A TDL Concrete Syntax for ETSI: Analysis Report

Prepared for: MTS Plenary #62
Prepared by: Philip Makedonski, STF 476
5 Apr 2014

Executive Summary

The present report summarises STF 476's findings and recommendations from the analysis of ETSI's needs and requirements for a concrete textual and tabular syntax for TDL. Submitted to the attention of TC MTS at the 62nd MTS Plenary, it will serve as basis for a decision on whether to pursue further activities towards a standardised concrete syntax for TDL which is tailored to support test specification processes at ETSI and 3GPP. Depending on this decision, a new Terms of Reference for the extension of the scope and resources of STF 476 may be submitted for the second half of 2014[1]. It will cover the necessary tasks and activities leading to a new Part 4 of ES 203 119 in December 2014, describing a concrete syntax for TDL addressing the current and future needs of the test specification processes at ETSI and 3GPP.

Introduction and Motivation

Based initial discussions with CTI and feedback on the current practices at ETSI, the following key use cases for test descriptions were identified:

1. **Documentation and communication** - the test descriptions serve as the basis for high-level discussions at often large meetings (80 to 100 participants), bridge the gap between management, core specifications experts, and testing experts. The level of unnecessary technical detail shall be reduced as much as possible. Furthermore, the notation is used as a primary constituent in technical specifications and related documents which play a central role in ETSI processes. While graphical representations may augment such documents in the future, they cannot be considered a viable replacement for the present notation.
2. **Basis for implementation** - the test descriptions serve as high-level designs on which executable test specifications are based. Given that test specifications were previously based either on 1-2 sentence objectives or on requirements directly, the currently used TPLan-like format is considered an improvement and has a reportedly high acceptance among test engineers. Additional technical details may be considered beneficial, especially if these can be derived by intelligent means, such as a standardised formalised transformation.

[1] Based on updated information from CTI and ETSI, an extension of 20 days (or less) can be requested and granted by the ETSI DG without going to the Board for approval, and thus also without the need for new Terms of Reference. Instead, a request with sufficient motivation shall be sent to Alberto (at any time, but the sooner the better).

Finding the right balance between the required level of detail for both use cases can be a difficult task. The TPLan-based notation currently in use at ETSI provides a natural language look and feel, with the necessary flexibility that allows users to determine the amount detail included. Nonetheless, the largely informal character of the notation also introduces certain challenges, such as potential inconsistencies and ambiguities in the specifications and corresponding maintenance overhead, poor or no tool support due to machine-supported analysis and transformation into other artifacts being difficult or impossible to achieve. Due to these challenges, TPLan as such has not been decisively adopted as the notation of choice for the specification of test purposes at ETSI. Rather, a variety of "dialects" have emerged which adopt certain notions from TPLan, yet do not strictly adhere to all principles defined in the TPLan standard.

CTI provided the STF with examples of this notation used in the *Conformance test specification for Co-operative Awareness Messages (CAM)* [CAM2011] and the *Conformance test specifications for Transmission of IP packets over GeoNetworking* [GeoNetworking2011] for study and as a showcase of how the notation is currently being used to address the two main use cases identified above. The fact that the notation currently used in the examples provided by CTI for study is related to TPLan, is never explicitly noted, which is indicative of the challenges noted above. In addition to the examples, CTI recommended consulting the TPLan standard [TPLan2009] for further details and ideas on what the expectations towards a concrete syntax for TDL would be.

Figure 1 shows the definition rules for such test purposes. While TPLan defines similar constructs, the document prescribes a tabular notion, where selected elements of TPLan are mapped to table cells and compartments. The TP Header defines meta-data, including hierarchical IDs (following a pre-defined naming conventions), a test objective, references to relevant technical specifications, and a boolean expression over PICS for test selection purposes. The TP behaviour defines relevant initial conditions, expected behaviour, and final conditions. Figure 2 shows an example test purpose taken from [CAM2011]. It includes metadata related to the objective being realised (checking that a "*CAM message includes DoorOpen information 30s after*" the door is closed), respective standard references ("*TS 102 637-2 [1], clauses 7.1 and 7.2*"), and PICS ("*PICS_PUBTRANSVEH*"). The metadata is followed by initial conditions describing the IUT state ("*initial state*" followed by the IUT sending "*a valid CAM message containing DoorOpen TaggedValue*") and the events which trigger responses expected from the IUT in order realise the test purpose objective (when "*the door is closed*", "*the IUT sends CAM messages containing DoorOpen TaggedValue during the 30s following the door closing event*").

