4th UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

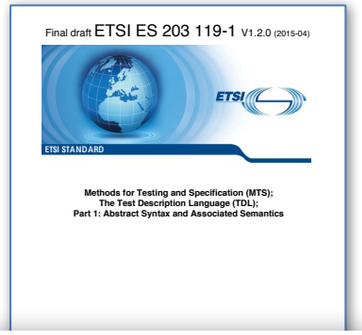
TESTING AND MODELING WITH TDL

Philip Makedonski, Gusztav Adamis, Martti Käärik, Finn Kristoffersen, Andreas Ulrich, Xavier Zeitoun

Overview

What is TDL?

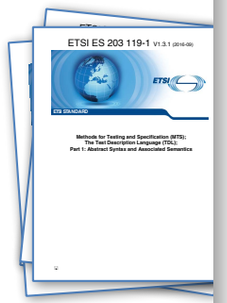
- Test Description Language
  - Design, documentation, and representation of formal test descriptions
  - Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)



3

Why TDL?

- For users
  - separate test specification from test implementation
  - amenable to tool-supported verification
  - adjustable to stakeholders
  - focus on what to test vs how



ange format
different domains
h existing tools
d value through interoperability

4th UCAAT

Where does TDL fit in?

Keyword-Driven Testing
MBT
Representation
Generation
Rail
Standards
Visualisation
Documentation
ITS
Interoperability
Conformance

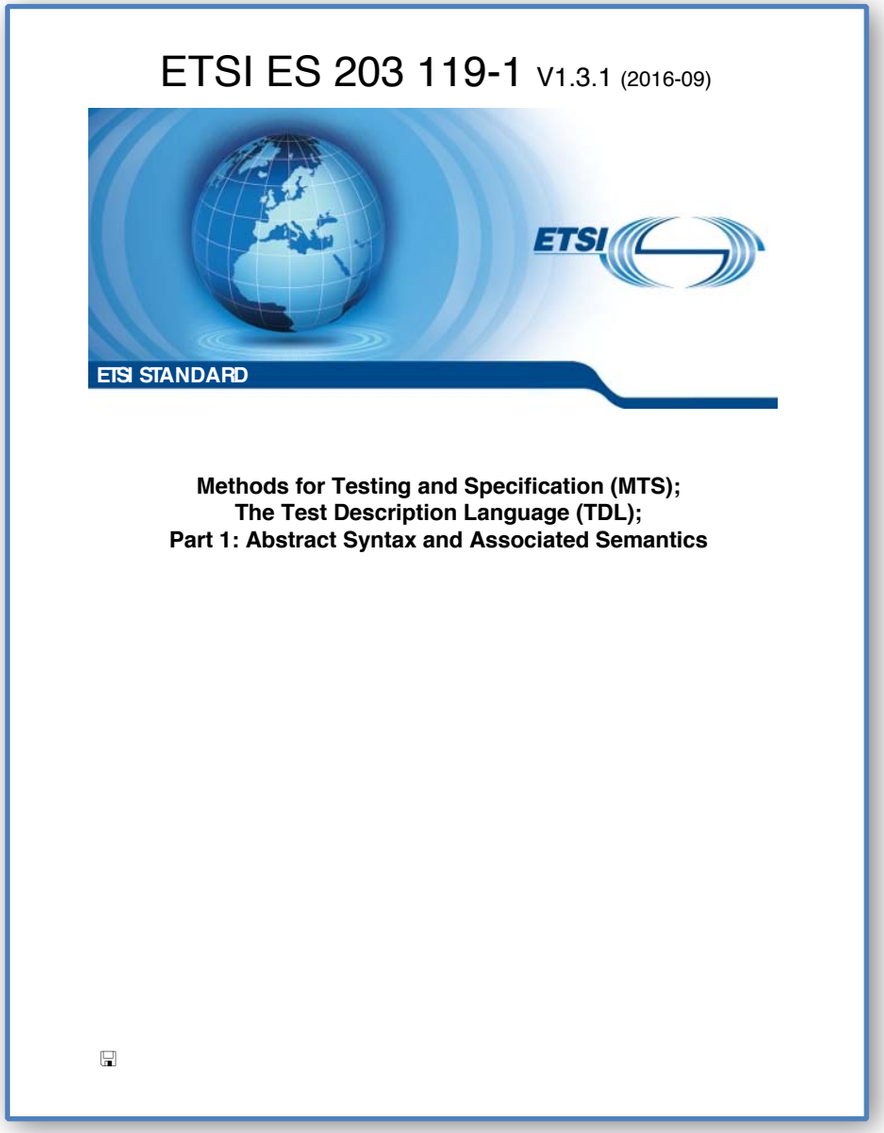
46

3rd UCAAT

4th UCAAT

What is TDL?

- Test Description Language
  - Design, documentation, and representation of formalised test descriptions
  - Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)

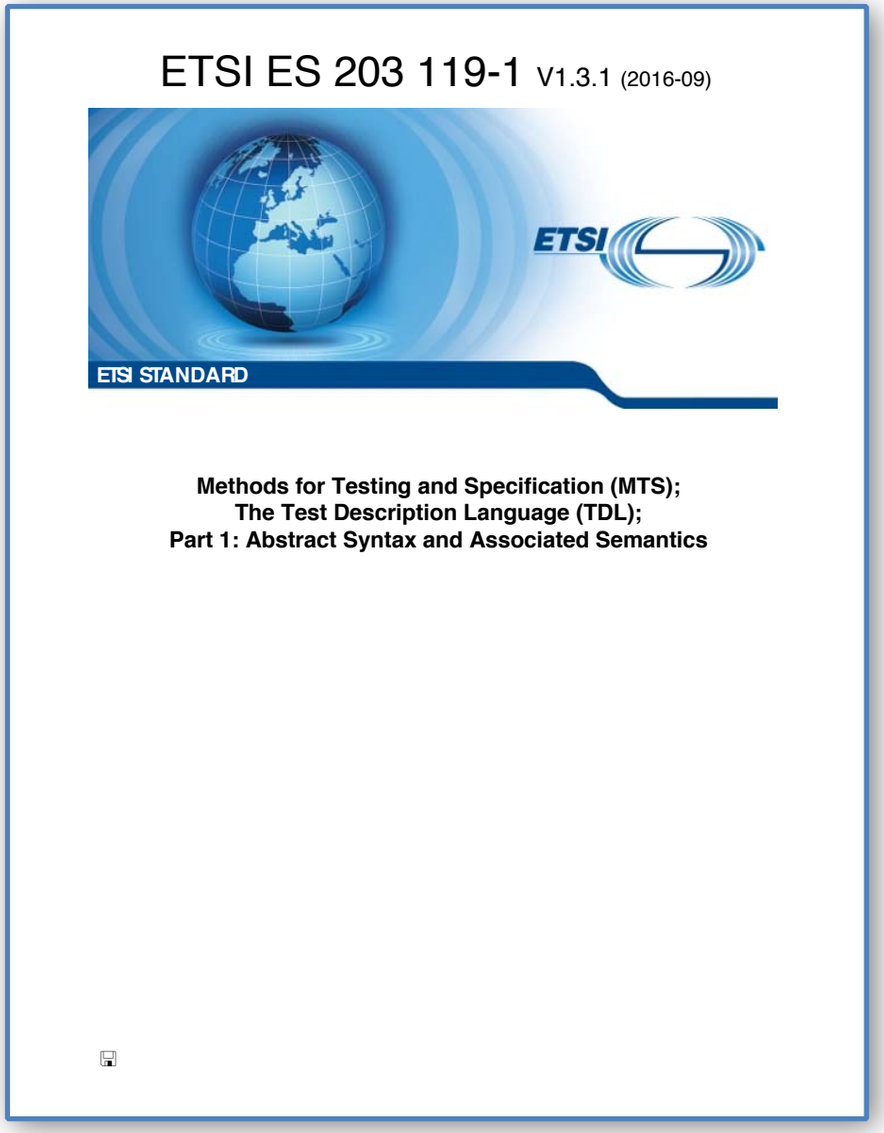

ETSI ES 203 119-1 V1.3.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

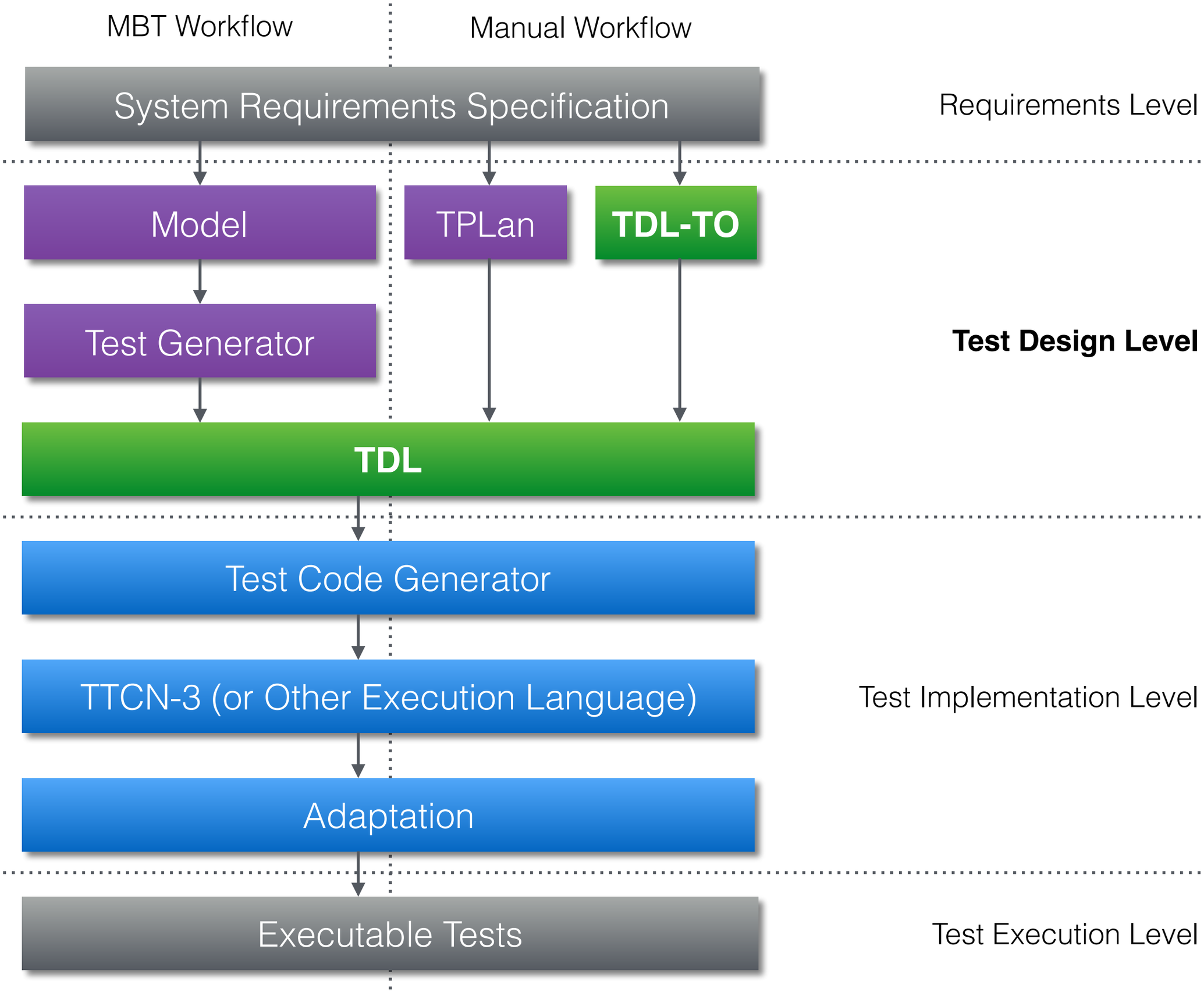
What is TDL?

- Design, documentation, representation?
  - ease development and review
  - improve productivity and quality
  - both industry and standardisation
  - reduce implementation details


ETSI ES 203 119-1 V1.3.1 (2016-09)

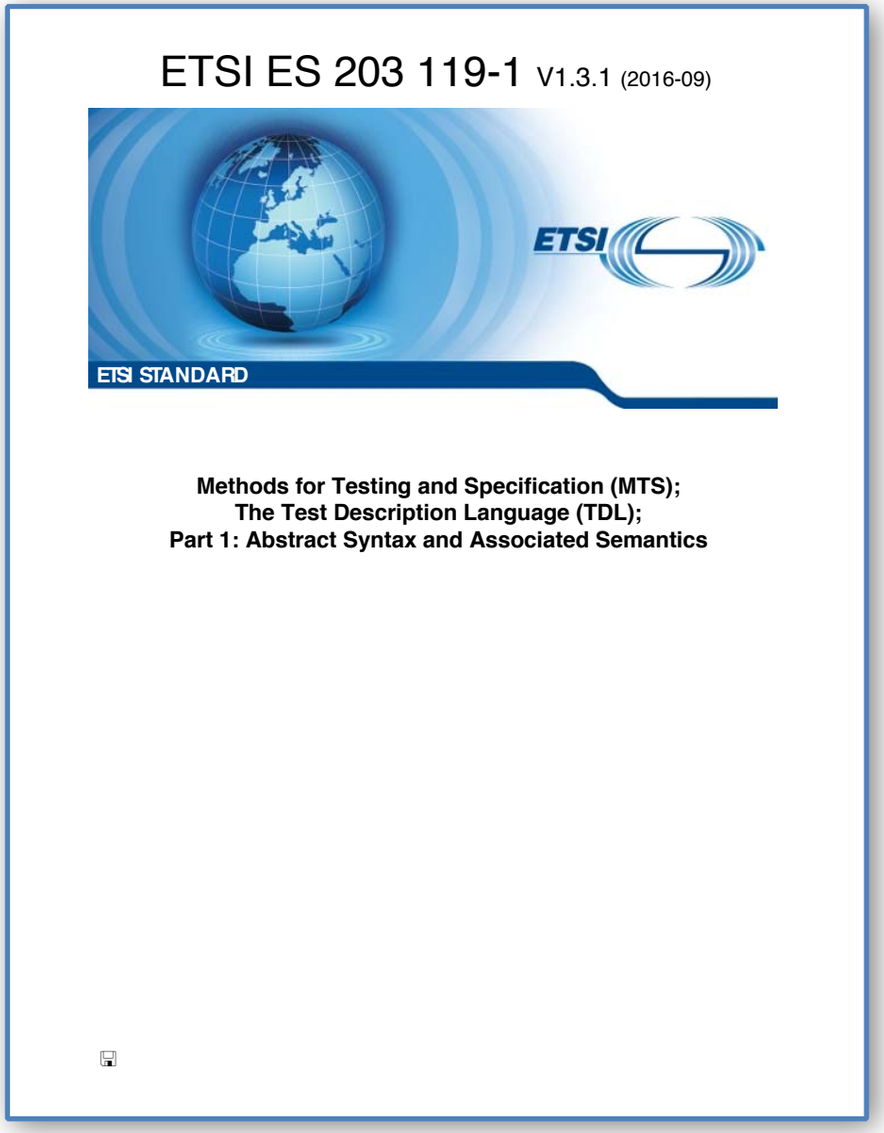
ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics


MBT Workflow
Manual Workflow
System Requirements Specification
Requirements Level
Model
TPLan
TDL-TO
Test Generator
Test Design Level
TDL
Test Code Generator
TTCN-3 (or Other Execution Language)
Test Implementation Level
Adaptation
Executable Tests
Test Execution Level

What is TDL?

- Scenario-based?
  - describe interactions with a system
  - attach test objectives to scenarios
  - derive and automate tests
- Reactive, distributed, real-time
  - common black-box testing concepts
  - domain adaptation
  - agile development

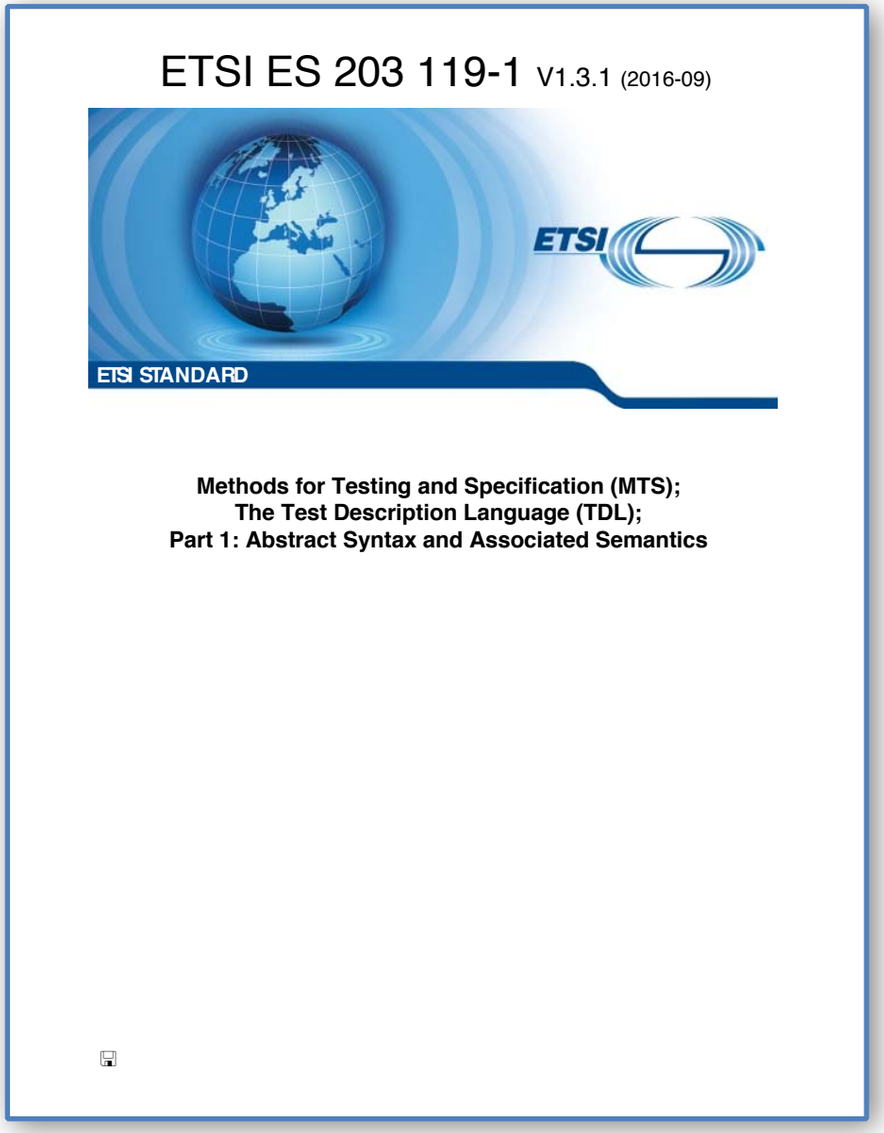

ETSI ES 203 119-1 V1.3.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

What is TDL?

- Standardised?
  - canonical reference
  - stable documentation
  - clear semantics
  - interoperability and independence
  - updated with user needs
  - maintenance commitment

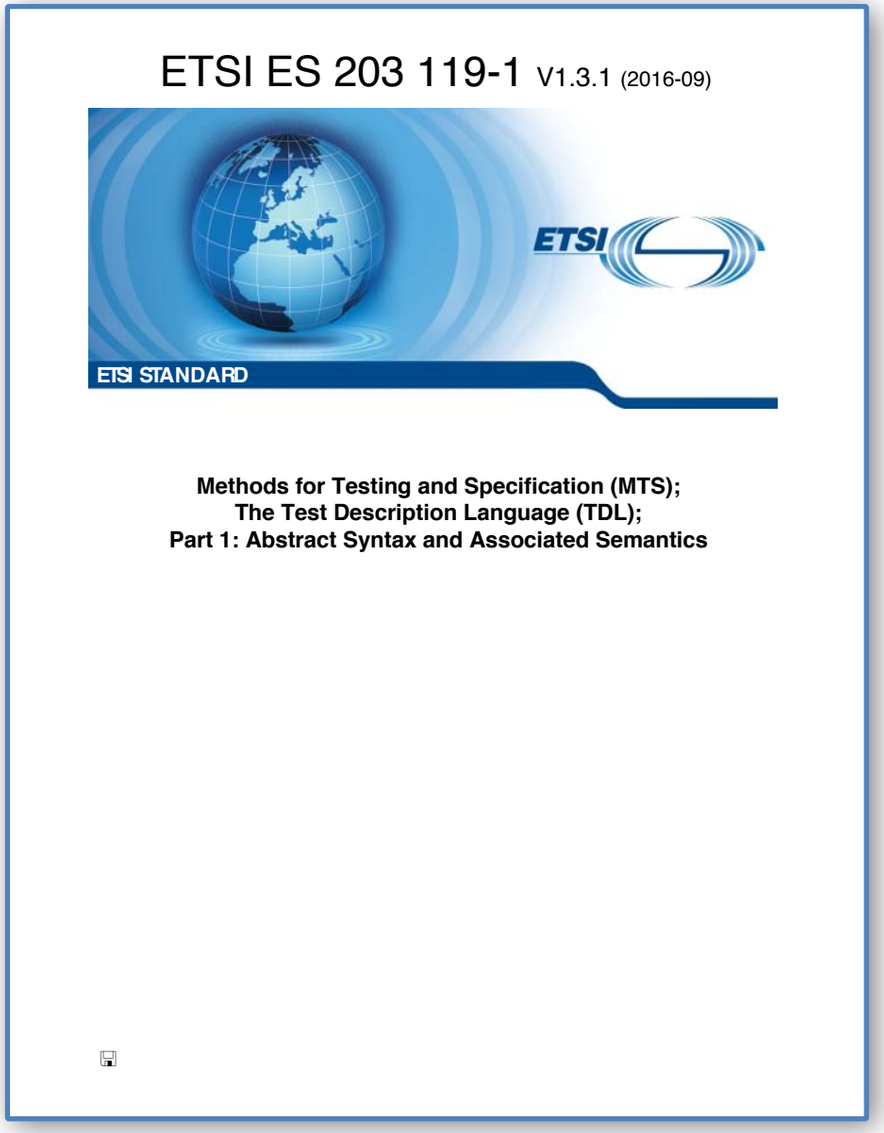

ETSI ES 203 119-1 V1.3.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

What is TDL?

- Contributions from:
  - Siemens AG, Ericsson Hungary
  - Fraunhofer FOKUS, ETSI CTI
  - CEA, University of Göttingen
  - OU Elvior, Cinderella ApS
- Guidance:
  - Steering Group, TC MTS


ETSI ES 203 119-1 V1.3.1 (2016-09)

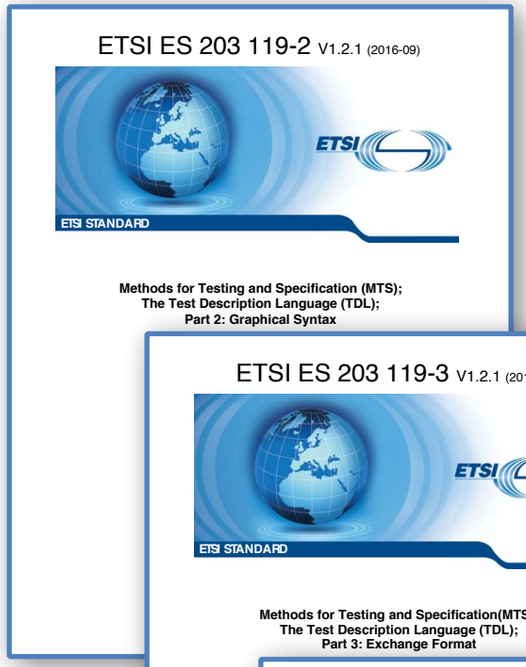
ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

What is TDL?

Part 1: MM Meta-Model and Semantics

+

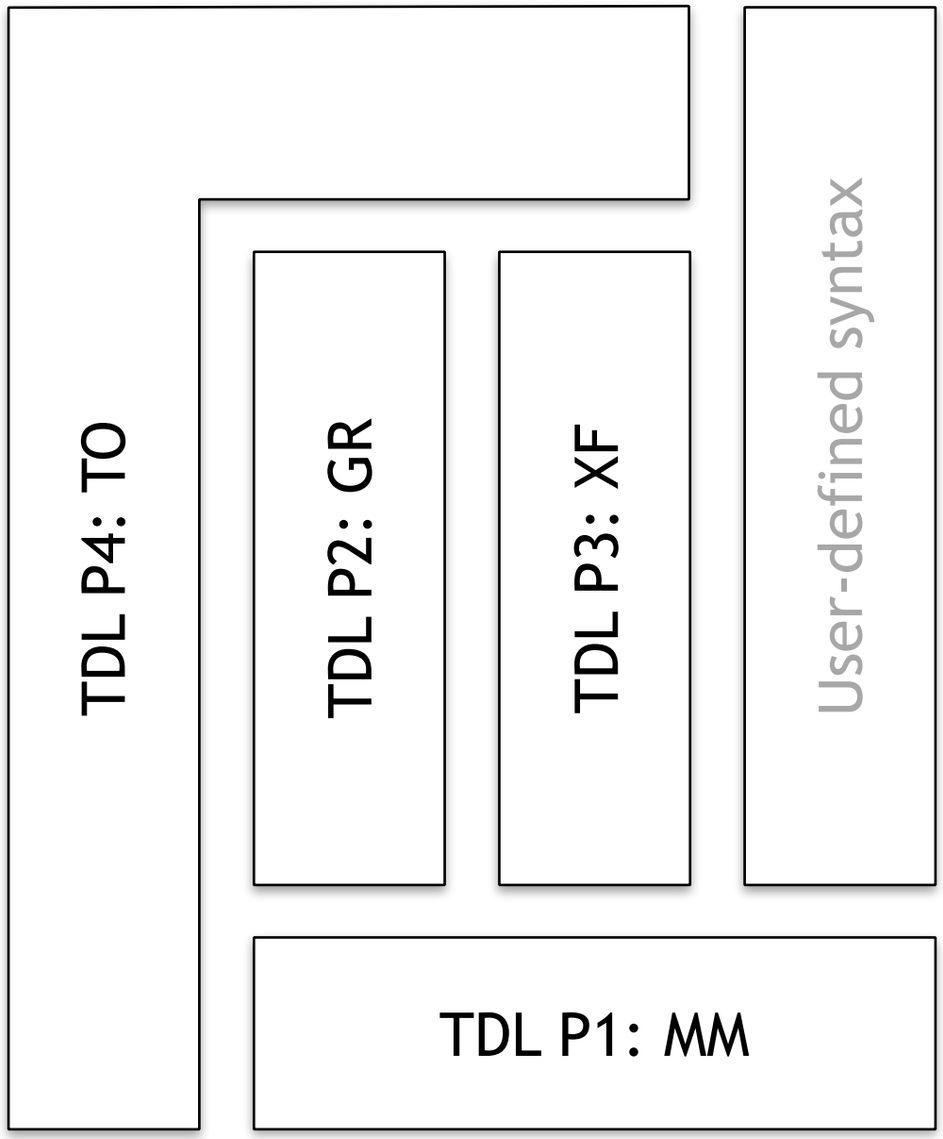


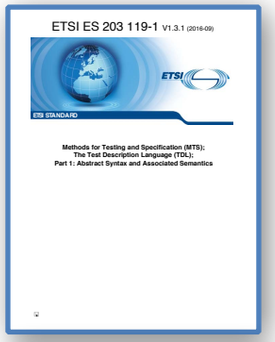
Part 2: GR Graphical Syntax

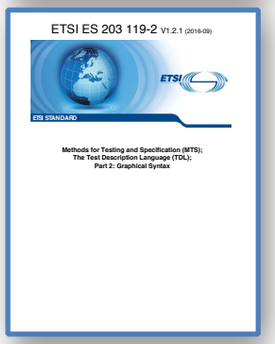
Part 3: XF Exchange Format

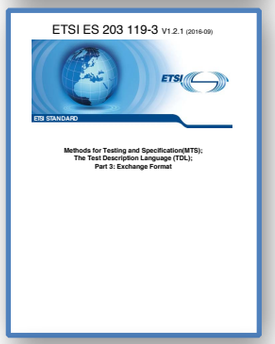
Part 4: TO Structured Test Objective Specification

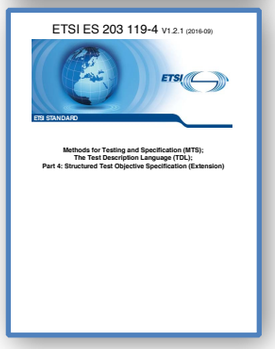
What is TDL?

