UML Testing Profile 2 FTF Errata (ptc/2018-05-34)

For

FTF Submission (ptc/18-05-28 and ptc/18-05-29)

The following entries correct errors that were done in the resolution of UTP 2 FTF issues. The editing instructions listed in that errata sheet override erroneous editing instructions of resolutions reported in the UTP 2 FTF Report (ptc/18-05-27). The FTF submission document (ptc/18-05-28) already include the correct editing instructions listed in this document.

The errata addresses corrections for the following issue resolutions:

- Issue UMLTP2-14
   Summary of the correction: The predefined verdict 'fault' was removed from the UTP 2 Types Library. The corresponding element 'Fault' of the conceptual model (Clause 7.6 Arbitration Specification) was not removed by the resolution. This concept must be removed to keep the conceptual model and the profile specification in synch.

- Issue UMLTP2-24
   Summary of the correction: An obsolete entry 'Detailed Semantics' of 8.7.1.1.2.2 ArbitrationSpecification needs to be removed from the specification. Additionally, the Figure 8.1 Language Architecture need to reflect the removal of the UTP Arbitration library (old Clause 9.3, which is removed by the resolution of Issue UMLTP2-24)
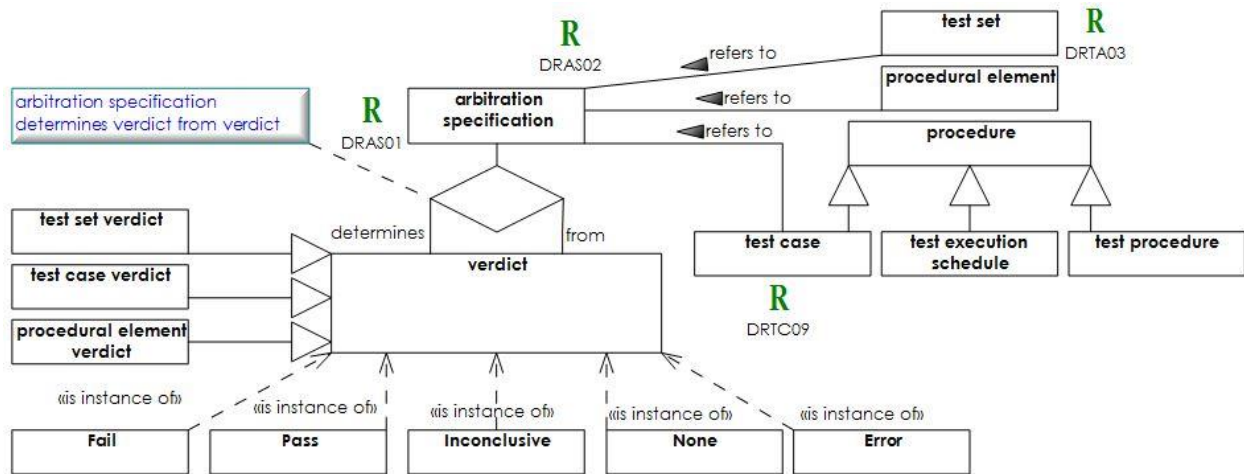
- Issue UMLTP2-27
   Summary of the correction: The newly introduced stereotype "ArbitrationResult" needs to be listed in the profile summary table in Clause 8.2
   Because of the introduction of a new stereotype, the numbering of the sub-clauses in clause 8.7.1.1.2 Stereotype Specifications needed to be updated. The new numbers that mentioned by the original issue resolution had a typo and were not correct. This is corrected by this errata document.

- Issue UMLTP2-29
   Summary of the correction: The newly introduced attribute 'expectationKind' of Clause 8.5.3.4.4 ExpectResponseAction was omitted by the issue resolution.