While the studied examples comprise test purposes only, CTI sees TDL as an opportunity to unify the means for the specification of both test purposes and test descriptions. A single notation that is suited both for specifying concise, yet better structured, test purposes and for transitioning to more detailed and refined test descriptions, while relying on the same underlying meta-model and benefiting from other related technologies built around this meta-model is expected to meet ETSI's needs. 3GPP will be involved at a later stage, with a potentially extended or custom-tailored concrete syntax.

Part of the work of STF 476 involves the definition of a graphical concrete syntax for TDL which covers all constructs of the TDL meta-model. While this concrete syntax seeks to address the needs of the TDL users at large, there are several key aspects may impact its adoption at ETSI in the near term:

1. By design, TDL requires a certain level completeness, including all relevant technical details, such as comprehensive descriptions of test configurations and test behaviour. The graphical concrete syntax being developed will also reflect this level of necessary detail.
2. In order to use a graphical concrete syntax effectively and efficiently, corresponding sophisticated tooling is needed.
3. All the stakeholders that interact with test descriptions need to be familiar with and fluent in the syntax being used.

Combined, the required level of detail, the need for new or extra tooling and the availability of such tooling, as well as the need to learn and master a new syntax, may have a negative impact on the acceptance of TDL among users that already have established processes and practices in place, especially in the initial stages of deployment of TDL.

Based on the discussions with CTI, the studied examples, the key aspects impacting the adoption of a graphical syntax, as well as earlier discussions during STF 454, it was determined that efforts towards an ETSI-specific concrete syntax shall concentrate on investigating the extent to which concepts from TPLan, and in particular from the TPLan-based notation used in the examples, can be mapped to TDL. It shall be considered as a simplified syntax, initially targeting a sufficient subset of TDL features and possibly also restricted to a subset of features of TPLan that can and shall be mapped to TDL in order to cover the studied examples. Possible limitations, as well as necessary adaptations to both the concrete syntax and to the TDL meta-model shall be identified, agreed upon, and implemented where applicable. CTI considers this notation as a full syntax, rather that an output format only, enabling users to directly work with test purposes and test descriptions in this notation, instead of using a separate notation as input, which is seen as a potential challenge to user acceptance.

The definition and standardisation of the proposed notation as a simplified means to access and use TDL in the form of a textual concrete syntax will make TDL accessible to users at ETSI early on. Furthermore, a tailored solution based on a study of processes at ETSI can help foster early and fast adoption during the initial stages of deployment, given that a similar notation is already in use. The notation itself does not require fluency in an advanced programming language, rather it relies mostly on loosely structured patterns of natural language expressions. The usage of the notation can benefit greatly from early tooling availability, additional formalisation (to a reasonable extent), and consistency. Finally, it is expected that raising awareness among users will help in establishing TDL as a brand and early adoption will create an initial demand that shall foster a favourable environment for tool vendors to get into.

The risk of potential confusion and the proposed notation being accepted as “the TDL notation” shall be considered early on, e.g. by establishing distinct branding, e.g. different name such as “TDL Base”, “Simple TDL”, or “TDLan”. Clear communication with and awareness among users needs to be ensured.

5.1.1 TP definition conventions

The TPs are defined by the rules shown in table 2.

Table 2: TP definition rules

TP Header	
TP ID	The TP ID is a unique identifier. It shall be specified according to the TP naming conventions defined in clause 5.1.2.
Test objective	Short description of test purpose objective according to the requirements from the base standard.
Reference	The reference indicates the sub-clauses of the reference standard specifications in which the conformance requirement is expressed.
PICS Selection	Reference to the PICS statement involved for selection of the TP. Contains a Boolean expression.
TP Behaviour	
Initial conditions	The initial conditions defines in which initial state the IUT has to be to apply the actual TP. In the corresponding Test Case, when the execution of the initial condition does not succeed, it leads to the assignment of an Inconclusive verdict.
Expected behaviour (TP body)	Definition of the events, which are parts of the TP objective, and the IUT are expected to perform in order to conform to the base specification. In the corresponding Test Case, Pass or Fail verdicts can be assigned there.
Final conditions	Definition of the events that the IUT is expected to perform or shall not perform, according to the base standard and following the correct execution of the actions in the expected behaviour above. In the corresponding Test Case, the execution of the final conditions is evaluated for the assignment of the final verdict.