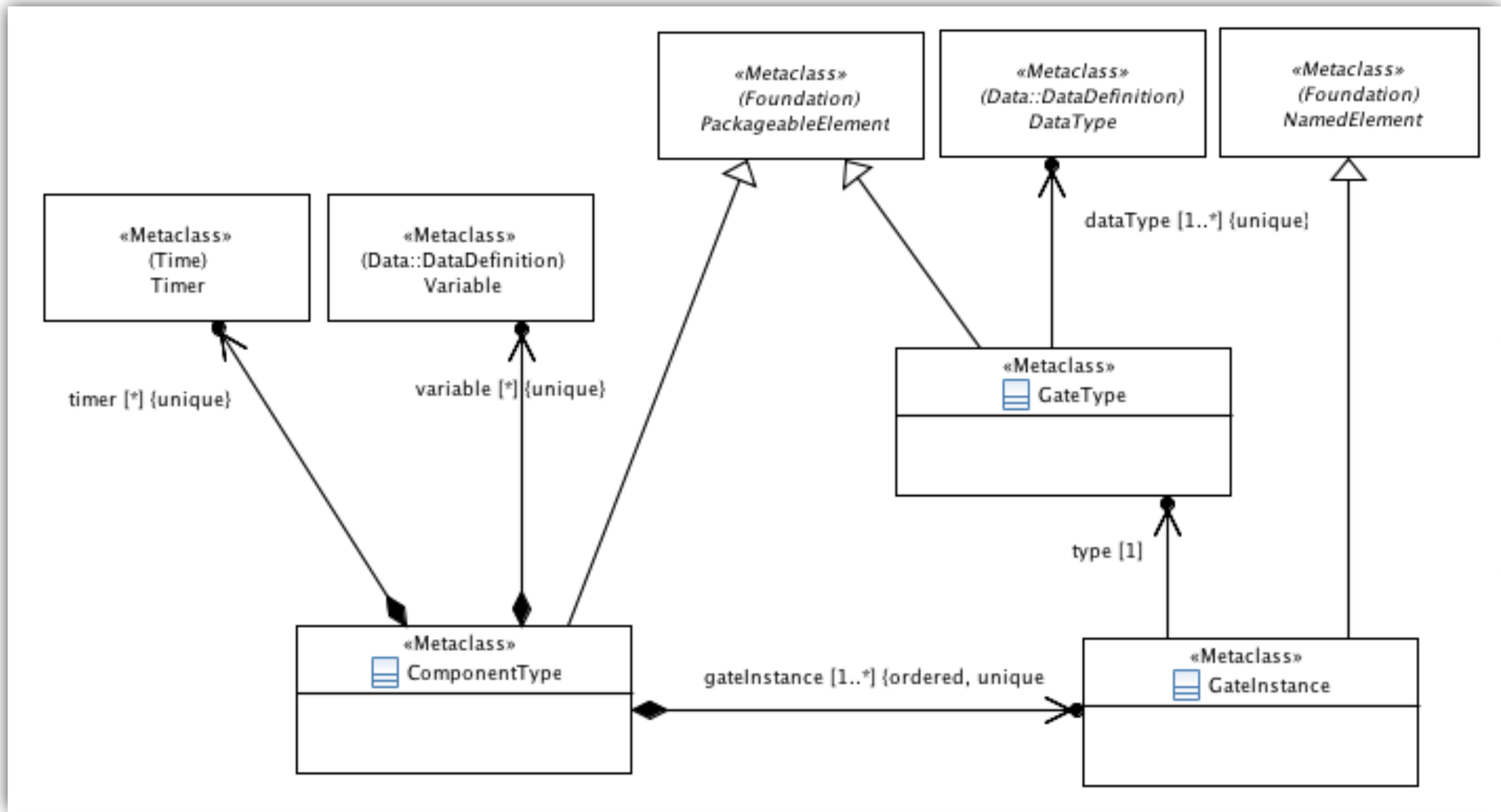

TDL P4: TO
TDL P2: GR
TDL P3: XF
User-defined syntax
TDL P1: MM

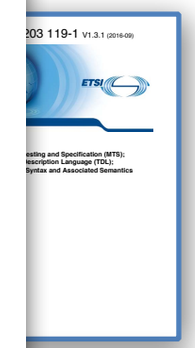

ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics


ETSI ES 203 119-2 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax

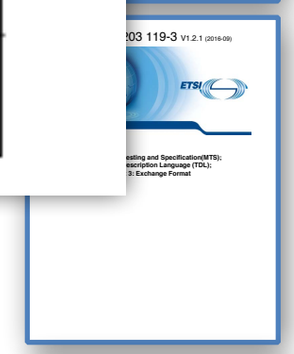

ETSI ES 203 119-3 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification(MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

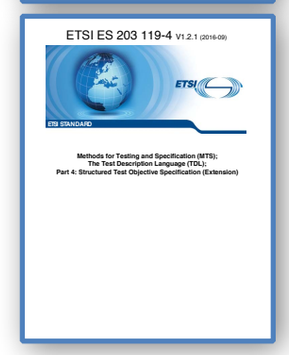

ETSI ES 203 119-4 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)


«Metaclass»
(Foundation)
PackageableElement
«Metaclass»
(Data::DataDefinition)
DataType
«Metaclass»
(Foundation)
NamedElement
«Metaclass»
(Time)
Timer
«Metaclass»
(Data::DataDefinition)
Variable
dataType [1..*] {unique}
«Metaclass»
GateType
timer [*] {unique}
variable [*] {unique}
type [1]
«Metaclass»
ComponentType
gateInstance [1..*] {ordered, unique
«Metaclass»
GateInstance

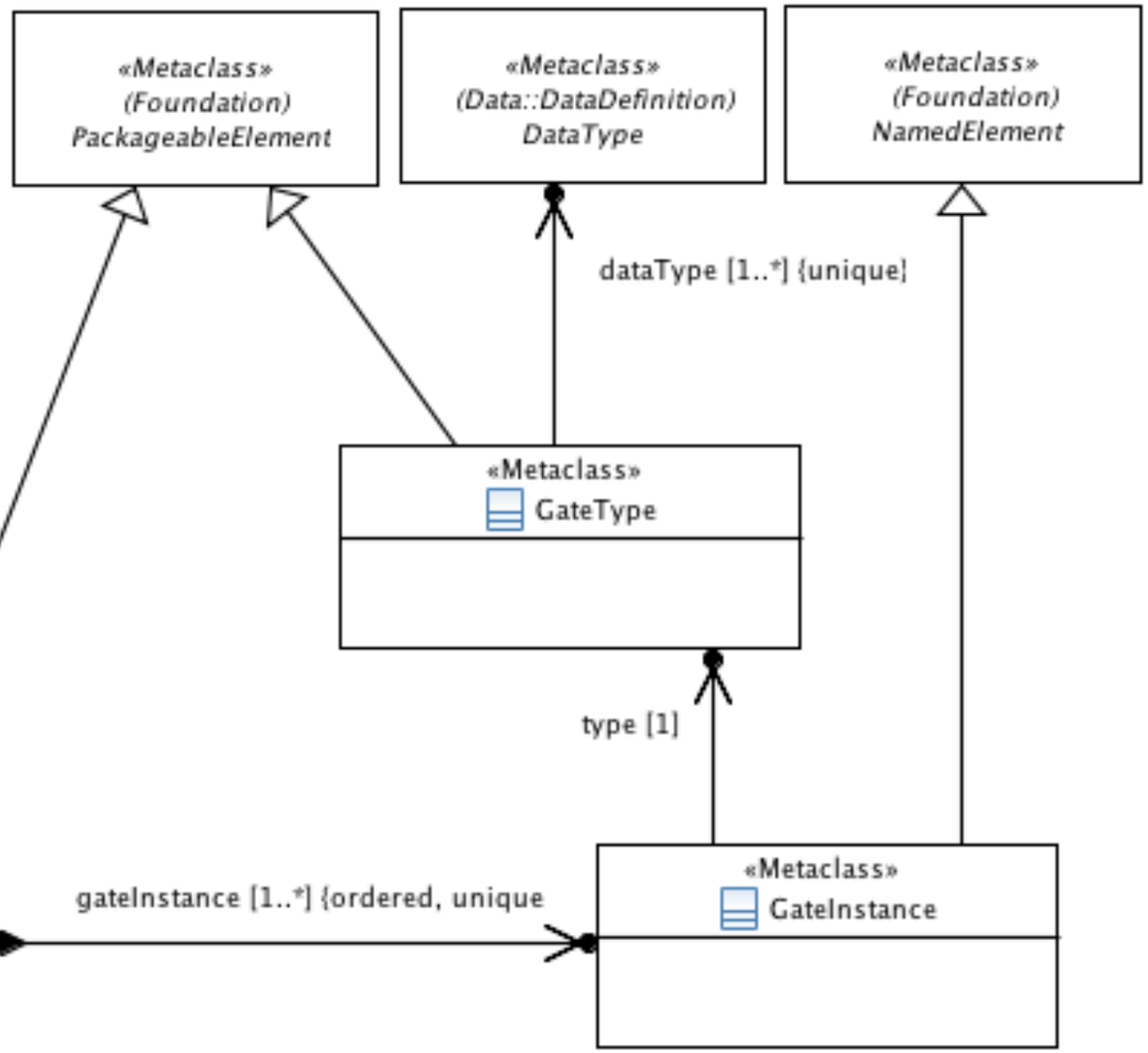

203 119-1 V1.3.1 (2016-09)
ETSI


203 119-2 V1.2.1 (2016-09)
ETSI


203 119-3 V1.2.1 (2016-09)
ETSI


ETSI ES 203 119-4 V1.2.1 (2016-09)
ETSI
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

TDL P1: MM



Semantics

A 'GateType' represents a type of communication points, called
'ComponentInstance's. A 'GateType' specifies the 'DataType's t
both directions.

Generalization

- PackageableElement

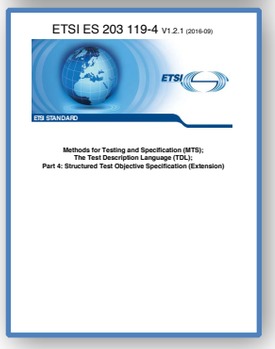
Properties

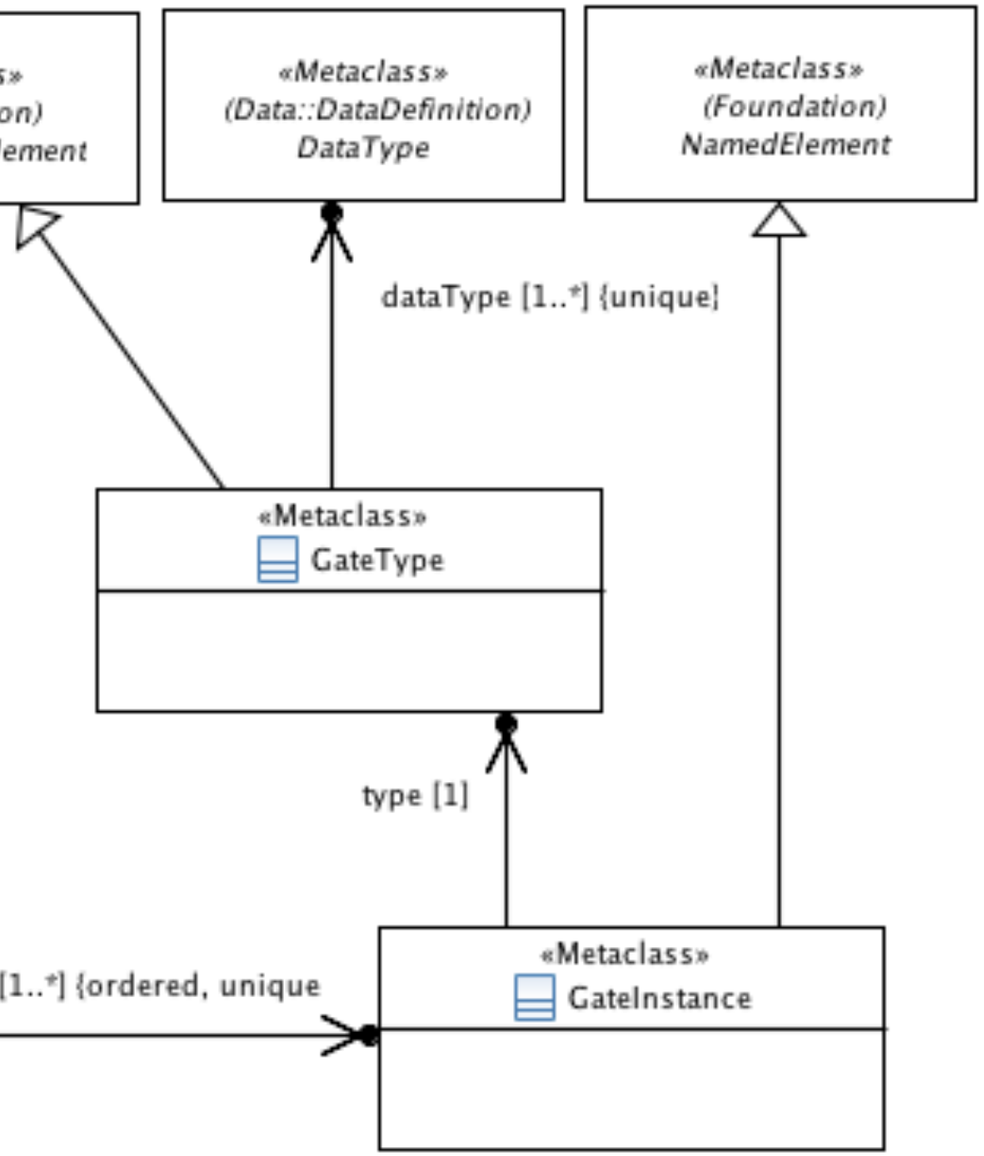
- dataType: DataType [1..*] {unique}
  The 'DataType's that can be exchanged via 'GateInstan
  shall adhere to the 'DataType's that are allowed to be e

Constraints

There are no constraints specified.

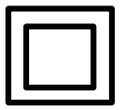
TDL P1: MM





6.4.2 GateType

Concrete Graphical Notation



GATETYPENAMELABEL
Data Type: DATATYPELISTLABEL

Formal Description

context GateType
GATETYPENAMELABEL **::=** self.name
DATATYPELISTLABEL **::=** self.dataType.name->**separator(',')**

Comments

No comments.

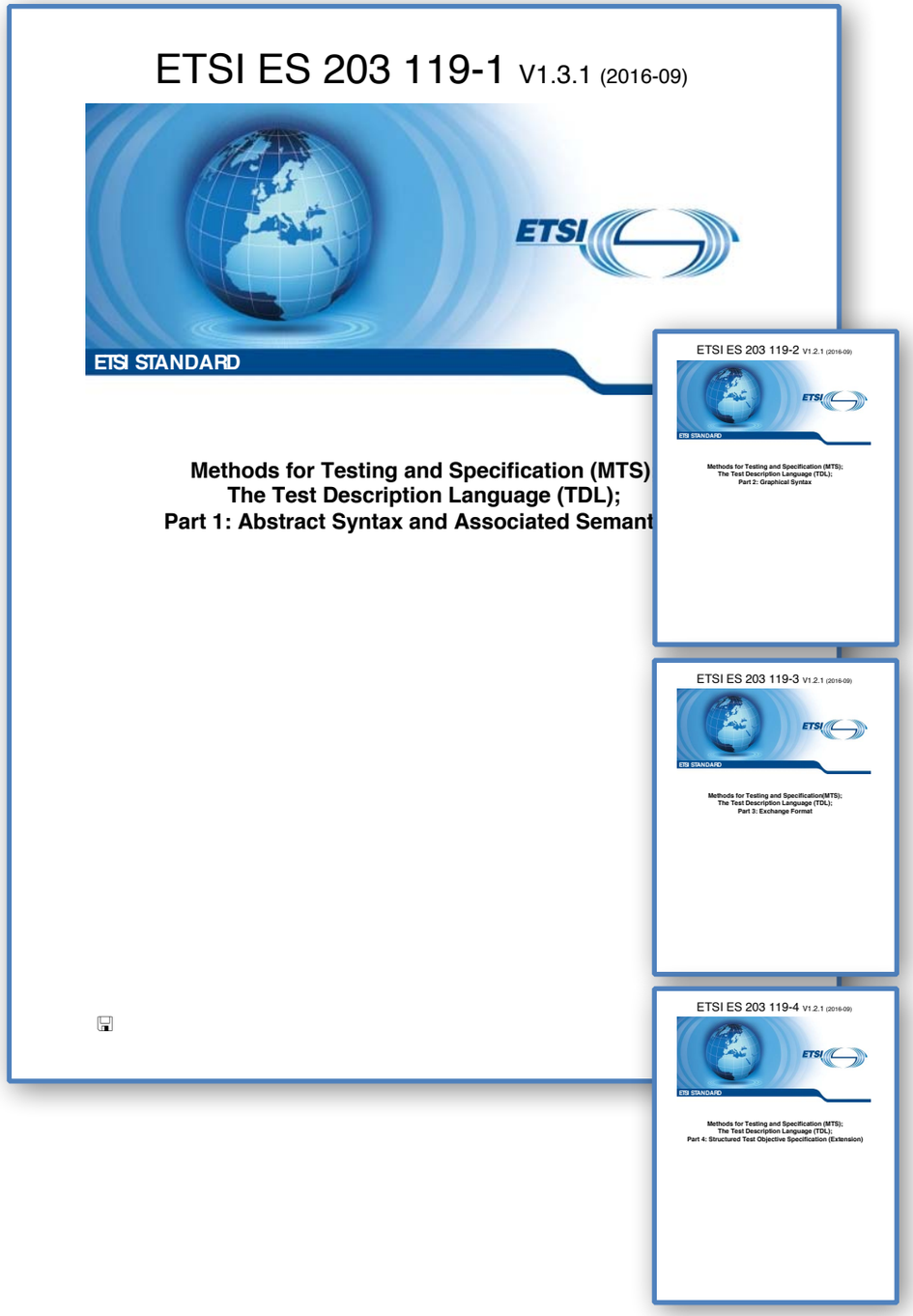
TDL P2: GR

TDL P1: MM

Radio
Data Type: Message, Signal

What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time



What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time

54 ETSI ES 203 119 V1.1.1 (2014-04)

Annex B (informative):
Examples of a TDL Concrete Syntax

B.1 Introduction

The applicability of the TDL meta-model that is described in the main part of the present document depends on the availability of TDL concrete syntaxes that implement the meta-model (abstract syntax). Such a TDL concrete syntax can then be used by end users to write TDL specifications. Though a concrete syntax will be based on the TDL meta-model, it can implement only parts of the meta-model if certain TDL features are not necessary to handle a user's needs.

This annex illustrates an example of a possible TDL concrete syntax in a textual format that supports all features of the TDL meta-model, called "TDLan". Three examples are outlined below - two examples translated from existing test descriptions taken from [i.2] and [i.3], as well as an example illustrating some of the TDL data parameter mapping concepts. The examples are accompanied by a complete reference description of the textual syn given in EBNF.

B.2 A 3GPP Conformance Example in Textual Sy

This example describes one possible way to translate clause 7.1.3.1 from TS 136 523-1 [i.2] into the prop textual syntax, by mapping the concepts from the representation in the source document to the correspon in the TDL meta-model by means of the proposed textual syntax. The example has been enriched with ad information, such as explicit data definitions and test configuration details for completeness where applic

```
//Translated from [i.2], Section 7.1.3.1
TDLan Specification Layer_2_DL_SCH_Data_Transfer {
   //Procedures carried out by a component of a test configuration
   //or an actor during test execution
   Action preCondition : "Pre-test Conditions:
        RRC Connection Reconfiguration" ;
   Action preamble : "Preamble:
            The generic procedure to get UE in test state Loopback
            Activated (State 4) according to TS 36.508 clause 4.5
            is executed, with all the parameters as specified in the
            procedure except that the RLC SDU size is set to return no
            data in uplink.
            (reference corresponding behaviour once implemented" ;

   //User-defined verdicts
   //Alternatively the predefined verdicts may be used as well
   Verdict PASS ;
   Verdict FAIL ;

   //User-defined annotation types
   Annotation TITLE ;          //Test description title
   Annotation STEP ;           //Step identifiers in source documents
   Annotation PROCEDURE ;      //Informal textual description of a test step
   Annotation PRECONDITION ; //Identify pre-condition behaviour
   Annotation PREAMBLE ;       //Identify preamble behaviour.

   //User-defined time units
   Time Unit seconds;

   //Test objectives (copied verbatim from source document)
   Test Objective TP1 {
      from : "36523-1-a20_s07_01.doc::7.1.3.1.1 (1)" ;
      description : "with { UE in E-UTRA RRC_CONNECTED state }
                  ensure that {
                   when { UE receives downlink assignment on the PDCCH
                       for the UE's C-RNTI and receives data in the
                             associated subframe and UE performs HARQ
                             operation }
                   then { UE sends a HARQ feedback on the HARQ
                             process }
                  }" ;
   }
```

ETSI

ETSI ES 203 119-2 V1.2.1 (2016-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax

ETSI ES 203 119-3 V1.2.1 (2016-09)

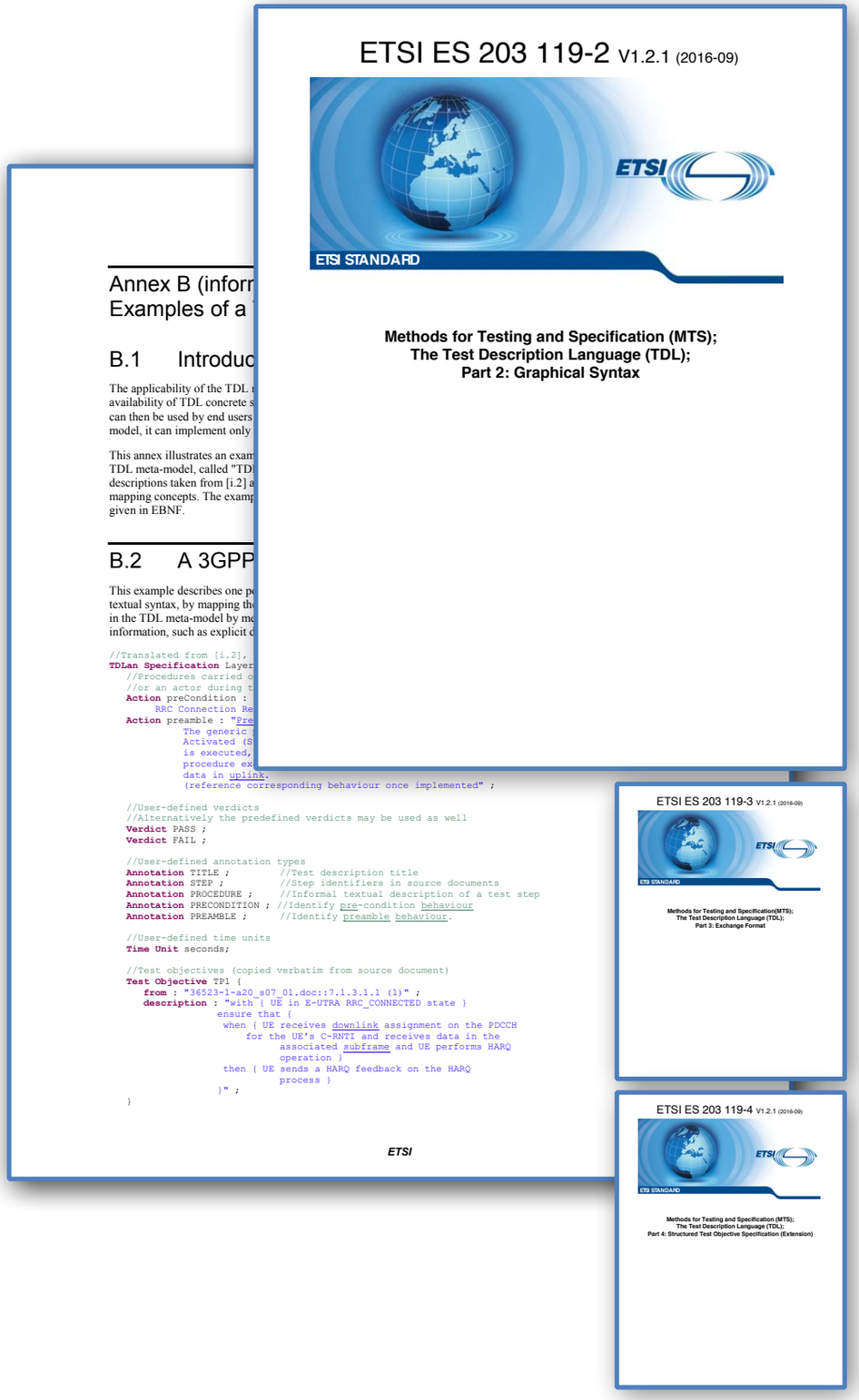
Methods for Testing and Specification(MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

ETSI ES 203 119-4 V1.2.1 (2016-09)

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time

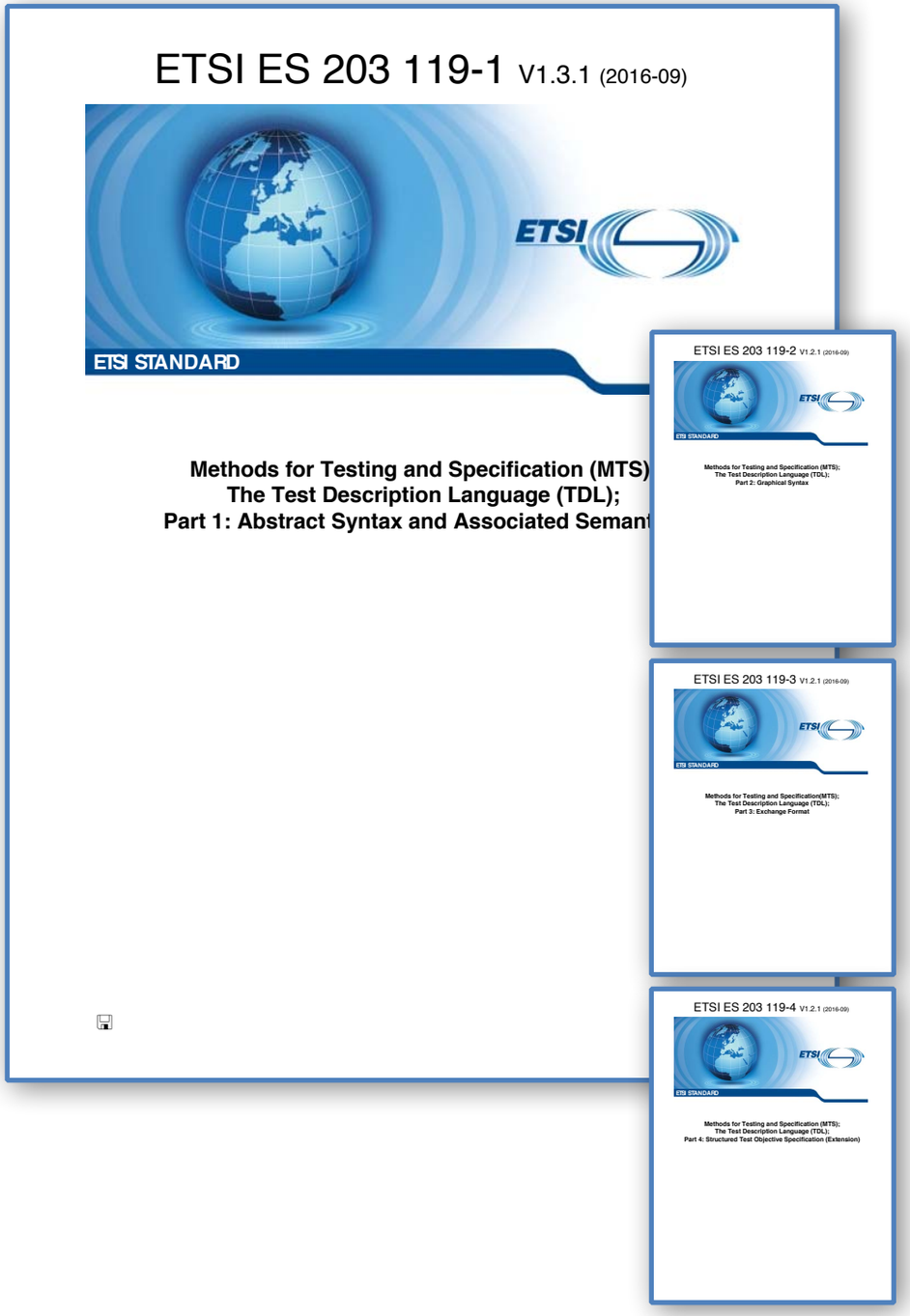


What is TDL? Part 1: MM

- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time

What is TDL? Part 1: MM

- Test data
  - data definition and data use
  - abstract types and instances
  - composed by using parameters
  - functions and actions
  - mappable to concrete data
  - variables and special values



What is TDL? Part 1: MM

Test Design

```
Type Login;
Login correct;
Login incorrect;

Use "data.ttcn3" as DATA ;
Map correct to "johnny_correct" in DATA as correct_ttcn3;
Map incorrect to "johnny_incorrect" in DATA as incorrect_ttcn3;
```

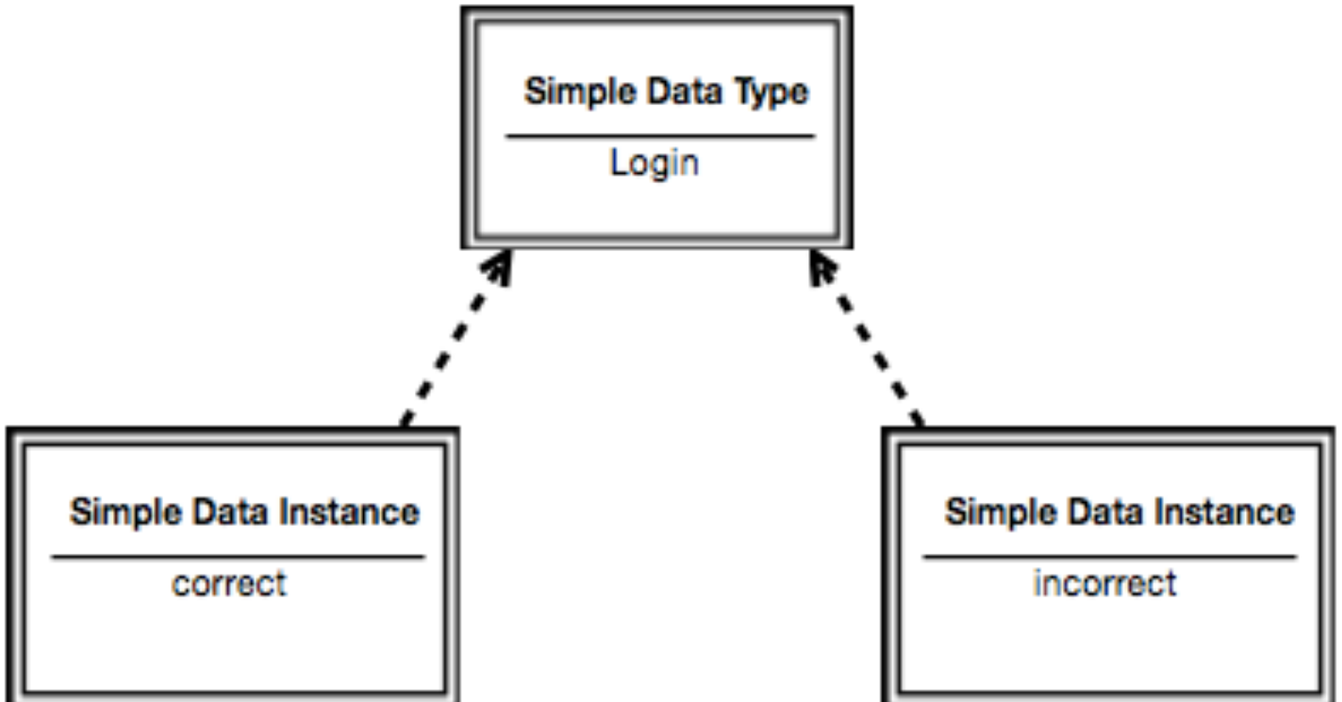
Test Implementation

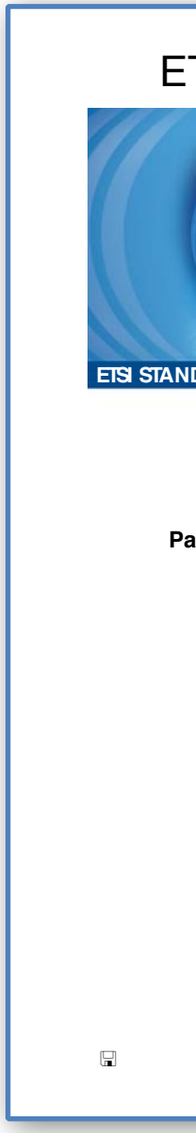
```
template Login johnny_correct := {
    user := "johnny",
    password := "apple",
    hint := "seed",
    id := 1000
}
template Login johnny_incorrect := {
    user := "johnny",
    password := "orange",
    hint := "second favourite fruit",
    id := 2000
}
```

