# Review of Precise Semantics of UML State Machines (PSSM) submission draft 1

**C. Dickerson, M. Li & S. Ji**

**31st October 2016**

**Gaps in meeting RFP objectives:**

1. Comments 2, 3 are concerned with how the draft submission proposal meets the high level objectives of the PSSM RFP.
2. “The objective of this RFP is to solicit specifications containing more precise semantics for UML state machines to enable execution, allow model checking, and reduce ambiguities in UML models.” (Objective of This RFP, OMG Document: ad/15-03-02)
   1. The revised submission (draft 1), in Requirement 6.5.1f, requires that “These new axioms shall have explicit relationships with existing axioms of fUML base semantics”, i.e. extensions of the fUML metamodel shall have….
   2. The response claims: “This proposal does not extend the fUML base semantics.” (6.5.1f Response)
   3. Is there an implicit (explicit?) assumption that fUML is sufficient to implement PSSM to meet the 3 RFP objectives?
   4. The test cases appear to support the objective of ‘enable execution’; but there are no test cases for ‘model checking and reducing ambiguity.’ We recommend writing test cases for model checking, and reduction of ambiguities in UML models; or citing fUML test cases if they are sufficient for PSSM.
3. The revised submission (draft 1) proposes 15 requirements. These requirements appear to be only constraints. Are these constraints sufficient to ensure that fUML supports model checking, and reduce ambiguities in UML models?

**Recommendations:**

1. Test cases need to be included to illustrate how to pass parameters between states. For example, Figure 7.4 shows the Value meta-class of PSSM. Value specification is precise by Expression and OpaqueExpression meta-classes that link with Behavior and Parameter. Does the Behavior refer to activities? Need to specify test cases to illustrate parameter allocation.
2. Figure 7.7 illustrates the PSSM redefinition of StateMachine. For Transition, can the meta-class in the diagram that triggers states can be referenced to Activity Diagram to represent more complex events that lead to change of system states? Also, can Transition link to Operations of a Class in Class Diagram? We believe that Transition should provide some degree of traceability to Activity and Class Diagrams.
3. How to attach formulae or parameters? If possible, the Behavior meta-class on Figure 7.7 should add more such as “+,-,\*, /”.
4. Add probability on each transaction; then analysis methods such as Markov Chains can be used.
5. On page 2, “A particular execution tool can then realize specific semantics by suitably constraining the unconstrained semantic aspects and providing specifications for any desired variation at semantic variation points.”
   * 1. When a variation point is mentioned, it should be necessary to also list variants to allow configuration specification and management according to the target execution environment. (Is it reasonable to require a configuration algorithm?)
     2. Traceability between variation points and variants is needed.
6. For the event pool, events need to be to be able to be time related. (Is it reasonable to require a scheduling algorithm?)
7. Can *priority* be manually specified by a user when multiple transitions are coming in or going out? (A priority algorithm could be further specified by user demand.)

**Identified Small Errors:**

1. Some figures need to be adjusted in size to show text properly, e.g. Figure 7.4 in page 11, the note on meta-class “Expression” that “The expression mus” should be “The expression must”. Also, in Figure 9.64, the stopping point symbol is not graphically complete.
2. On page 85, Step 4 of RTC steps table should be “CE(S1.1)” rather than “CE(1.1)”.
3. What is the meaning of the greyed out parts of the metamodel diagrams (e.g. in Figure 7.7)?