**RFP Requirements for Requirements and Verification**

**2/22/17 (Draft)**

The following are the draft RFP requirements that reflect the intent of the Requirements and Verification concepts in the SECM.

Note: The original UML for SE Requirements are included below as well.

**1. Requirement.**

1.1 Requirement definition group.

1.1.1 Requirement definition name. The SysML v2 specification shall provide the capability to model a requirement definition that can be used to constrain a solution.

1.1.1 Requirement unique identifier. The definition of a requirement shall contain a unique identifier.

1.1.2 Requirement attributes. The definition of a requirement shall contain the following optional attributes. (Note: this needs to be reconciled with other metadata and model element attributes that apply more generally)

* Status
* Priority
* Risk
* Originating Author
* Owner
* User defined attributes

1.1.3 Requirement text statement. The definition of a requirement shall contain an optional text statement.

1.1.4. Restricted requirement text statement group.

1.1.4.1 Restricted requirement statement. The definition of a requirement shall contain an optional restricted requirement statement which may include predefined sentence structures and key words.

1.1.4.2. Restricted requirement statement extensibility. The predefined sentence structure and key words for a restricted requirement statement shall be extensible.

1.1.5 Formal requirement statement group. 1.1.5.1 Formal requirement statement. The definition of a requirement shall contain an optional formal requirement statement.

1.1.5.2 Constraint expression. A formal requirement statement shall include one or more expressions to specify the constraints on an acceptable solution.

1.1.5.3 Assumptions. A formal requirement statement shall include one or more expressions to specify the assumptions for acceptable solutions (e.g., does the weight of a car include the fuel weight?)

1.2 Requirement group. The SysML v2 specification shall provide the capability to model a group of requirements that are used to constrain a solution.

1.2.1 Requirement usage identifier. Each requirement in a requirement group is defined by an optional requirement definition with a unique identifier.1.2.2 Requirement ordering. Each requirement in a requirement group shall have an order within the group that can be independent of its unique id.

1.2.3 Requirement usage (localized). The values associated with a requirement usage may over-ride the values of its definition.

1.2.4. Logical constraint. A logical constraint (e.g., And, Or, Exclusive Or, Not) can be applied to the requirements in a Requirement group. By default, the logical constraint in a requirement group is assumed to be a logical ‘And’.

1.3 Requirement definition relationships.

1.3.1 Requirement specialization

1.4. Requirement relationships. The following relate a requirement or requirement group to other model elements. (*confirm these apply to usage and not definition*).

1.4.1 Requirement composition. A r provides the means to aggregate constraints on a solution space.

1.4.2 Requirement satisfaction. Relates a requirement to a realization element that is asserted to satisfy it.

1.4.3 Requirement verification. Relates a verification case to the requirement it is intended to verify.

1.4.4 Requirement derivation. Relates a derived requirement to a source requirement.

1.4.5 Requirement allocation. Specifies a portion of a requirement to be satisfied by a component.

1.4.6 Requirement refinement. Relates a more precise requirement to a less precise requirement.

1.4.7 Requirement trace. A dependency relationship with a requirement or requirement group on at least one end of the relationship.

1.4.8 Requirement group relationships. Any external relationship applied to a requirement group applies to each member of the requirement group.

1.5 Logical constraint. A logical constraint (e.g., And, Or, Exclusive Or, Not) can be applied to one or more requirements relationships of the same kind, with an associated completeness property (e.g., complete satisfaction or partial satisfaction). By default, the logical constraint group is assumed to be a logical ‘And’.

1.X.X Requirement Support Information. SysML V2 shall provide the ability to associate supporting information to a requirement, requirement definition or a requirement group.

Definitions:

Because we did the concept diagrams first, most terms are already available in the glossary.

Supporting Information: Supporting Information provides additional information to help better understand the intent of a model element and specifically for a requirement or requirement group. This information can include items such as an introductions, one or more goals, a reference to further readings, justification, rationales, examples, diagrams, pictures, graphs, tables, etc.

**2. Verification.**

2.1 Verification objectives. The SysML v2 specification shall provide the capability to model the verification objectives.

2.2 Verification case. The SysML v2 specification shall provide the capability to model a verification case to verify that one or more requirements are satisfied by a unit under verification.

2.2.1 Verification success criteria. The verification case shall include the criteria used to evaluate whether the requirements are satisfied.

2.2.2. Verification methods. The verification case shall define the methods used to verify the requirements, including inspection, analysis, demonstration, and test.