Figure 1: Test purpose definition rules from [CAM2011]

16 ETSI TS 102 868-2 V1.1.1 (2011-03)

TP Id	TP/CAM/INA/DOP/BV/02
Test objective	Checks that CAM message includes DoorOpen information 30s after closed
Reference	TS 102 637-2 [1], clauses 7.1 and 7.2
PICS Selection	PICS_PUBTRANSVEH

Initial conditions

```
with {
    the IUT being in the "initial state" and
    the IUT having sent a valid CAM message
        containing DoorOpen TaggedValue
}
```

Expected behaviour

```
ensure that {
    when {
        the door is closed
    }
    then {
        the IUT sends CAM messages
            containing DoorOpen TaggedValue during the 30s following the door closing event
    }
}
```

Figure 2: Test purpose example from [CAM2011]

Proposed Approach

The proposed approach seeks to address the realisation of a concrete syntax that suits ETSI's needs and targets the integration of TPLan and TPLan-based notations with TDL. It is based on identifying relevant concepts and relationships among them in the targeted notation. A domain-specific meta-model reflecting these concepts and relationships is defined and used in a set of standardised mappings to corresponding textual (in the form of a BNF, based on TPLan) and tabular elements, as well as in standardised mappings to and from TDL, with corresponding constraints and requirements towards target TDL models. Subsequent work may target mappings and means for documentation generation and TTCN-3 skeleton generation, if such mappings are not already defined for TDL or cannot be inherited from TDL once available.

The first step towards realising concrete syntax based on the findings above involves the identification of relevant concepts that describe the abstract structure of the test purposes. While these concepts can be partially derived from TPLan, additional structural patterns can be identified and represented explicitly. While imposing minor restrictions on how content needs to be structured, such structural patterns will provide better foundation for tool support and more comprehensive transformation into other artifacts.

Three basic levels of formalisation were identified for consideration with corresponding advantages and disadvantages:

1. **Highly structured and well-defined** - TDL at its core can be considered highly structured and well-defined, where there is a need for some upfront effort, e.g. for the definition of necessary concepts in advance (which can be reused). Structure is more rigid but also more consistent. Recognition and transformation, as well as corresponding tool support can be very sophisticated. Translation to and from other highly structured notations can be very comprehensive.
2. **Loosely structured** - Mark-up based languages can be considered loosely structured. More freedom in the way how statements are expressed, recognition and transformation is restricted only to essential parts, with little need for upfront definitions, at the cost of (manual) mark-up. Potential ambiguities, undefined relationships between concepts, and limitations in the extent of translation to and from other (and in particular highly-structured) notations, are some of the major drawbacks of this approach.
3. **Mixed** - characterised by a loose structure that provides some level of freedom, as long as certain conventions and well-defined patterns that define explicit relationships among concepts are maintained. Based on these conventions and patterns, key structural properties and relationships among concepts can be preserved, leaving the extent to which these are translated into a different notation to the specific mapping between the notations. Semantically weak qualifiers can be added to certain concepts in order to refine them or as a semantically void glue between expressions. The amount of necessary definitions in advance is limited to bare essentials that can be reused across large number of test purposes within a defined domain.

Based on the examples under study, discussions with CTI, as well as preliminary experiments with the different options above, it was determined that the **mixed** approach will be best suited for the needs of ETSI, by providing a good balance between level of formalisation and convenience of use, while maintaining the natural language look and feel of the notation currently in use, with only minor restrictions and adjustments necessary.

The concepts for the overall framework of the proposed notation are fairly straightforward. One particular source of complexity, ambiguity, and lack of consistency at present, is the notation for actions. While the present notation is loosely based on TPLan which prescribes certain principles for expressing actions (divided into stimuli and responses, as well as user defined events), these are not always reflected in the studied examples. One way to capture a large part of the examples with minor or no modifications, in addition to preserving relationships between concepts is by using the roughly the following pattern (expressed here in BNF terms):

`Action`	`::=`	`SubjectReference PredicateReference OriginReference?`
`SubjectReference`	`::=`	`InlineArticleQualifier SubjectID`
`PredicateReference`	`::=`	`InlineQualifier* PredicateID Argument?`
`OriginReference`	`::=`	`‘from’ SubjectReference`
`Argument`	`::=`	`InlineArticleQualifier? InlineQualifier* NAME Content?`
`InlineQualifier`	`::=`	`NAME`
`InlineArticleQualifier`	`::=`	`‘a’ ǀ ‘an’ ǀ ‘the’`
`Content`	`::=`	`...`

where `Term*` denotes zero or more occurrences of the term and `Term?` denotes an optional term. The semantically weak `InlineQualifier` does not need to be defined in advance and may be used as glue or to refine a related terms. Similarly, the `NAME` in an `Argument` specification need not be defined in advance, although doing so may be beneficial for the consistent use of e.g. messages. The contents of an argument can be refined to reflect the needs for the partial specification of relevant data structures and data contents. Additionally, constraints may be added to actions to enable the explicit and structured specification of timing and other constraints.