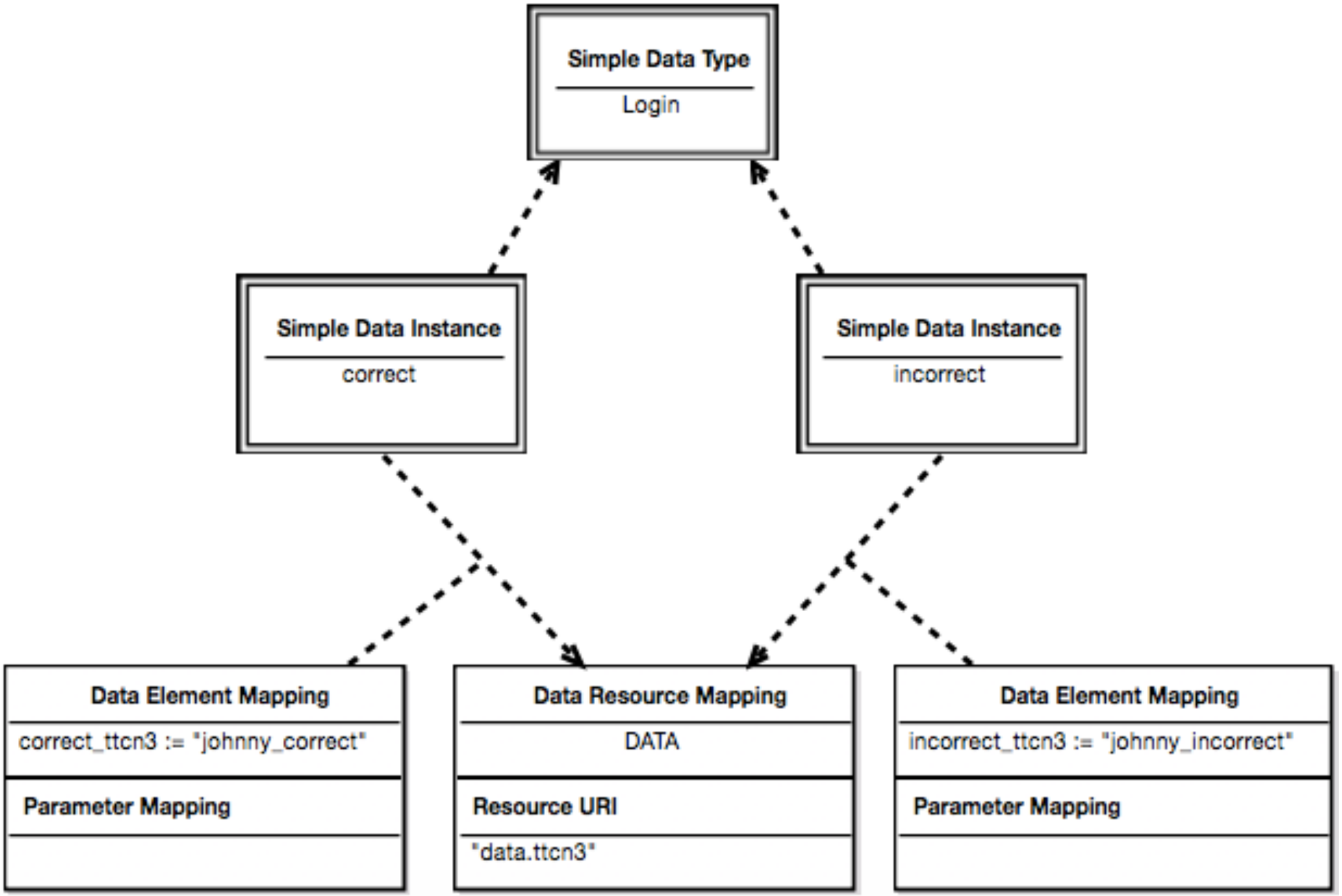
```
type record Login {
  charstring user,
  charstring password,
  charstring hint,
  integer id
} with {
  encode "xpath=//div[@id='login']";
  encode (user) "relative=/div/dd[3]";
  encode (password) "relative=/div/dd[4]";
};
```

What is TDL? Part 1: MM

```
Type Login;
Login correct;
Login incorrect;

Use "data.ttcn3" as DATA ;
Map correct to "johnny_correct" in DATA as correct_ttcn3;
Map incorrect to "johnny_incorrect" in DATA as incorrect_ttcn3;
```

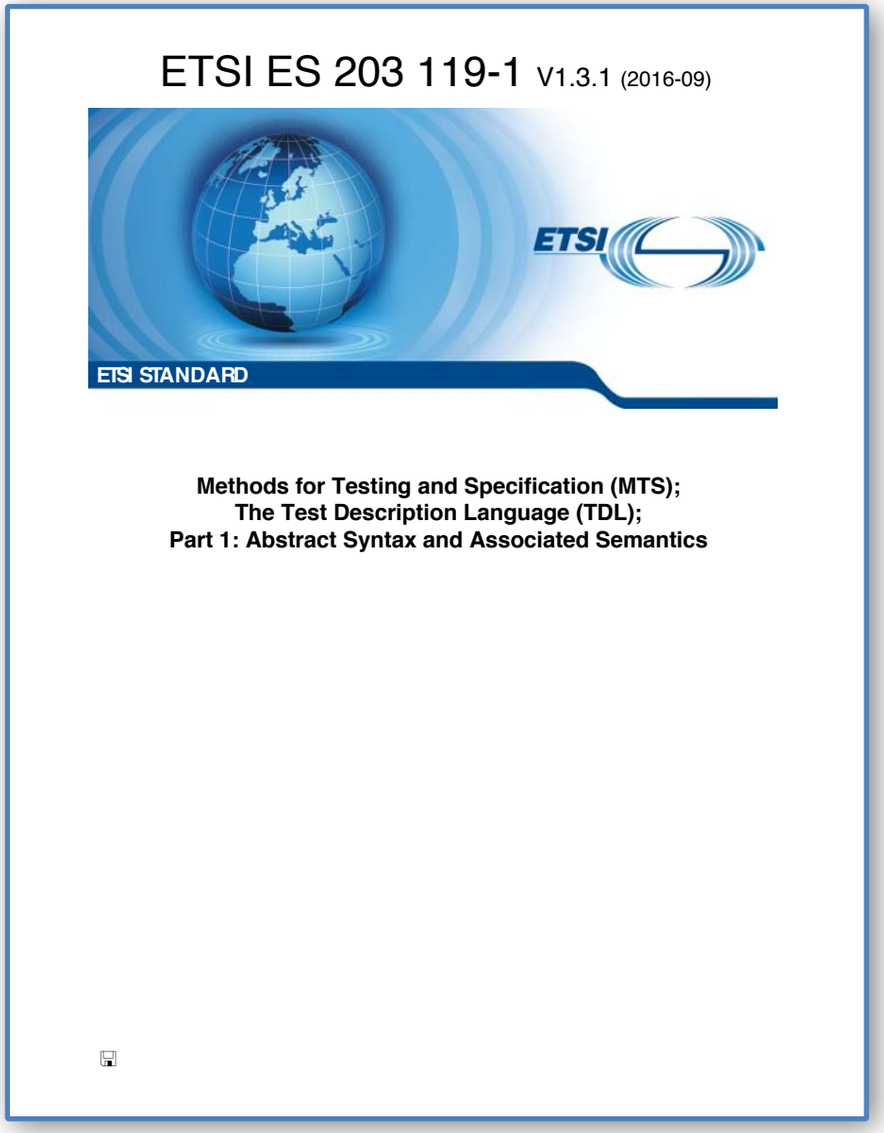





Simple Data Type
Login
Simple Data Instance
correct
Simple Data Instance
incorrect
Data Element Mapping
correct_ttcn3 := "johnny_correct"
Parameter Mapping
Data Resource Mapping
DATA
Resource URI
"data.ttcn3"
Data Element Mapping
incorrect_ttcn3 := "johnny_incorrect"
Parameter Mapping

What is TDL? Part 1: MM

- Test configuration
  - typed components and gates
  - timers and variables
  - connections among gates
  - component roles

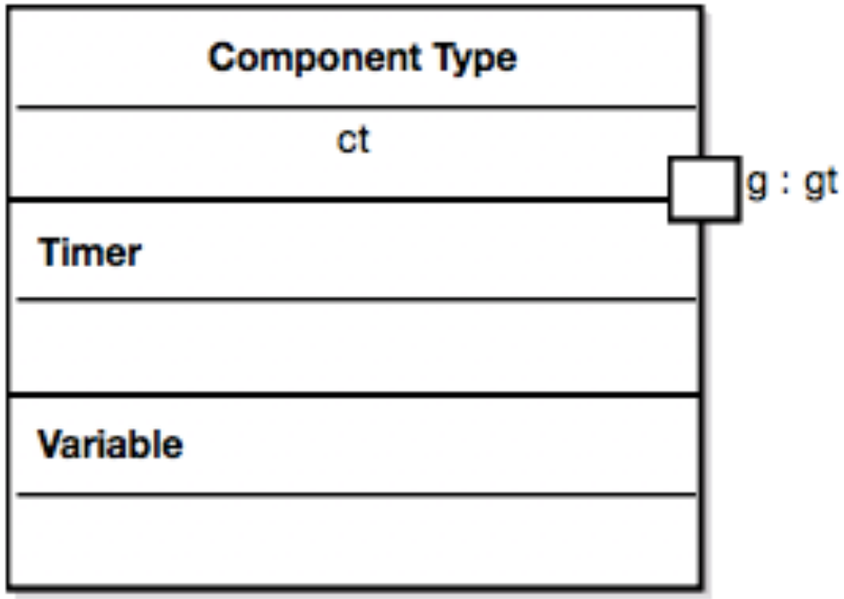

ETSI ES 203 119-1 V1.3.1 (2016-09)

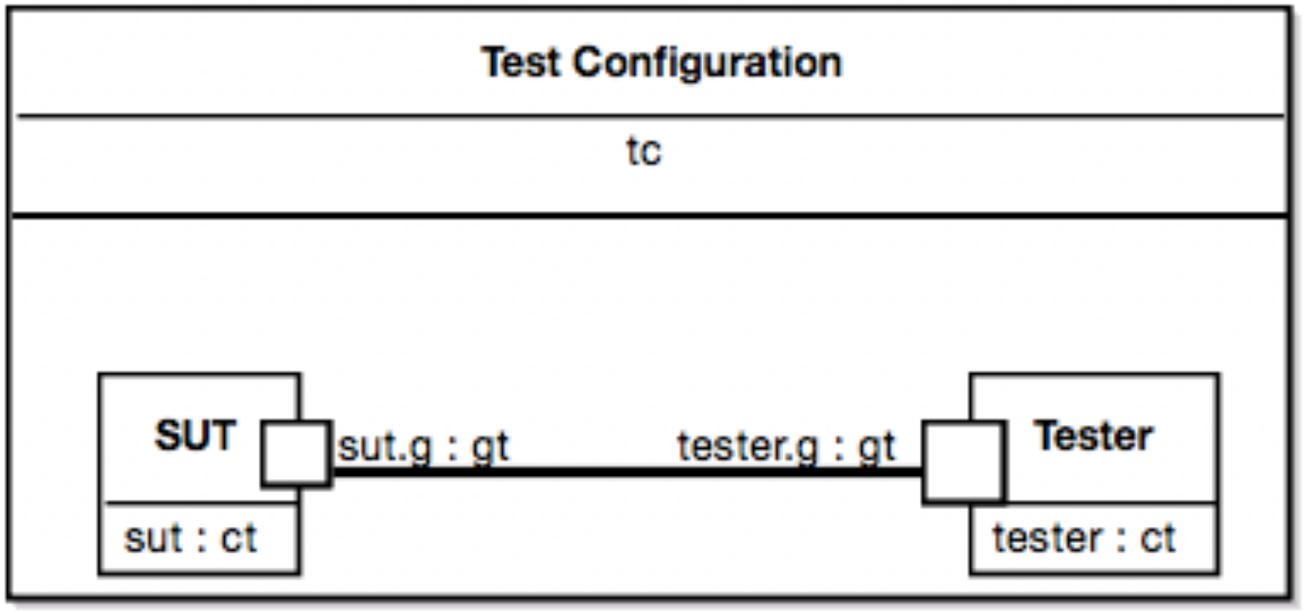
ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

What is TDL? Part 1: MM

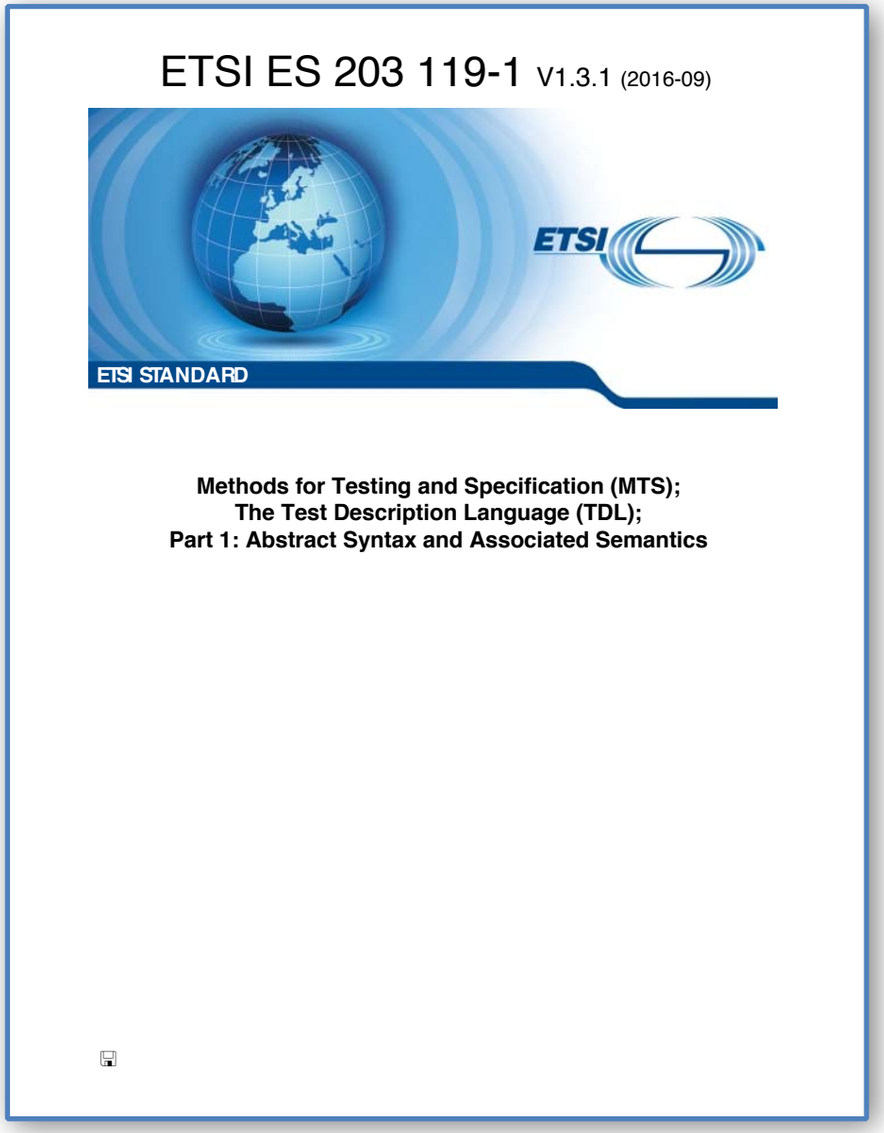
```
Gate Type gt accepts Login, Response;

Component Type ct having {
    gate g of type gt;
}

Test Configuration tc {
    create Tester tester of type ct;
    create SUT sut of type ct;
    connect tester.g to sut.g;
}
```





What is TDL? Part 1: MM

- Test behaviour
  - defines expected behaviour
  - failure upon deviations by default
  - actions and interactions
  - alternative, parallel, iterative, conditional
  - defaulting, interrupting, breaking


ETSI ES 203 119-1 V1.3.1 (2016-09)

ETSI STANDARD

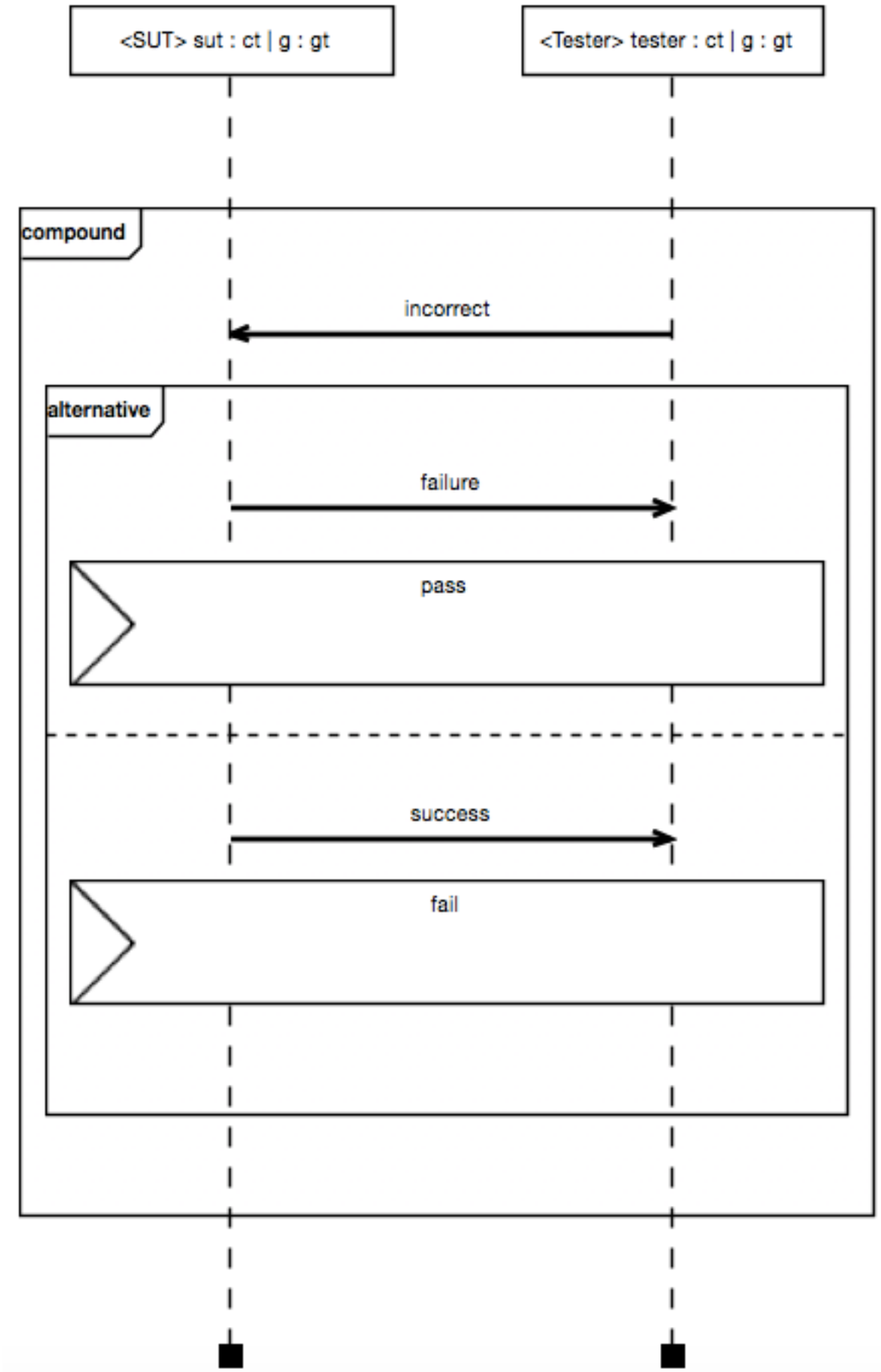
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

What is TDL? Part 1: MM

```
Test Description td (p of type Login)
  uses configuration tc {
    tester.g sends incorrect to sut.g;
    alternatively {
        sut.g sends failure to tester.g with {
            test objectives : tp;
        };
        set verdict to pass;
    } or {
        sut.g sends success to tester.g;
        set verdict to fail;
    }
}
```

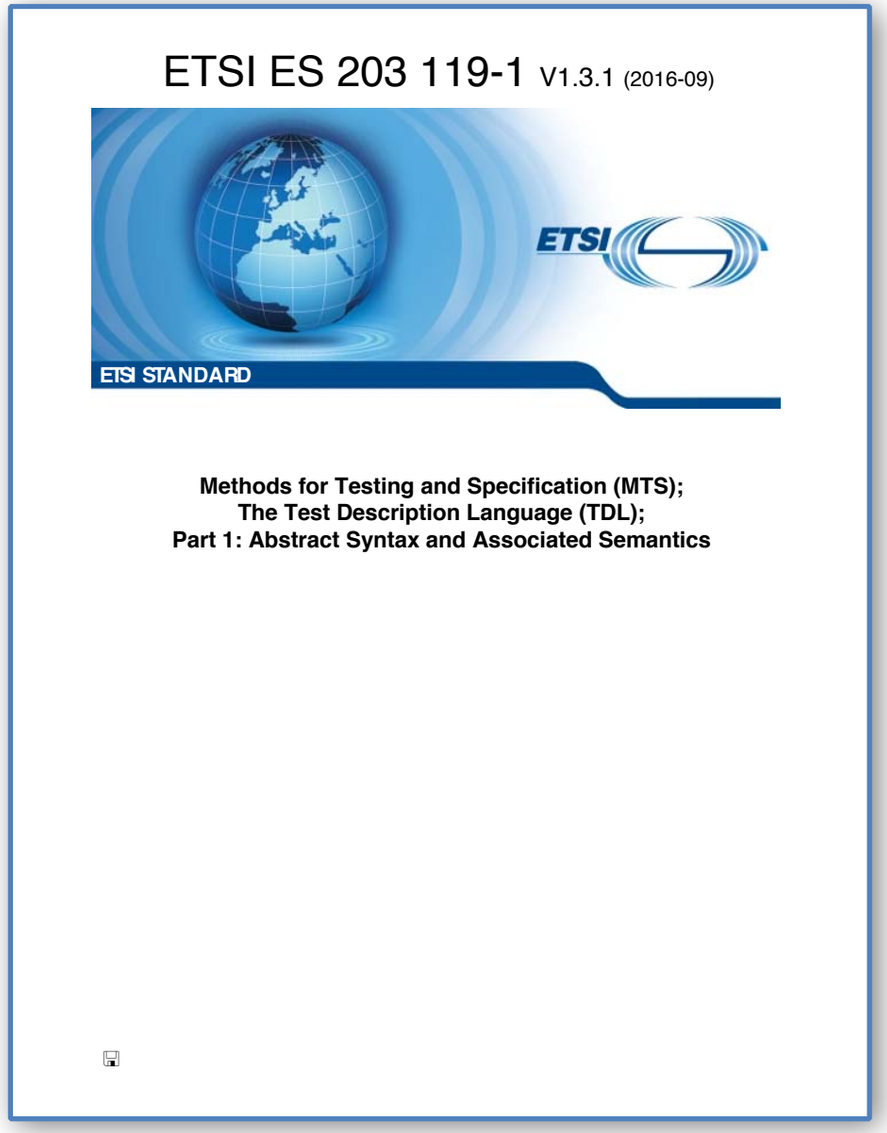
or simply (relying on the default semantics):

```
Test Description td_default (p of type Login)
  uses configuration tc {
    tester.g sends incorrect to sut.g;
    sut.g sends failure to tester.g with {
        test objectives : tp;
    };
}
```



What is TDL? Part 1: MM

- Test objectives
  - may be attached to
    - behaviour (atomic or compound)
    - whole test description
  - contain description and reference

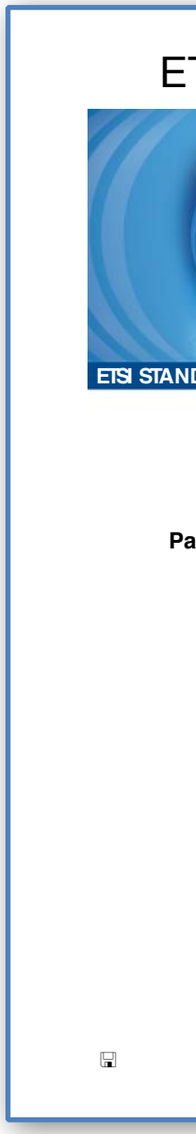

ETSI ES 203 119-1 V1.3.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

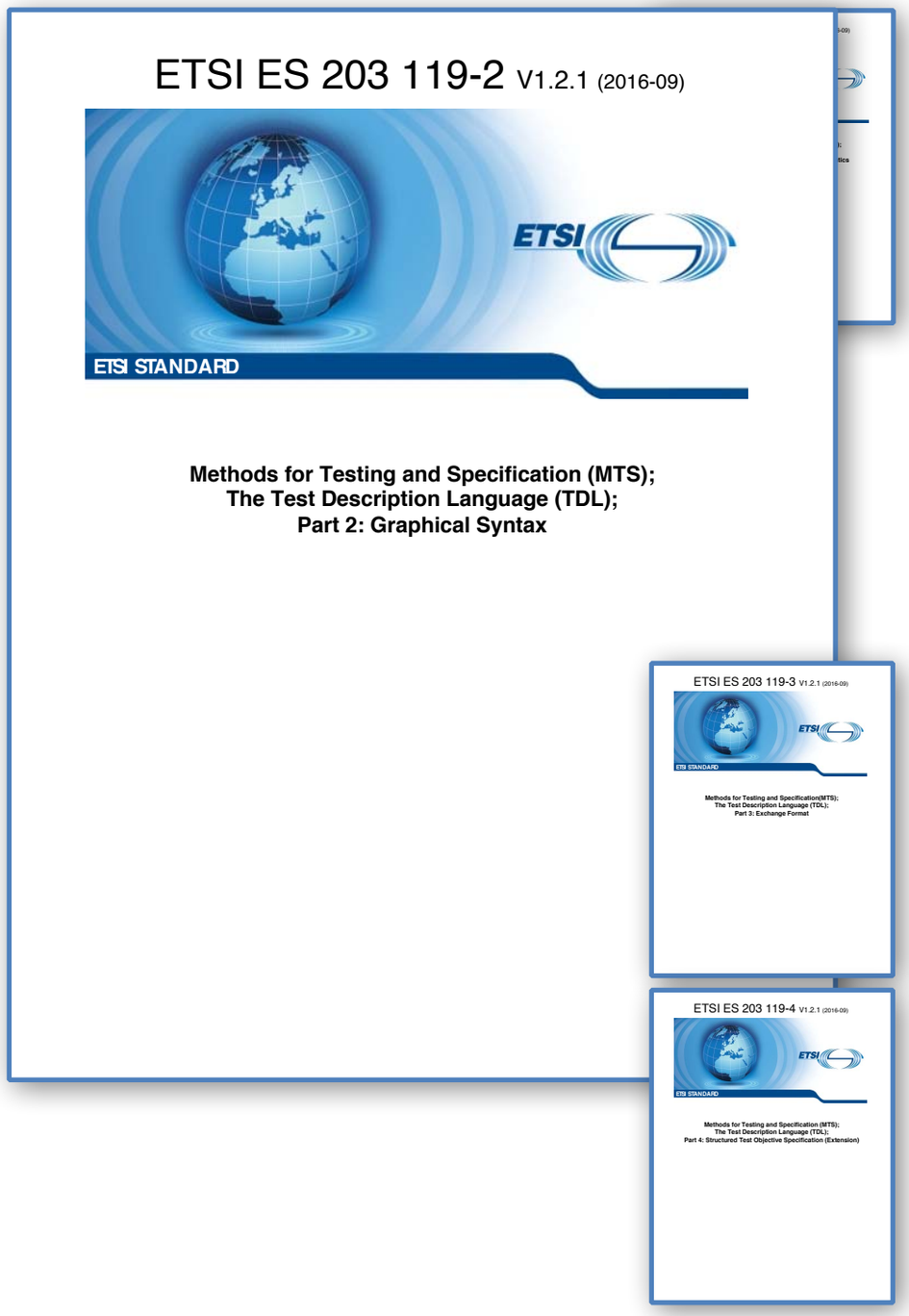
What is TDL? Part 1: MM

```
Test Objective tp {
    description : "ensure that
                   when incorrect login is provided
                   a failure response is sent";
}
Test Description td (p of type Login)
  uses configuration tc {
    tester.g sends incorrect to sut.g;
    alternatively {
        sut.g sends failure to tester.g with {
            test objectives : tp;
        };
        set verdict to pass;
    } or {
        sut.g sends success to tester.g;
        set verdict to fail;
    }
}
```



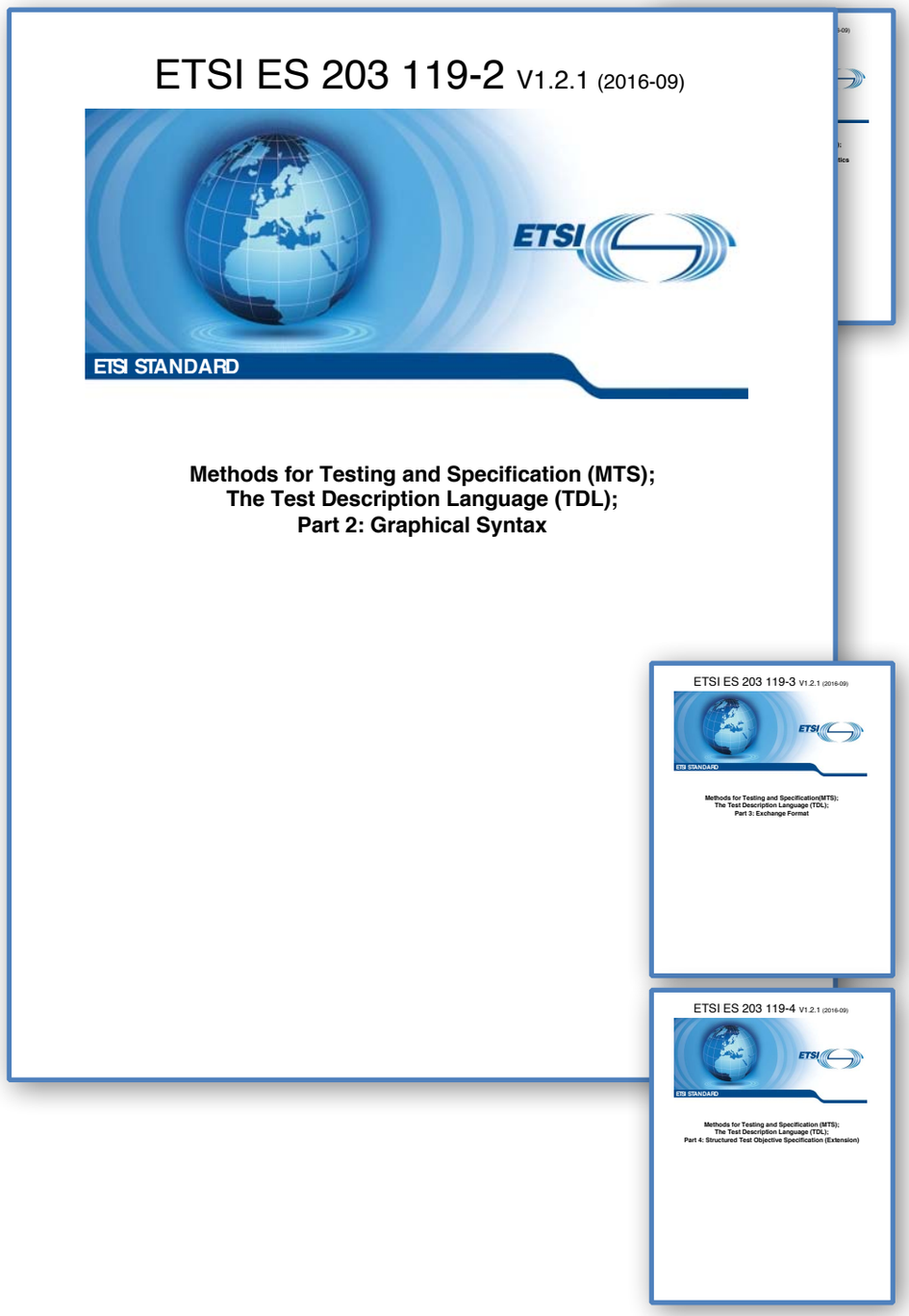
What is TDL? Part 2: GR

- Graphical languages
  - common in (test) modelling
  - ease communication
- TDL Graphical Syntax
  - hybrid graphical language
  - simple shapes, compartments
  - textual visualisation of contents