2.2.3 Verification activities and results. The verification case shall include the activities to collect and evaluate the verification data, and the verification results that are output from the activities.

2.3 Verification system. The SysML v2 specification shall provide the capability to model the system and associated environment that is used to verify the unit under verification. (Note: the verification system may include verification elements that are combinations of operational and simulated hardware, software, people, and facilities.)

2.4 Verification relationships.

2.4.1 Requirement verification. (refer to above). (Note: Should there be a relationship between the verification case and the unit under verification?)

2.4.2 Verification objectives to verification cases. This relationship relates the verification cases to their verification objectives.

**UML FOR SE Requirements**

* + 1. **Requirement**

#### 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:

1. Operational
2. Functional
3. Interface (inputs and outputs, ports, etc.)
4. Performance
5. Activation/deactivation
6. Storage
7. Physical
8. Design constraint or resource constraint
9. Specialized (i.e. safety, reliability, maintainability, usability, security, cost, other life cycle requirements, etc.)
10. Measure of effectiveness (MOE)

Note 1: Requirements should include values and associated tolerances, where applicable.

Note 2: A stakeholder need, which represents a desired capability, is typically expressed as a high-level requirement, which may be further defined in terms of operational, functional, performance and other requirement types. The representation of high-level capabilities may include use cases, or other abstract models, augmented with text.

Note 3: The representation of system requirements is intended to address the complete life cycle process, from concept through disposal of a system, and as such may include requirements for the enabling systems (i.e. the production system, support system, etc), as well as the operational system.

#### Requirement properties

UML for SE shall provide the capability to associate properties to a requirement.

Note: The properties may include a reference to requirement criticality or weighting, level of uncertainty, risk, verification status, and/or other defined model elements. These may include elements of related models, such as the risk from a risk model.

#### Requirement relationships

UML for SE shall provide the capability to associate a requirement to one or more model elements, which include associations between:

1. Derived requirements and their source requirements (trace)
2. Requirements and the model elements that realize and/or implement the requirements

Note: This includes the allocation of requirements to components.

1. Requirements and goals of a system by hierarchical decomposition into lower level requirements and sub-goals

Note: This form of analysis is often used to identify high-level requirements before any system-level modeling takes place.

#### 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.

#### Problem association

UML for SE shall provide the capability to associate a problem with one or more model elements.

Note 1: A problem can be associated with the behavior, structure, and/or properties of a system or element at any level of the hierarchy.

Note 2: A problem can be associated either with the as-is system, which has the problem, or the to-be system, which is intended to correct the problem.

#### Problem cause

UML for SE shall provide the capability to model a relationship between a problem and its source problems (i.e. cause).

Note: This can be used to represent cause-effect relationships that are often depicted in fish-bone diagrams, failure modes and effects analysis, or fault tree analysis.

* + 1. **Verification**

#### 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:

1. The system requirements have been properly allocated to the system components, such that the system requirements are satisfied if the components satisfy their requirements.
2. The implemented/realized system satisfies its requirements.
3. The requirements have been specified correctly to satisfy the higher-level needs (i.e. validation).

Note 1: Verification methods include inspection, analysis, demonstration, test, or similarity.

Note 2:Validation methods may include focus groups, market testing, market surveys, prototyping, field demonstrations, and other elicitation methods.

#### 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.

Note 1: The test case can be a test scenario, which replicates the behavior of the external environment interacting with the system, to demonstrate that the system satisfies its functional, interface, and performance requirements. Alternatively, the test case can be a measurement of a physical characteristic, or an analysis that demonstrates that the system satisfies its requirements.

Note 2: Test criteria may include non-functional aspects, such as performance, and other requirement types indicated in 6.5.4.1.

Note 3: Test cases may be grouped into test runs to accomplish a specific test objective. A single test case may appear in more than one test run. Test cases are sometimes sequenced in a test run, reflecting dependencies between test cases. Failure of a test case in a sequence may result in the remainder of the test run not being executed.

#### Verification result

UML for SE shall provide the capability to specify the outcome from executing one or more test cases or test runs.

#### Requirement verification

UML for SE shall provide the capability to model the comparison between a requirement and the verification results.

Note: The comparison may yield a result of pass, fail, or not executed.

#### Verification procedure

UML for SE shall provide the capability to model the functions needed to support execution of a test case or test run.

Note: This can include the functions to generate an input stimulus and monitor an output response.

#### Verification system

UML for SE shall provide the capability to model the system that implements the verification procedures.

Note: The verification system can include test hardware and software, such as simulators and measuring devices, test facilities, and test operators (users).