3rd UCAAT User Conference on Advanced Automated Testing

ETSI World Class Standards

Sophia Antipolis, French Riviera
20-22 October 2015

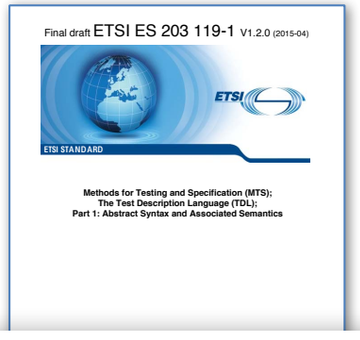
APPLYING TDL IN PRACTICE

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

Overview

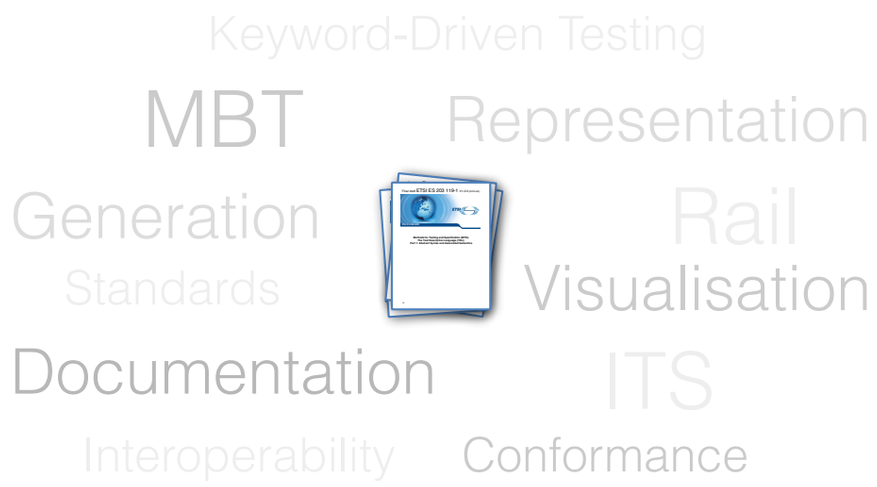
What is TDL?

- 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)



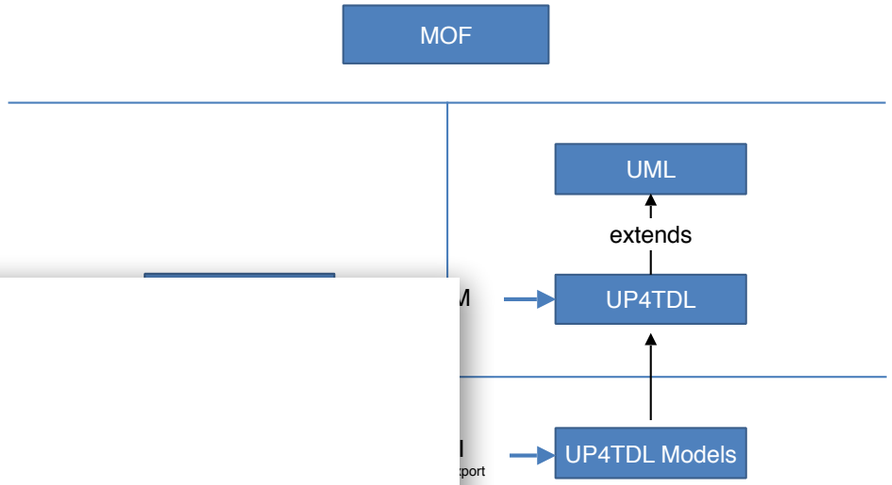
3

Where does TDL fit in?



46

UP4TDL, TDL or UML?



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)


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

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


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

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

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


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

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

What is TDL?

- Standardised?
  - clear semantics
  - interoperability of tools and test specifications
  - updated with user needs
  - maintenance commitment


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

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

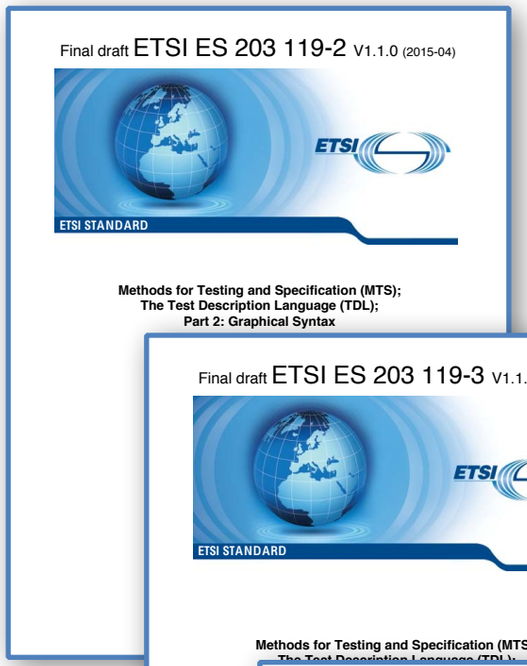

Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

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

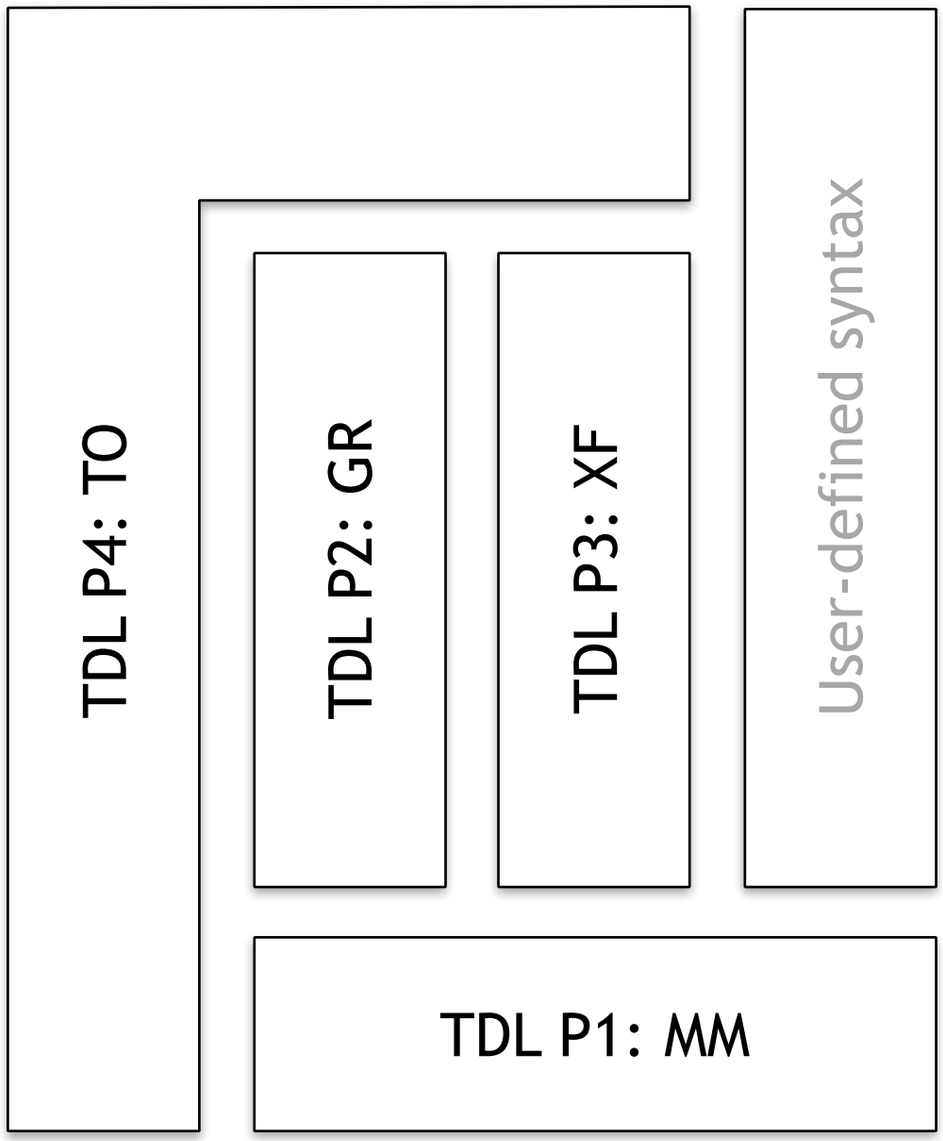


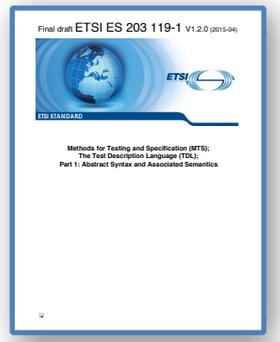
Part 2: GR Graphical Syntax

Part 3: XF Exchange Format

Part 4: TO Structured Test Objective Specification

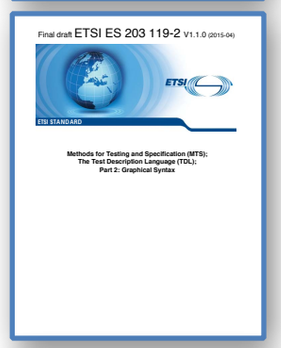
What is TDL?




Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

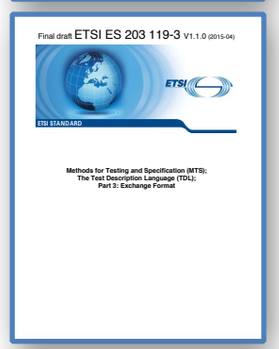
ETSI STANDARD

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


Final draft ETSI ES 203 119-2 V1.1.0 (2015-04)

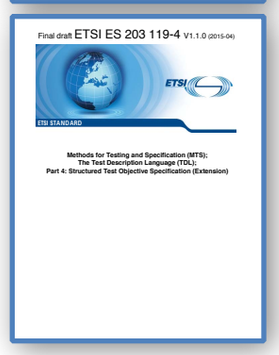
ETSI STANDARD

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


Final draft ETSI ES 203 119-3 V1.1.0 (2015-04)

ETSI STANDARD

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

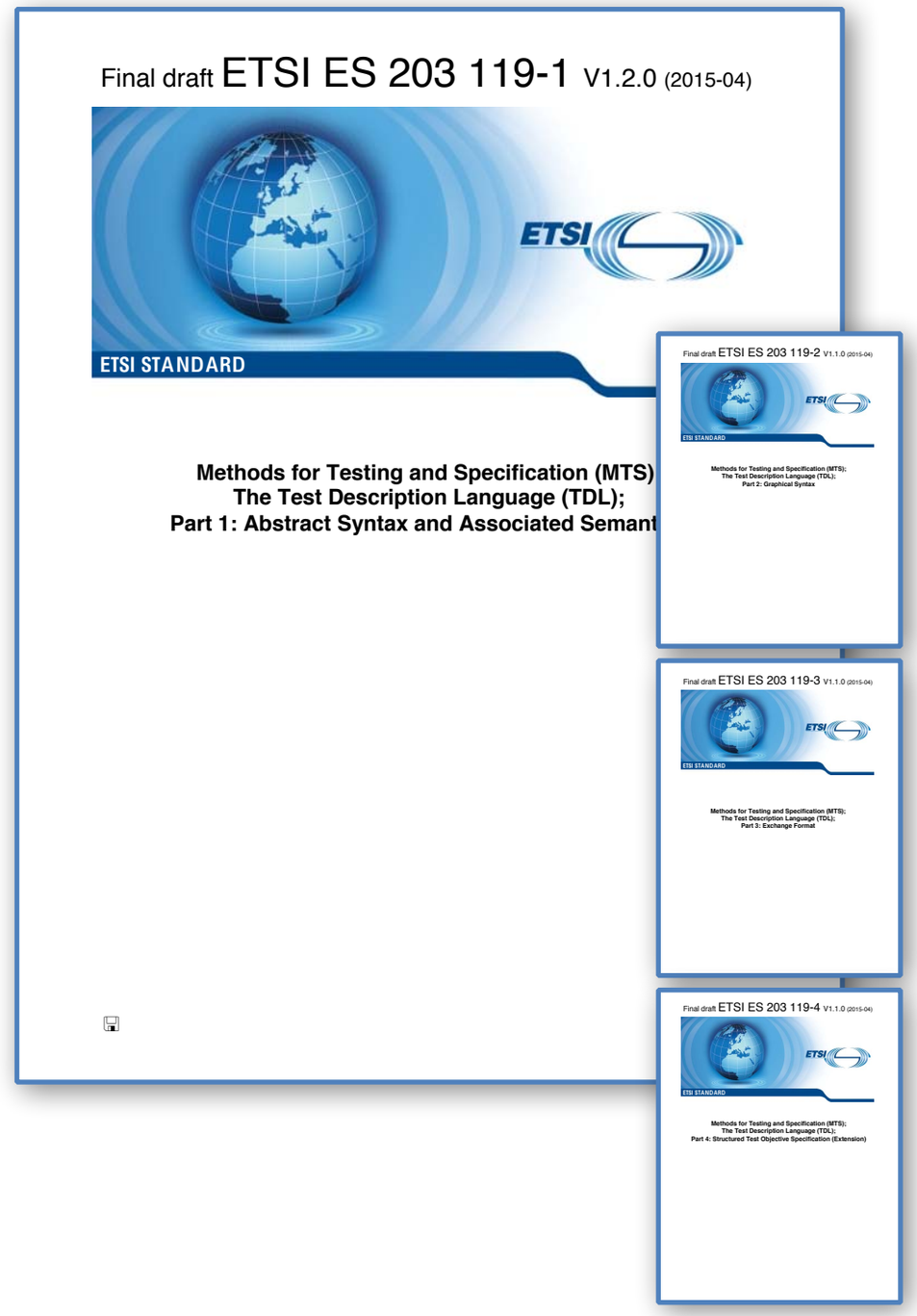

Final draft ETSI ES 203 119-4 V1.1.0 (2015-04)