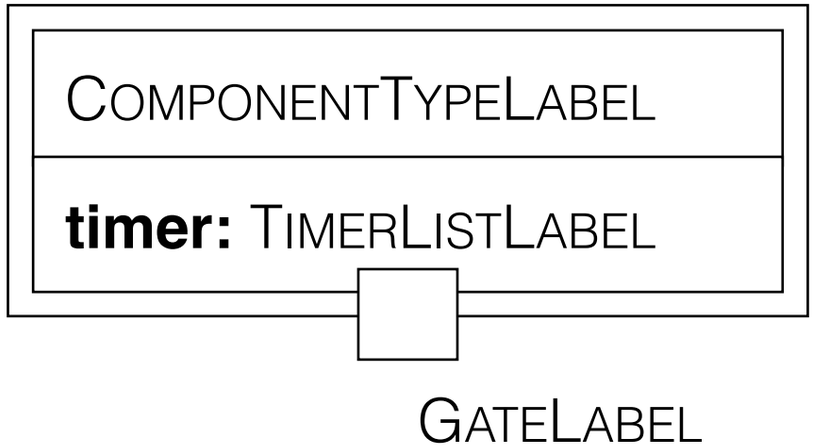


What is TDL? Part 2: GR

- Aligned with UML
  - distinct where semantics differ
- One diagram to rule them all!
- BNF-like label specification
- Considers both ease of use and implementation
- Prototyped with Sirius



What is TDL? Part 2: GR

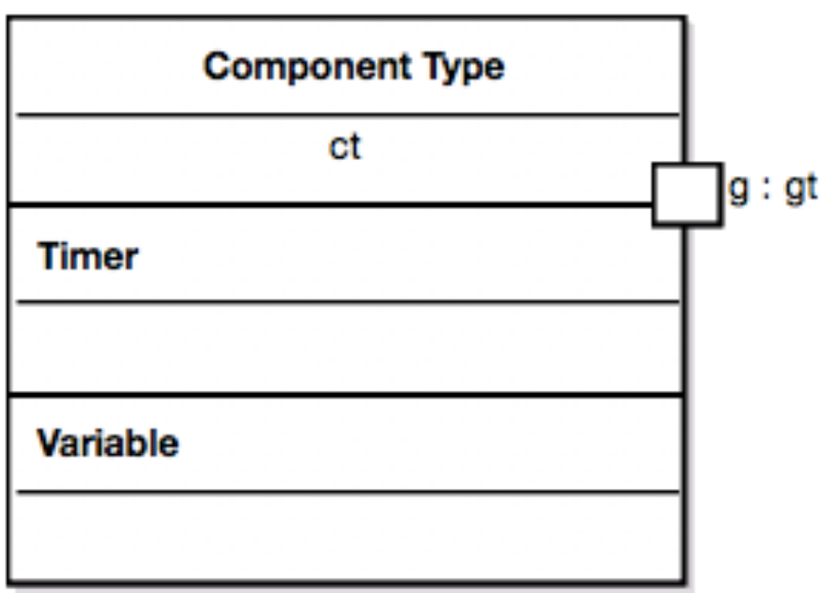


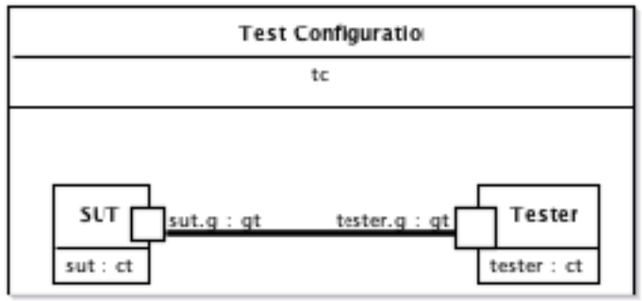
context: ComponentType

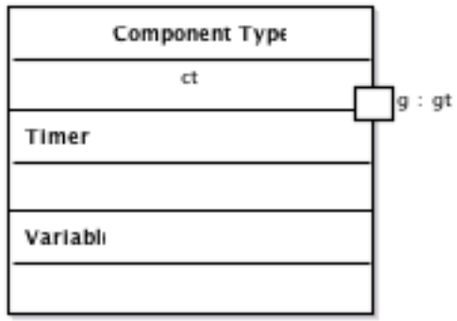
COMPONENTTYPELABEL ::= self.name

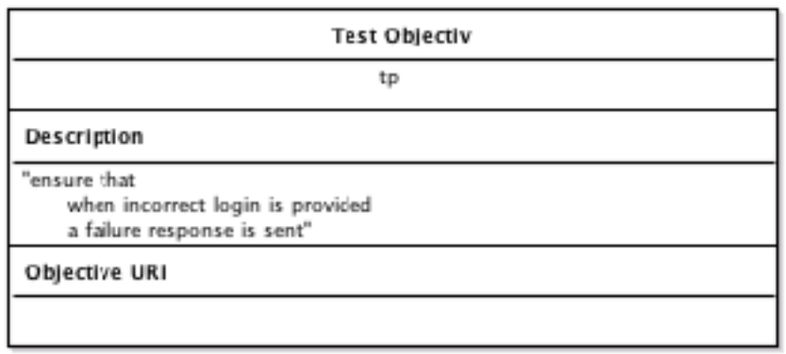
TIMERLISTLABEL ::= self.timer.name

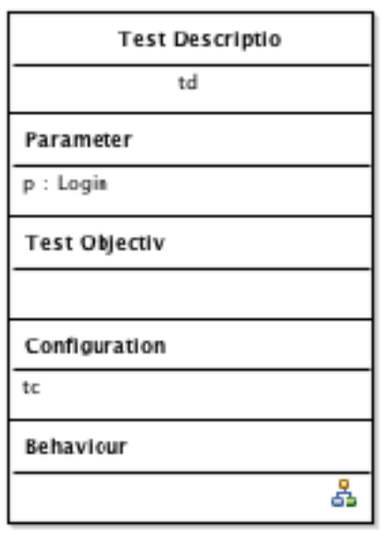
…

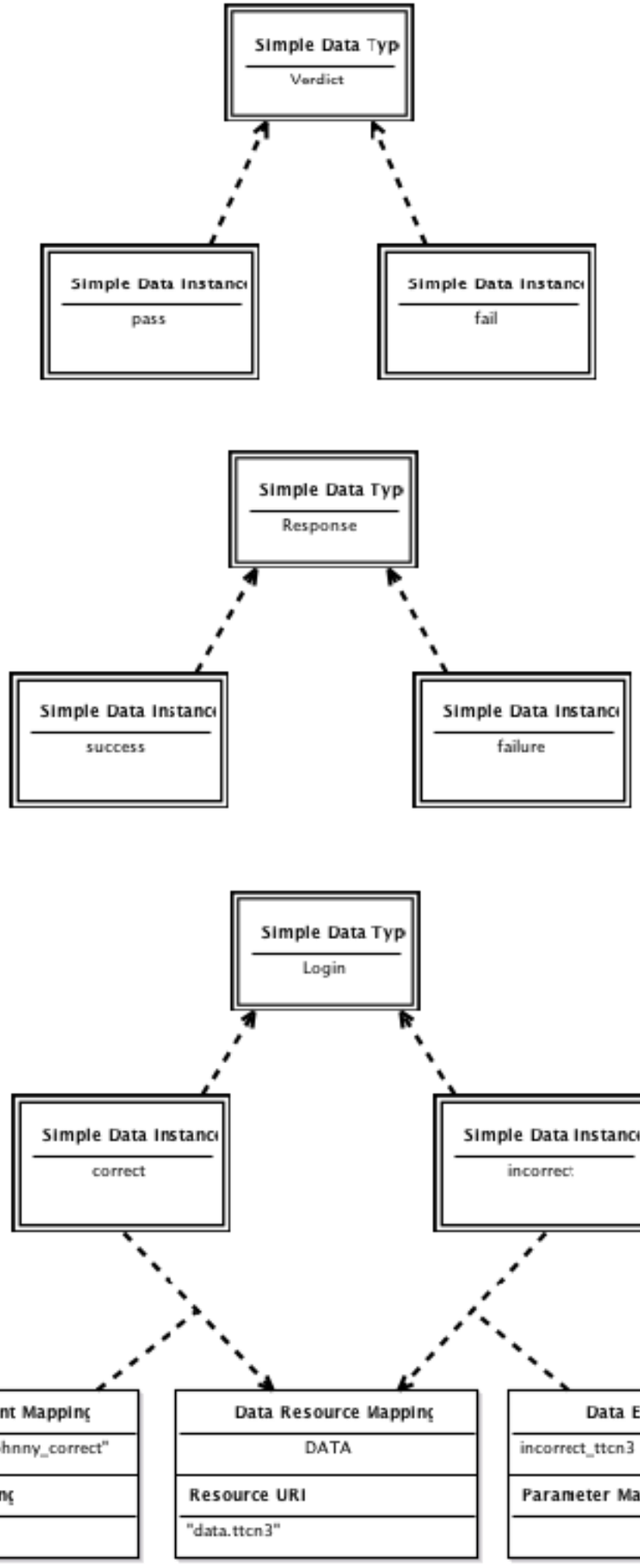


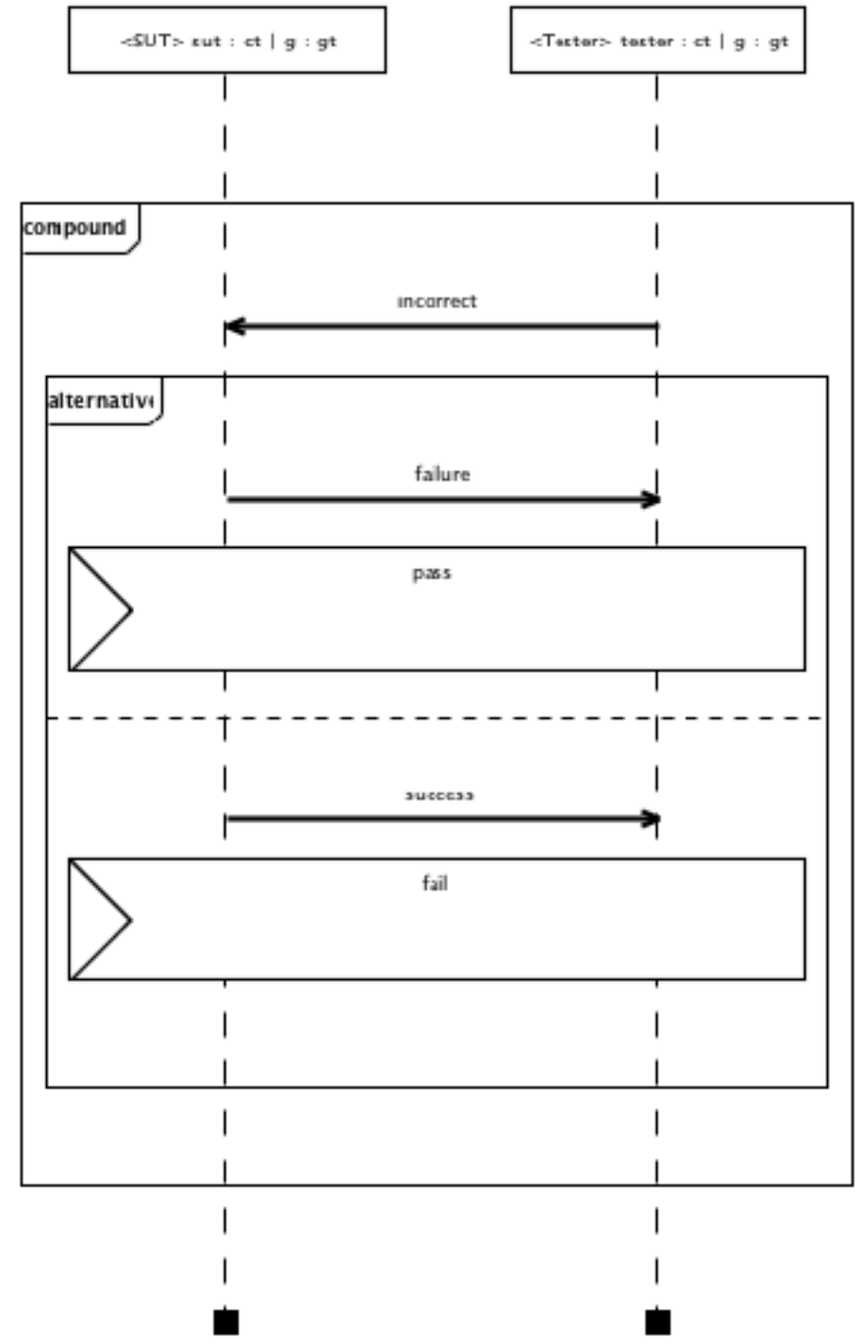

Test Configuratio
tc
SUT
sut : ct
sut.g : gt
tester.g : gt
Tester
tester : ct


Component Type
ct
g : gt
Timer
Variabl


Test Objectiv
tp
Description
"ensure that
when incorrect login is provided
a failure response is sent"
Objective URI

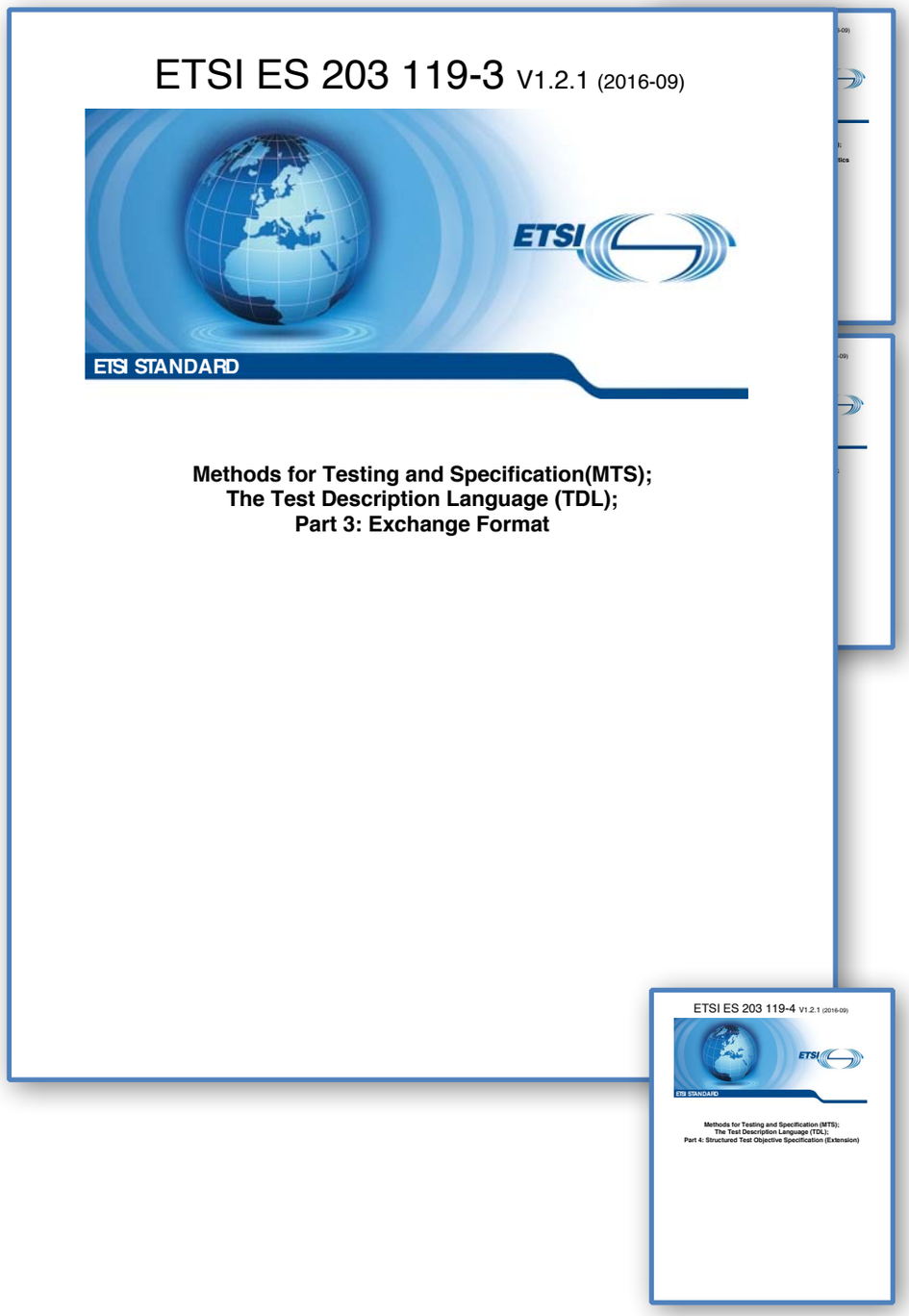

Test Descriptio
td
Parameter
p : Login
Test Objectiv
Configuration
tc
Behaviour


Simple Data Typ
Verdict
Simple Data Instanc
pass
Simple Data Instanc
fail
Simple Data Typ
Response
Simple Data Instanc
success
Simple Data Instanc
failure
Simple Data Typ
Login
Simple Data Instanc
correct
Simple Data Instanc
incorrect
Data Element Mapping
correct_ttcn3 := "johnny_correct"
Parameter Mapping
Data Resource Mapping
DATA
Resource URI
"data.ttcn3"
Data Element Mapping
incorrect_ttcn3 := "johnny_incorrect"
Parameter Mapping


<SUT> sut : ct | g : gt
<Tester> tester : ct | g : gt
compound
incorrect
alternativ
failure
pass
success
fail

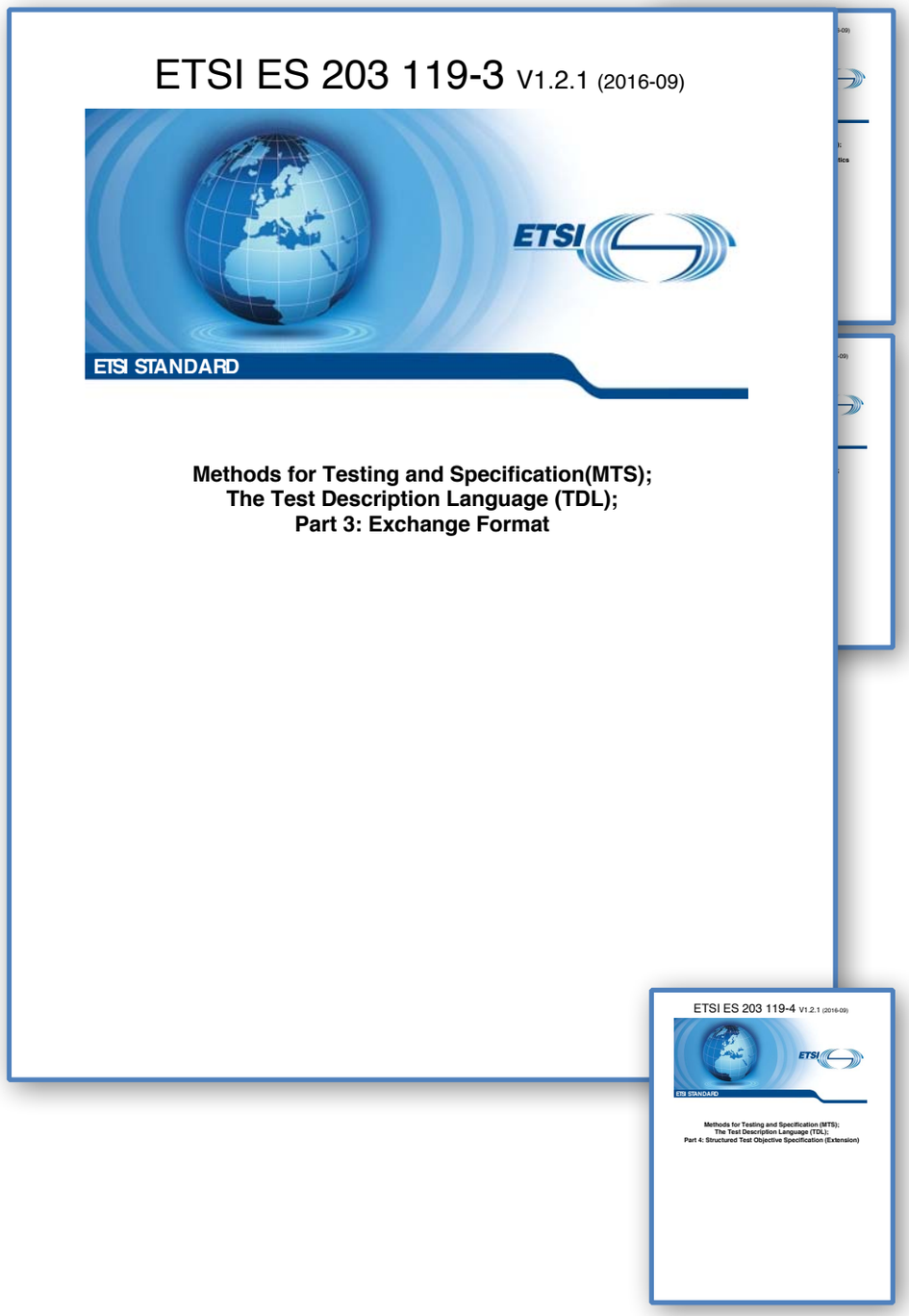
What is TDL? Part 3: XF

- Based on OMG XMI
  - XML: Metadata Interchange
  - Serialisation of MOF models
  - Exchange among MOF tools
- XMI concerns
  - complex, many options



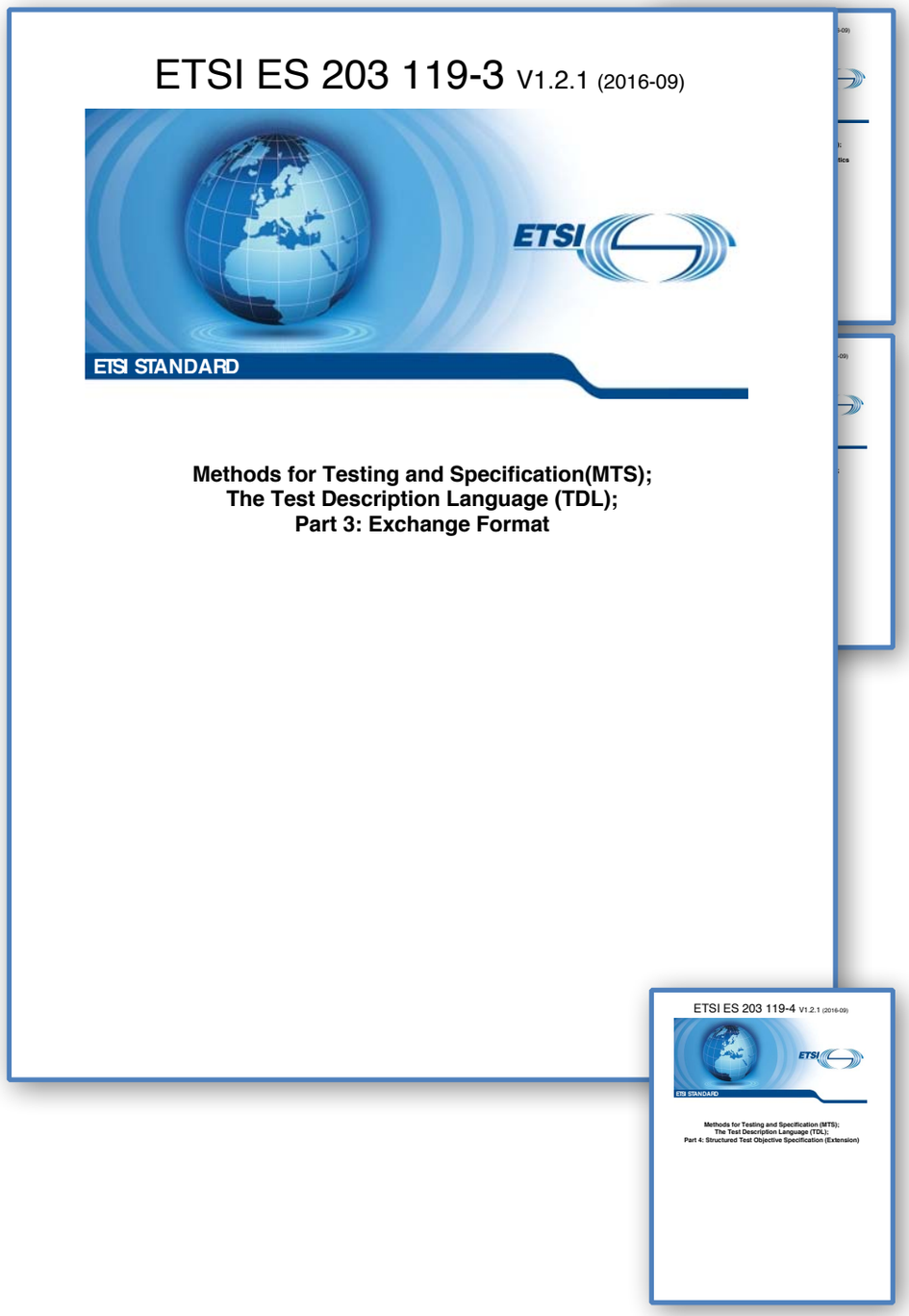
What is TDL? Part 3: XF

- TDL specific XMI structure
  - exchange of TDL models
  - canonical TDL XMI structure
    - meta-class representations
    - multiplicity, associations, inheritance
  - restrict flexibility of XMI
  - syntactical validity only!



What is TDL? Part 3: XF

- Syntactical validity only?
  - two-step validation
  - syntax: XMI Schema
  - semantics: MOF model validation


ETSI ES 203 119-3 V1.2.1 (2016-09)

ETSI STANDARD

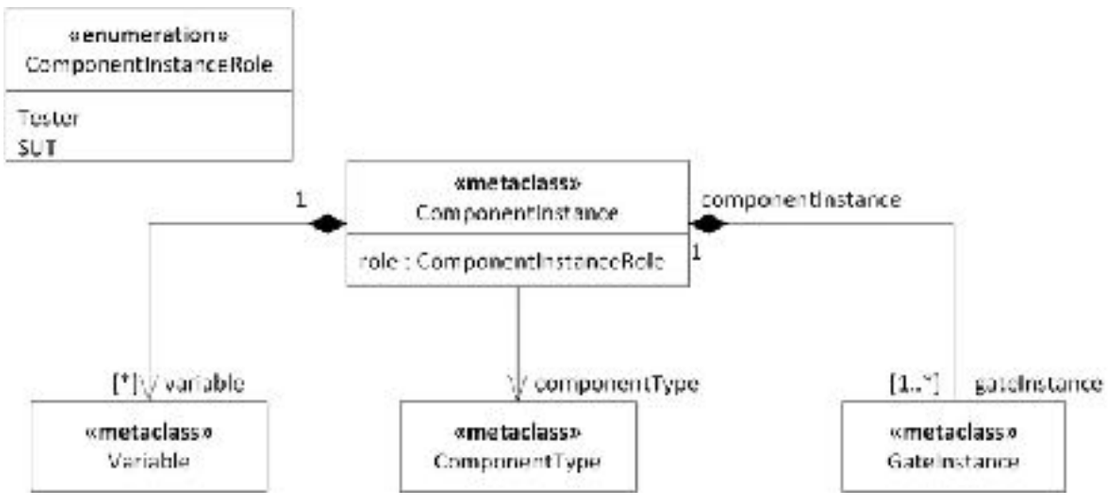
Methods for Testing and Specification(MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

ETSI ES 203 119-4 V1.2.1 (2016-09)

ETSI STANDARD

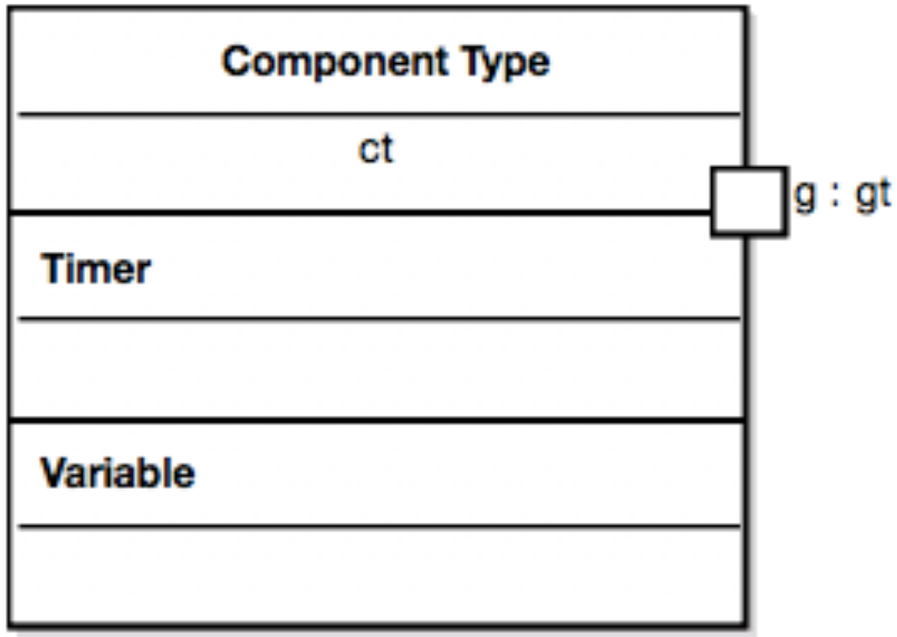
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