Corrections of editing instructions of resolution for Issue UMLTP2-14

Remove entry 'Fault' from the table in Clause 4 Terms and Definitions
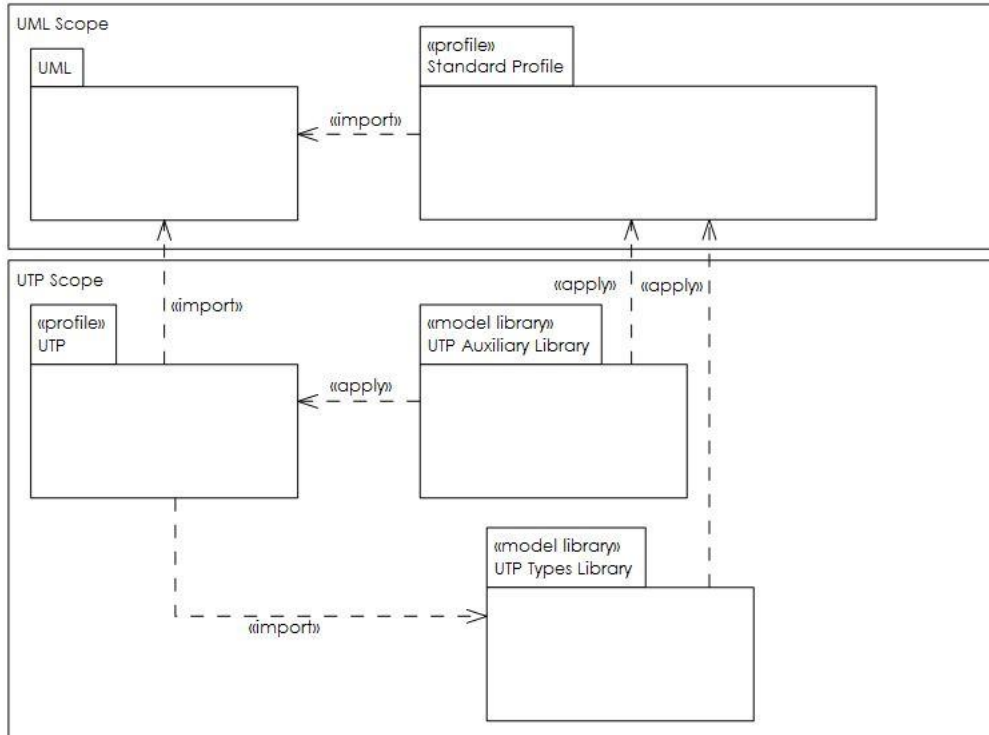
Replace Figure 7.9 Arbitration & Verdict Overview with

Remove subsection 'Fault' from Clause 7.6.1.2 Concept Descriptions

Corrections of editing instructions of resolution for Issue UMLTP2-24

Remove subsection 'Detailed Semantics' of new Clause 8.7.1.1.2.2 ArbitrationSpecification

Replace Figure 8.1 Language Architecture with

Corrections of editing instructions of resolution for Issue UMLTP2-27

Add new entry 'ArbitrationResult' into the table of Clause 8.2 Profile Summary between the entries 'AnyValue' and 'ArbitrationSpecification' with details:

Stereotype	UML 2.5 Metaclass	Concepts
ArbitrationResult	InstanceSpecification	

Replace the erroneous clause number changes made by the resolution with the following numbers for the respective clauses
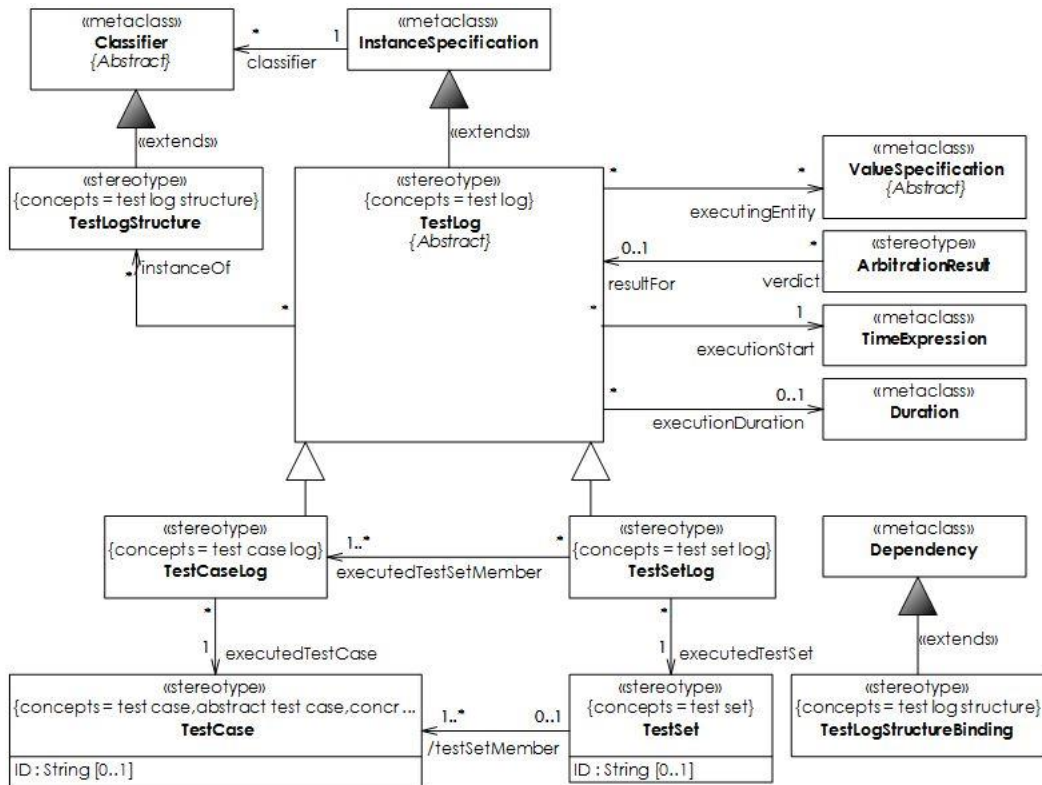
Clause 8.7.1.1.2.1 ArbitrationResult – instead of 8.7.1.1.1 ArbitrationResult

Clause 8.7.1.1.2.2 ArbitrationSpecification – instead of 8.7.1.1.2 ArbitrationSpecification

Clause 8.7.1.1.2.3 TestCaseArbitrationSpecification – instead of 8.7.1.1.3 TestCaseArbitrationSpecification

Clause 8.7.1.1.2.4 TestSetArbitrationSpecification – instead of 8.7.1.1.4 TestSetArbitrationSpecification

Replace the erroneous Figure 8.24 of the resolution with

Correction of editing instructions of resolution for Issue UMLTP2-29

Add new subsection 'Attributes' to Clause 8.5.3.4.4 ExpectResponseAction before subsection 'Associations'

Add new entry to subsection 'Attributes' of Clause 8.5.3.4.4 ExpectResponseAction with text

`expectationKind : ImplicitExpectationKind [1] = implicitForbid`

The expectation kind determines which of the three explicit sets in the context of an ExpectResponseAction is implicitly merged (union) with the complement set of the union of the other two sets. The following possibilities are:

- forbidden elements are implicitly unified (implicitForbid): Any received element that does not belong to the set of expected or ignored elements will be unified with the explicit set of forbidden elements during test execution. This prevents (or reduces the likelihood of) 'false negatives'.
- ignored elements are implicitly unified (implicitIgnore): Any received element that does not belong to the set of expected or forbidden elements will be unified with the explicit set of ignored elements during test execution. Care must be taken when going for this mechanism, since it is prone to 'false negative' results in case a forbidden element was forgotten to be explicitly defined in the corresponding set.
- expected elements are implicitly unified (implicitExpect): Any received element that does not belong to the set of ignored or forbidden elements will be unified with the explicit set of expected elements during test execution. Care must be taken when going for this mechanism, since it is prone to 'false negative' results in case a forbidden element was forgotten to be explicitly defined in the corresponding set.