It is worth noting that while further formalisation may negatively impact the speed with which test purposes are initially created, by causing authors to think about how to express a certain action within the scope and constraints of the provided language framework, rather than using natural language plain and (not always so) simple, this initial hurdle may contribute to more consistent, less ambiguous, and altogether better formed test purposes in the long term, precisely due to authors taking a bit more time to think and phrase their expressions in loosely structured, yet formalised terms, besides (and in part due to) the benefits of tool support.

Once the level of formalisation is established and the relevant concepts and their relationships have been identified and adapted to fit the level of formalisation, these need to be mapped to TDL. There are fundamentally two ways to approach this: **direct mapping** of concrete syntax elements to TDL, by encoding the relationships of the concrete concept representations to the relevant meta-model elements of TDL in the concrete syntax specification, or **indirect mapping** of he concepts to a domain-specific meta-model, and then mapping the domain-specific concepts to the elements of TDL at the meta-model level, by utilising

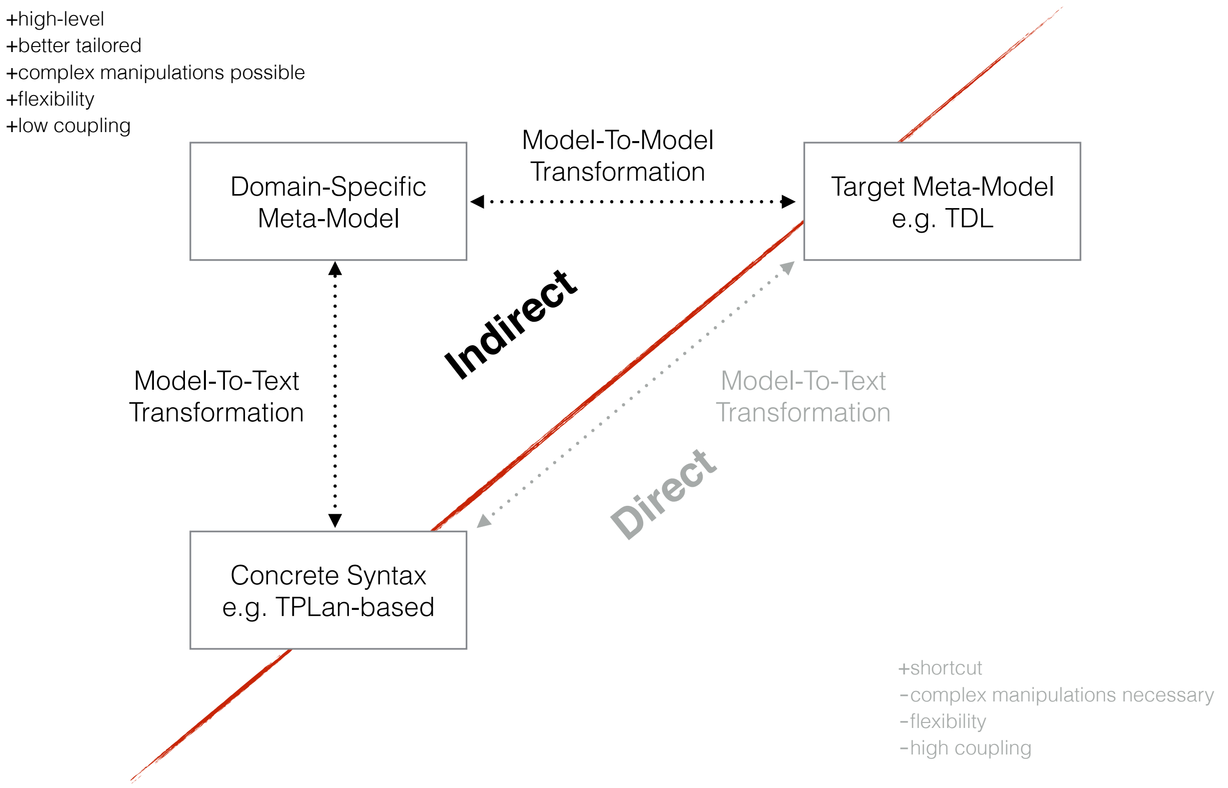

Figure 3: Differences between direct and indirect mapping approaches

common Model-to-Model transformation approaches. While the direct approach is perfectly suited for concrete syntax notations that are close to the target meta-model, given the observed differences between the main use cases for the studied examples and TDL, and the corresponding differences between the relevant concepts and their level of detail (lots of implicit information, lack of test configurations, lack of data definitions, partial interaction specifications), the indirect approach is better suited and also commonly adopted in practice. It allows more sophisticated and flexible bi-directional transformations at a higher level of abstraction, as opposed to "stubborn" attempts to bend both the target meta-model and the concrete syntax to make them fit. Figure 3 summarises the differences between the direct and indirect approaches. Note that whether the direct or indirect approach is chosen has generally little to no direct impact on how the user is exposed to the language, they do not necessarily need to be aware of the indirection.