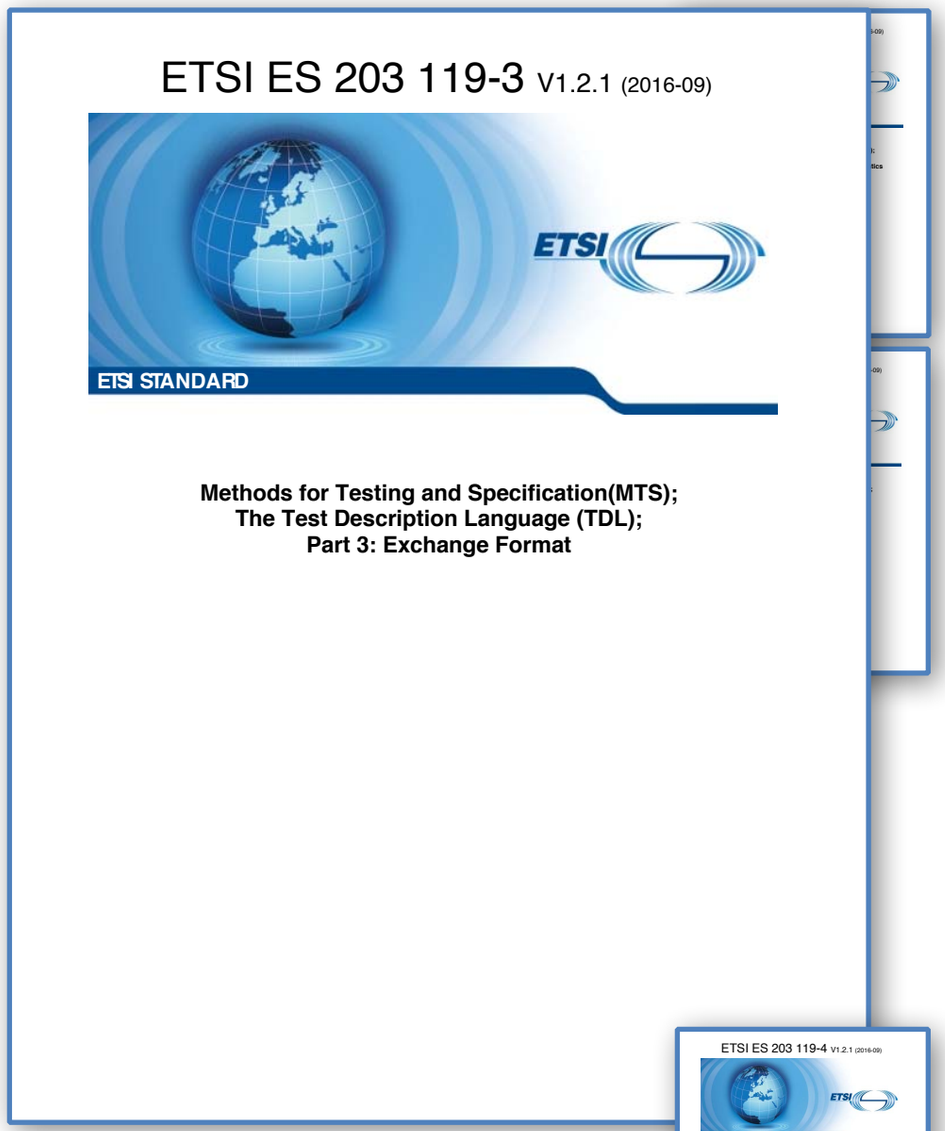
What is TDL? Part 3: XF



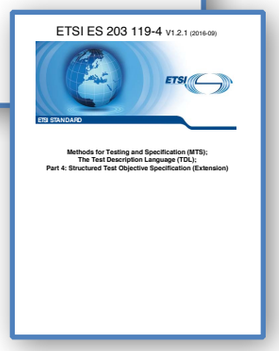

```
<xsd:complexType name="ComponentInstance">
  <xsd:complexContent>
    <xsd:extension base="tdl:Element">
      <xsd:choice maxOccurs="unbounded" minOccurs="0">
        <xsd:element name="gateInstance" type="tdl:GateInstance"/>
        <xsd:element name="variable" type="tdl:Variable"/>
      </xsd:choice>
      <xsd:attribute name="componentType" type="xsd:anyURI">
      <xsd:attribute name="role" type="tdl:ComponentInstanceRole">
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

What is TDL? Part 3: XF



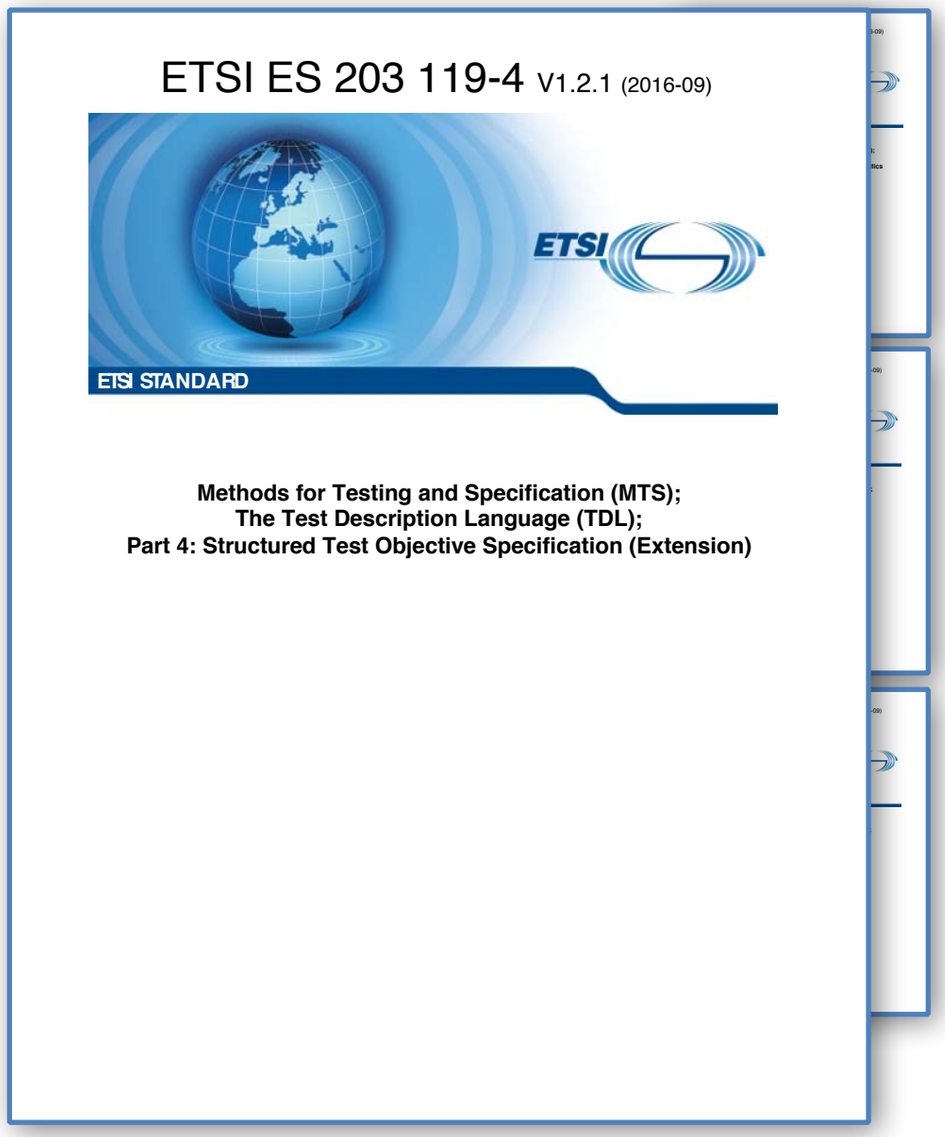


```
<packagedElement xsi:type="tdl:ComponentType" xmi:id="_qKt233asEeWrfP0MdfQNpg" name="ct">
  <gateInstance xmi:id="_qKt24HasEeWrfP0MdfQNpg" name="g" type="_qKt23nasEeWrfP0MdfQNpg"/>
</packagedElement>
```



What is TDL? Part 4: TO

- Based on TPLan
  - refine test objectives
  - formalise specification
  - integrate and unify test description and test purpose specification

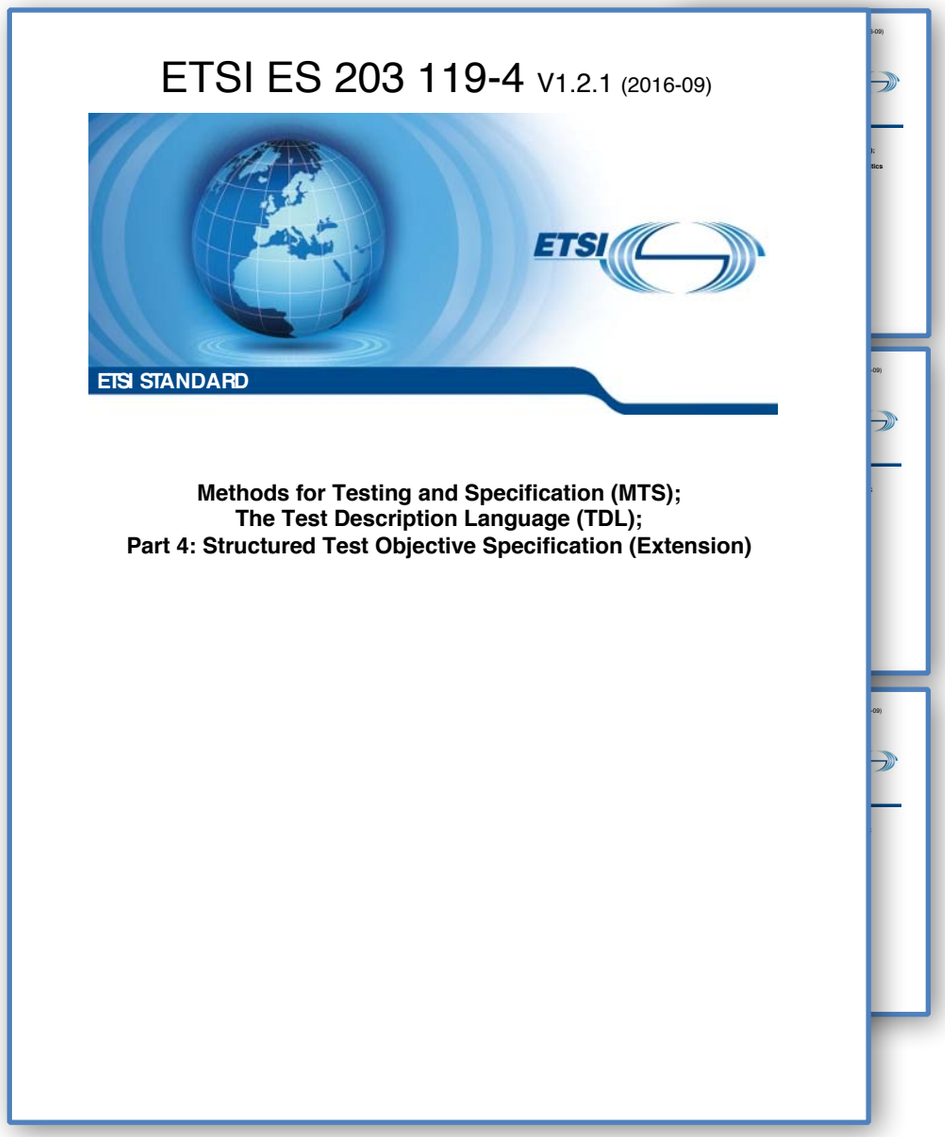

ETSI ES 203 119-4 V1.2.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

What is TDL? Part 4: TO

- Base Standard Specification
- Identification of Requirements
- Creation of ICS/IFS
- Definition of TSS
- Specification of Test Purposes
- Specification of Test Descriptions
- Specification of Test Cases
- Validation

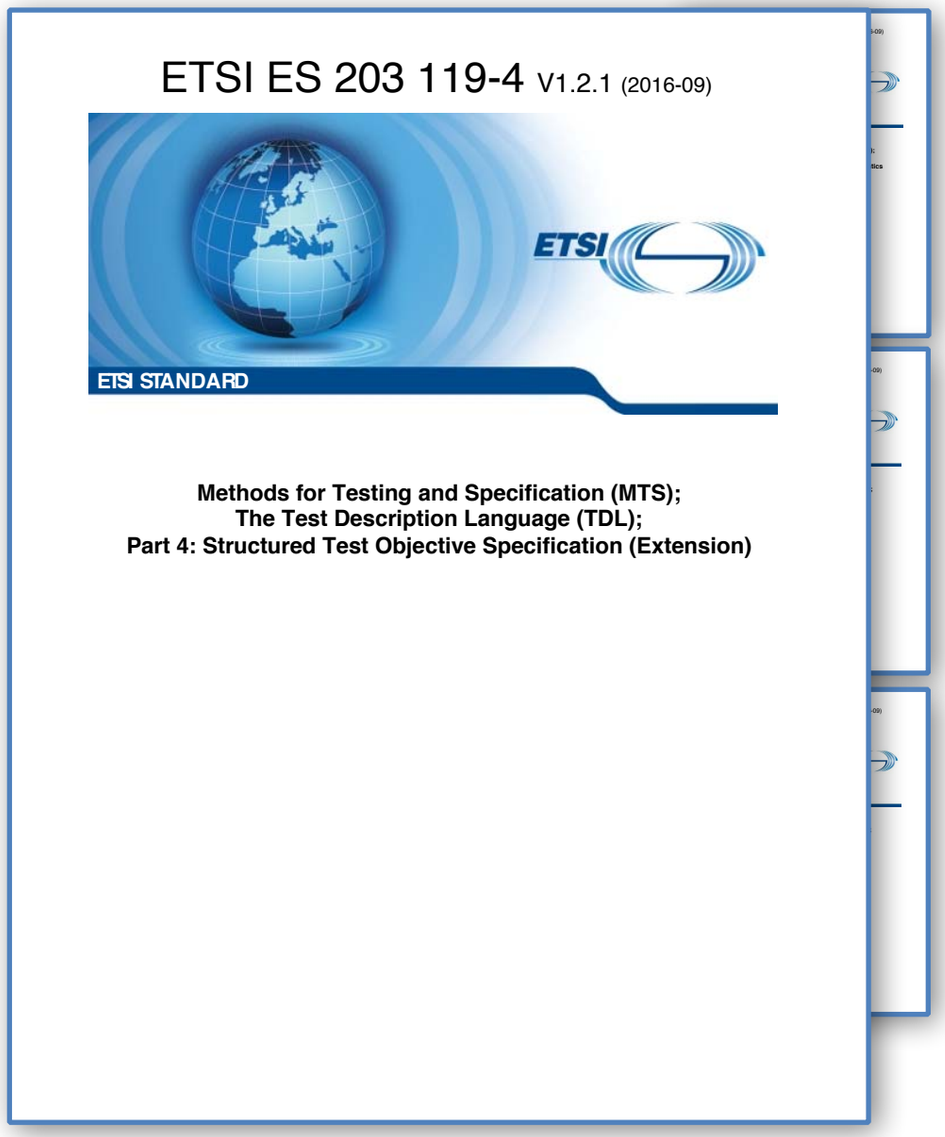

ETSI ES 203 119-4 V1.2.1 (2016-09)

ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

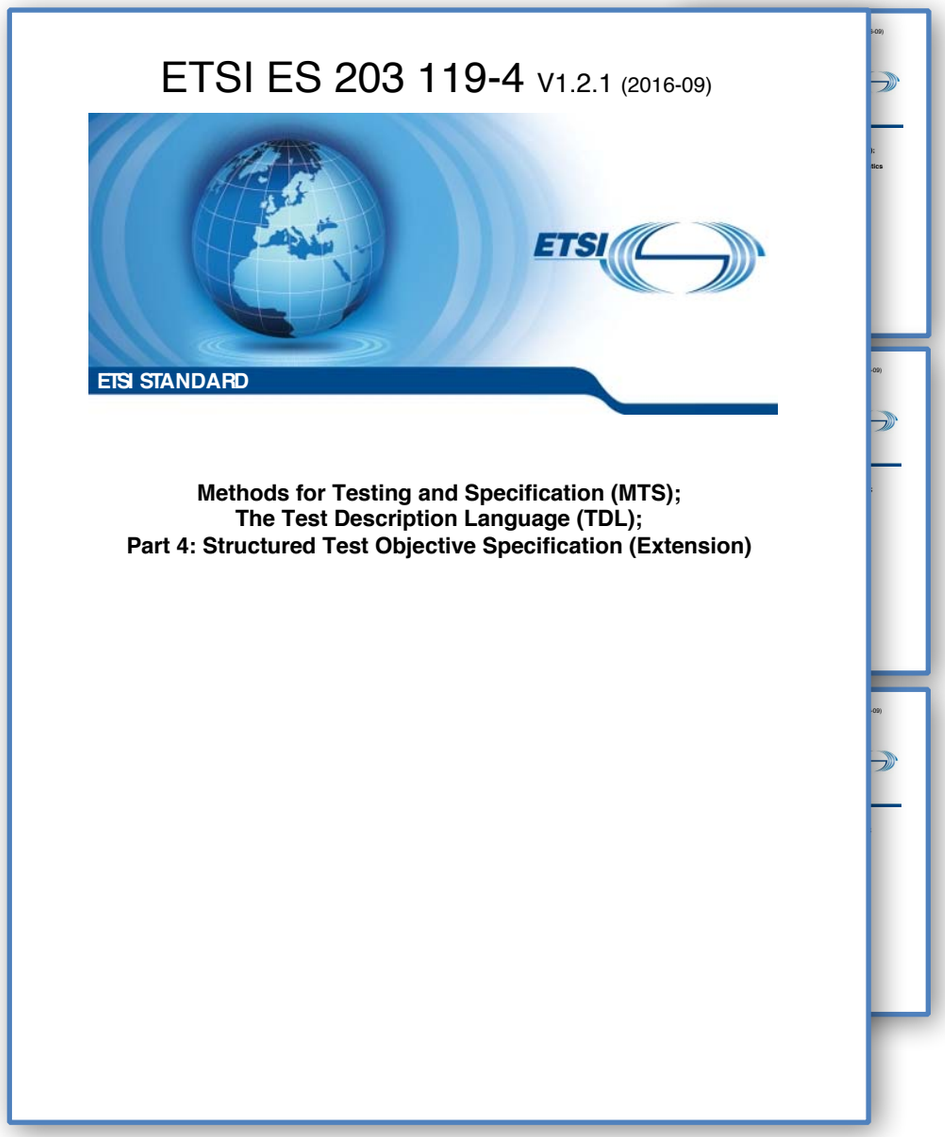
What is TDL? Part 4: TO

- Base Standard Specification
- Identification of Requirements
- Creation of ICS/IFS
- Definition of TSS
- Specification of Test Purposes
- **Specification of Test Descriptions**
- Specification of Test Cases
- Validation



What is TDL? Part 4: TO

- Base Standard Specification
- Identification of Requirements
- Creation of ICS/IFS
- Definition of TSS
- Specification of Test Purposes
- Specification of Test Descriptions
- Specification of Test Cases
- Validation

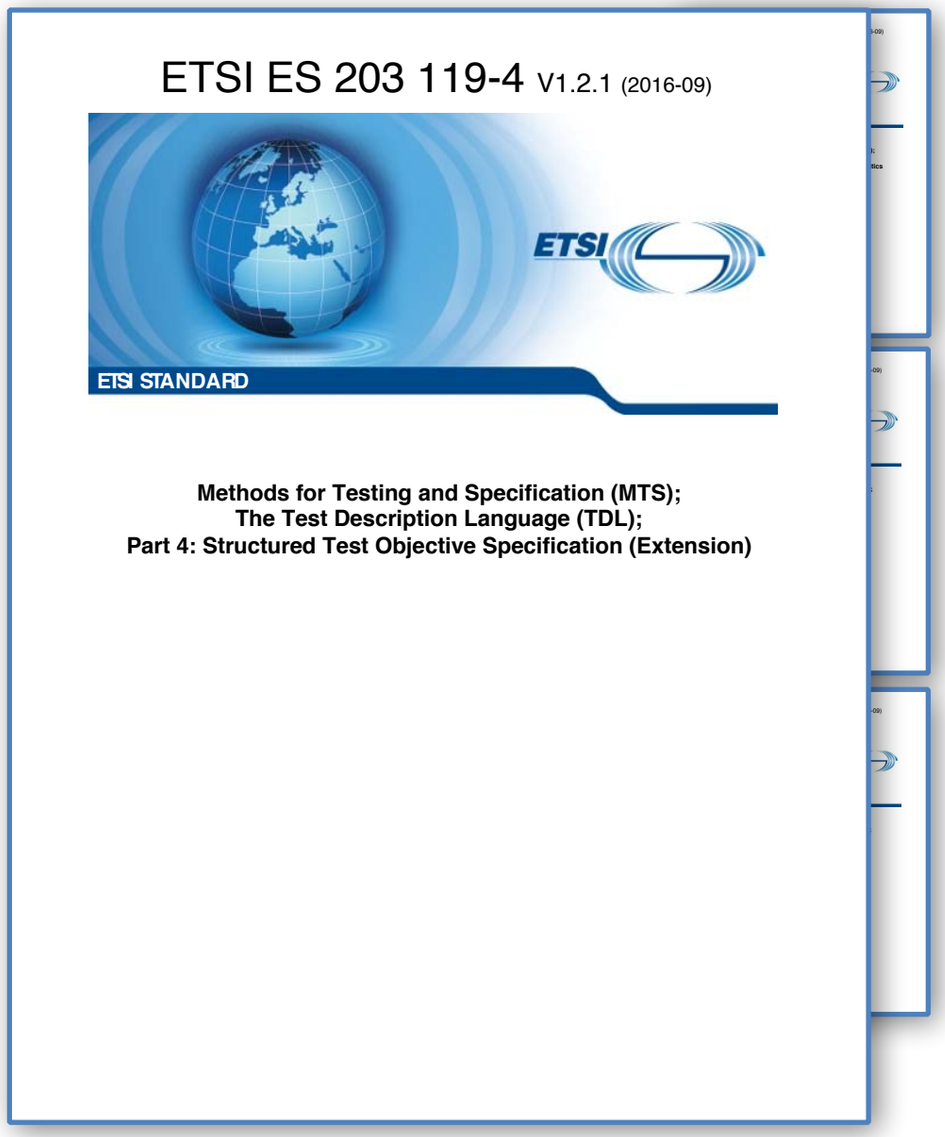

ETSI ES 203 119-4 V1.2.1 (2016-09)

ETSI STANDARD

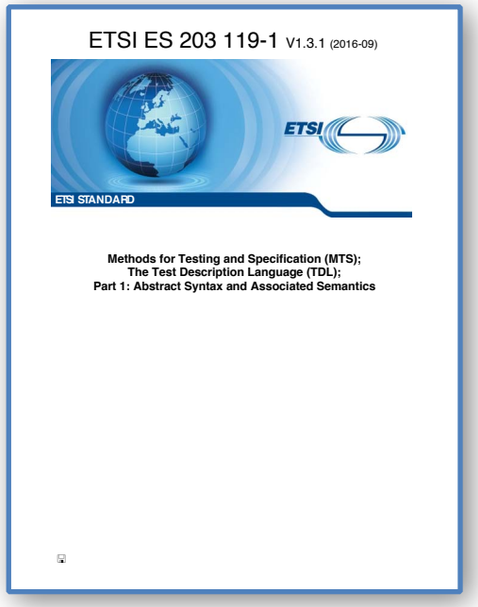
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

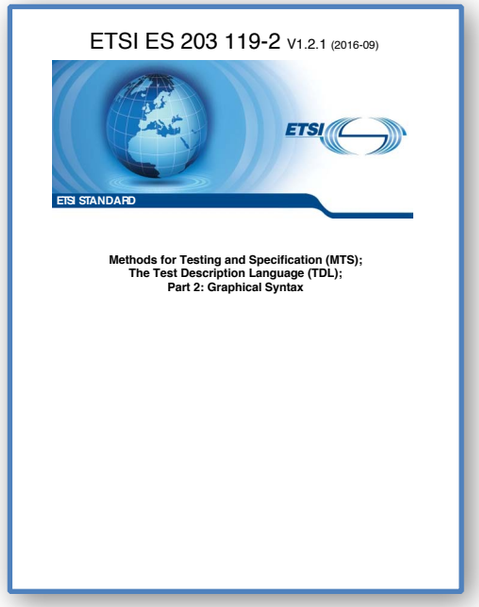
What is TDL? Part 4: TO