ETSI STANDARD

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

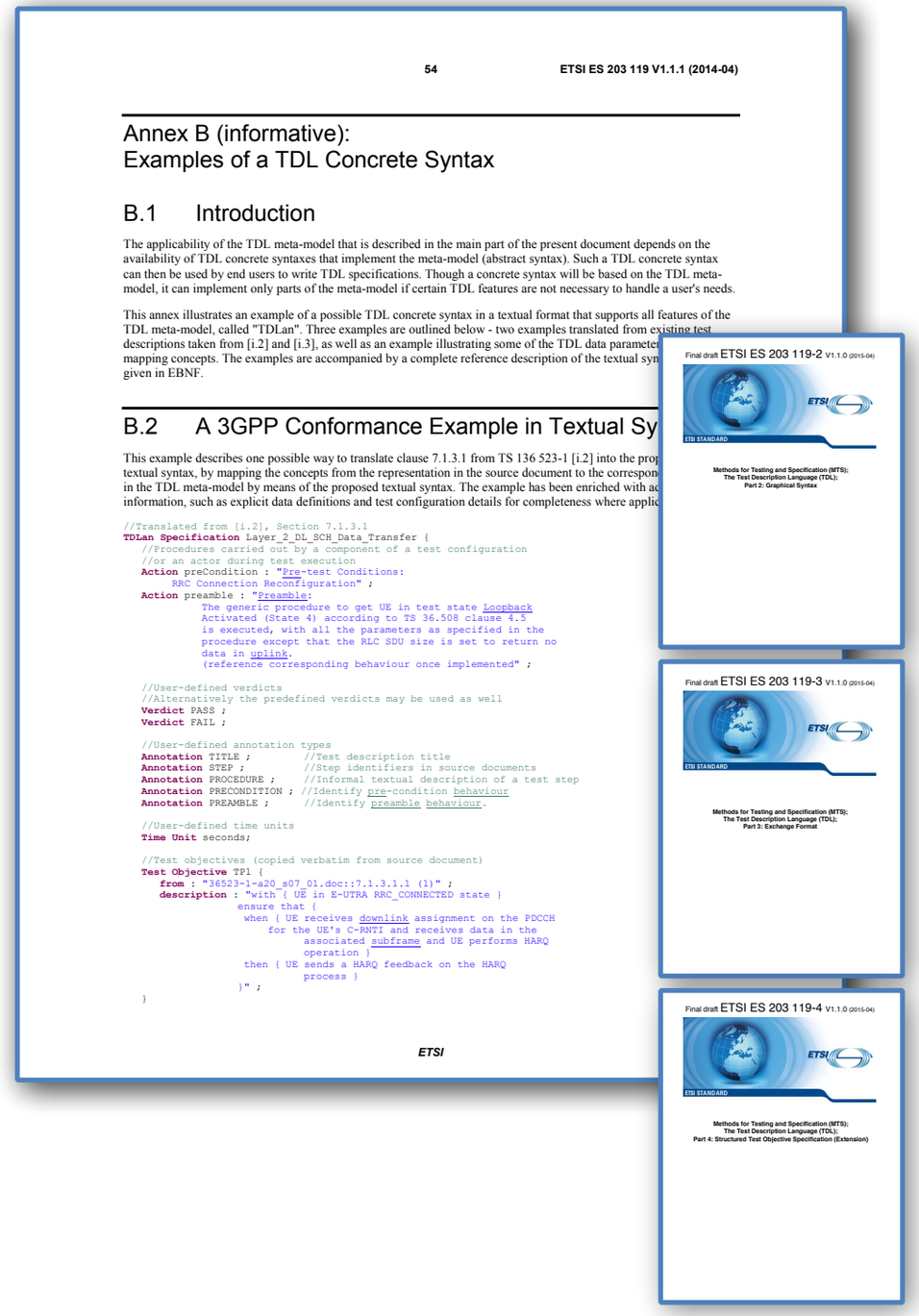
What is TDL? Part 1: MM

- 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

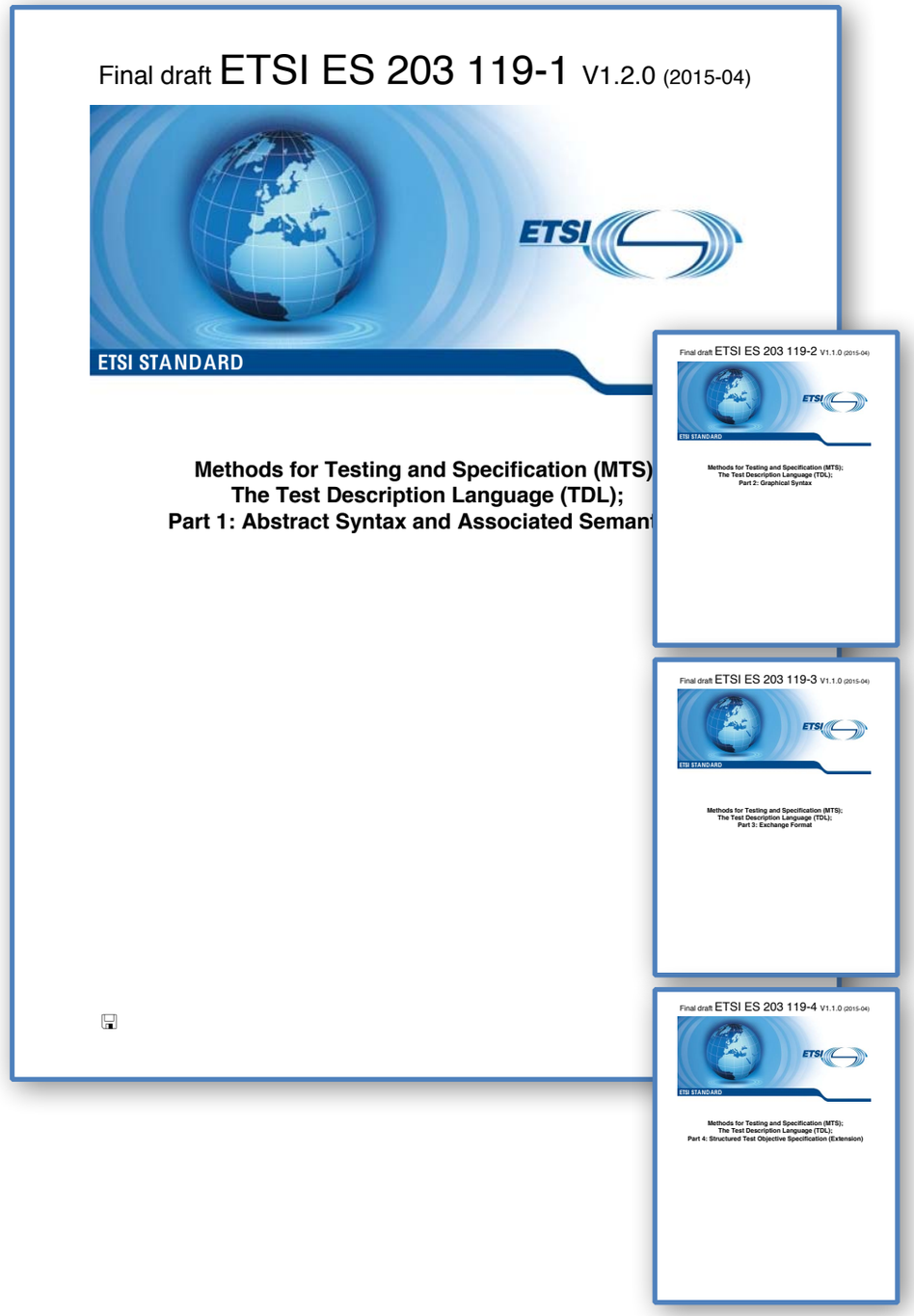
- TDL main ingredients
  - Test data
  - Test configuration
  - Test behaviour
  - Test objectives
  - Time

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



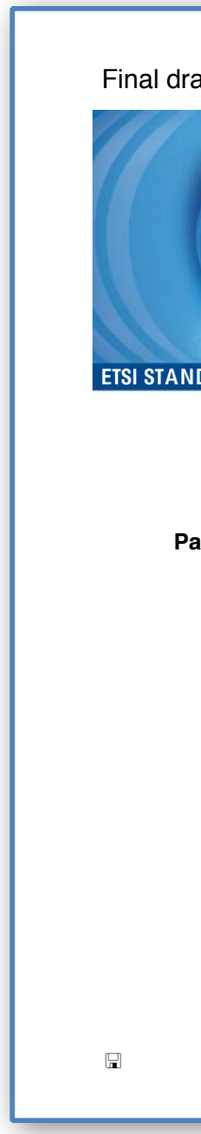
What is TDL? Part 1: MM

```
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;
```

```
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
}
```

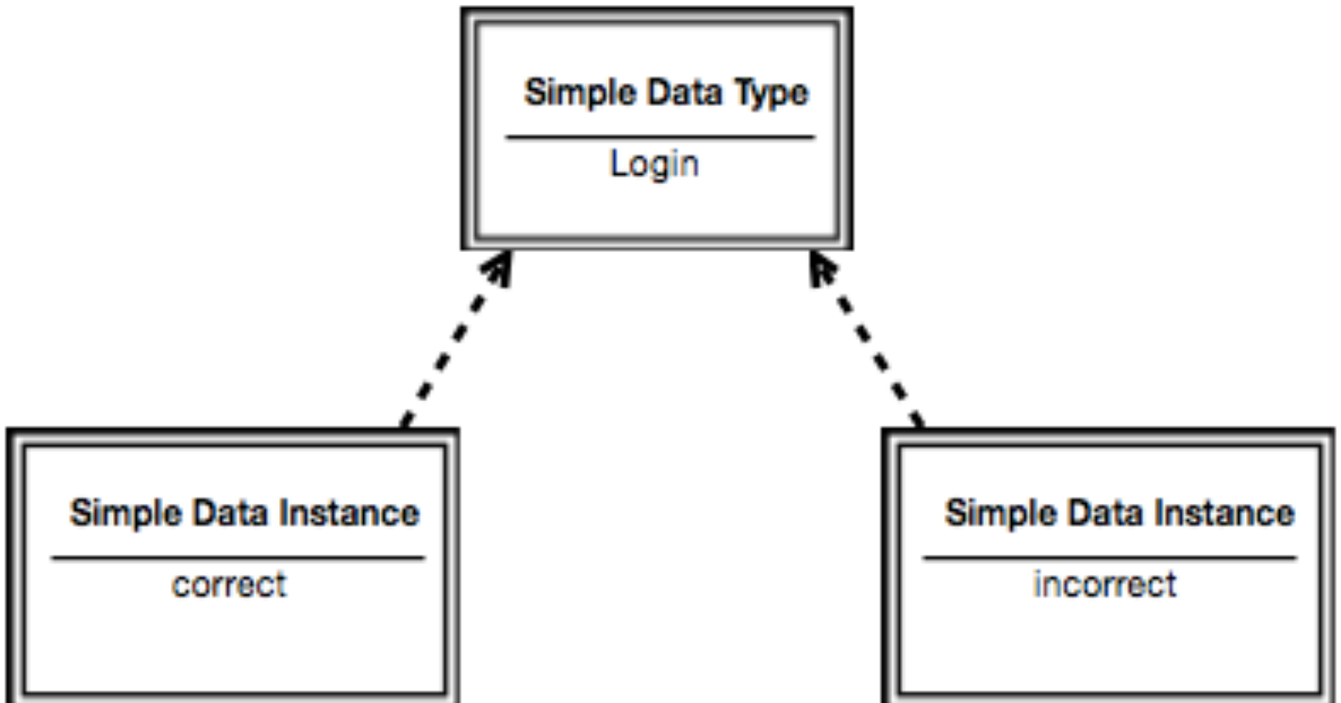
```
type record Login {
  charstring user,
  charstring password,
  charstring hint,
  integer id
} with {
  encode "xpath=//div[@id='login']";
  encode (user) "relative=/div/dd[3]";
  encode (password) "relative=/div/dd[4]";
};
```