In case the indirect approach is chosen, the relevant concepts shall be captured in a domain-specific meta-model. Such a meta-model was sketched out as part of the analysis task in order to explicitly define and explore the relevant concepts and how they relate to TDL, and how well the different approaches are suited for the task. A partial representation of some of the concepts is visualised on Figure 9 in the Annex. Based on this domain-specific relevant mappings shall be defined next - to the concrete textual and tabular syntax, and to the TDL meta-model. Examples illustrating the application of the notation and the mappings as well as a migration strategy for existing documents and processes can be outlined as informative parts where applicable. The annex to this document includes several examples illustrating the textual part of the proposed concrete syntax, which is then incorporated into a tabular presentation by textual blocks to different table compartments. The examples are derived from the documents provided for study.

Summary and Recommendations

The definition of a standardised concrete syntax tailored for the needs of ETSI, and later 3GPP, will contribute to direct overall improvements in the test specification process with reduced effort for the development and maintenance of standardised test descriptions. In addition, the quality and consistency of the test purposes and test descriptions can be checked and improved based on a set of guidelines defined on the meta-model level. The integration with TDL will yield further benefits from ongoing and future developments on TDL, such as seamless translation between different views on a model (graphical, textual, tabular), automated generation of consistent documentation, and generation of executable test skeletons (e.g. in TTCN-3). The proposed standardised mapping will also serve as a showcase and reference for mapping other languages and models to TDL.

Given the importance of such a notation and the corresponding use cases to CTI and ETSI, further activities in to address these current and concrete needs for the adoption of TDL at ETSI are recommended. The choice of a mixed level of formalisation and an indirect mapping to TDL have been identified as the preferable solutions to the key technical challenges for the proposed notation. Apart from the benefits for CTI and ETSI at large, this is seen as an opportunity to gain early recognition and awareness for TDL by establishing it as a key technology within ETSI for the foreseeable future. Recommended next steps are outlined on Page 9, including tasks and milestones for subsequent work towards a standardised concrete syntax for TDL tailored for the needs of ETSI.

Tasks and Milestones

The work organisation for the proposed extended scope of STF 476 involves the following tasks:

Task	Description	Start	End	Days
1	Identification of relevant concepts and implementation into a domain-specific meta-model	07/2014	10/2014	7
2	Mapping of domain-specific meta-model elements to a concrete syntax notation	08/2014	10/2014	8
3	Definition of bi-directional mapping between the domain-specific meta-model the TDL meta-model	09/2014	12/2014	10
	Total resources:			**25**

For the proposed extended scope of STF 476, two milestones are defined as follows:

- Early Draft in September, 2014 (aligned with stable drafts for STF 476)
  - Stable domain-specific meta-model
  - Stable concrete syntax notation specification
  - Early mapping definitions
- Stable Draft in December, 2014 (aligned with final drafts for STF 476, submitted for approval at MTS #64)
  - Final domain-specific meta-model (normative)
  - Final concrete syntax notation specification (normative)
  - Final mapping definitions (normative)

Budget and Resources

The estimated resources cover the costs for the contracted experts. Travels will be aligned with travels related to the activities of STF 476, thus no additional budged for travel is requested.

Description	Days	Rate	Cost
Contracted experts (remunerated)	20	€ 600	€ 12.000
Contracted experts (voluntary, 20% from total)	5	€ 0	€ 0
CTI staff (voluntary)	10	€ 0	€ 0
Total manpower cost	**35**		**€ 12.000**
Total cost			**€ 12.000**