```
Test Purpose {
    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 entity having reached an initial_state
        and
        the IUT entity having sent a valid CAM message
            containing DoorOpen TaggedValue;
    }
    Expected behaviour
    ensure that {
        when {
            the door entity is closed
        }
        then {
            the IUT entity sends a new CAM message
                containing DoorOpen TaggedValue;
        }
    }
}
```



What is TDL?

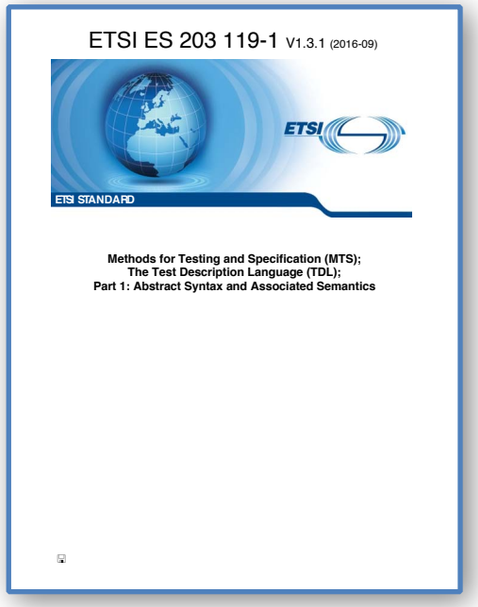

ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

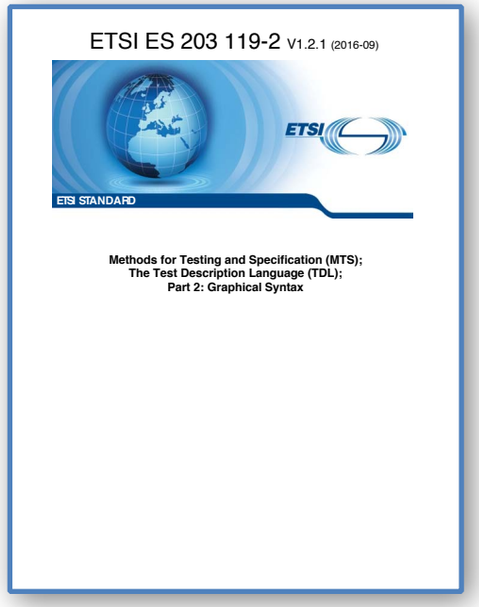

ETSI ES 203 119-2 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax


ETSI ES 203 119-3 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification(MTS);
The Test Description Language (TDL);
Part 3: Exchange Format


ETSI ES 203 119-4 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

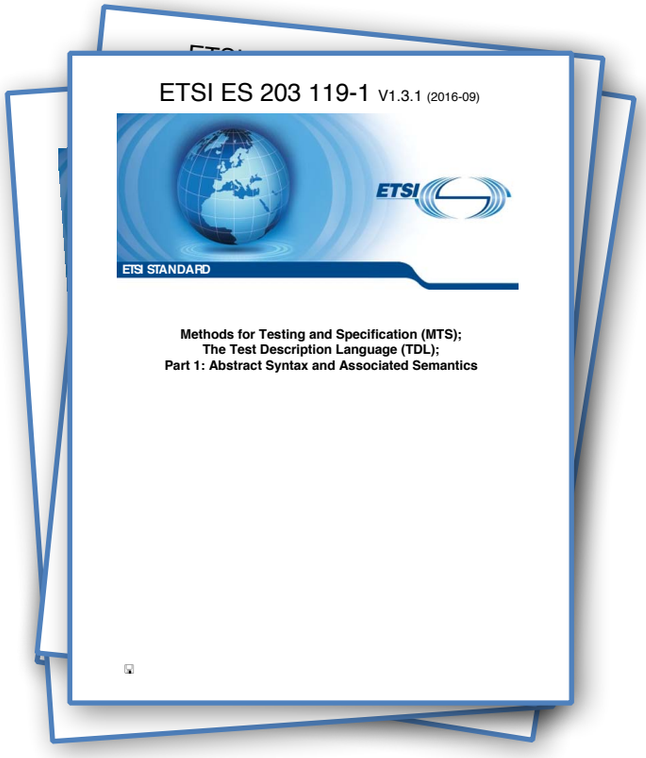
What is TDL?

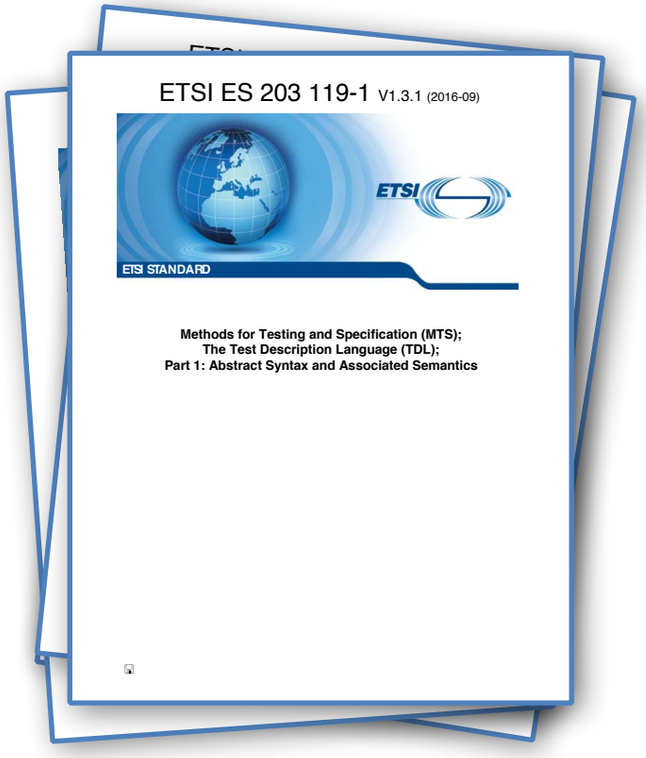

ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics


ETSI ES 203 119-2 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 2: Graphical Syntax


ETSI ES 203 119-3 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification(MTS);
The Test Description Language (TDL);
Part 3: Exchange Format

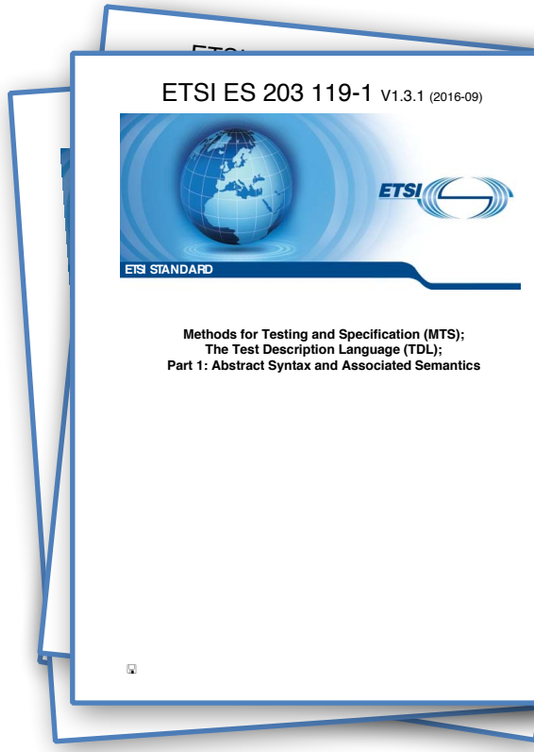

ETSI ES 203 119-4 V1.2.1 (2016-09)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)


ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics


ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

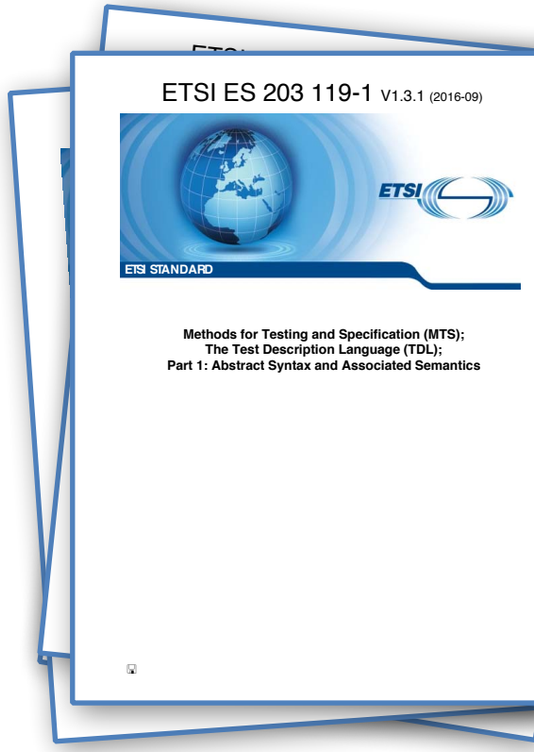
Why TDL?

- Model-based Testing
  - modelling features to describe and generate abstract tests
- Test automation
  - support for common testing patterns
  - clear semantics for generating concrete tests
- Agile Development
  - test-driven and behaviour-driven development
  - scenario-based testing from user stories
  - multiple representation formats for different stakeholders



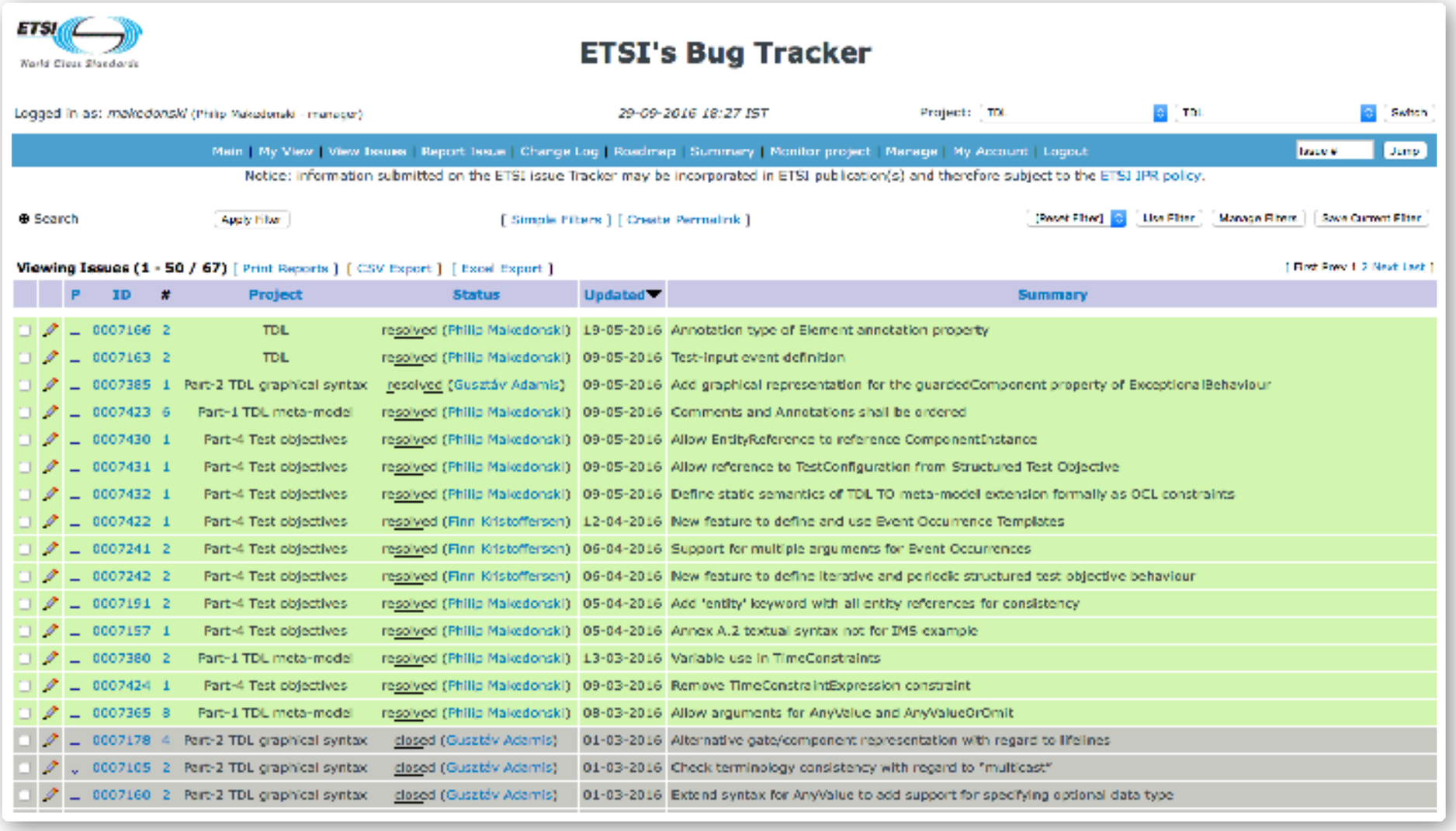
Why TDL?

- For users
  - separate test specification from test implementation
  - amenable to tool-supported verification
  - adjustable to stakeholders
  - focus on what to test vs how
- For tool vendors
  - universal standardised exchange format
  - support for customers from different domains
  - reuse of and integration with existing tools
  - focus on core expertise, add value through interoperability



Why TDL?

- Transparent change management



ETSI's Bug Tracker

Logged in as: makedonski (Philip Makedonski - manager) 29-09-2016 18:27 IST Project: TDL TDL Switch

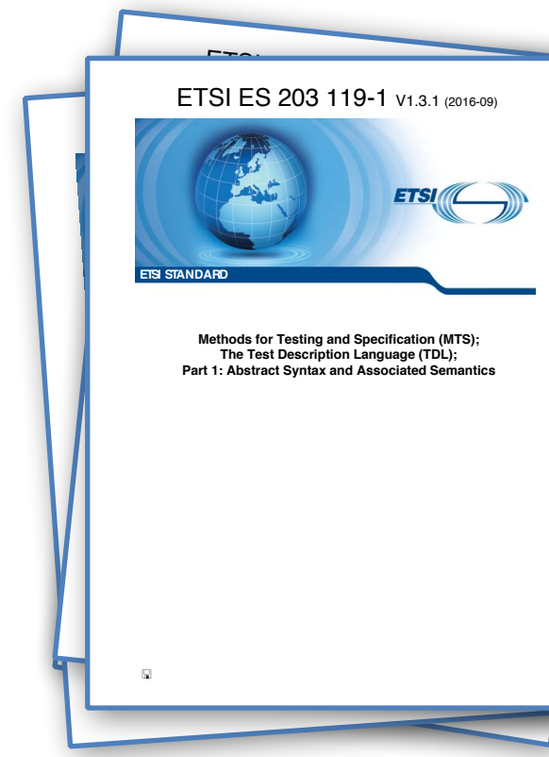
Main | My View | View Issues | Report Issue | Change Log | Roadmap | Summary | Monitor project | Manage | My Account | Logout Issue # Jump

Notice: Information submitted on the ETSI Issue Tracker may be incorporated in ETSI publication(s) and therefore subject to the ETSI IPR policy.

Search Apply Filter [Simple Filters] [Create Permalink] [Reset Filter] Use Filter Manage Filters Save Current Filter

Viewing Issues (1 - 50 / 67) [Print Reports] [CSV Export] [Excel Export] [First Prev 1 2 Next Last]

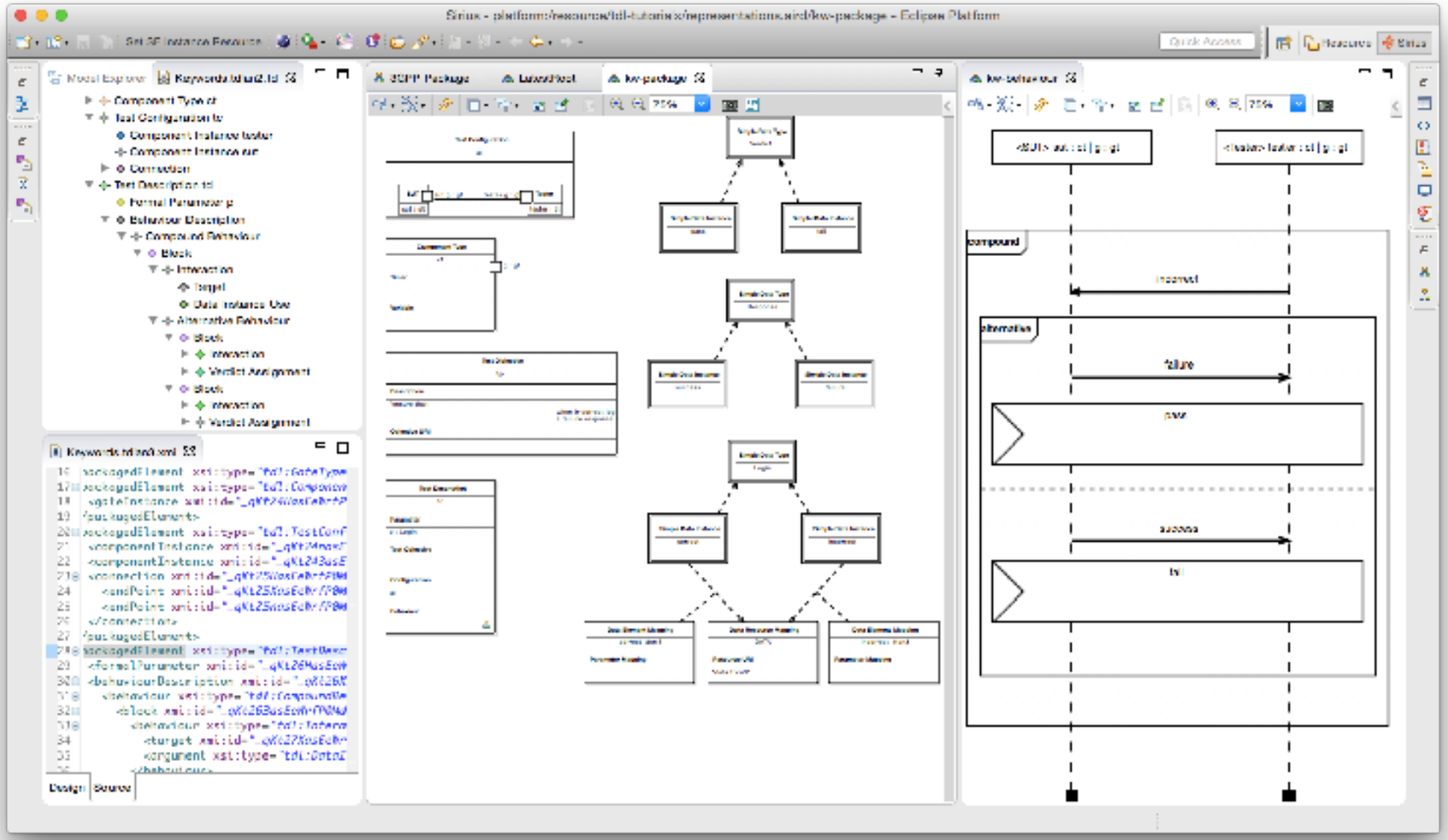
P	ID	#	Project	Status	Updated	Summary
_	0007166	2	TDL	resolved (Philip Makedonski)	19-05-2016	Annotation type of Element annotation property
_	0007163	2	TDL	resolved (Philip Makedonski)	09-05-2016	Test-input event definition
_	0007385	1	Part-2 TDL graphical syntax	resolved (Gusztáv Adamis)	09-05-2016	Add graphical representation for the guardedComponent property of ExceptionalBehaviour
_	0007423	6	Part-1 TDL meta-model	resolved (Philip Makedonski)	09-05-2016	Comments and Annotations shall be ordered
_	0007430	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Allow EntityReference to reference ComponentInstance
_	0007431	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Allow reference to TestConfiguration from Structured Test Objective
_	0007432	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Define static semantics of TDL TO meta-model extension formally as OCL constraints
_	0007422	1	Part-4 Test objectives	resolved (Finn Kristoffersen)	12-04-2016	New feature to define and use Event Occurrence Templates
_	0007241	2	Part-4 Test objectives	resolved (Finn Kristoffersen)	06-04-2016	Support for multiple arguments for Event Occurrences
_	0007242	2	Part-4 Test objectives	resolved (Finn Kristoffersen)	06-04-2016	New feature to define iterative and periodic structured test objective behaviour
_	0007191	2	Part-4 Test objectives	resolved (Philip Makedonski)	05-04-2016	Add 'entity' keyword with all entity references for consistency
_	0007157	1	Part-4 Test objectives	resolved (Philip Makedonski)	05-04-2016	Annex A.2 textual syntax not for IMS example
_	0007380	2	Part-1 TDL meta-model	resolved (Philip Makedonski)	13-03-2016	Variable use in TimeConstraints
_	0007424	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-03-2016	Remove TimeConstraintExpression constraint
_	0007365	8	Part-1 TDL meta-model	resolved (Philip Makedonski)	08-03-2016	Allow arguments for AnyValue and AnyValueOrOmit
_	0007178	4	Part-2 TDL graphical syntax	closed (Gusztáv Adamis)	01-03-2016	Alternative gate/component representation with regard to lifelines
_	0007105	2	Part-2 TDL graphical syntax	closed (Gusztáv Adamis)	01-03-2016	Check terminology consistency with regard to "multicast"
_	0007160	2	Part-2 TDL graphical syntax	closed (Gusztáv Adamis)	01-03-2016	Extend syntax for AnyValue to add support for specifying optional data type

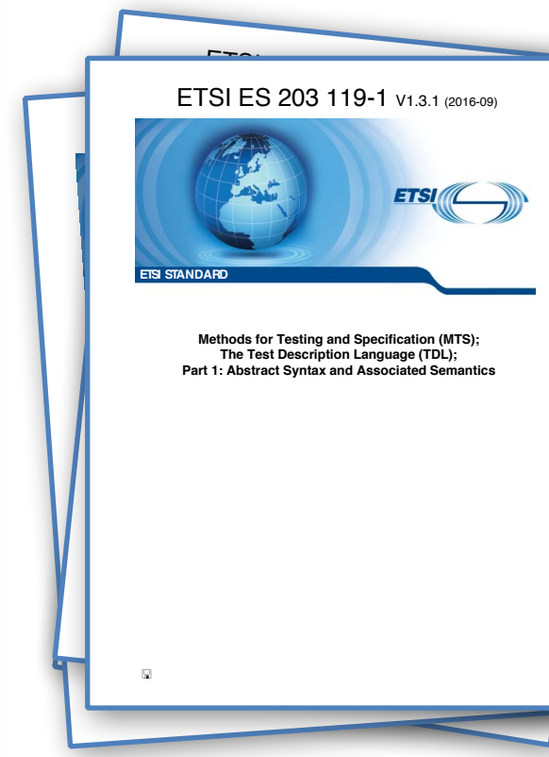


TDL Project at http://forge.etsi.org/mantis/view_all_bug_page.php

Why TDL?

- Proof by implementation

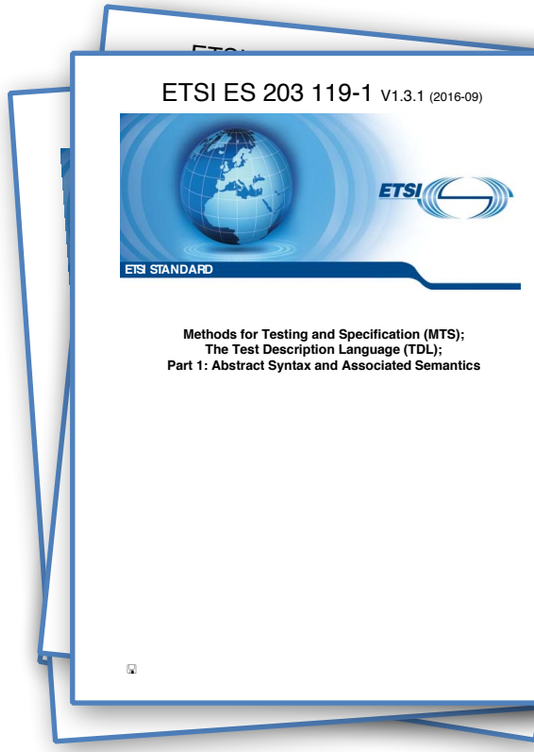




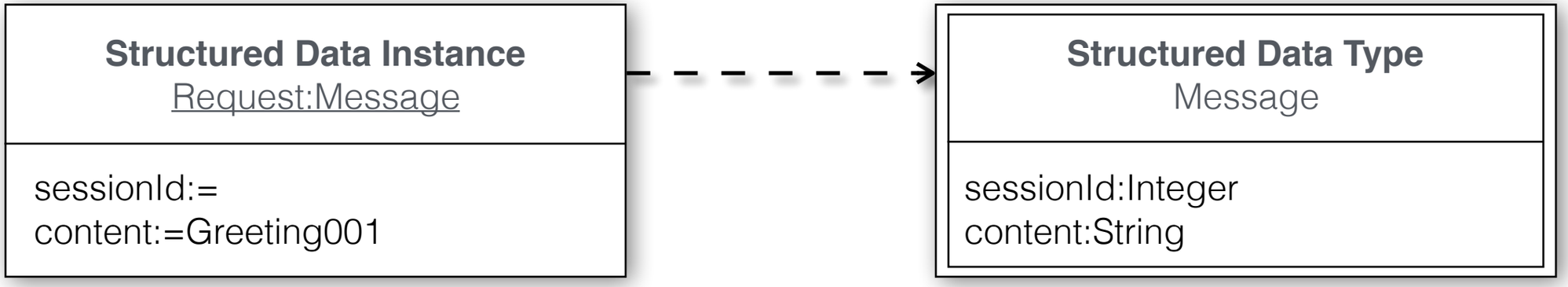
TDL Website at tdl.etsi.org

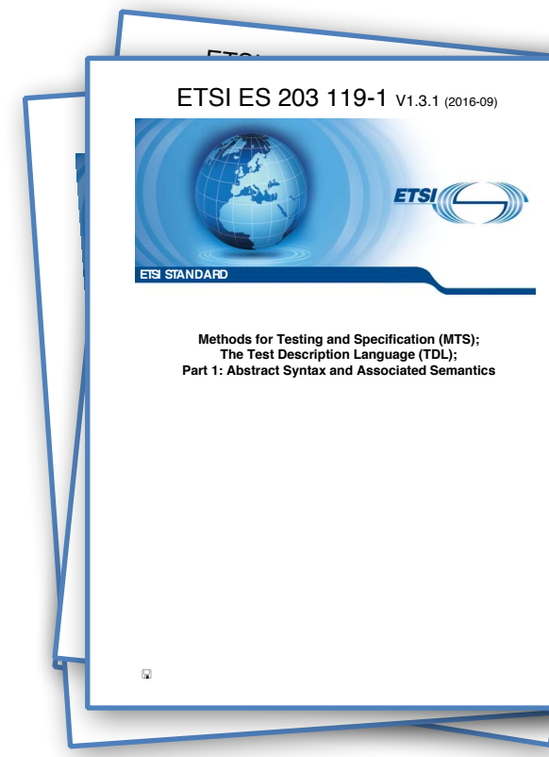
Why TDL?

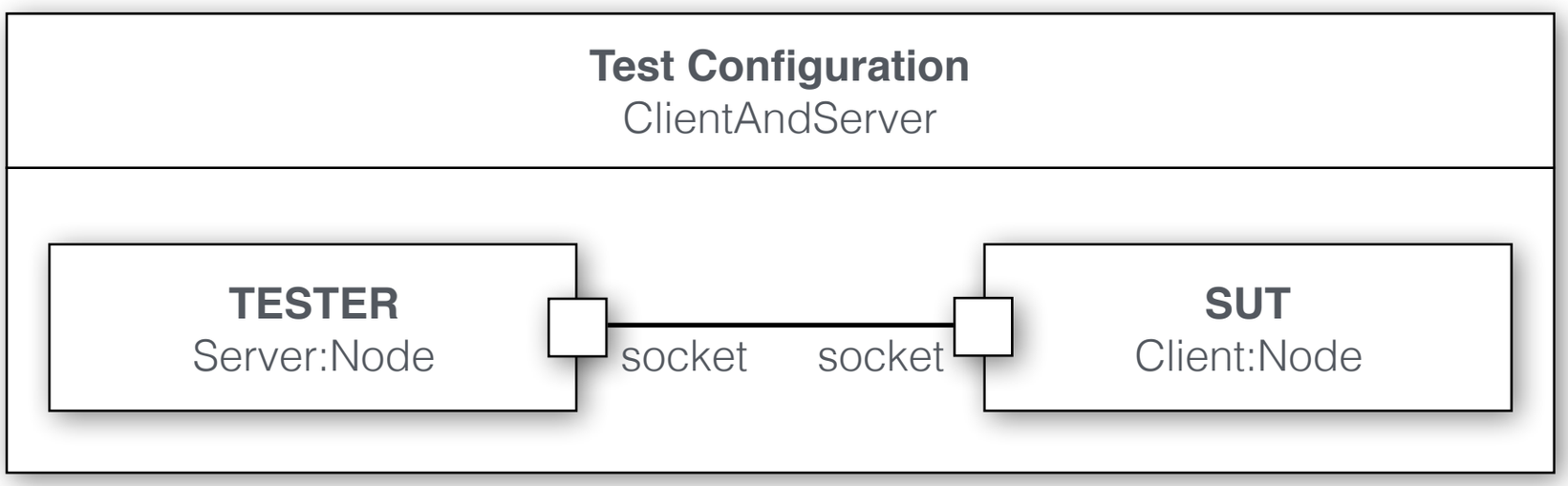
- Why not UTP?
  - requires knowledge and understanding of UML
  - loose semantics inherited from UML
  - tool-specific implementations
  - poor transferability as a consequence
  - additional profiles for timing aspects



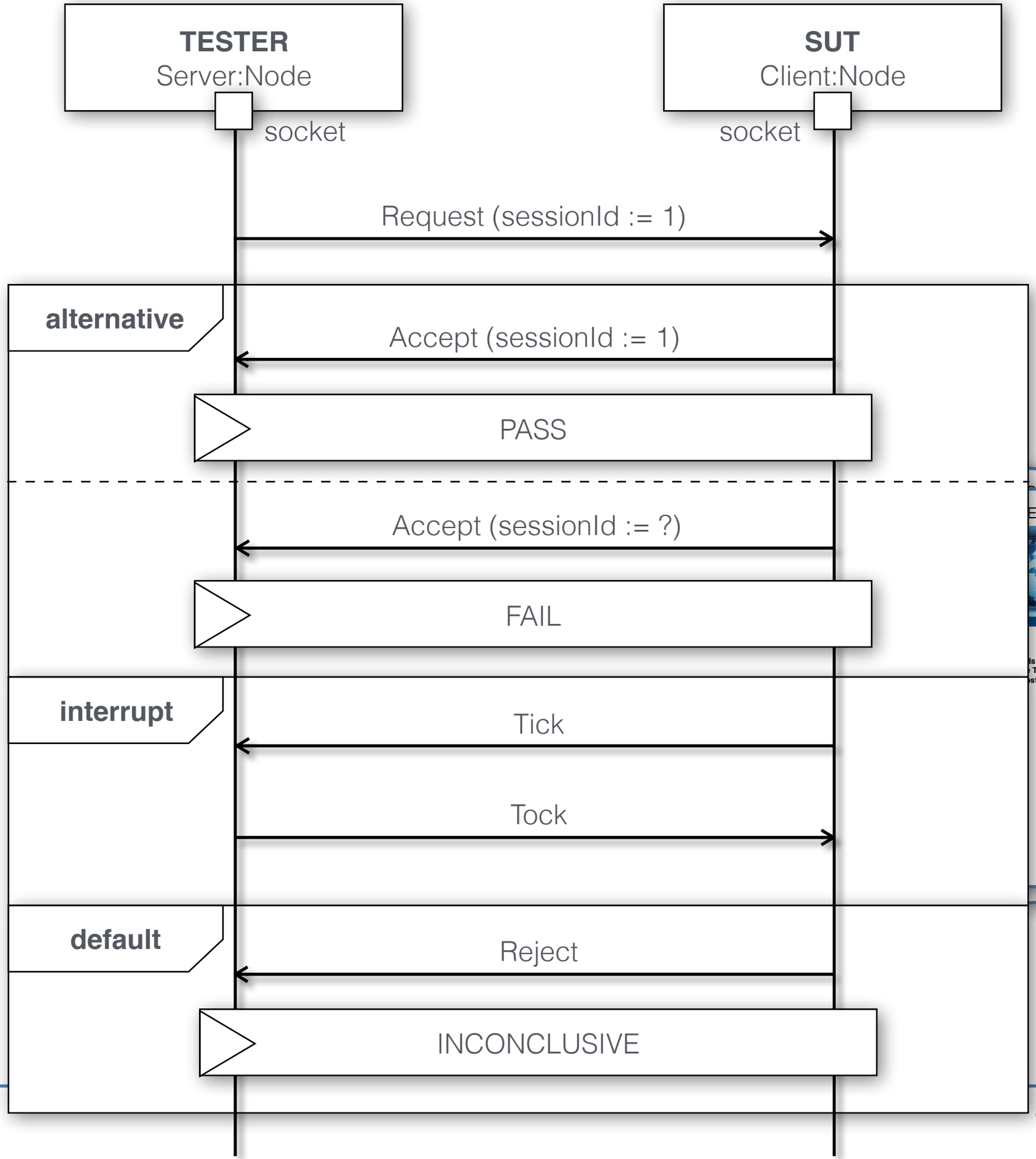
A more comprehensive example…

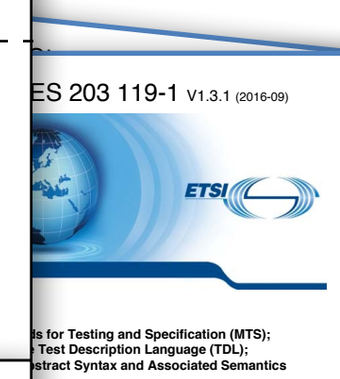


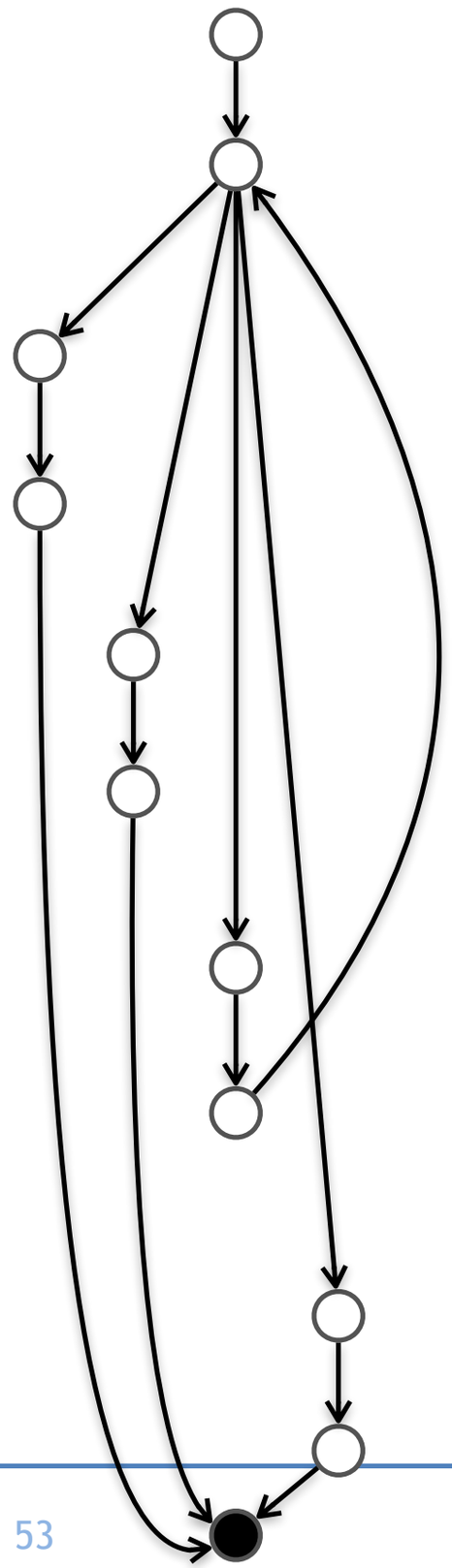


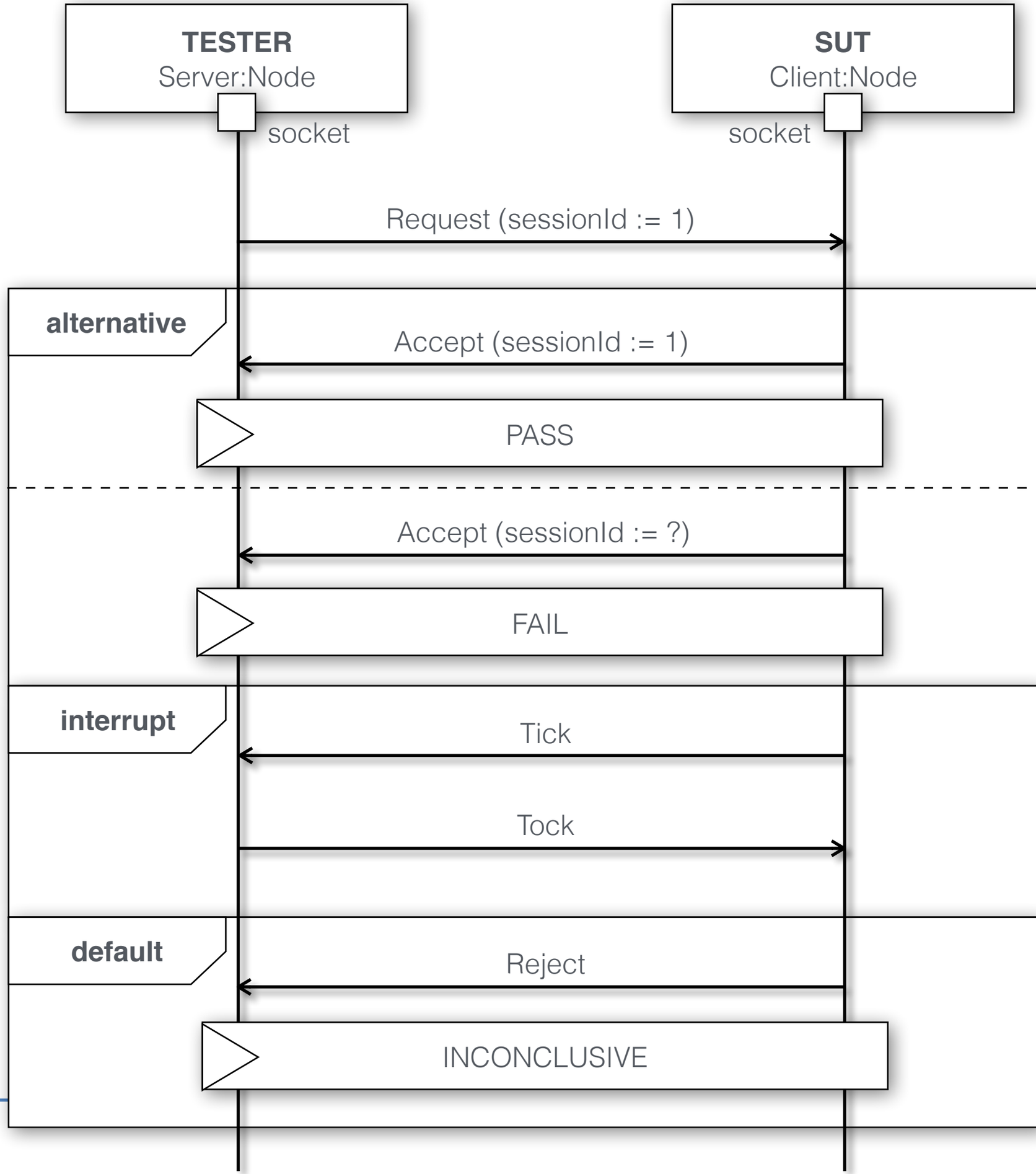


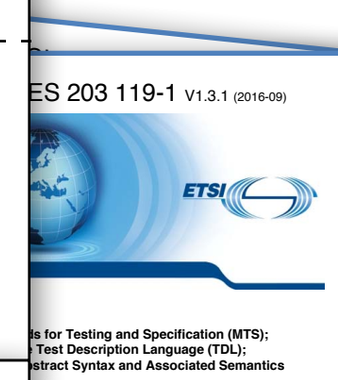
- Atomic
- Combined
- Exceptional

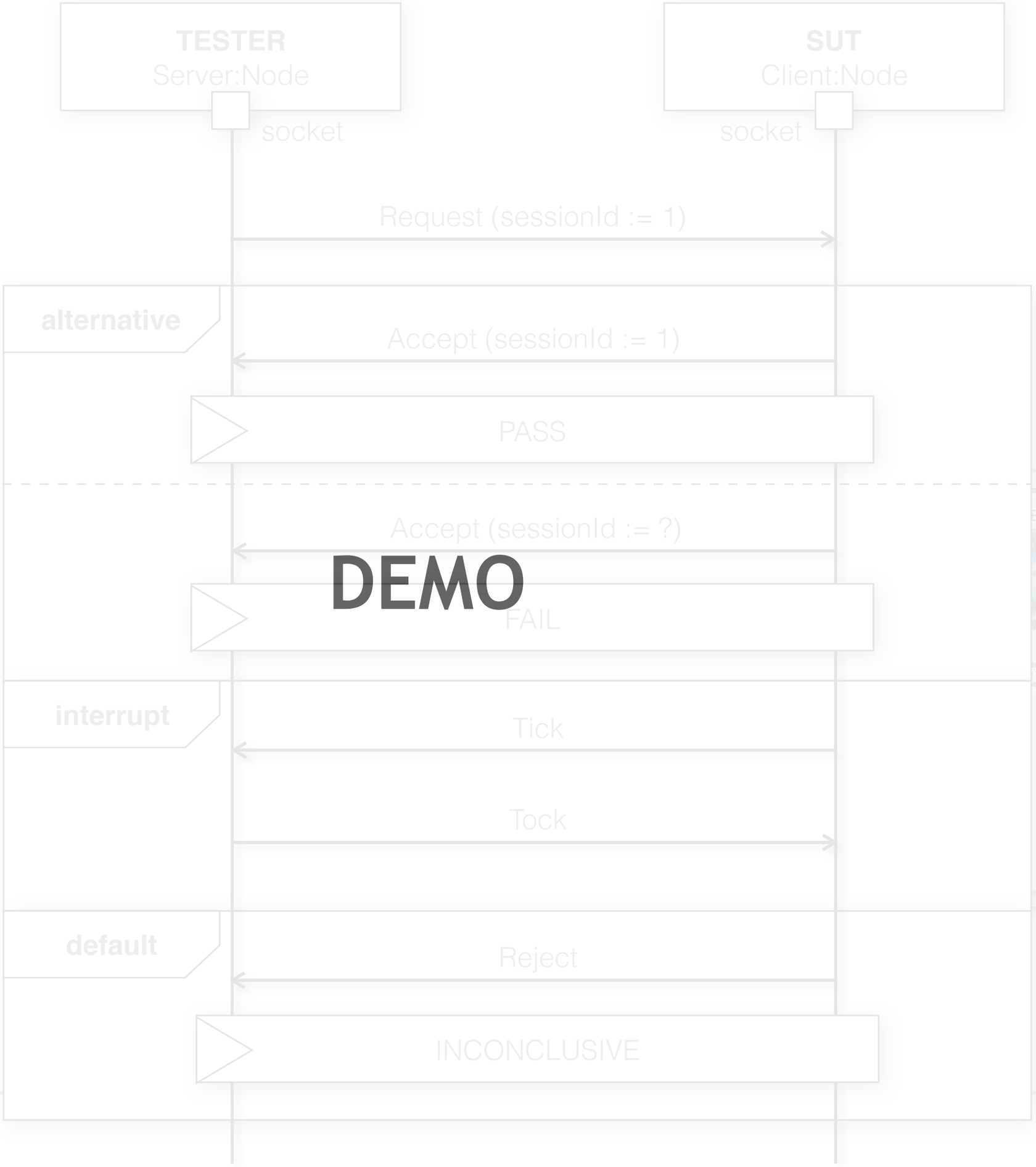




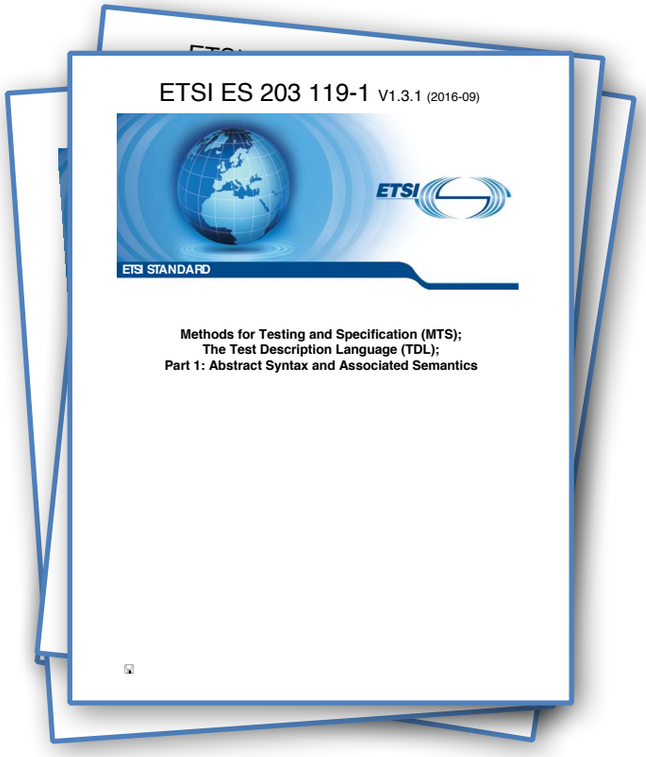



TESTER
Server:Node
socket
SUT
Client:Node
socket
Request (sessionId := 1)
alternative
Accept (sessionId := 1)
PASS
Accept (sessionId := ?)
FAIL
interrupt
Tick
Tock
default
Reject
INCONCLUSIVE

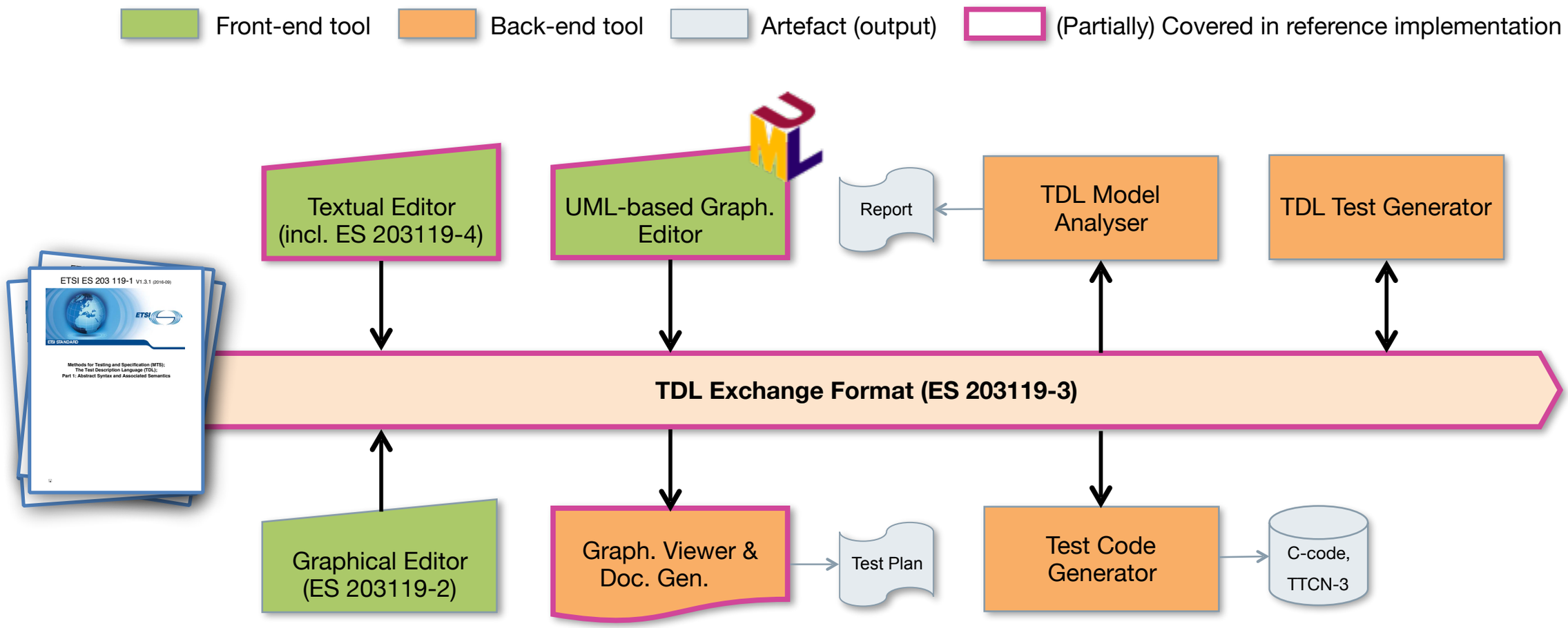

ES 203 119-1 V1.3.1 (2016-09)
ETSI
is for Testing and Specification (MTS);
Test Description Language (TDL);
stract Syntax and Associated Semantics


TESTER
Server:Node
SUT
Client:Node
socket
socket
Request (sessionId := 1)
alternative
Accept (sessionId := 1)
PASS
Accept (sessionId := ?)
FAIL
interrupt
Tick
Tock
default
Reject
INCONCLUSIVE

DEMO


ETSI ES 203 119-1 V1.3.1 (2016-09)
ETSI
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

Where does TDL fit in?



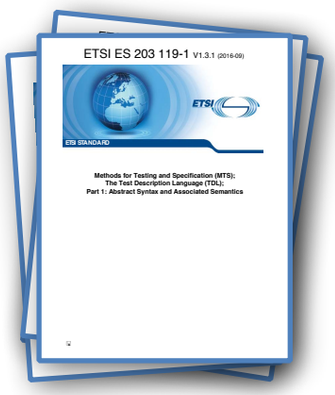
Where does TDL fit in?

Keyword-Driven Testing

MBT

Representation

Generation



Visualisation

Standards

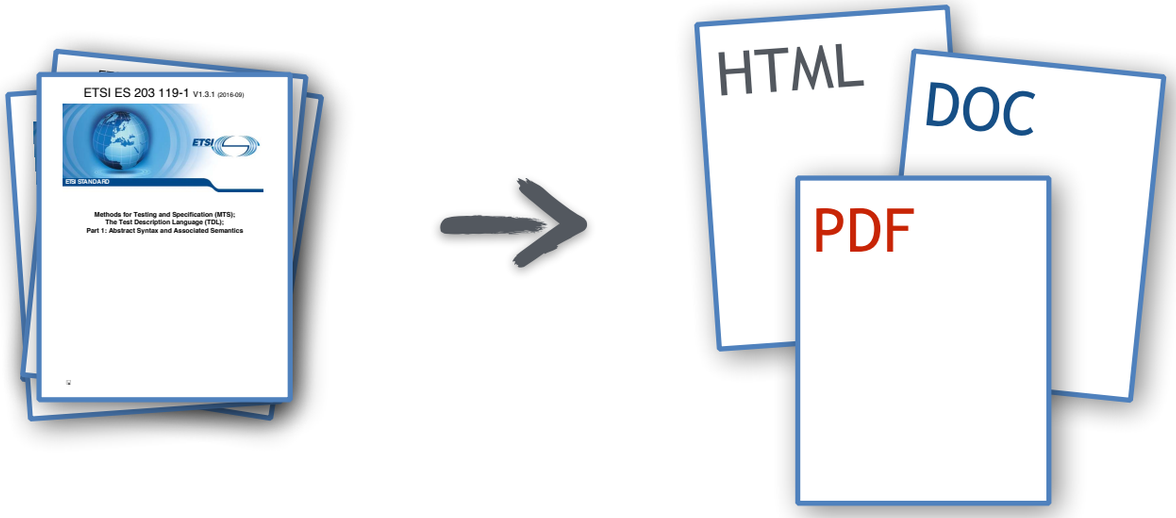
Documentation

ITS

Interoperability

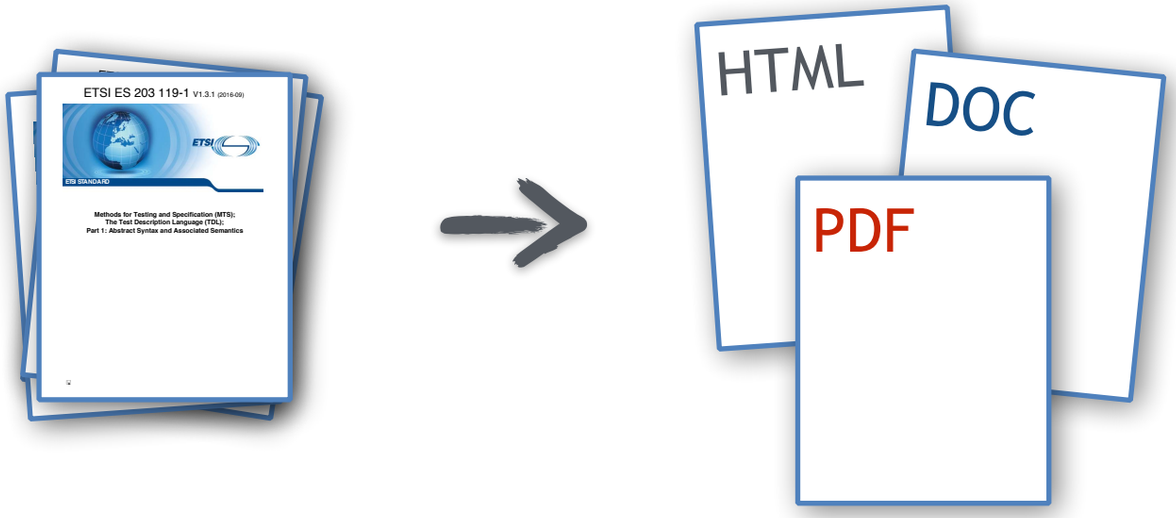
Conformance

Where does TDL fit in?



Documentation

Where does TDL fit in?



- Context
  - Conformance and interoperability test descriptions
  - Standardised test specifications for various ETSI technologies
  - Typically protocol oriented, used in certification schemes
  - End-to-end interoperability of systems involving different equipment

- Stakeholders
  - High-level discussions at large meetings (80-100 participants)
    - ETSI Technical Committees, 3GPP, other standards organisations, CTI Plugtests team and participants, industrial fora and equipment vendors
    - all need to be familiar with and fluent in the syntax being used
    - different notions of “good” test
  - Better comprehension among developers with little or no testing expertise
    - bridge the gap between management, core specifications experts, testing experts

- Challenges
  - Informal (Word, Excel) or semi-formal (TPLan) approaches
    - considered inadequate, no test descriptions as a consequence
    - no single consistent approach, varying level of quality, detail, difficult maintenance
    - certification requires completeness and accuracy, test descriptions are the design stage before developing TTCN-3 test cases
  - Acceptance for more rigorous approaches among Technical Committees
    - applicable to a wide range of technologies (protocols, services, applications)

- TDL
  - Standardised approach improves consistency
  - Tools offer faster development, higher quality, easier maintenance
  - Direct link to TTCN-3
- Initial run within ITS, expand to other Technical Committees

From 3GPP TS 36.523-1 V10.2.0 (2012-09):

7.2.2.3 UM RLC / Reassembly / 5-bit SN / LI value > PDU size

7.2.2.3.1 Test Purpose (TP)

(1)

