation						
6.6.4.1 Spatial reference						
UML for SE may provide the capability to assign the position of an element relative to a reference coordinate system.	True	Full	An attribute of a block may be used to capture the position of the element relative to a reference coordinate system.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	
6.6.4.2 Geometric relationships						
UML for SE may provide the capability to model simple geometric relationships, such as the containment of one element within another.	True	Full	Composition associations imply containment of the part within the whole. Properties of the part may be used to capture specifics of the relationship.	Blocks	Table 4 Graphical Paths for Blocks Table 3 Graphical Nodes for Blocks	Figure B-18 Block Definition Diagram: Equipment Breakdown Structure
6.6.5 Dynamic structure						
UML for SE may provide the capability to model structure, which changes with time and is dynamically reconfigured, including:	True		See sub-requirements			
a) Create and destroy elements	True	Full	Parts may be dynamically created and destroyed dynamically.			
b) Time varying hierarchies	True	Full	Parts may be created and destroyed dynamically.			
c) Time varying interconnections	True	Full	Links (connectors) may be created dynamically.			
d) Time varying deployment	True	Full	Parts may be dynamically created and destroyed.			
6.6.6 Executable semantics						
UML for SE may provide the capability to provide fully executable models to support simulation and analysis, which may include reference to a constraint language.	True	Partial	SysML provides the provides all the required modeling concepts and elements to create executable models. The specific action language is not specified, but various action languages (such as C, C++, Java, tool specific) can be used.			
6.6.7 Other behavior modeling paradigms						
UML for SE may provide the capability to model other behavior modeling paradigms beyond state-based and function-based behaviors.	True	Non-Compliant	Don't understand the requirement.			

Table 2: Statement of Compliance to Design Goals (Optional Requirements) of RFP						
OMG UML for Systems Engineering RFP	**Optional Req**	**SOC**	**Requirement Satisfaction**	**Chapter Reference**	**Abstract and Concrete Syntax Reference**	**Sample Problem Reference**
6.6.8 Integration with domain-specific models						
UML for SE may provide the capability to integrate with domain-specific models, including electrical, mechanical, and specialty engineering models, such as reliability, safety, etc.	True	Partial	Alignment with AP-233 provides the basis for integration with other domain specific languages. Tools are expected to provide integrations with other models as well.	App F: ISO AP233 Model Interchange		
6.6.9 Testing model						
UML for SE may provide the capability to integrate with test analysis models to support automated verification.	True	Partial	SysML is compatible with the UML Profile for Testing.	Requirements	Table 19 Graphical Paths for Requirements Table 18 Graphical Nodes for Requirements	Figure B-9 Requirement Diagram: Requirement Verification
6.6.10 Management model						
UML for SE may provide the capability to integrate with management models, (i.e. schedules, work breakdown structures, risk identification and assessment, etc.).	True	Partial	Tools are expected to integrate with management models (ex. MS Project, Primavera, etc.).			