|  |
| --- |
|  |
| **Title\*:** | Ericsson’s proposals to resolve issues raised to TDL MM draft |
|  | 04 February 2015 |
| from **Source**\*: | L.M.Ericsson |
| Contact: | György Réthy  |
|  |  |
| input for **Committee**\***:** | MTS |
|  |  |
| Contribution **For\*:** | Decision | **X** |  |
|  | Discussion |  |  |
|  | Information |  |  |
|  |  |
| Submission date**\***: | 2015-02-04 |
|  |  |
| Meeting & Allocation: | [**MTS-TDL\_Steering Group meeting**](http://webapp.etsi.org/MeetingCalendar/MeetingDetails.asp?mid=16052) **on 05th February 2015** |
|  |

Please find below the result of the analysis and proposed solutions to the issues raised by Siemens in an email following MTS#64. For completeness the whole email text is attached at the end of this TD.

***1) A sentence in clause 9.4.6 Interaction***

**Issue raised**: The statement "The occurrence of the <undefined> value within the 'DataUse' specification of 'argument' causes undefined semantics of the 'Interaction' behaviour at runtime." has the same meaning as stating that the argument shall be fully specified.

**Conclusions**:

- The sentence can be understood as if <undefined> value in the argument of an Interaction \***caused in all cases**\* an undefined runtime semantics of the Interaction itself. This is certainly not the case. An <undefined> value, just like e.g. an \_*incorrectly defined concrete*\_ value, may cause an unexpected runtime behaviour of the SUT. But this is an error in the TDL specification or in the underlying infrastructure, not a direct consequence of the existence of <undefined>.

- This concrete question has been discussed and decided by TB MTS to allow partially defined messages.

**Solution**:

The above **sentence** has been deleted in paragraph 5 of clause 9.4.6 and re-inserted again as paragraph 7. It shall be **deleted** in the latest final draft as well; adding a note, warning the users though may be useful (strikethrough font is by Gyorgy, in the draft the text is in normal font):

“~~The occurrence of the <undefined> value within the 'DataUse' specification of 'argument' causes undefined semantics of the 'Interaction' behaviour at runtime.~~

NOTE: The  <undefined> value within the 'DataUse' specification of 'argument' is resolved outside of the TDL specification that may leave to an unexpected runtime behaviour of the SUT.”

***2) Semantics of OmitValue***

**Issue raised**: ”It 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.”

**Conclusion**: The constraint in clause 6.3.8 is clear enough: “'OmitValue' can be assigned only to optional 'Member' s of 'StructuredDataInstance's.” Therefore - like violation of any other constraints - an attempt to assign OmitValue to variables, parameters or mandatory fields shall cause an error. 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.

Consequently, the example:
“var x = msg.field;
If (x == omit) then doSomething();”
shall cause an error.

On the other hand the example can be changed to have the same behaviour:

var x = msg;

if (x.field == omit) then doSomething();

**Solution**:

- There are two TDs for the TDL MM draft on the ETSI server: TD MTS(15)64\_003r4 was uploaded at 29-01-2015 08:24, and it contains the above constraint. However, there is also another TD: MTS(15)64\_046, uploaded at 29-01-2015 11:40, just at the beginning of the TDL discussion, which doesn’t contain the above constraint.

TD MTS(15)64\_003 has been agreed at the beginning of MTS#64 as the TD containing the TDL MM draft. MTS(15)64\_046 has not been agreed by the meeting to include it into the agenda (it would not be possible anyway as experts would have had zero time to review it). Therefore it shall be reinforced that the version discussed and agreed at MTS#64 is **MTS(15)64\_003r4**.

- 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.”.

***3) Assigning OmitValue to variables and mandatory fields runtime***

**Issue raised**: “needs to be stated in the relevant clauses what it means when OmitValue is assigned to a variable or a mandatory member at runtime”

**Conclusion**: OmitValue can be assigned to optional members only (see previous item). No new feature to be able to assign OmitValue to parameters or variables (as a whole) or mandatory members has been discussed or agreed by the TDL Technical session on 28th January. See the minutes of the meeting in TD MTS(15)64\_043. If users request such new features, it shall be discussed and included into TDL in a consistent way.

**Solution**:

Assigning OmitValue to parameters, variables and mandatory members is not allowed. The following changes are required in MTS(15)64\_003r4:

- In clause **6.3.10** FunctionCall **delete** **the newly inserted sentence** (strikethrough font is by Gyorgy, in the draft the text is in normal font):
“~~If one of the 'DataUse' specifications provided in the 'ParameterBinding' of the 'FormalParameter's has the value <undefined>, the 'FunctionCall' represents also the <undefined> value.~~”

- In clause **6.3.12** VariableUse **delete the newly inserted sentence** (strikethrough font is by Gyorgy, in the draft the text is in normal text):

“~~If the accessed 'Variable' has the value <undefined> assigned to it, the 'VariableUse' represents also the <undefined> value.~~”

- Regarding mandatory fields no change is needed, the constraint in clause 6.3.8 forbids assigning OmitValue to mandatory members..

***4) 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…”

**Conclusion**: OmitValue is defined in TDL as a value, therefore (in a concrete syntax example)
variable.field != omit;
is possible. As Andreas has used this syntax in his example as well, we consider that this issue is pointless.

**Solution**: No change is needed.