```
with { UE in E-UTRA RRC_CONNECTED state }
ensure that {
  when { UE receives a 5 bit SN configured RLC PDU with Length Indicator value larger than RLC PDU
size }
    then { UE discards the RLC PDU }
            }
```

7.2.2.3.3.2 Test procedure sequence

Table 7.2.2.3.3.2-1: Main behaviour

St	Procedure	Message Sequence		TP	Verdict
		U - S	Message		
-	EXCEPTION: the behaviour described in table 7.2.2.3.3.2-2 runs in parallel with steps 1 to 5 below.	-	-	-	-
1	The SS transmits UMD PDU#1 containing first segment of RLC SDU#1.	<--	UMD PDU#1 (SN=0)	-	-
2	The SS transmits UMD PDU#2 containing last segment of RLC SDU#1 and first segment of RLC SDU#2.	<--	UMD PDU#2 (SN=1)	-	-
3	The SS transmits UMD PDU#3 containing last segment of RLC SDU#2, first segment of RLC SDU#3 and with Length Indicator that points beyond the end of the UMD PDU#3.	<--	UMD PDU#3 (SN=2)	-	-
4	The SS transmits UMD PDU#4 containing last segment of RLC SDU#3.	<--	UMD PDU#4 (SN=3)	-	-
5	The SS transmits UMD PDU#5 containing RLC SDU#4.	<--	UMD PDU#5 (SN=4)	-	-

From 3GPP TS 36.523-1 V10.2.0 (2012-09):

7.2.2.3 UM RLC / Reassembly / 5-bit SN / LI value > PDU size

7.2.2.3.1 Test Purpose (TP)

(1)

```
with { UE in E-UTRA RRC_CONNECTED state }
ensure that {
  when { UE receives a 5 bit SN configured RLC PDU with Length Indicator value larger than RLC PDU
size }
    then { UE discards the RLC PDU }
            }
```

7.2.2.3.3.2 Test procedure sequence

Table 7.2.2.3.3.2-1: Main behaviour

St	Procedure	Message Sequence		TP	Verdict
		U - S	Message		
-	EXCEPTION: the behaviour described in table 7.2.2.3.3.2-2 runs in parallel with steps 1 to 5 below.	-	-	-	-
1	The SS transmits UMD PDU#1 containing first segment of RLC SDU#1.	<--	UMD PDU#1 (SN=0)	-	-
2	The SS transmits UMD PDU#2 containing last segment of RLC SDU#1 and first segment of RLC SDU#2.	<--	UMD PDU#2 (SN=1)	-	-
3	The SS transmits UMD PDU#3 containing last segment of RLC SDU#2, first segment of RLC SDU#3 and with Length Indicator that points beyond the end of the UMD PDU#3.	<--	UMD PDU#3 (SN=2)	-	-
4	The SS transmits UMD PDU#4 containing last segment of RLC SDU#3.	<--	UMD PDU#4 (SN=3)	-	-
5	The SS transmits UMD PDU#5 containing RLC SDU#4.	<--	UMD PDU#5 (SN=4)	-	-

Table 7.2.2.3.3.2-2: Parallel behaviour

St	Procedure	Message Sequence		TP	Verdict
		U - S	Message		
1	The UE transmits RLC SDU#1.	-->	(RLC SDU#1)	-	-
2	Check: Does the UE transmit RLC SDU#2?	-->	(RLC SDU#2)	1	F
3	Check: Does the UE transmit RLC SDU#3?	-->	(RLC SDU#3)	1	F
4	The UE transmits RLC SDU#4.	-->	(RLC SDU#4)	-	-

From 3GPP TS 36.523-1 V10.2.0 (2012-09):

Resource - tdl-scopes/3GPP-7.1.3.tdlan2.tdl - Eclipse Platform

3GPP-7.1.3.tdlan2

```
92      //Test description definition
93      Test Description TD_7_1_3_1 uses configuration defaultTC {
94          //Pre-conditions and preamble from the source document
95          perform action preCondition with { PRECONDITION ; } ;
96          perform action preamble  with { PREAMBLE ; } ;
97
98          //Test sequence
99          SS.g sends pdcch (c_rnti=ue) to UE.g with {
100             STEP : "1" ;
101             PROCEDURE : "SS transmits a downlink assignment
102                          including the C-RNTI assigned to
103                          the UE" ;
104         } ;
105         SS.g sends mac_pdu to UE.g with {
111         UE.g sends harq_ack to SS.g with {
117         set verdict to PASS ;
118         SS.g sends pdcch (c_rnti=unknown) to UE.g with {
124         SS.g sends mac_pdu to UE.g with {
```

3GPP-7.1.3.tdlan2.tdl

- Simple Data Type C_RNTI
- Simple Data Instance ue
- Simple Data Instance unknown
- Structured Data Type PDCCH
- Simple Data Instance pdcch
- Simple Data Type CONFIGURATION
- Simple Data Instance RRCConnectionReconfiguration
- Time SECONDS
- Time MS
- Simple Data Instance five
- Gate Type defaultGT
- Component Type defaultCT
- Test Configuration defaultTC
- Test Description TD_7_1_3_1
  - Behaviour Description
    - Compound Behaviour
      - Comment
      - Block
        - Action Reference
        - Action Reference
        - Interaction
          - Target
            - Annotation
            - Annotation
          - Data Instance Use
            - Parameter Binding
              - Data Instance Use
        - Interaction

History | Properties

Property	Value
Annotated Element	Target
Key	Annotation Type PROCEDURE
Name	
Value	"SS transmits a downlink assignment ...

Selected Object: Annotation

From ETSI TS 186 011-2 V3.1.1 (2011-06):

4.5.1 General Capabilities

4.5.1.1 SIP messages longer than 1 500 bytes

Interoperability Test Description		
Identifier:	TD_IMS_MESS_0001	
Summary:	IMS network shall support SIP messages greater than 1 500 bytes	
Configuration:	CF_INT_CALL	
SUT	IMS_B	
References	**Test Purpose**	**Specification Reference**
	TP_IMS_4002_1	TS 124 229 [1], clause 4.2A ¶1
Use Case ref.:	UC_05_I	
Pre-test conditions:	• HSS of IMS_A and of IMS B is configured according to table 1 • UE_A and UE_B have IP bearers established to their respective IMS networks as per clause 4.2.1 • UE_A and IMS_A configured to use TCP for transport • UE_A is registered in IMS_A using any user identity • UE_B is registered user of IMS_B using any user identity • MESSAGE request and response has to be supported at II-NNI (TS 129 165 [16] see tables 6.1 and 6.3)	
Test Sequence:	**Step**	
	1	User A sends message to User B with at least 1 500 characters
	2	Verify that user B receives message from user A
Conformance Criteria:	**Check**	
	1	TP_IMS_4002_01 in CFW step 4 (MESSAGE) *ensure that {* *when { UE_A sends a MESSAGE to UE_B* *containing a Message_Body greater than 1 300 bytes }* *then { IMS_B receives the MESSAGE* *containing the Message_Body greater than 1 300 bytes }* *}*

Step	Direction								Message	Comment
	User A	UE A	IMS A	IBCF A	IBCF B	IMS B	UE B	User B		
1		→								User A sends an instant message to user B
2			→						MESSAGE	UE_A sends MESSAGE to IMS_A
3				→					**MESSAGE**	**IMS_A sends MESSAGE to IBCF_A**

From ETSI TS 186 011-2 V3.1.1 (2011-06):

4.5.1 General Capabilities

4.5.1.1 SIP messages longer than 1 500 bytes

Interoperability Test Description		
Identifier:	TD_IMS_MESS_0001	
Summary:	IMS network shall support SIP messages greater than 1 500 bytes	
Configuration:	CF_INT_CALL	
SUT	IMS_B	
References	**Test Purpose**	**Specification Reference**
	TP_IMS_4002_1	TS 124 229 [1], clause 4.2A ¶1

Step	Direction								Message	Comment
	User A	UE A	IMS A	IBCF A	IBCF B	IMS B	UE B	User B		
1	→									User A sends an instant message to user B
2		→							MESSAGE	UE_A sends MESSAGE to IMS_A
3			→						**MESSAGE**	**IMS_A sends MESSAGE to IBCF_A**
4				→					**MESSAGE**	**IBCF_A sends MESSAGE to IBCF_B**
5					→				**MESSAGE**	**IBCF_B sends MESSAGE to IMS_B with via header indicating TCP**
6						→			MESSAGE	IMS_B sends MESSAGE to UE_B
7							→			User B is informed about the instant message
8						←			200 OK	UE_B sends 200 OK to IMS_B
9					←				200 OK	IMS_B sends 200 OK to IBCF_B
10				←					200 OK	IBCF_B sends 200 OK to IBCF_A
11			←						200 OK	IBCF_A sends 200 OK to IMS_A
12		←							200 OK	IMS_A sends 200 OK to UE_A
13	←									Optional: User A is presented a delivery report

From ETSI TS 186 011-2 V3.1.1 (2011-06):

Resource - tcl-examples/IMS-4.5.1.tdlan2.tcl - Eclipse Platform

3GPP-7.1.3.tdlan2 | IMS-4.5.1.tdlan2