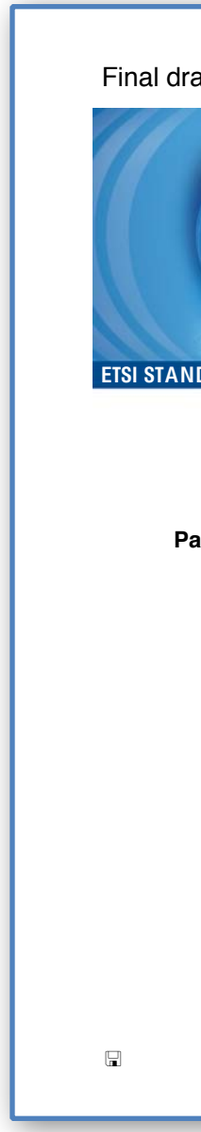


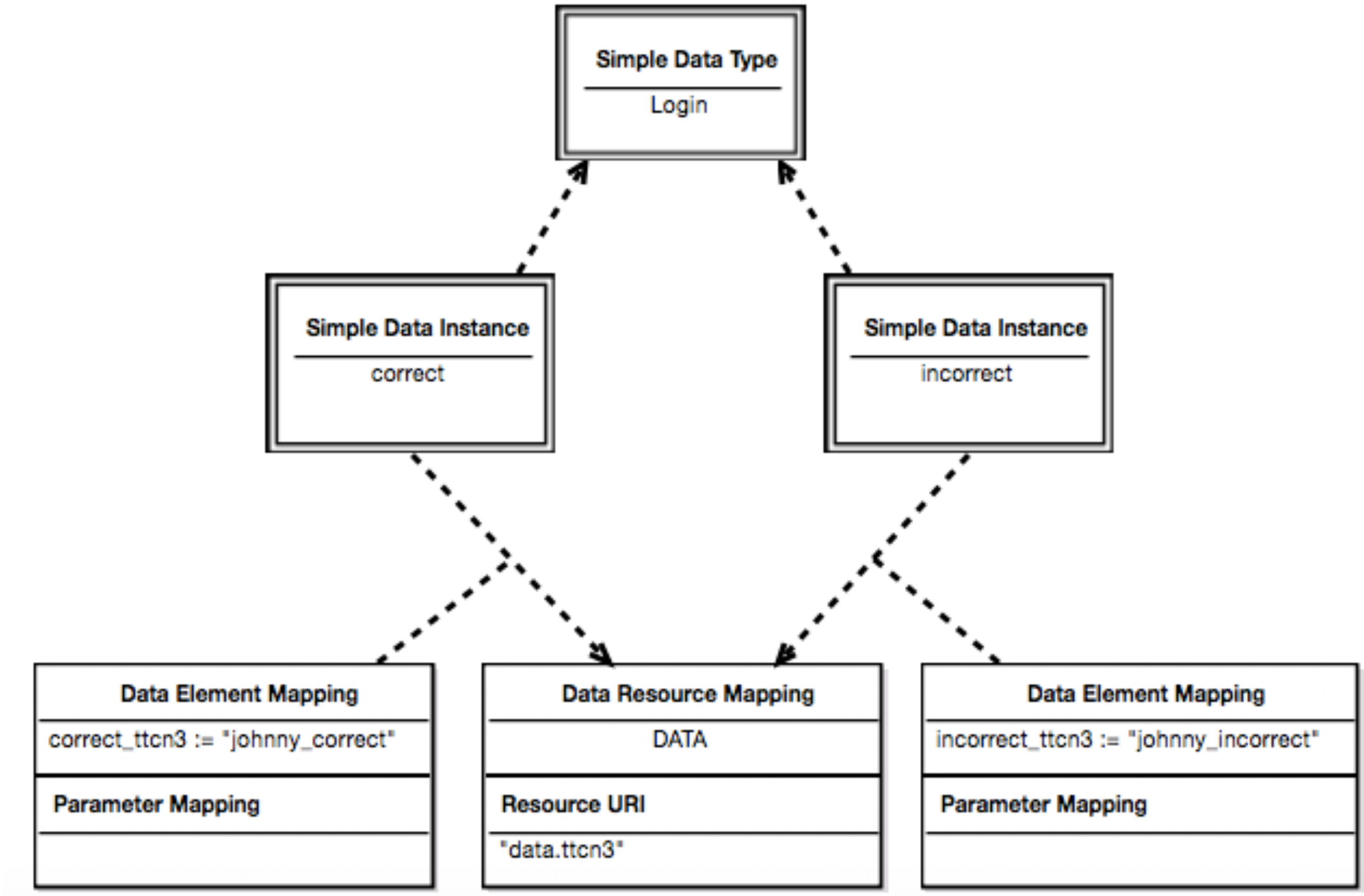
What is TDL? Part 1: MM

```
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


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

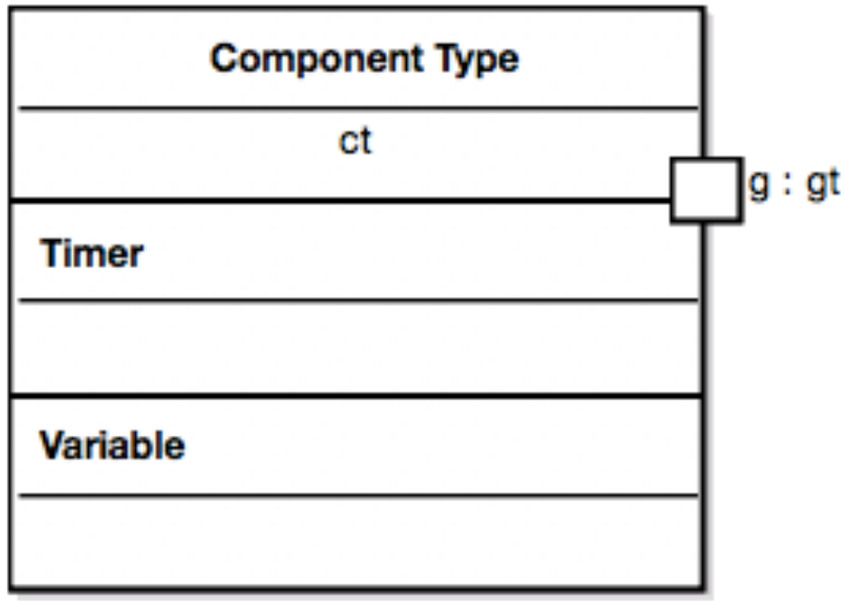
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

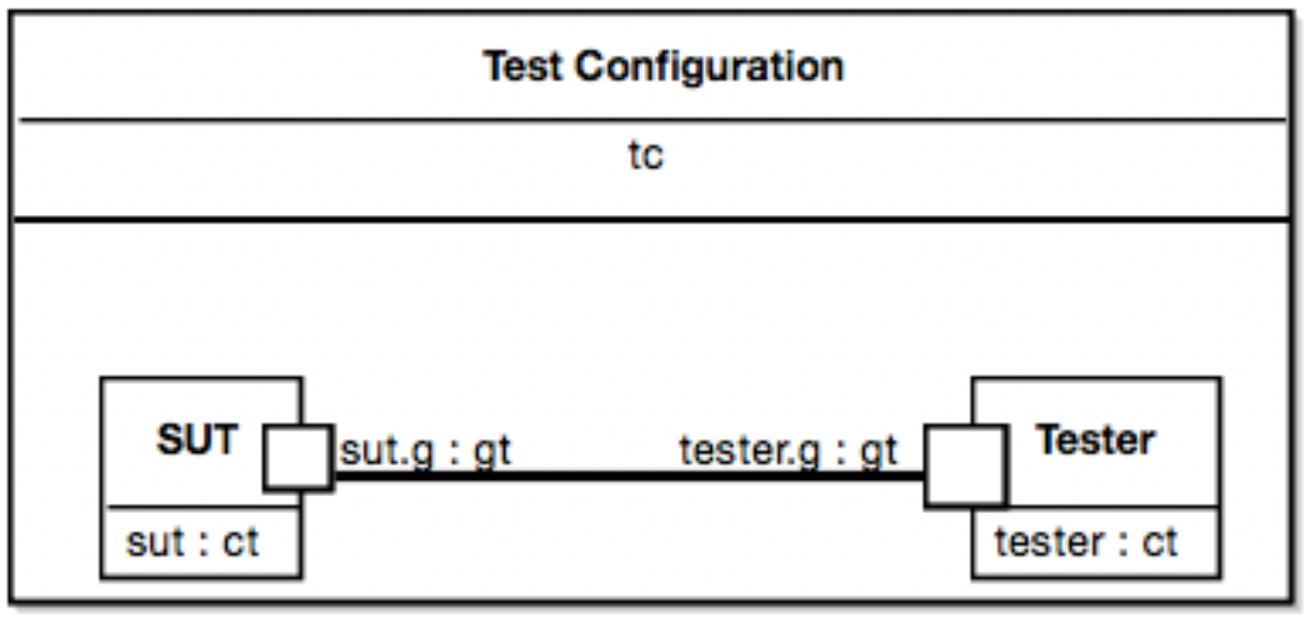
What is TDL? Part 1: MM