References

- **[TPLan2009]** ETSI ES 202 553 V1.2.1 (2009-06), *Methods for Testing and Specification (MTS); TPLan: A notation for expressing Test Purposes*, ETSI 2009
- **[CAM2011]** ETSI TS 102 868-2 V1.1.1 (2011-03), *Intelligent Transport Systems (ITS); Testing; Conformance test specification for Co-operative Awareness Messages (CAM); Part 2: Test Suite Structure and Test Purposes (TSS&TP)*, ETSI, 2011
- **[GeoNetworking2011]** ETSI TS 102 859-2 V1.1.1 (2011-03), *Intelligent Transport Systems (ITS); Testing; Conformance test specifications for Transmission of IP packets over GeoNetworking; Part 2: Test Suite Structure and Test Purposes (TSS&TP)*, ETSI, 2011
- **[TDL2014]** ETSI ES 203 119 V1.1.1 (2014-04), *The Test Description Language (TDL); Specification of the Abstract Syntax and Associated Semantics*, ETSI, 2014

Annex

This annex includes several examples illustrating the application of the proposed syntax to the studied examples, which were constructed as part of the analysis for this report. A translation of the example from [CAM2011] shown on Figure 2 into a textual representation based on a meta-model sketch also constructed as part of the analysis for this task is shown on Figure 4. Apart from minor adjustments and a minimal set of definitions in advance covering semantically significant concepts which can be reused across related test purposes, the example has been mostly copied verbatim from the source (adding quotation marks where applicable, and a column after predicates). While not visible in the textual representation directly, different terms are identified and related to each other at the domain-specific model level, so that `a valid CAM message` represents the `Argument` specification with `valid` and `CAM` qualifying the `message`. A conceptualised tabular presentation based on this textual notation, closely resembling the original test purpose, is shown on Figure 5. A possible colour highlights for predicates as semantically relevant terms is also shown on the figure. Based on this model, the transformation of this example into a TDL model is illustrated on Figure 6, represented here by means of the example syntax in Annex B of [TDL2014]. The transformation is achieved by a prototypical implementation of the proposed approach, created for exploring the different mapping options. The original test purpose fragments are added for reference and comparison as `SOURCE` annotations to relevant concepts in the TDL model corresponding to the essential concepts in the TPLan-based notation (also identified by the `BLOCK` annotation). It is not complete as indicated by the omission of certain types of events such as the transformation of actions describing a `state`, or comprehensive based on the oversimplified default establishment of connections and component instances. It also showcases potential ambiguities, as indicated by `the door is: closed` action, which likely refers to a signal originating from the tester, or targeting the IUT (assuming that the door is a tester component instance), but may be erroneously translated to a signal targeting the tester by an oversimplified transformation assumptions. This can be resolved by either constraining the syntax further or adopting a standardised mapping. The translation of a similar example from [GeoNetworking2011] is showcased on Figure 7, where relationships among data structures are expressed in a complex manner which varies among different test purposes. An artificially constructed example combining different complex data specifications based on examples [GeoNetworking2011] is illustrated on Figure 8. Finally, a partial visualisation of the domain-specific meta-model implementing the core concepts identified from the examples and from TPLan, which was used for the analysis, is shown on Figure 9.

```
Package CAM {

    //reusable definitions that can be shared among related test purposes
    pics :
        - PICS_PUBTRANSVEH ( "A.2/3 [2]" )
    ;

    subjects :
        - IUT
        - door
    ;

    //may be used to describe predicates in order to specify target mappings
    predicate types :
        - state
        - interaction
    ;

    predicates :
        - in (state)
        - is (interaction)
        - sent (interaction)
        - sends (interaction)
    ;

    //actual test purposes
    Group "Message Generation" {

        TestPurpose "TP/CAM/INA/DOP/BV/02" {
            TP Id "TP/CAM/INA/DOP/BV/02"
            Test objective "Checks that CAM message includes DoorOpen information 30s after closed"
            Reference "TS 102 637-2 [1], clauses 7.1 and 7.2"
            PICS Selection PICS_PUBTRANSVEH
            Initial conditions
            with {
                the IUT being in: the initial state and
                the IUT having sent: a valid CAM message
                    containing DoorOpen TaggedValue
            }
            Expected behaviour
            ensure that {
                when {
                    the door is: closed
                }
                then {
                    the IUT sends: a CAM message
                        containing DoorOpen TaggedValue during the 30.5s following the door is closed event
                }
            }
        }
    }
}
```

Figure 4: Textual test purpose syntax implementing a TP based on Figure 2

TP Id	TP/CAM/INA/DOP/BV/02
Test objective	Checks that CAM message includes DoorOpen information 30s after closed
Reference	TS 102 637-2 [1], clauses 7.1 and 7.2
PICS Selection	PICS_PUBTRANSVEH
Initial conditions	

```
with {
    the IUT being in: the initial state and
    the IUT having sent: a valid CAM message
        containing DoorOpen TaggedValue
}
```

