|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **Title\*:** | MTS TDL meeting Minutes | | |
|  | 09 Feb 2015 | | |
| from **Source**\*: | MTS Chairman | | |
| Contact: | Stephan Schulz | | |
|  |  | | |
| input for **Committee**\***:** | MTS | | |
|  |  | | |
| Contribution **For\*:** | Decision | **X** |  |
|  | Discussion |  |  |
|  | Information |  |  |
|  |  | | |
| Submission date**\***: | 2015-02-10 | | |
|  |  | | |
| Meeting & Allocation: | **MTS-TDL meeting** | | |
|  | | | |

## Review of TDL SG Feb 5 meeting minutes and decisions MTS(15)000020

## Decisions on issues related to “OmitValue”

### Semantics of OmitValueMTS(15)000012r1 and MTS(15)0000019

**Issue raised**: [The standard] explains the meaning of OmitValue only in context of optional members in a structured data type. However OmitValue can be assigned indirectly also to variables or other mandatory members at runtime. The provision of static semantics rules (constraints) is not sufficient to forbid this case.

Ericsson position is that constraint in clause 6.3.8 is clear enough: “'OmitValue' can be assigned only to optional 'Member' s of 'StructuredDataInstance's.” The draft **\*doesn’t specify that Constraints are relevant for the static semantics only\***, therefore runtime violation of constraints shall cause an error as well. To be aligned to ETSI editing rules, “can” shall be changed to “shall” in the constraint: “'OmitValue' shall be assigned only to optional 'Member' s of 'StructuredDataInstance's.”.

Siemens states we cannot extend a static semantics rule to mean also execution semantics automatically.

Note: TDL SC discussion on Feb 5 reached consensus that a clear statement about meaning of Semantics and Constraints subheadings needs to be explicitly added (at some point) to the conformance clause in part 1. Common knowledge should in this case not be assumed.

Decision: This issue can not be resolved and should be resolved as part of TDL Phase 3

AP Andreas: Create Mantis CR.

### Assigning OmitValue to variables and mandatory fields runtime / use of omit as an argument MTS(15)000012 and MTS(15)0000019

**Issue raised**: [The standard] needs to state in the relevant clauses what it means when OmitValue is assigned to a variable or a mandatory member at runtime.

Ericsson proposal is to delete newly introduced sentences referring to use of “<undefined>” from 6.3.10 FunctionCall and 6.3.12 VariableUse.

Siemens position is that it is ok to delete the newly introduced statements on undefined. There needs to be a restriction in semantics of DynamicDataUse that a data use expression that evaluates to omit at runtime causes undefined semantics at TDL level.

Decision: Remove the newly introduced sentences

### Checking if an optional member is omitted or not

**Issue raised**: A user needs to have the possibility to check at runtime whether a variable or a member have the OmitValue assigned to it.

Ericsson position is that OmitValue is defined in TDL as a value and therefore no change is needed.

## AOB

Decision: Create Mantis CR on clarifying use of sections in the standard clauses for specifying static and dynamic semantics

Decision: New revision of MTS 000014 to be uploaded today including above decisions and decisions from previous TDL SG meeting. Review open to all Wed Feb 11.

Launch of AbC approval for all standard parts targeted at Feb 12