```
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


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

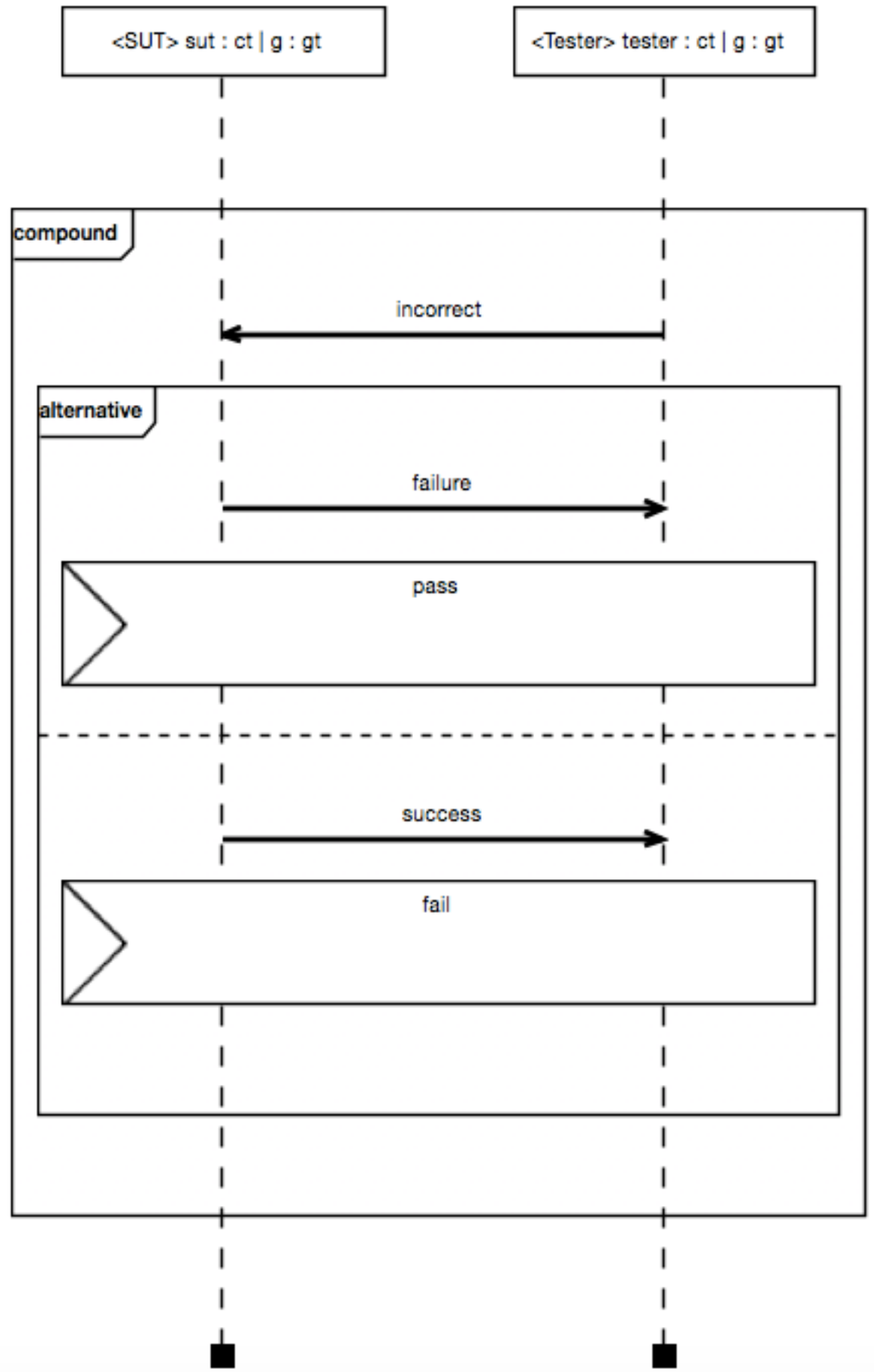
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

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

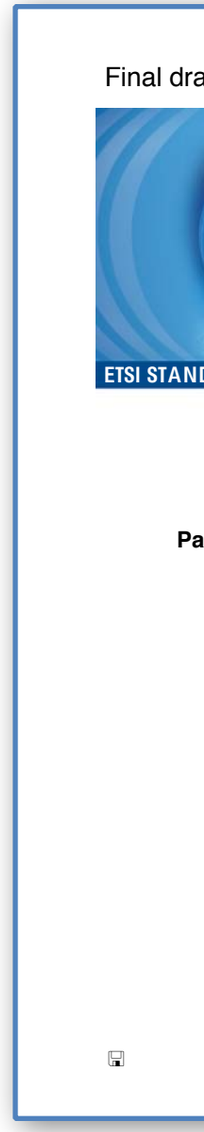

Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

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

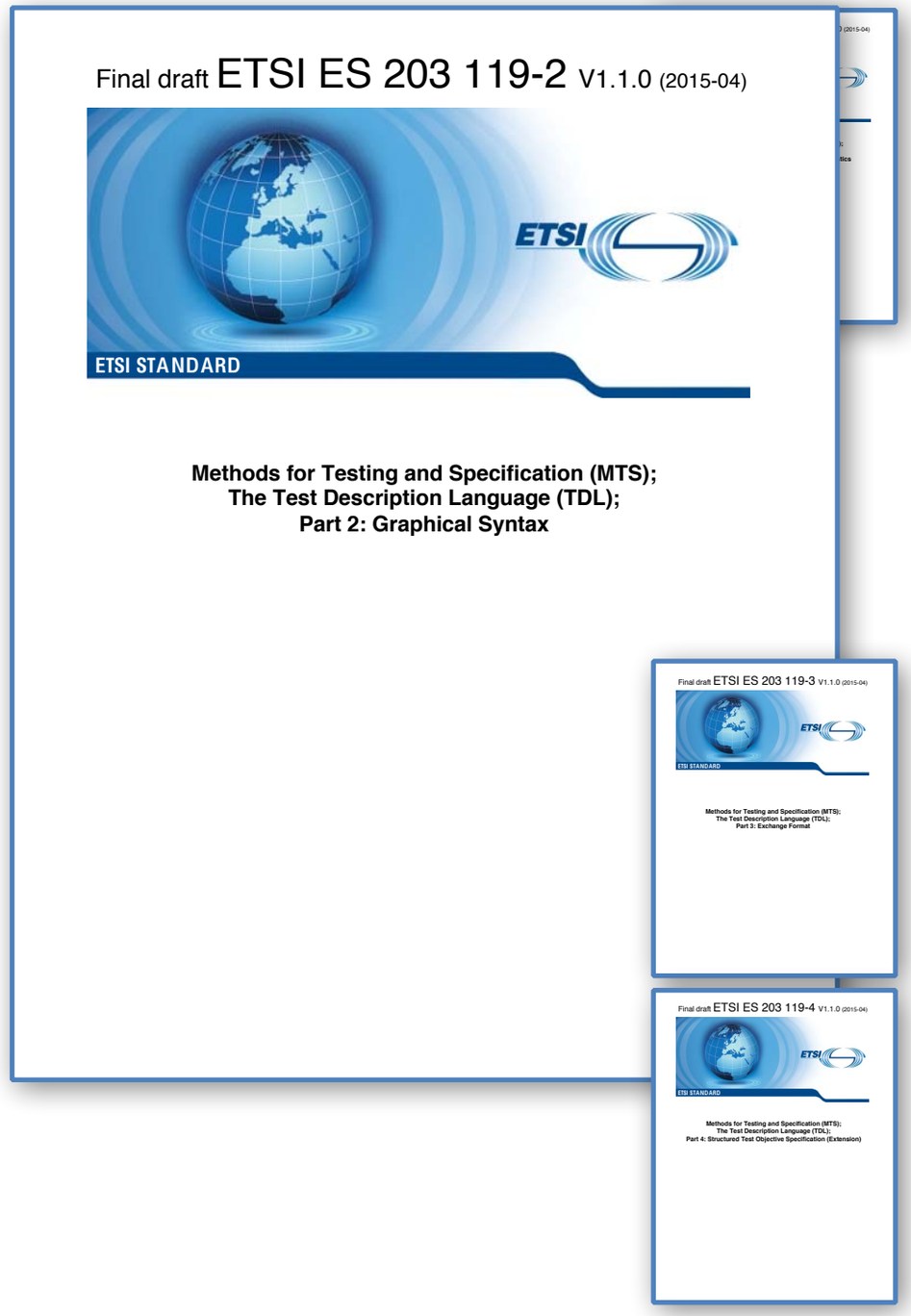
What is TDL? Part 1: MM

```
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;
    }
}
```