Expected behaviour

```
ensure that {
    when {
        the door is: closed
    }
    then {
        the IUT sends: a CAM message
            containing DoorOpen TaggedValue during the 30.5s following the door is closed event
    }
}
```

Figure 5: “Textual in Tabular” test purpose syntax implementing a TP based on Figure 2

```
TDLan Specification GlobalPackage {
    Annotation SOURCE ;
    Annotation BLOCK ;
    Data Set defaultSet {
        instance closed ;
    }
    Data Set state {
        instance initial ;
    }
    Data Set message {
        instance CAM ;
    }
    Gate Type defaultGate accepts defaultSet, state, message ;
    Component Type defaultComponent {
        gate types : defaultGate ;
    }
    Test Configuration defaultConfiguration {
        component c_TESTER as Tester of type defaultComponent having {
            gate TESTER of type defaultGate ;
        }
        component c_IUT as SUT of type defaultComponent having {
            gate IUT of type defaultGate ;
        }
        component c_door as SUT of type defaultComponent having {
            gate door of type defaultGate ;
        }
        connect IUT to TESTER ;
        connect door to TESTER ;
    }
    Package CAM {
        Package Message_Generation {
            Test Description TP_CAM_INA_DOP_BV_02 {
                use configuration : defaultConfiguration ;
                {
                    {
                        IUT sends instance CAM to TESTER ;
                    } with {
                        BLOCK "InitialConditions" ;
                        SOURCE
                        "
                            Initial conditions
                            with {
                                the IUT being in: the initial state and
                                the IUT having sent: a valid CAM message
                                    containing DoorOpen TaggedValue
                            }
                        " ;
                    }
                    {
                        door sends instance closed to TESTER ;
                        IUT sends instance CAM to TESTER ;
                    } with {
                        BLOCK "ExpectedBehaviour" ;
                        SOURCE
                        "
                            Expected behaviour
                            ensure that {
                                when {
                                    the door is: closed
                                }
                                then {
                                    the IUT sends: a CAM message
                                        containing DoorOpen TaggedValue during the 30.5s following the door is closed event
                                }
                            }
                        " ;
                    }
                }
            }
        }
```

Figure 6: TDLan representation of TP from Figure 2, translated from Figure 4 (incomplete)

```
Package sample_its {
    pics :
        - PICS_GVL
        - PICS_Ethernet
        - PICS_MIB_ManualAssigned
    ;
    subjects :
        - IUT
        - "IUT's Upper Layer"
    ;
    predicates :
        - having
        - use
        - receives
        - sends
    ;

    Group G2 {
        TestPurpose TP2 {
            TP Id TP/IPv6GEO/MG/GVL/BV/01
            Test objective "Checks that an IPv6 multicast message is carried out over a GeoBroadcast message into the correct
                            geographical area, with a GVL manually configured"
            Reference "TS 102 636-6-1 [1], clauses 5.2.2, 5.3.3.1, 8.2.1 and 9.1.2"
            PICS Selection PICS_GVL and PICS_Ethernet and PICS_MIB_ManualAssigned
            Initial conditions
              with {
                the IUT having: a manually configured GVL as GVL1 and
                the "IUT's Upper Layer" being manually configured to use the virtual interface associated to GVL1
              }
            Expected behaviour
            ensure that {
              when {
                the IUT receives: an IPV6 packet
                    containing "destination address" field
                        indicating value "a multicast IPv6 address"
                from the "IUT's Upper Layer"
              }
              then {
                the IUT sends: a valid GeoNetworking GeoBroadcast message containing the "geographical Destination area"
                  corresponding to GVL1
                    containing NH field indicating value '2'
                    containing HT field indicating value '4'
                    containing LT field indicating value '0'
                    containing TC field
                        //indicating value derived from NH
                            containing Priority field carrying the packet received from Upper Layer as payload

              }
            }
        }
    }
}
```

Figure 7: Textual test purpose syntax implementing a TP from [GeoNetworking2011]

```
Package sample {
    pics :
        - PICS1
        - PICS2
        - PICS3
    ;
    subjects :
        - IUT
        - UpperLayer
    ;
    predicate types :
        - action
        - configuration
        - state
    ;
    predicates :
        - receives (action)
        - received (action)
        - sends (action)
        - responds (action)
        - having (state)
        - is (state)
    ;

    Group G1 {
        TestPurpose TP1 {
            TP Id ID
            Test objective "Some objective"
            Reference "Some standard clause"
            PICS Selection PICS1 and PICS2 and PICS3
            Initial conditions
            with {
                the IUT having received a MESSAGE containing the ADDRESS
            }
            Expected behaviour
            ensure that {
                when {
                    the IUT is: online and
                    the IUT has received: a valid IPTV message
                        containing HT field
                            indicating value MyValue
                        containing CT field
                            indicating value "Your Value"
                        containing FT field
                            indicating value 1 and
                    the UpperLayer is: properly configured and
                    the IUT receives: a valid instruction
                        containing PT field
                            containing XT field
                                indicating value corresponding to HT
                }
                then {
                    the IUT sends: the IPTV message
                        containing the packet
                            indicating value corresponding to ADDRESS
                }
            }
        }
    }
}
```