```
99      //Test description definition
100     Test Description TD_IMS_MESS_0001 uses configuration CF_INT_CALL {
101             //Pre-conditions from the source document
102             perform action preConditions with { PRECONDITION ; };
103
104             //Test sequence
105             USER_A.g sends MESSAGE to UE_A.g with { STEP : "1" ; } ;
106             UE_A.g sends MESSAGE to IMS_A.g with { STEP : "2" ; } ;
107             IMS_A.g sends MESSAGE to IBCF_A.g with { STEP : "3" ; } ;
108             IBCF_A.g sends MESSAGE to IBCF_B.g with { STEP : "4" ; } ;
109             IBCF_B.g sends MESSAGE (TCP = tcp) to IMS_B.g with { STEP : "5" ; } ;
110             IMS_B.g sends MESSAGE to UE_B.g with { STEP : "6" ; } ;
111             UE_B.g sends DING to USER_B.g with { STEP : "7" ; } ;
112             UE_B.g sends M_200_OK to IMS_B.g with { STEP : "8" ; } ;
113             IMS_B.g sends M_200_OK to IBCF_B.g with { STEP : "9" ; } ;
114             IBCF_B.g sends M_200_OK to IBCF_A.g with { STEP : "10" ; } ;
115             IBCF_A.g sends M_200_OK to IMS_A.g with { STEP : "11" ; } ;
116             IMS_A.g sends M_200_OK to UE_A.g with { STEP : "12" ; } ;
117             alternatively {
```

3GPP-7.1.3.tdlan2.tcl | IMS-4.5.1.tdlan2.tcl

- Simple Data Instance top
- Time seconds
- Simple Data Instance default_timeout
- Gate Type defaultGT
- Component Type USER
- Component Type UE
- Component Type IMS
- Component Type IBCF
- Test Configuration CF_INT_CALL
- Test Description TD_IMS_MESS_0001
  - Behaviour Description
    - Compound Behaviour
      - Annotation
      - Block
        - Action Reference
        - Interaction
          - Target
            - Annotation
            - Data Instance Use
        - Interaction
        - Interaction
        - Interaction
        - Interaction
        - Interaction
        - Interaction
        - Interaction
        - Interaction

History | Properties

Property	Value
Annotated Element	Target
Key	Annotation Type STEP
Name	
Value	"1"

Selected Object: Annotation

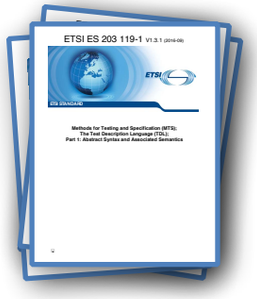
From ETSI TS 102 868-2 V1.1.1 (2011-03):

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	

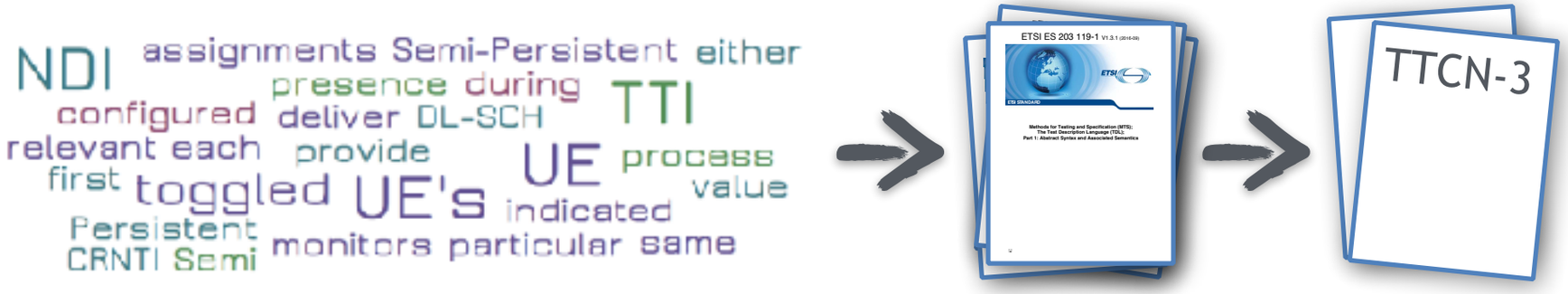
with {
the IUT being in the "initial state" and
the IUT having sent a valid CAM message
containing DoorOpen TaggedValue
}	

Expected behaviour	

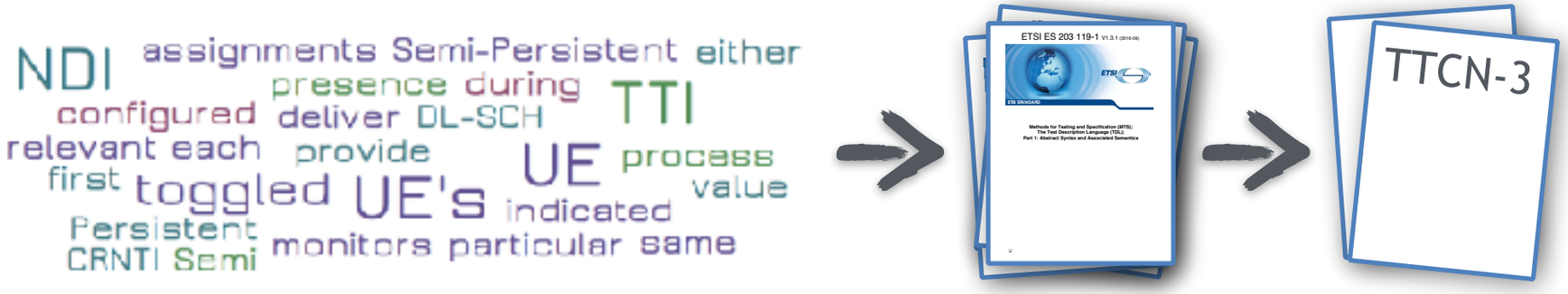
ensure that {
when {
the door is closed
}
then {
the IUT sends CAM messages
containing DoorOpen TaggedValue during the 30s following the door closing event
}
}	



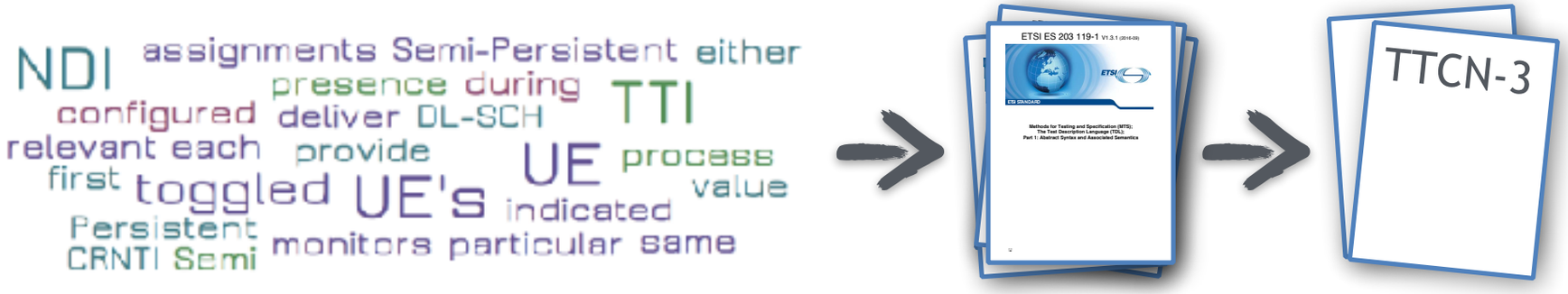
Keyword-Driven Testing



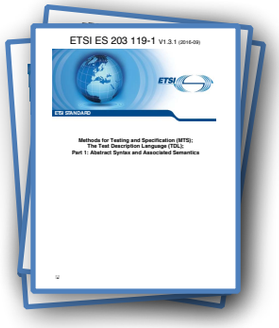
- Context
  - TDL in MBT: Keyword driven UI testing
  - Create behavioural model of the SUT using symbolic action descriptions
    - define keywords once
    - map abstract keyword definitions to keyword implementations in execution language
  - Generate abstract test sequences by means of MBT
  - Convert abstract test sequences to a test execution language



- Challenges
  - Generated test sequences
    - proprietary format - not accessible, tool-specific integrations to requirements management, test planning
    - straight to executable code - loss of meta-data, difficult parameterisation
  - Mapping between abstract (symbolic) and real test system interface
    - implicit - error-prone
    - implemented in test execution language - additional overhead, language limitations



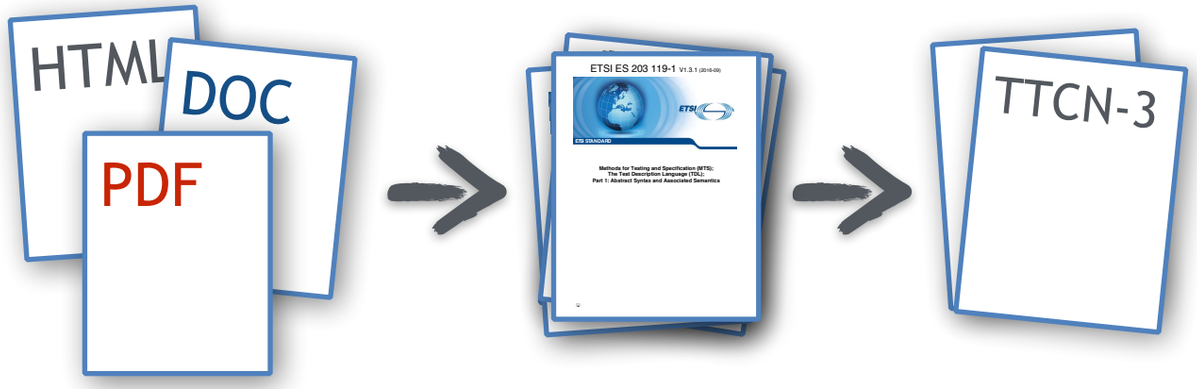
- TDL
  - Interoperability with requirements management by explicit test objectives
  - Parameterisation of test descriptions and symbolic data representations
  - Explicit data mapping to underlying data system of execution language
- Advantages over alternatives
  - Less ambiguity, testing specific (e.g. break, stop, default concepts)



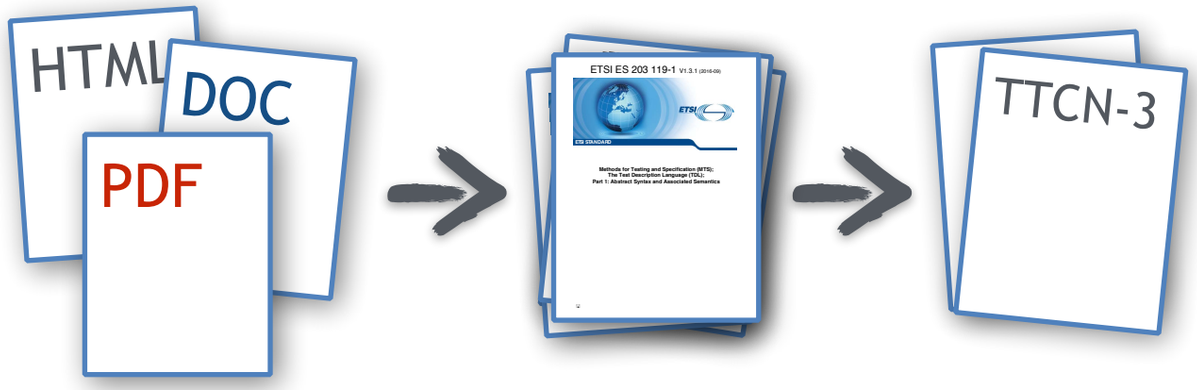
Representation

Generation

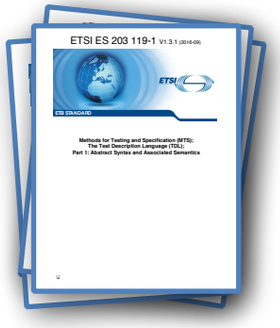
Visualisation



- Context
  - Test automation tools for performance and load tests
- Challenges
  - Textual test specifications with sequence diagram-like examples (or using a different graphical notation)
  - Manual derivation of TTCN-3 code and configuration settings
  - Too wide a gap between input and output!



- TDL
  - Raises the abstraction level of the test description
    - multiple levels of test specification (from system to implementation), iterative and agile development
  - Concentrate on the problems themselves rather than programming details
- Application
  - Visualisation of test case behaviour
  - Automatic generation of TTCN-3 code from TDL test descriptions



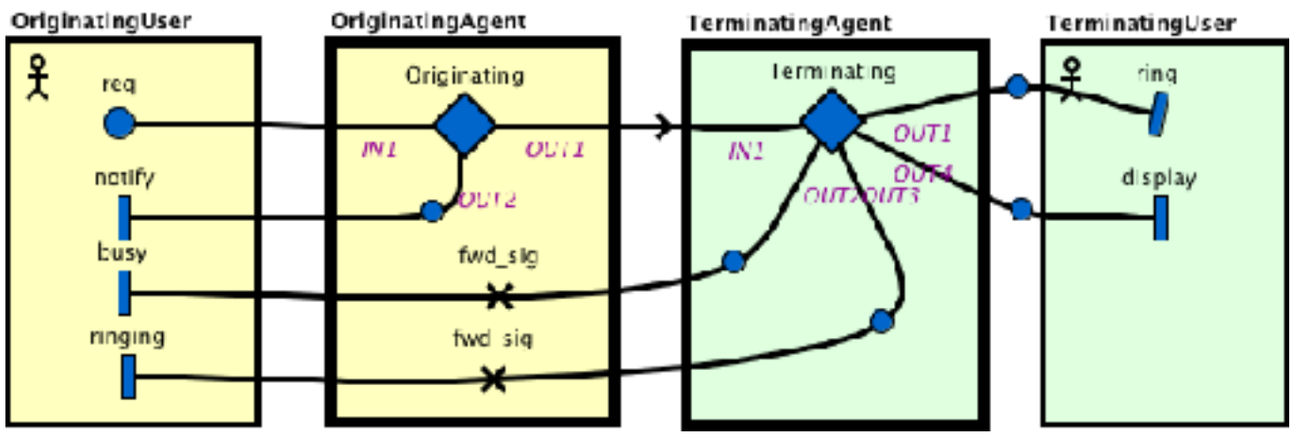
MBT

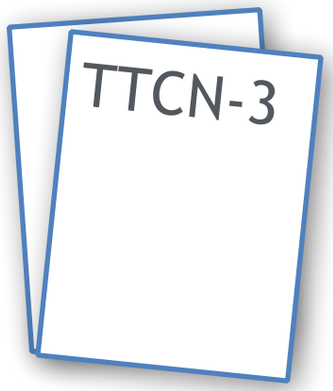
Representation

Generation

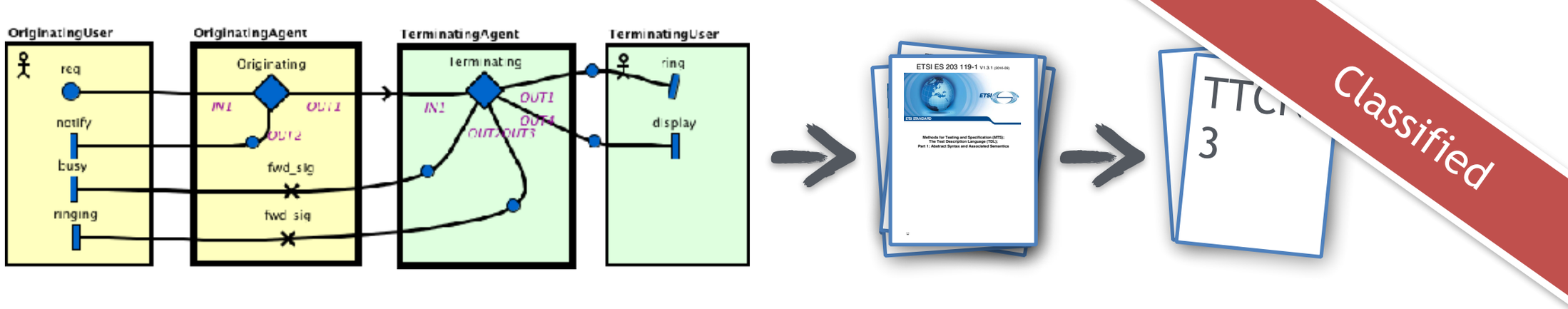
Standards

Where does TDL fit in?

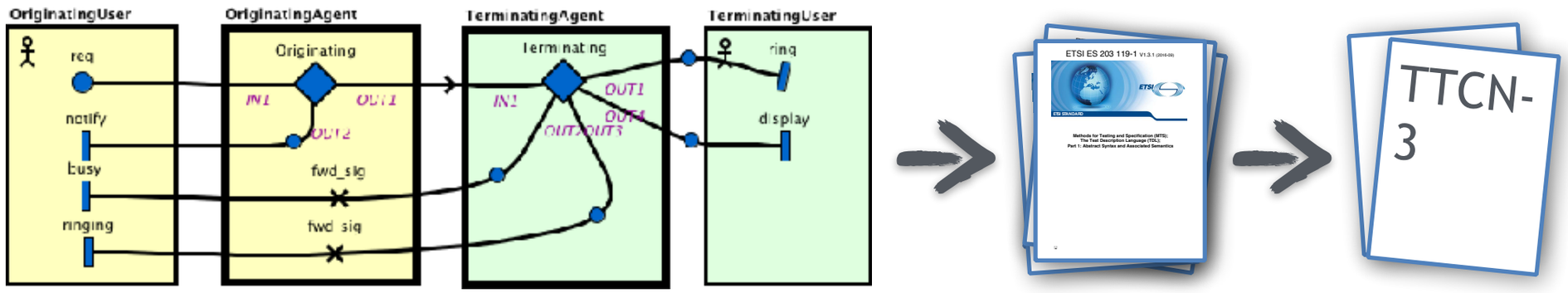




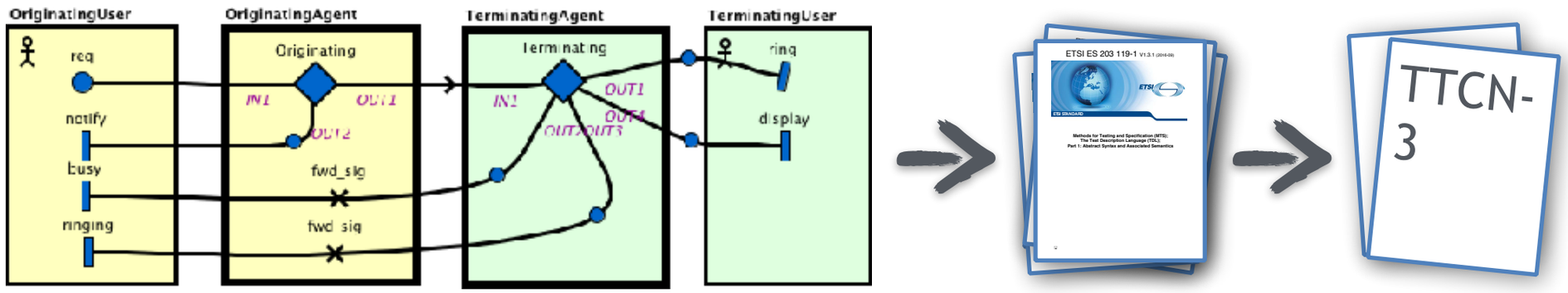
- User Requirements Notation (URN)
  - Elicitation, analysis, specification, and validation of requirements
  - Complementary views - goals (GRL) and scenarios (UCM)
  - ITU-T Recommendation Z.151 (10/12)



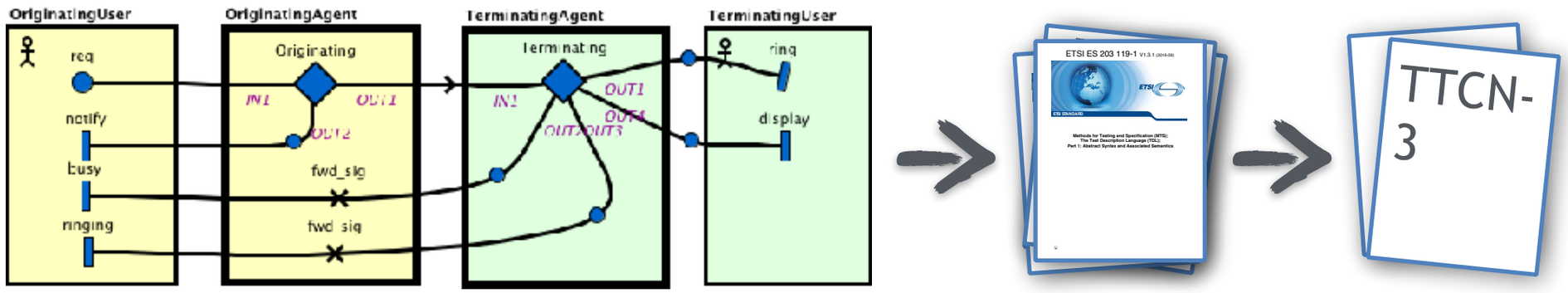
- Context
  - Test systems for cockpit systems and avionics solutions
  - Alternative means for
    - standards-based and model-based test generation and test automation
    - replace proprietary solutions
  - Transformation from high-level requirements and scenarios in UCM to TDL
  - Transformation from TDL to TTCN-3



- Goals
  - URN/UCM suitable starting point for modelling requirements?
  - TDL appropriate intermediate representation or even starting point?
  - TTCN-3 viable technology in the avionics industry?
- Stakeholders
  - Research, industry, agencies
  - Test engineers, test developers, test managers, analysts and modellers

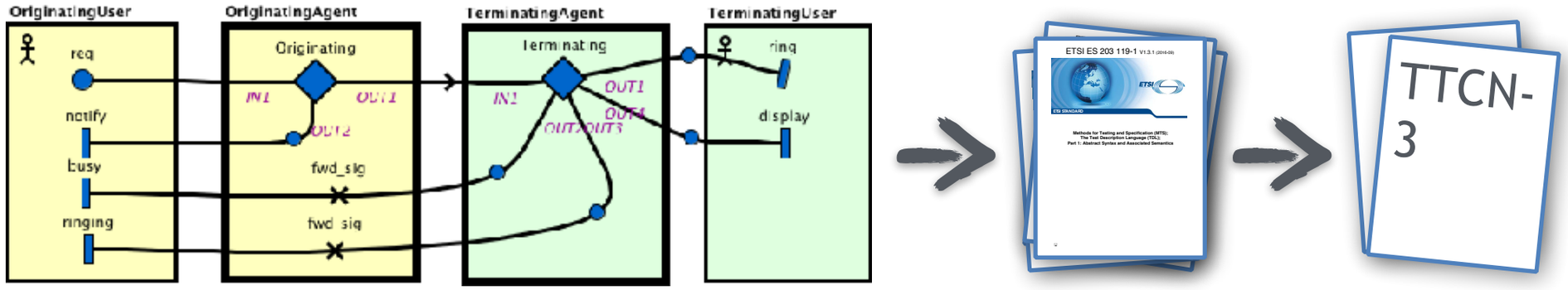


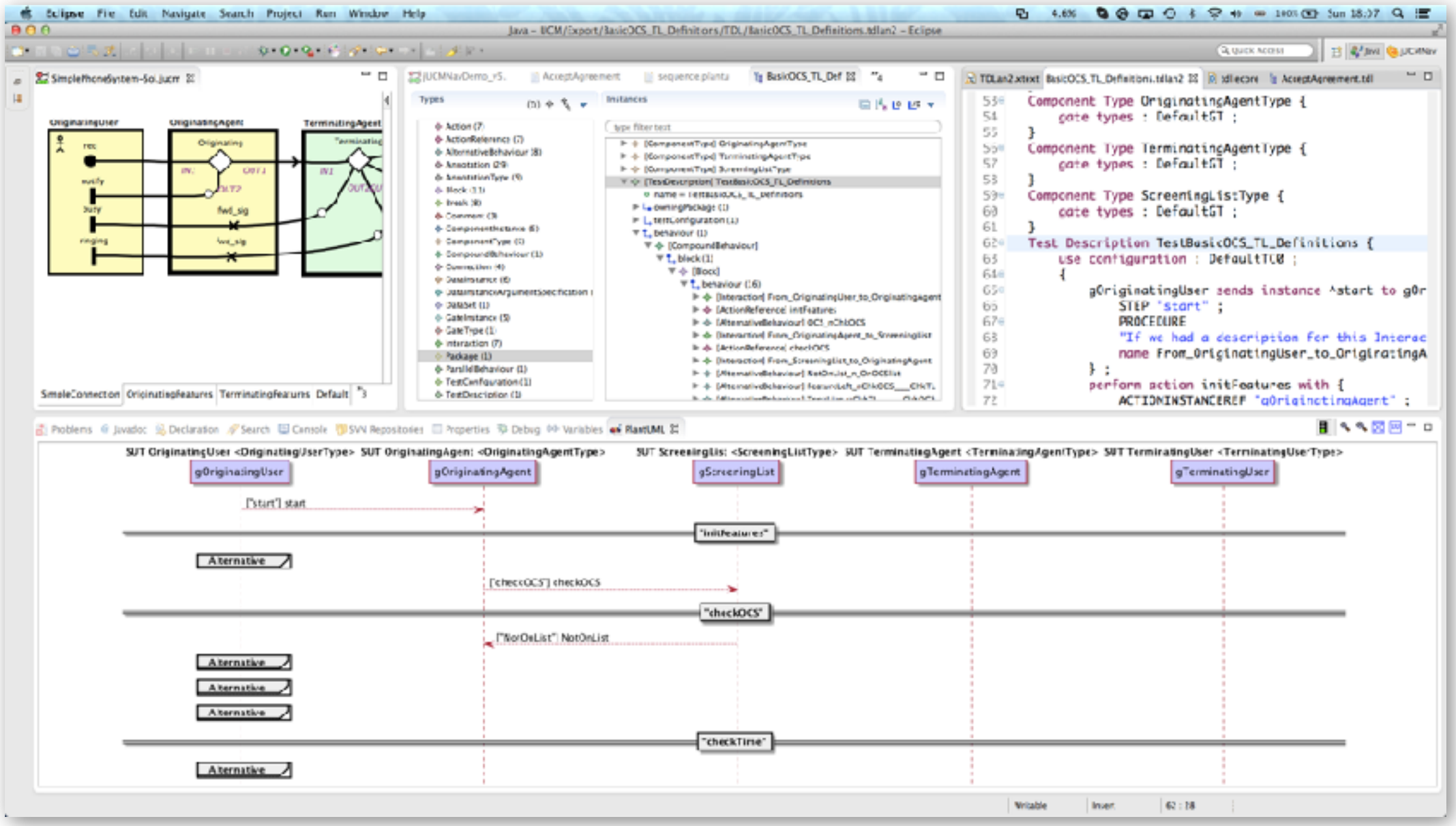
- Motivation
  - Tree-like structure of tests
    - TDL/TTCN-3 reflect this, existing transformations from UCM to e.g. MSC/UML do not
  - UCMs do not include much data information
    - appropriate stage to add data for executable test cases (UCM/TDL/TTCN-3/other)?
  - Peculiarities of the domain
    - support testing in an environment where an unknown number of sensors can send alarms (over unreliable channels) and messages in parallel



- TDL
  - Close enough to UCM for test generation
  - Close enough to TTCN-3 for generating executable test cases and test configurations
- Prototype
  - Part of jUCMNav (v6.0.0), developed at EECS (University of Ottawa)
  - Support for sequence and concurrent events (no alternatives yet)

http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/ExportTdlUserGuide



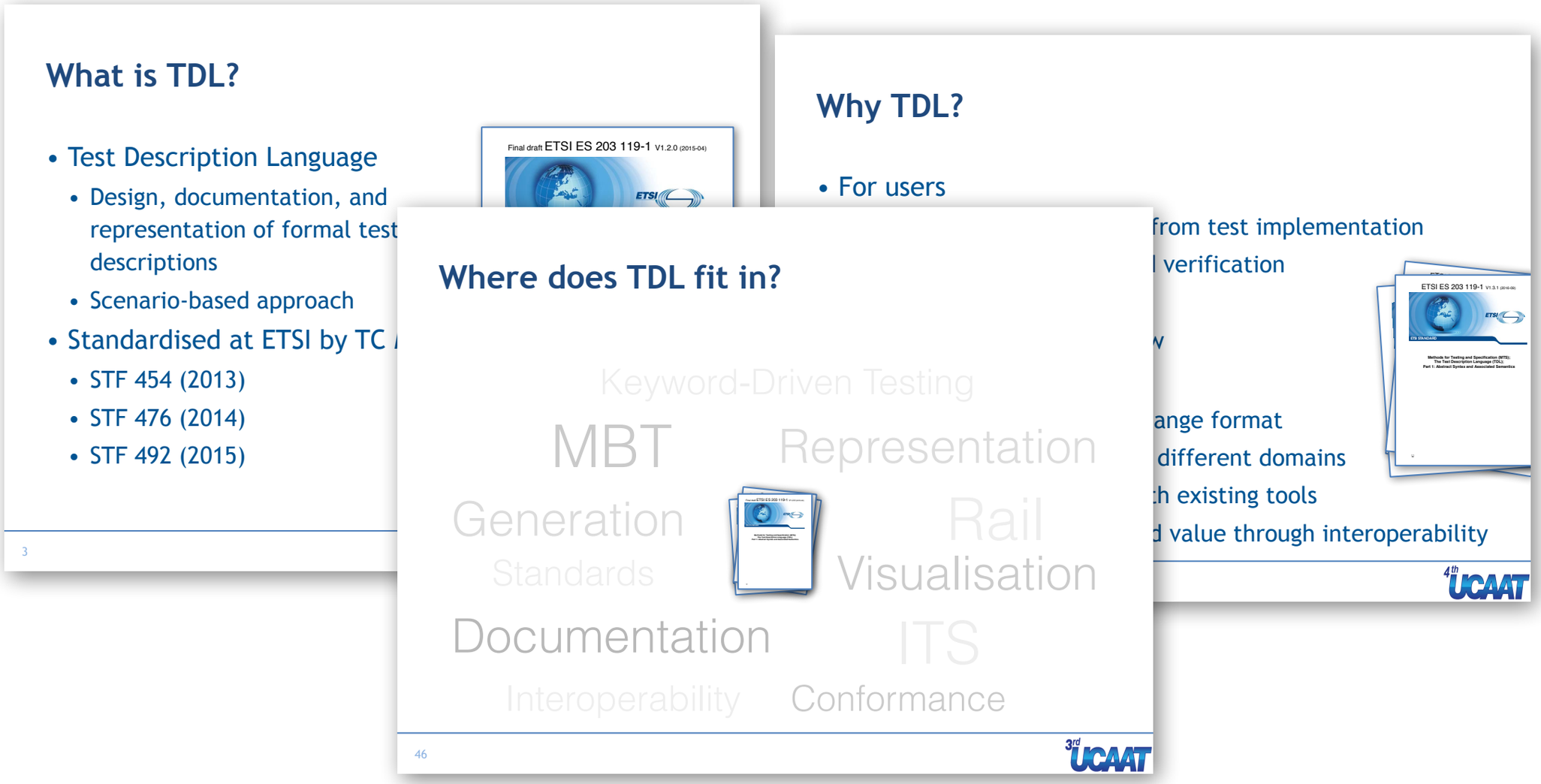


http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/ExportTdlUserGuide

Concluding remarks

- New technology, growing rapidly
- Open-source reference implementation
  - lower barrier to entry, accelerate adoption
  - commercial tool support not yet available
- Custom tools can be put together in a matter of hours
  - basic yet capable
  - make early adoption easier
- Advanced solutions still require additional effort
  - not immediately necessary to get started with using TDL

Summary

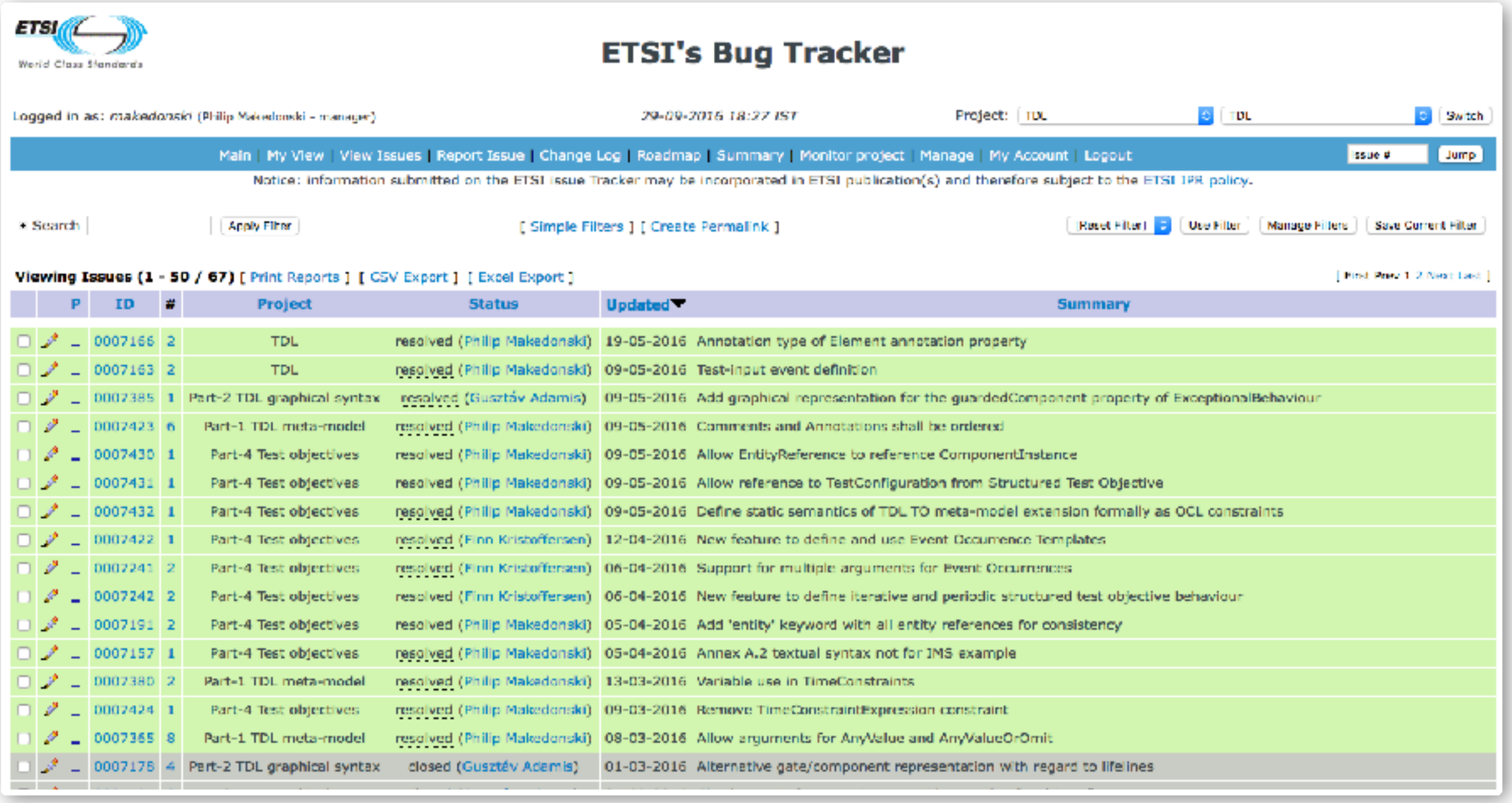


Where would you consider using TDL?

tdl.etsi.org

4th UCAAT

What would you want to see in TDL?



ETSI's Bug Tracker

Logged in as: makedonski (Philip Makedonski - manager) 29-09-2016 18:27 IST Project: TDL TDL Switch

Main | My View | View Issues | Report Issue | Change Log | Roadmap | Summary | Monitor project | Manage | My Account | Logout issue # Jump

Notice: Information submitted on the ETSI Issue Tracker may be incorporated in ETSI publication(s) and therefore subject to the ETSI IPR policy.

+ Search | Apply Filter [Simple Filters] [Create Permalink] Reset Filter | Use Filter | Manage Filters | Save Current Filter

Viewing Issues (1 - 50 / 67) [Print Reports] [CSV Export] [Excel Export] [First Prev 1 2 Next Last]

P	ID	#	Project	Status	Updated	Summary
_	0007166	2	TDL	resolved (Philip Makedonski)	19-05-2016	Annotation type of Element annotation property
_	0007163	2	TDL	resolved (Philip Makedonski)	09-05-2016	Test-input event definition
_	0007385	1	Part-2 TDL graphical syntax	resolved (Gusztáv Adamis)	09-05-2016	Add graphical representation for the guardedComponent property of ExceptionalBehaviour
_	0007423	6	Part-1 TDL meta-model	resolved (Philip Makedonski)	09-05-2016	Comments and Annotations shall be ordered
_	0007430	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Allow EntityReference to reference ComponentInstance
_	0007431	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Allow reference to TestConfiguration from Structured Test Objective
_	0007432	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-05-2016	Define static semantics of TDL TO meta-model extension formally as OCL constraints
_	0007422	1	Part-4 Test objectives	resolved (Finn Kristoffersen)	12-04-2016	New feature to define and use Event Occurrence Templates
_	0007241	2	Part-4 Test objectives	resolved (Finn Kristoffersen)	06-04-2016	Support for multiple arguments for Event Occurrences
_	0007242	2	Part-4 Test objectives	resolved (Finn Kristoffersen)	06-04-2016	New feature to define iterative and periodic structured test objective behaviour
_	0007191	2	Part-4 Test objectives	resolved (Philip Makedonski)	05-04-2016	Add 'entity' keyword with all entity references for consistency
_	0007157	1	Part-4 Test objectives	resolved (Philip Makedonski)	05-04-2016	Annex A.2 textual syntax not for IMS example
_	0007380	2	Part-1 TDL meta-model	resolved (Philip Makedonski)	13-03-2016	Variable use in TimeConstraints
_	0007424	1	Part-4 Test objectives	resolved (Philip Makedonski)	09-03-2016	Remove TimeConstraintExpression constraint
_	0007365	8	Part-1 TDL meta-model	resolved (Philip Makedonski)	08-03-2016	Allow arguments for AnyValue and AnyValueOrOmit
_	0007175	4	Part-2 TDL graphical syntax	closed (Gusztáv Adamis)	01-03-2016	Alternative gate/component representation with regard to lifelines

tdl.etsi.org

Testing and Modeling with TDL

Philip Makedonski, Gusztav Adamis, Martti Käärik,
Finn Kristoffersen, Andreas Ulrich, Xavier Zeitoun

tdl.etsi.org