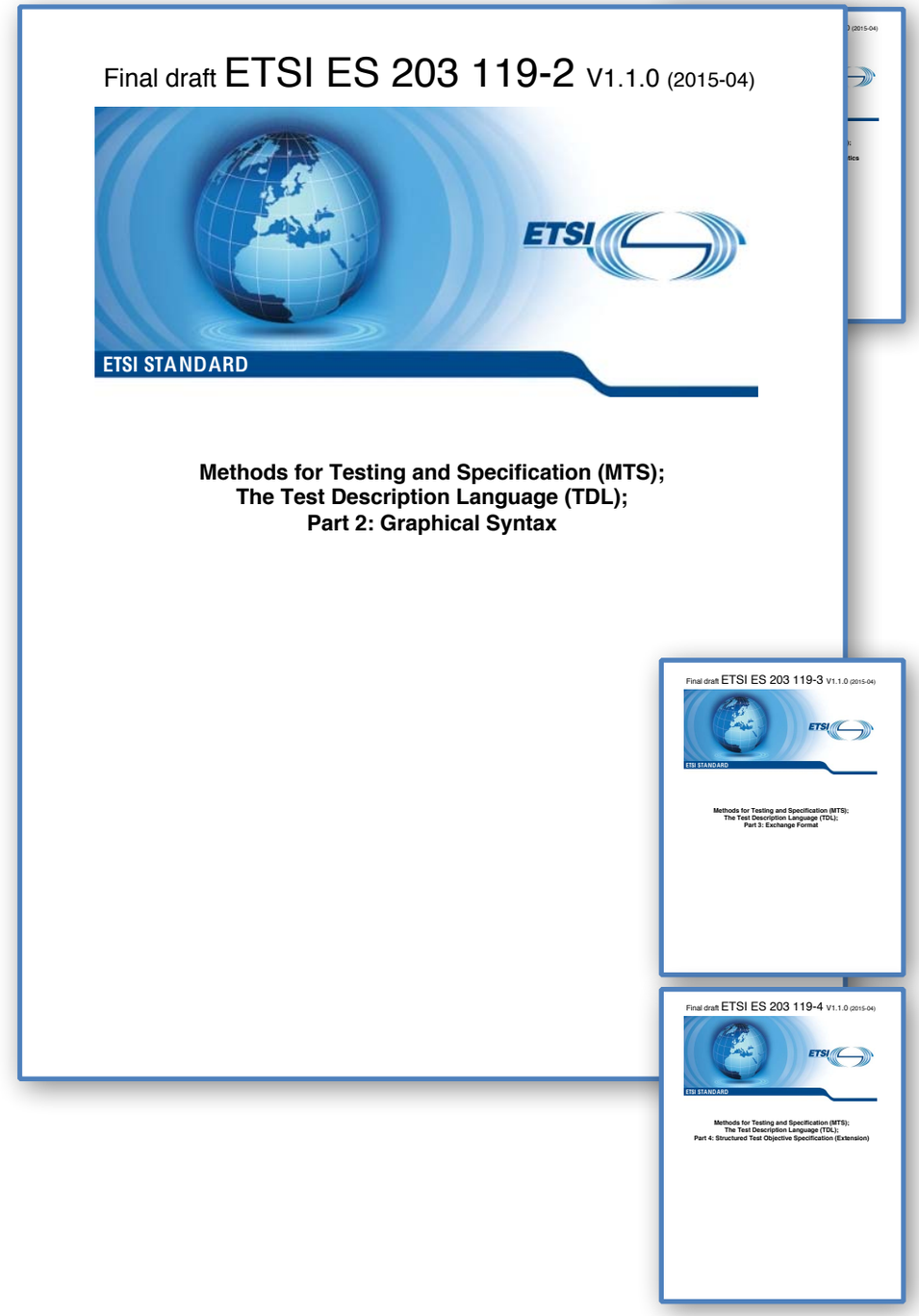
What is TDL? Part 2: GR

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

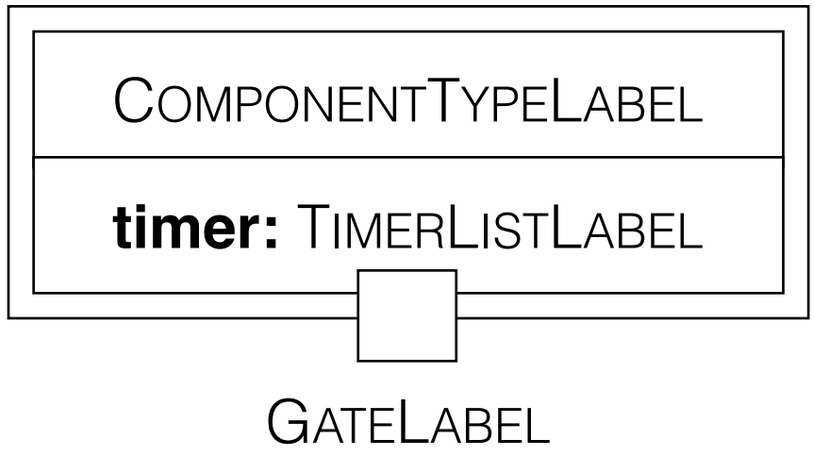


What is TDL? Part 2: GR

- 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

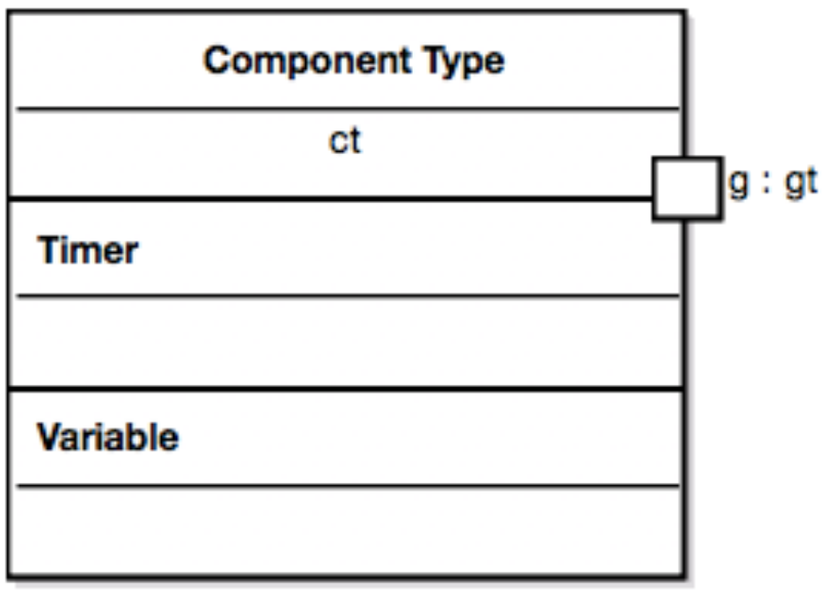


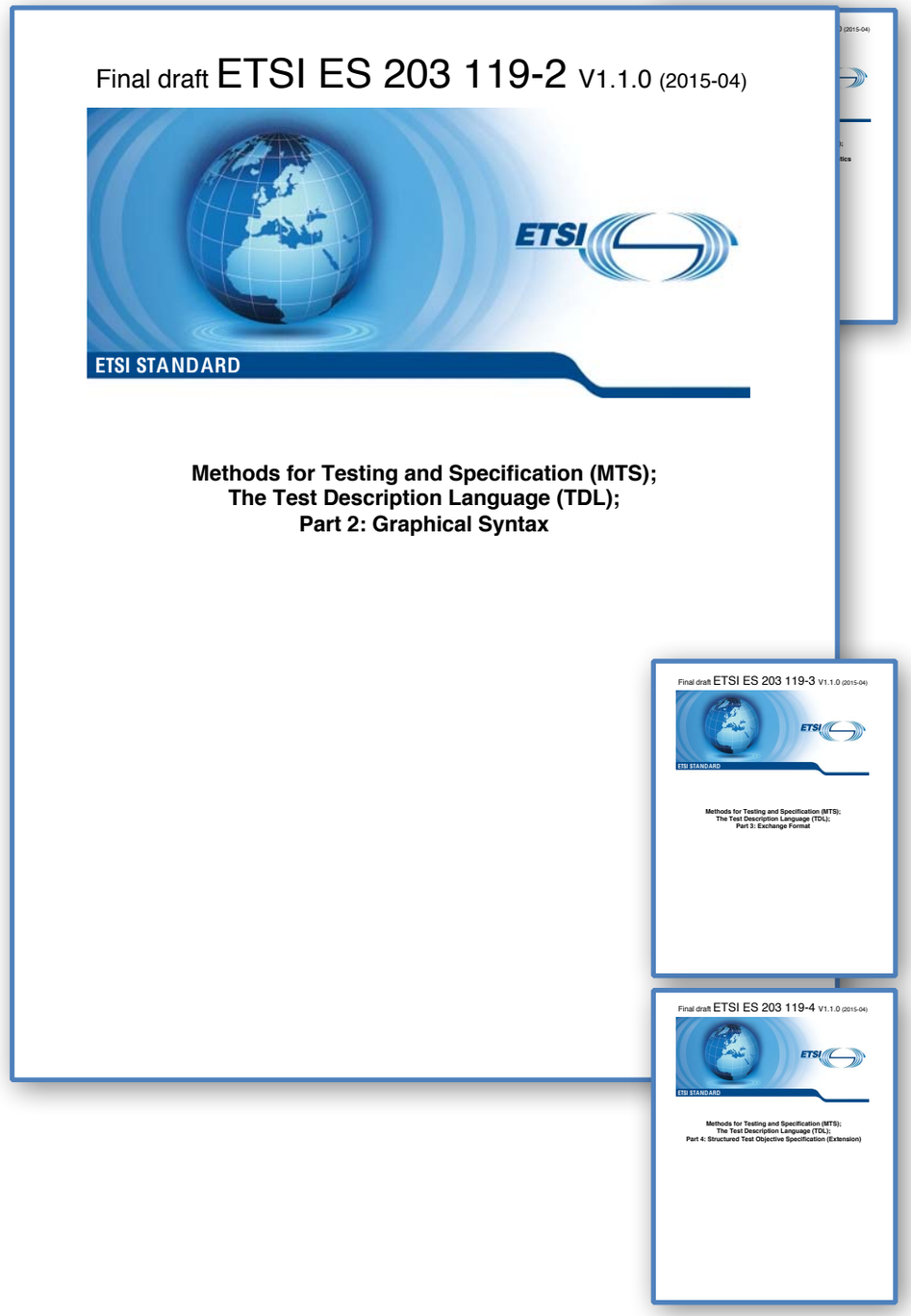
context: ComponentType

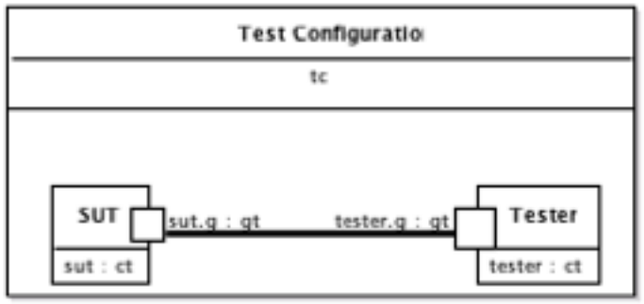
COMPONENTTYPELABEL ::= self.name

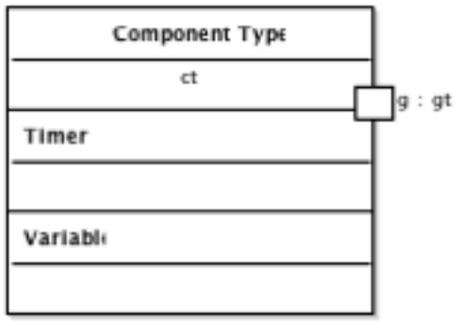
TIMERLISTLABEL ::= self.timer.name

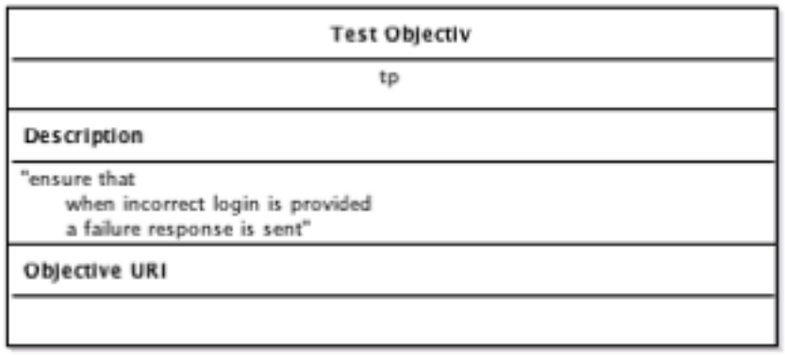