Figure 8: Artificial example showcasing complex and interrelating data specifications

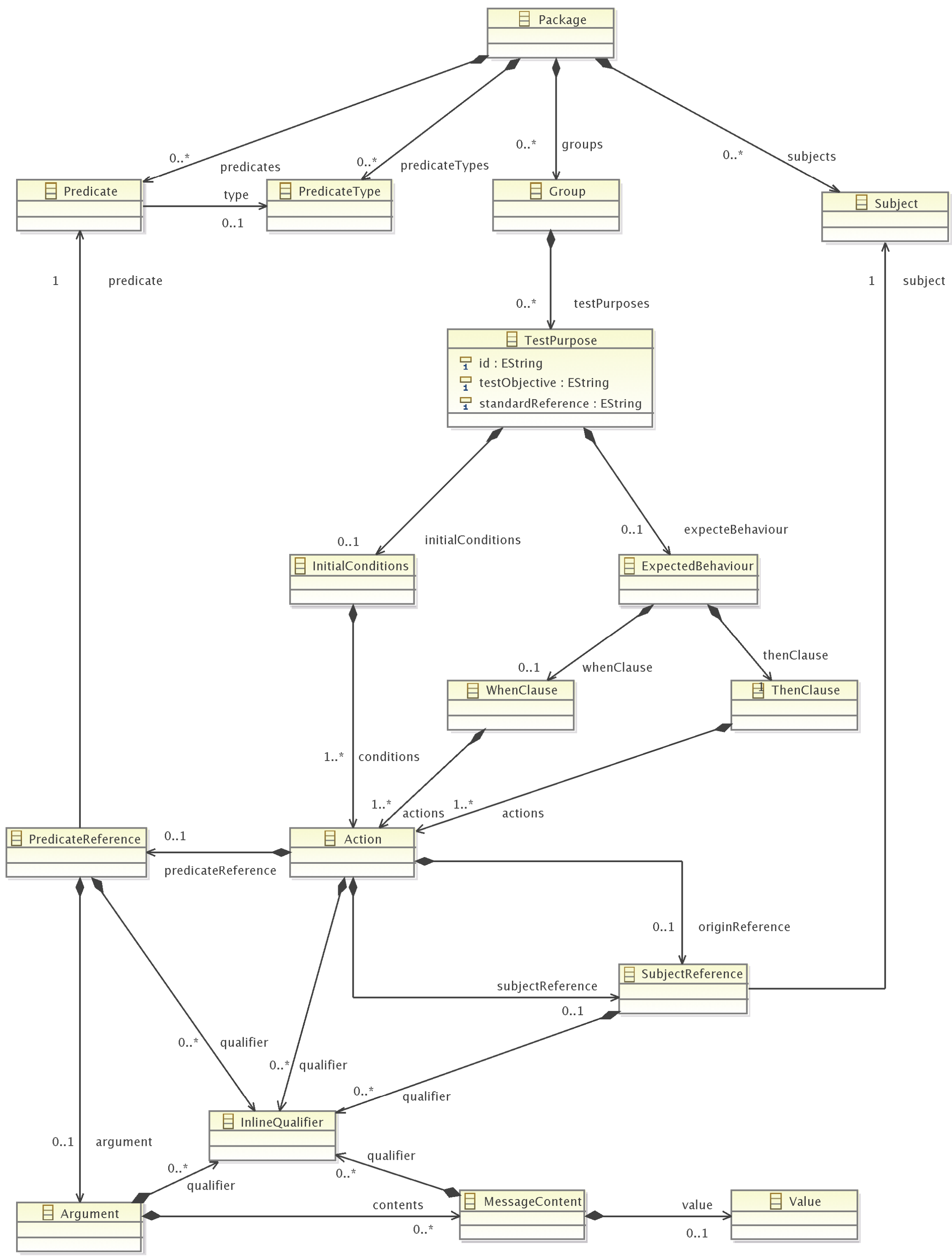

Figure 9: Domain-specific meta-model (partial sketch)