# Document changes and Comments to AG's revision of the TDL-MM doc from 12-Jan-2015 by AU

### 3.1 Definitions

AU: Clauses added to definitions when appropriate. New definition: tester-input event (was: trigger event):

"**tester-input event**: an event that occurs at a component in the role tester and determines the subsequent behaviour of this tester component; it typically impacts the test verdict; a tester-input event is one of the following: an interaction which targets a tester component (clause 9.4.6), a timeout (clause 7.2.12), or a quiescence (clause 7.2.7)"

### 5.2.2 NamedElement

GA: I guess, it is a concrete syntax related question or if remains here some restriction shall be done, since in the name property of Element it is stated: " name: String [0..1]

The name of the 'Element'. It can contain any character, including white-spaces. Having no name specified is different from an empty name (which is represented by an empty string).

" It includes that the name itself can contain '::'

AU: This text survived from TDL1. I have no strong opinion about it and keep it as it is. Should be discussed in TDL3.

### 5.2.5 ElementImport

GA: **Meaning of empty imported elements list**If 'importedElement' set is empty, it implies that all elements of the imported 'Package' are imported.

AU: This is not a language constraint, but part of the semantics description.

### 6.2.11 StructuredDataInstance

AU: Confusing statement on "uninitialized members" deleted.

### 6.2.13 Parameter

GA: Why not NamedElement? Is there any practically relevant situation when a parameter has no name? How can it be used then?

AU: NamedElement implies also qualified name, which is not needed for a parameter. Moreover, the name can be skipped in practical situations! Example: A TDL function declaration only needs to provide an ordered list of parameter data types. The name can be implicitly derived from the parameter position but is actually not needed when the function is called, e.g. consider the addition function "+" for integers.

### 6.2.15 Variable

AU: Confusing statement on "uninitialized variables" deleted.

### 6.3.7 AnyNoneValue

AU: Definition simplified to: "An 'AnyNoneValue' denotes an undefined symbolic value from the union set of 'AnyValue' and 'NoneValue'."

### 7.2.1 Time

GA: It is required that the same instance of the 'Time' data type shall be used in all 'DataUse' expressions within a TDL model.

AU: As we discussed, this requirement is not needed at the level of abstract syntax. The MM is robust for the use of multiple time units and also the semantics does not suffer. TDL has no implicit type conversion, therefore the use of multiple time units is save.

The problem is similar to multiple but similar data type definitions, e.g. byte, int16, uint32. A user will be able to derive the correct data type of, say, "1" only from context. It is rather a usability issue (which could be improved by adding the type name to a data instance in a concrete syntax, not only for time values).

Therefore the problem is best addressed at concrete syntax level. For example, the EBNF rules of a concrete syntax can be drafted such that Time can be instantiated only once. I revised the text to make this point clear:

"When designing a concrete syntax from the TDL meta-model, it is recommended that the 'Time' data type can be instantiated at most once by a user and the same 'Time' instance shall be used in all 'DataUse' expressions within a TDL model; let it be the predefined instance 'Second' or a user-defined instance. This assures a consistent use of time-related concepts throughout the TDL model."

Though you are right, the provided example EBNF definitions in B.5.2 allow multiple instances of Time (which I cannot change quickly without making a likely mistake).

### 7.2.7 Quiescence

GA: Let's not use the term 'trigger' in connection with the first event of alternative/exception, because 'trigger' means different in interactions, therefore could be misunderstood. I propose the term 'trigger event' but if somebody has a better idea, welcome.

AU: Excellent hint! After some contemplation I decided for the term "tester-input event" which I added also to the definition part in 3.1.

### 8.2.4 ComponentInstance

AU: Confusing statement on "uninitialized variables" deleted.

### 8.2.8 TestConfiguration

GA: GateReference [instead of GateInstance] is more appropriate.

AU: Gate reference is owned by the connection. Therefore it needs to be gate instance, indeed. Some text refinement done.

### 9.4.6 Interaction

GA: Can be an argument with variables where one of them belongs to sender and other to target? Not only one of them (the 'Tester' side)?

AU: First, variables, like timers, can be used only from tester components. Second, using a variable means using its value (see clause 6.3.12 on variable use). If the variable is on the receiving side, the value of this variable needs to be matched by this interaction. In case of tester-to-tester communication, if the variable is on the sender side, the effect is that the value stored in this variable is transferred to the receiving tester. (To store this transferred value, another variable referenced under Target on the receiving side is needed.)

AU: Some further clarifications on the occurrence of NoneValue in the argument specification of an interaction.

### 9.4.7 Target

GA: Matching data type for 'argument' and 'variable'
The 'DataUse' specification of the 'argument' and the referenced 'Variable' of a 'Target' shall refer to the same 'DataType'.

AU: This constraint is already defined for Interaction above. It is therefore duplicated. Moreover, the data use specification of the argument is not known by the target.

# B.2 A 3GPP Conformance Example in Textual Syntax

GA: What is the meaning of underlying?

AU: I guess it is an artifact left over from the tool. Deleted.

GA: Shall it [verdict type] be defined explicitly, since it is predefined? Shall they [verdicts] be defined explicitly, since they are predefined, and with lowercase letters?

AU: While verdicts are predefined, there is no import statement I would expect to make standard definitions known to this TDL spec. Moreover, the comment above states "user-defined verdicts".

GA: Shall it [Time SECONDS] be defined explicitly, since it is predefined, and the predefined spelling is : 'Second'?

AU: Similar to above. Consider it as user-defined.

## B.5.2 TDL Textual Syntax EBNF Production Rules

GA: Predefined Second missing

AU: Predefined instances of TDL meta-classes will be provided as a library. This library needs to be imported to make use of it. They do not need to be part of the concrete syntax definition. Similar to Java and the Java SDK.