…

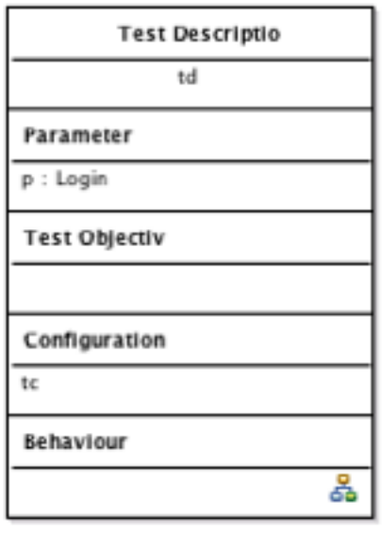


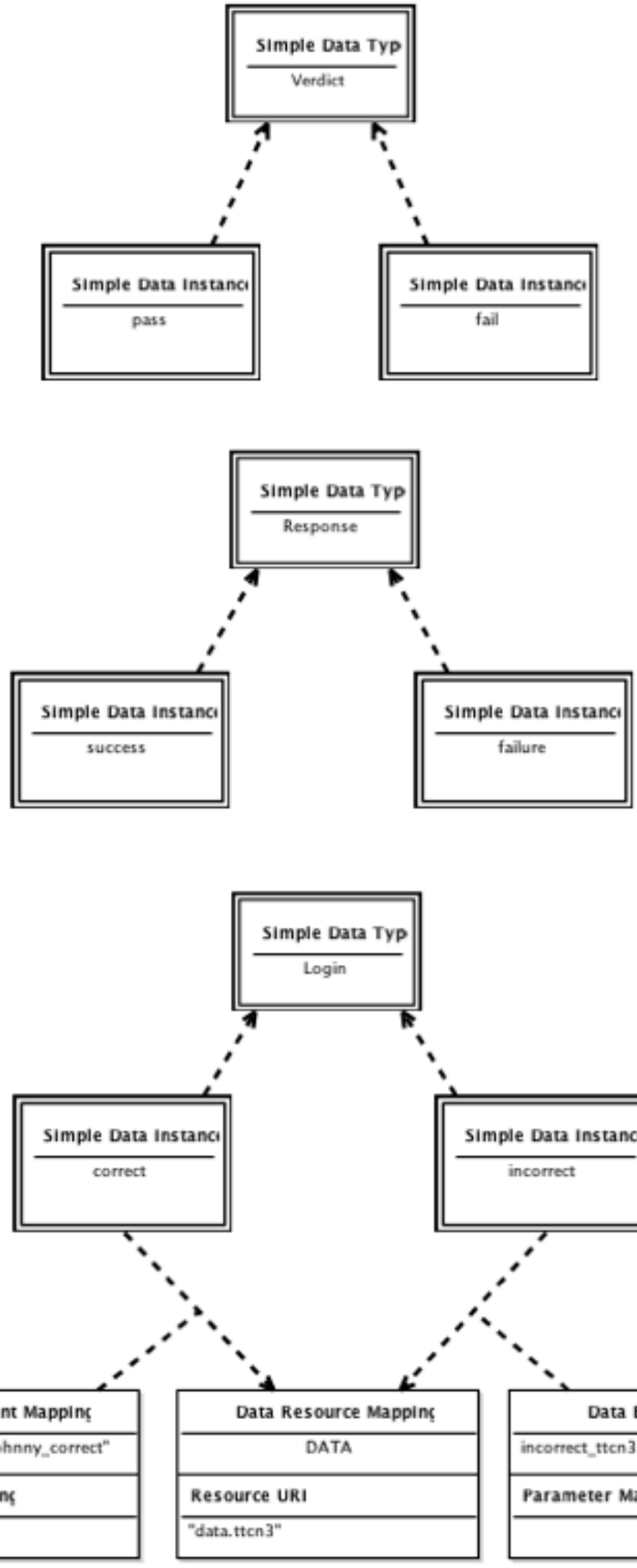


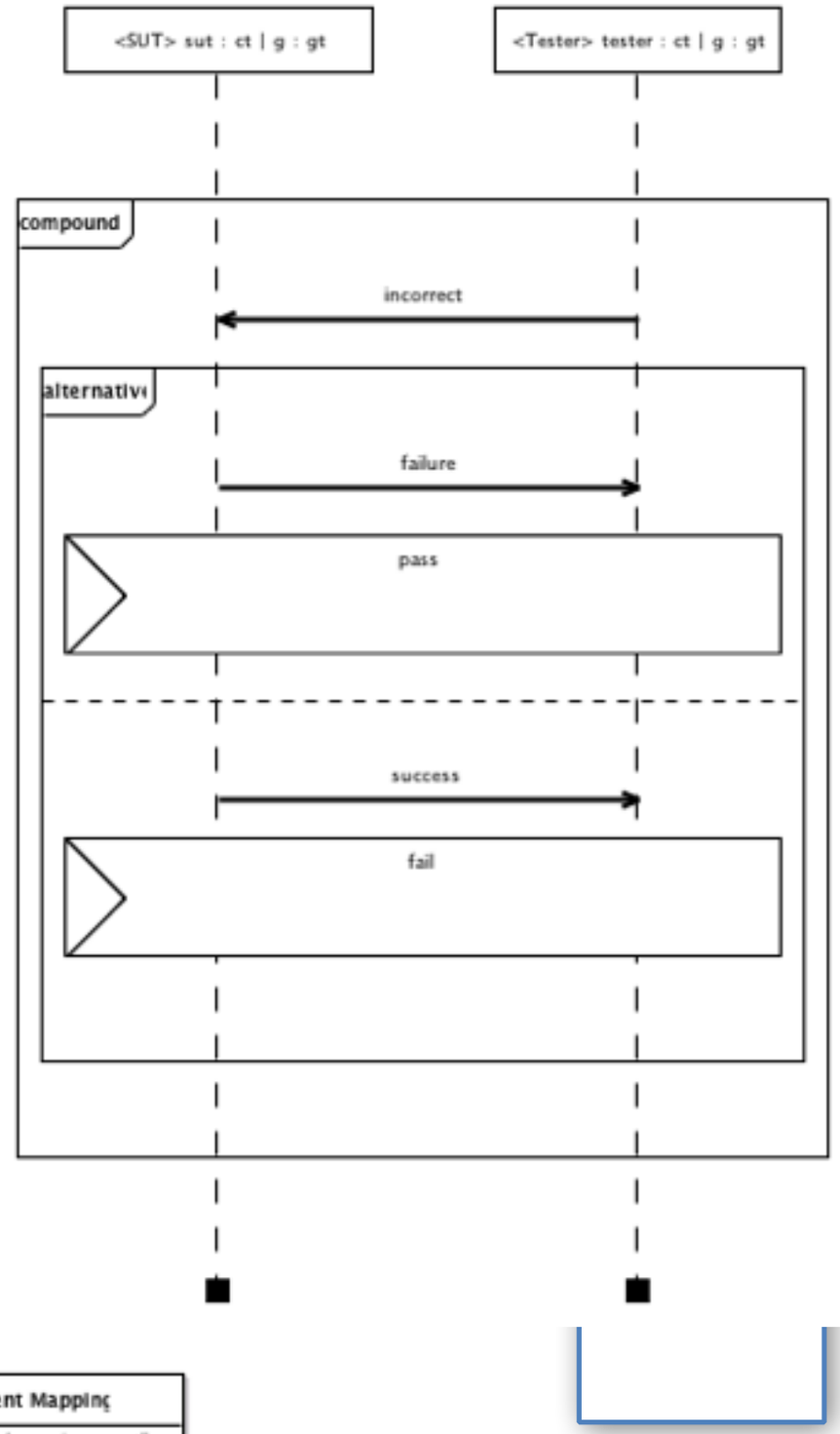

Test Configuratio
tc
SUT
sut : ct
sut.g : gt
tester.g : gt
Tester
tester : ct


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

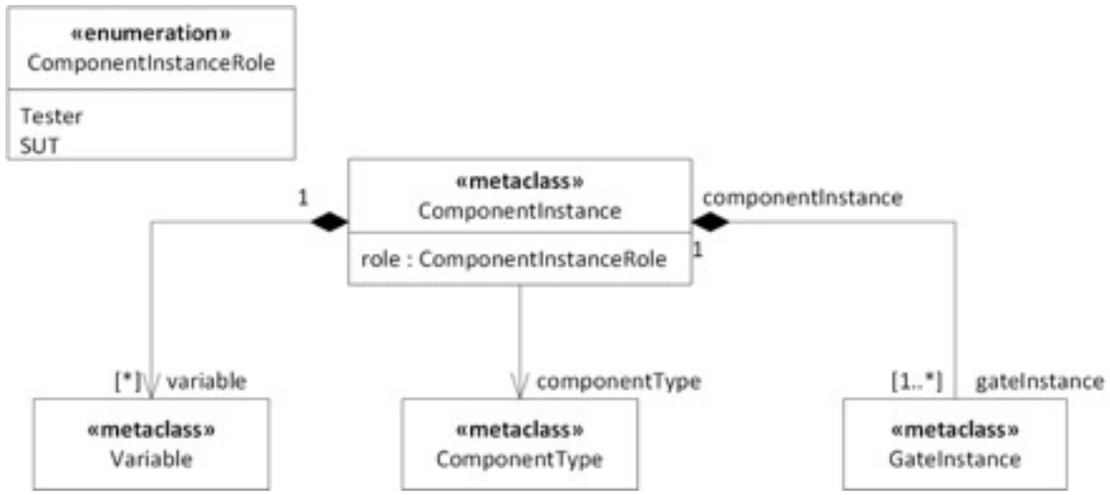
What is TDL? Part 3: XF

- 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

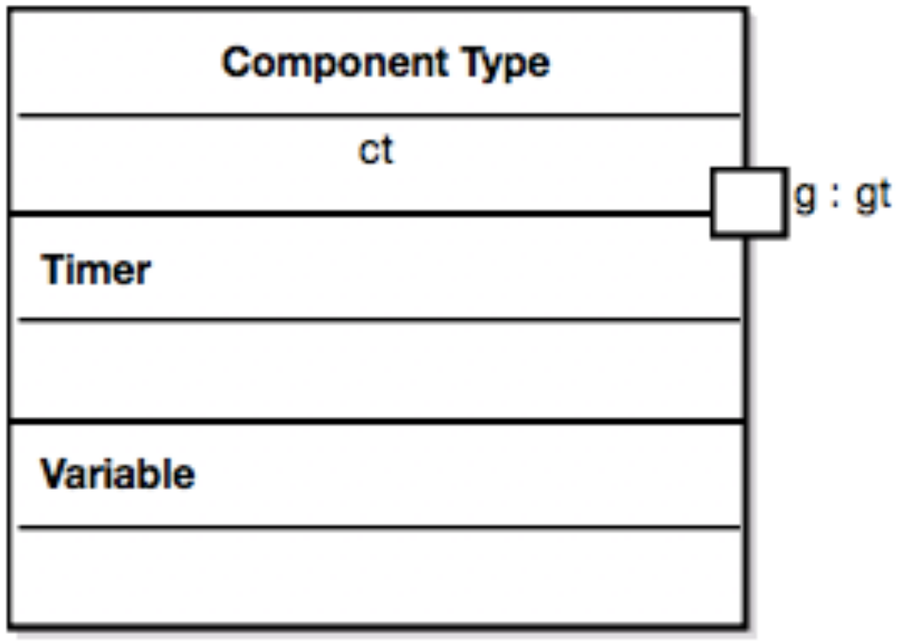
- Syntactical validity only?
  - two-step validation
  - syntax: XMI Schema
  - semantics: MOF model validation

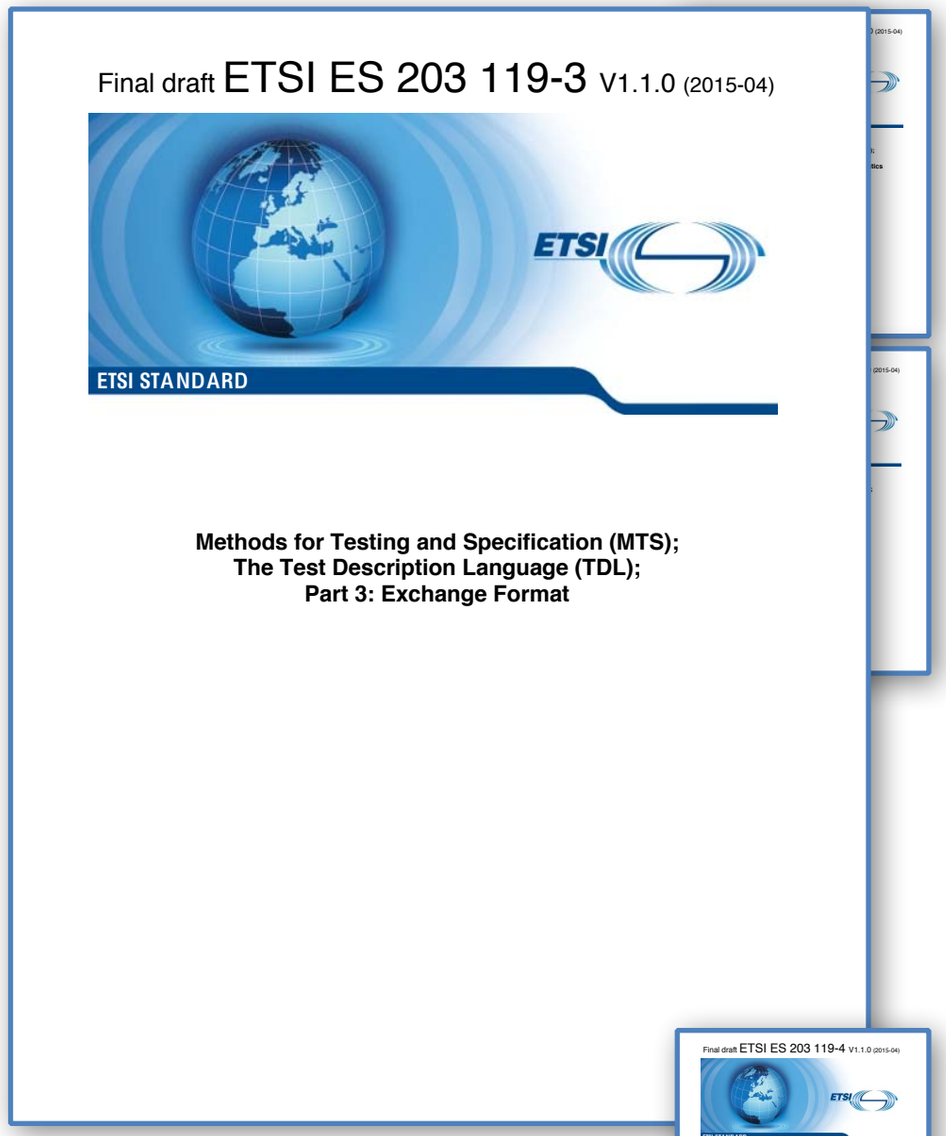
What is TDL? Part 3: XF



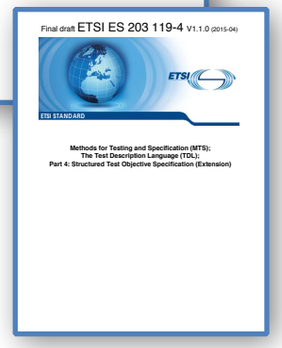

```
<xsd:complexType name="ComponentInstance">
   <xsd:complexContent>
     <xsd:extension base="tdl:Element">
       <xsd:choice maxOccurs="unbounded" minOccurs="0">
         <xsd:element name="gateInstance" type="tdl:GateInstance"/>
         <xsd:element name="variable" type="tdl:Variable"/>
     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



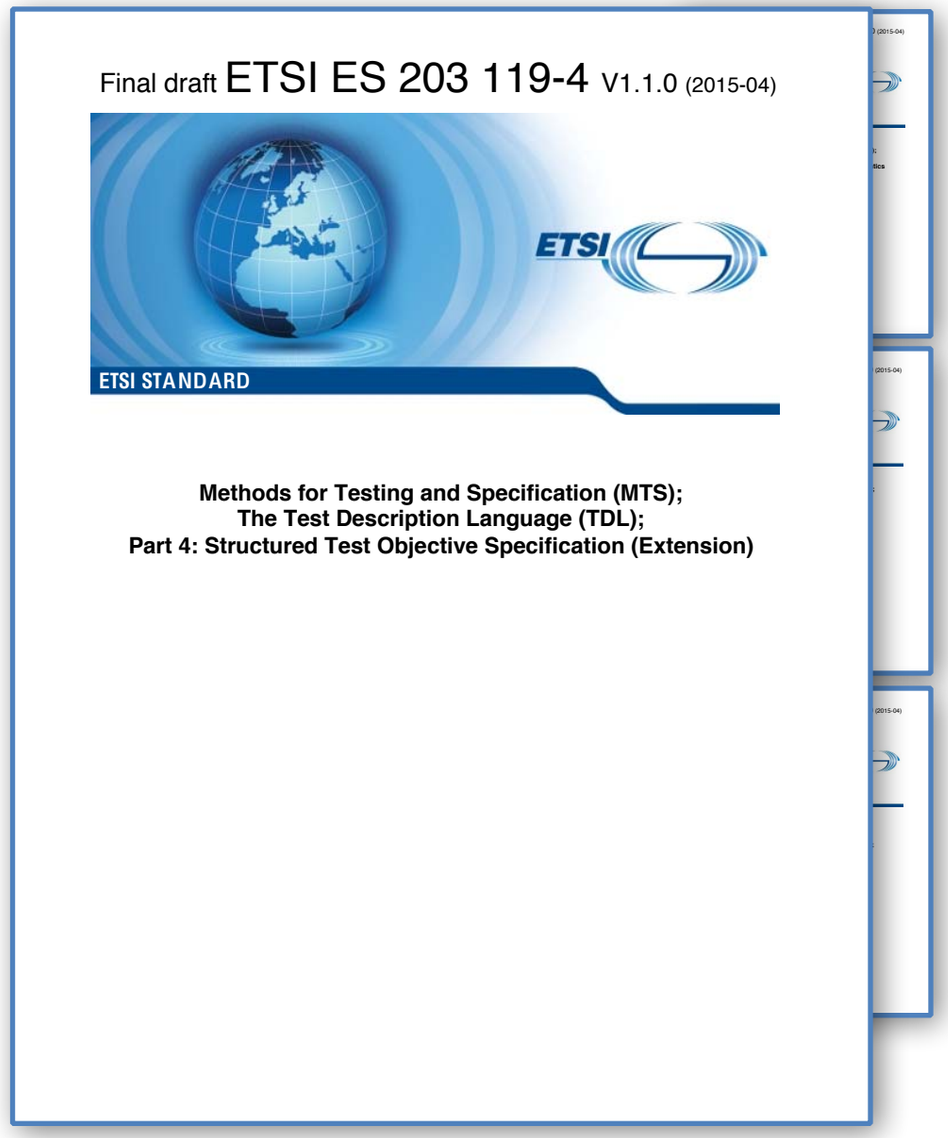


```
<packagedElement xsi:type="tdl:ComponentType" xmi:id="_qKt233asEeWrfP0MdfQNpg" name="ct">
  <gateInstance xmi:id="_qKt24HasEeWrfP0MdfQNpg" name="g" type="_qKt23nasEeWrfP0MdfQNpg"/>
</packagedElement>
```



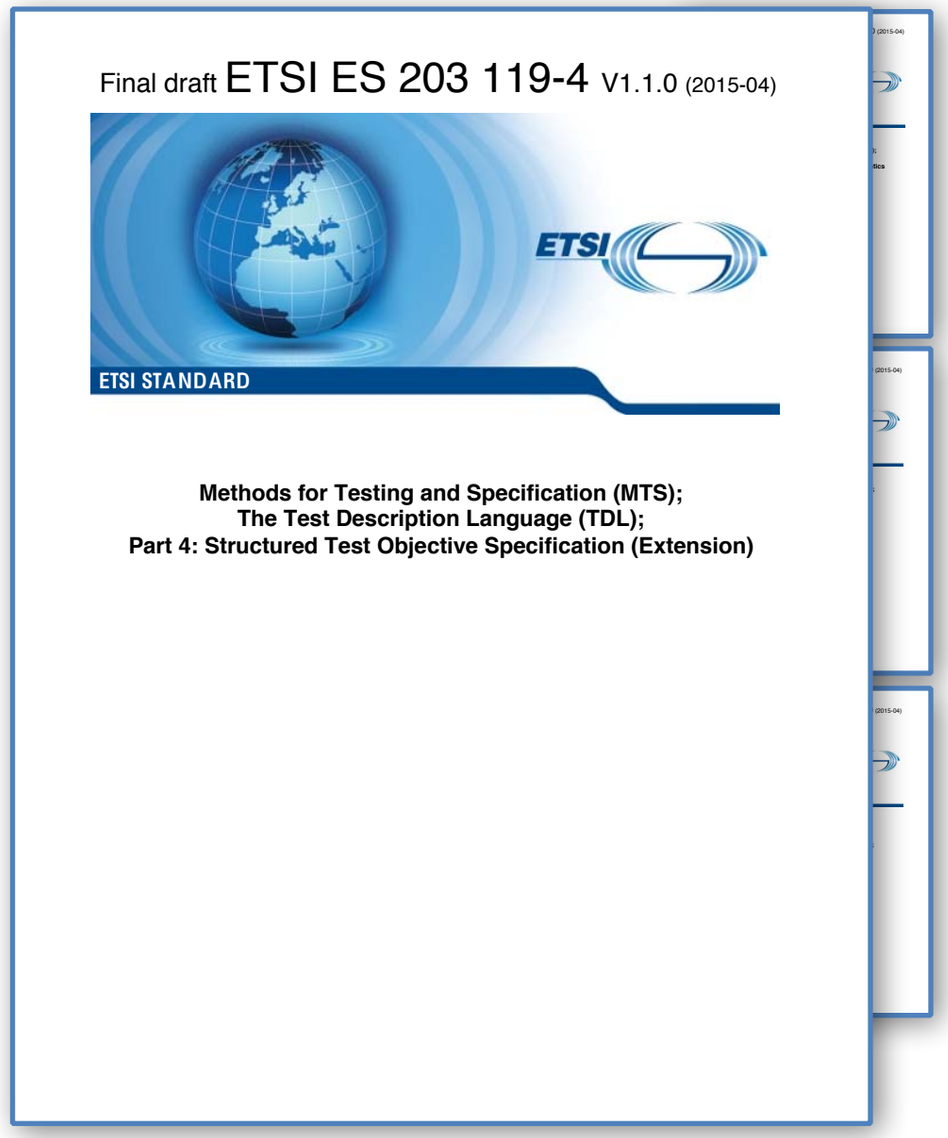
What is TDL? Part 4: TO

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



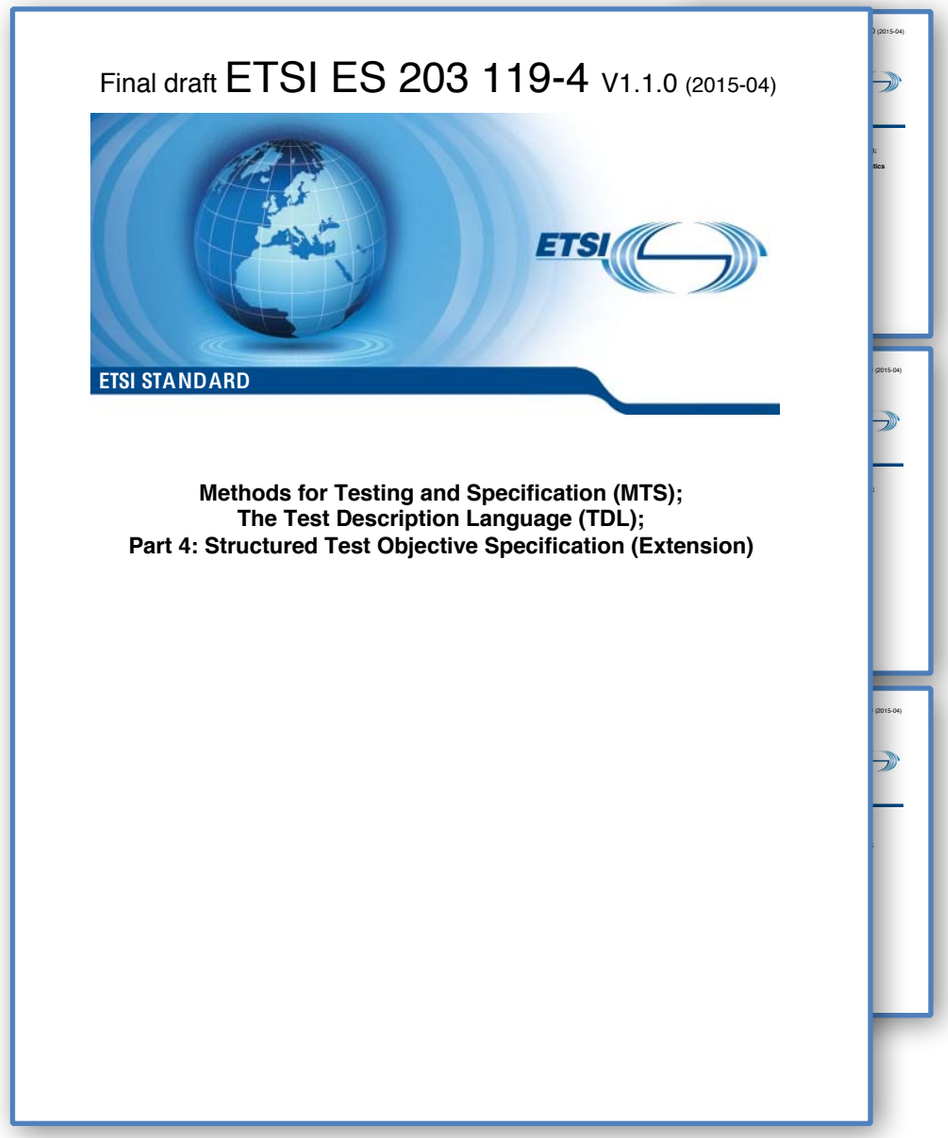
What is TDL? Part 4: TO

- 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



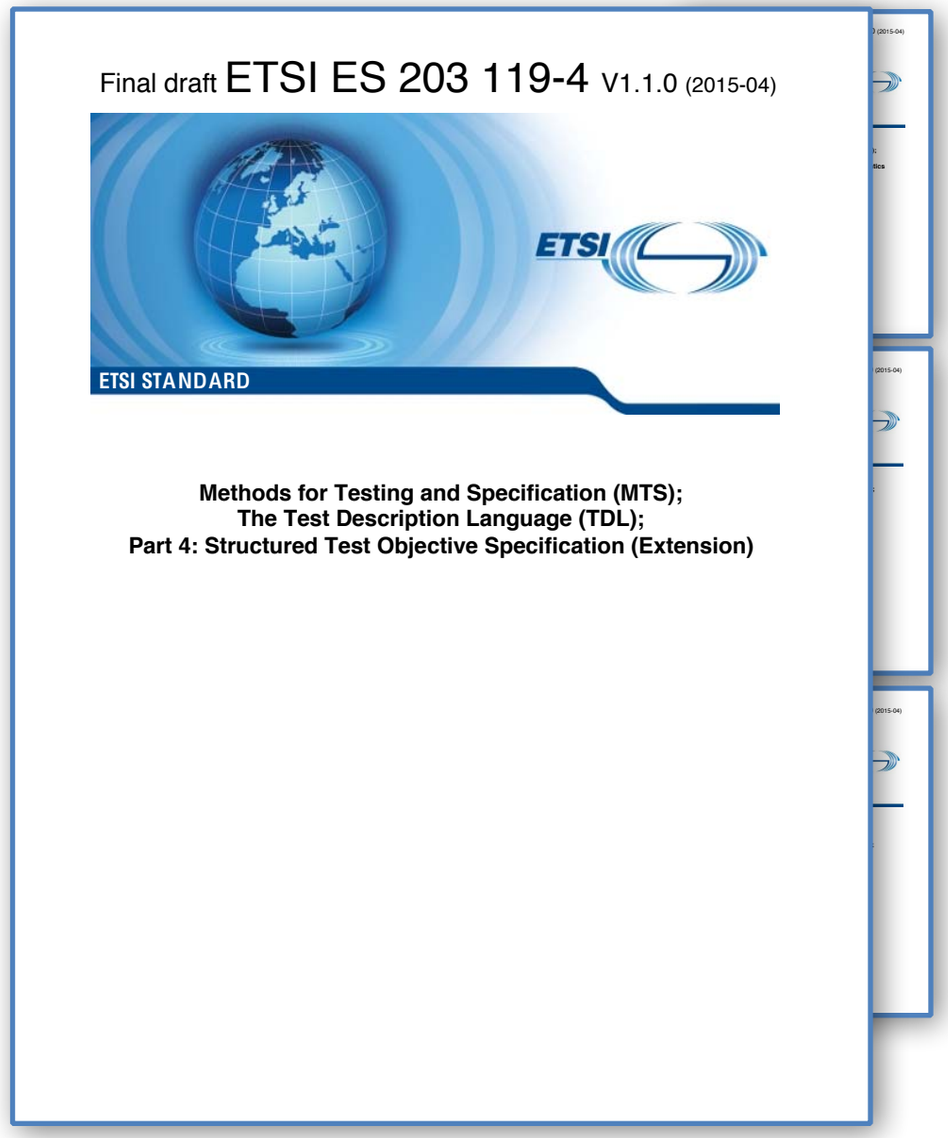
What is TDL? Part 4: TO

- 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



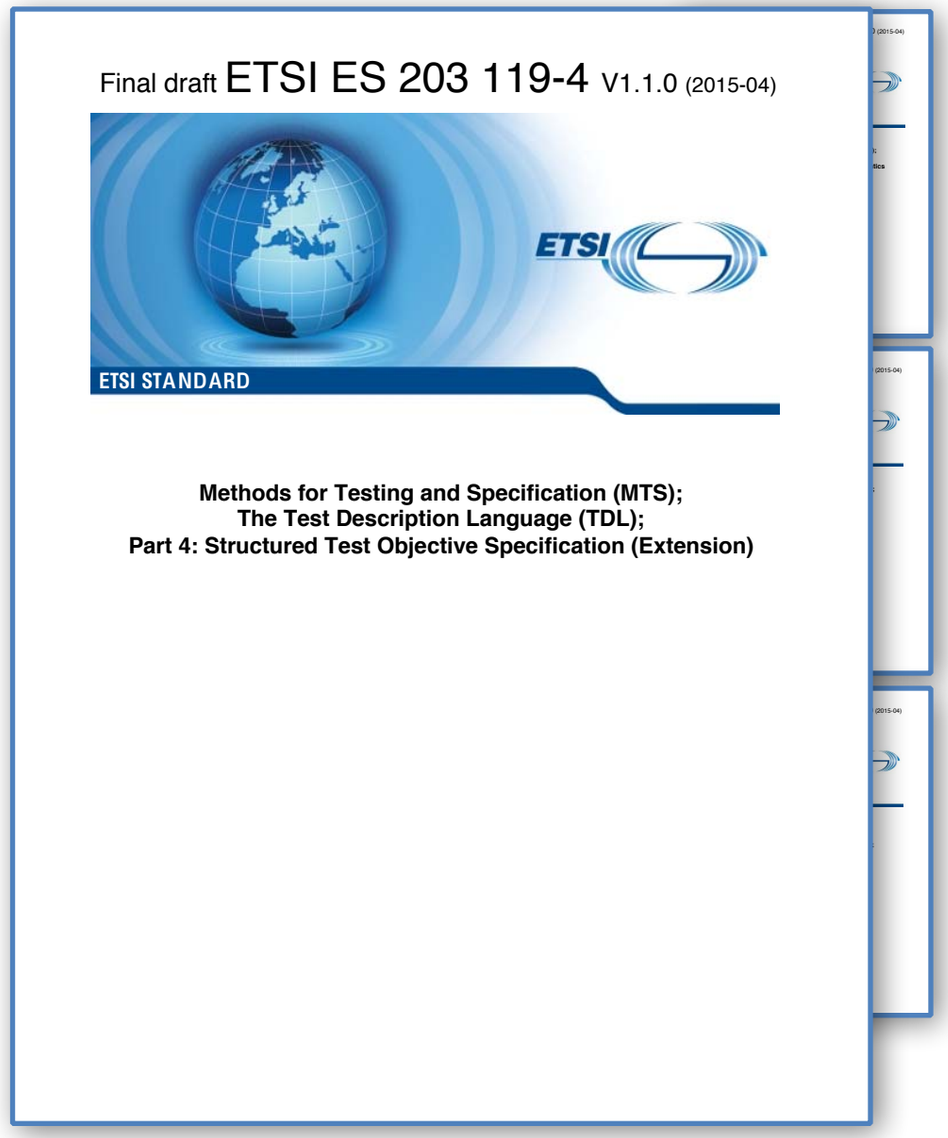
What is TDL? Part 4: TO

- 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



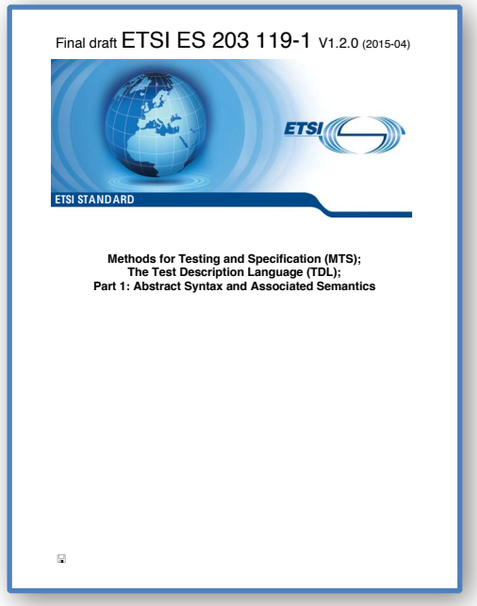
What is TDL? Part 4: TO

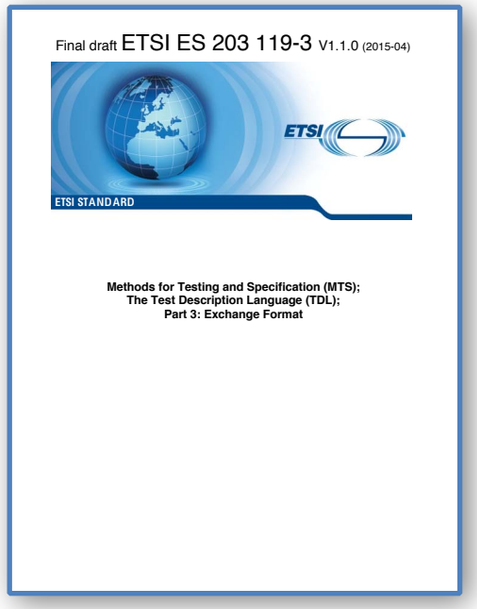
```
Test Purpose {
    TP Id "TP/CAM/INA/DOP/BV/02"
    Test objective "Checks that CAM message includes
                    DoorOpen information 30s after closed"
    Reference "TS 102 637-2 [1], clauses 7.1 and 7.2"
    PICS Selection PICS_PUBTRANSVEH
    Initial conditions
    with {
        the IUT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

- Ultimately standards need…
  - maturity
  - transparent change management
  - proof by implementation
  - validation by tests





Transparent Change Management

ETSI's Bug Tracker

Logged in as: *makedonski* (Philip Makedonski - manager) *13-09-2014 22:00 IST* Project: TDL TDL Switch

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

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

Viewing Issues (1 - 10 / 10) [Print Reports] [CSV Export] [Excel Export]

	P	ID	#	Project	Severity	Status	Updated	Summary
	⇈	0006768	1	TDL meta-model	**major**	assigned (Andreas Ulrich)	01-08-2014	New MM element as the starting point of the Behaviour Description of a Test Description
	–	0006773		TDL	feature	assigned (Andreas Ulrich)	31-07-2014	Accessing DataProxy arguments
	⇈	0006765		TDL meta-model	**major**	assigned (Andreas Ulrich)	31-07-2014	Time Observation
	–	0006764	1	TDL meta-model	minor	assigned (Andreas Ulrich)	31-07-2014	Description of VerdictType shall be modified
	–	0006763		TDL meta-model	minor	resolved (Andreas Ulrich)	31-07-2014	Blocks of ParallelBehaviour should be able to declare Guards
	–	0006767	1	TDL	minor	resolved (Andreas Ulrich)	11-07-2014	Allow to reference test descriptions that run on a different test (sub-) configuration
	–	0006772		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Variable assignment from Interaction and ActionReference
	–	0006771		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Component variables
	–	0006770		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Named parameters
	–	0006769		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Move parameters from DataInstance to DataSet

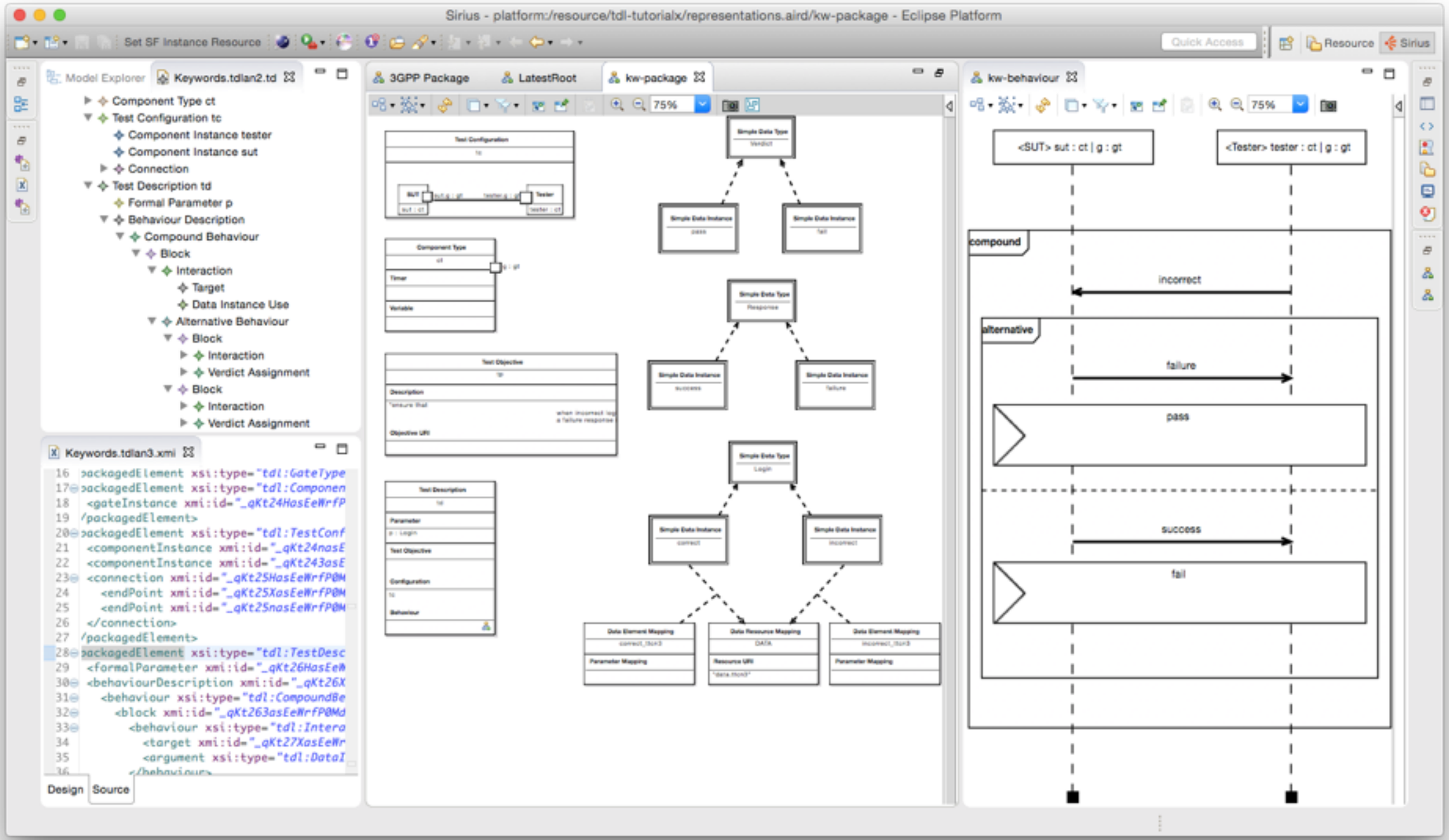
Select All Move OK

new feedback acknowledged confirmed assigned resolved closed

MantisBT 1.2.14 [^]
Copyright © 2000 - 2014 MantisBT Team
denis.filatov@etsi.org

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

Proof by Implementation



TDL Website at tdl.etsi.org

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)


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

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

What is TDL?

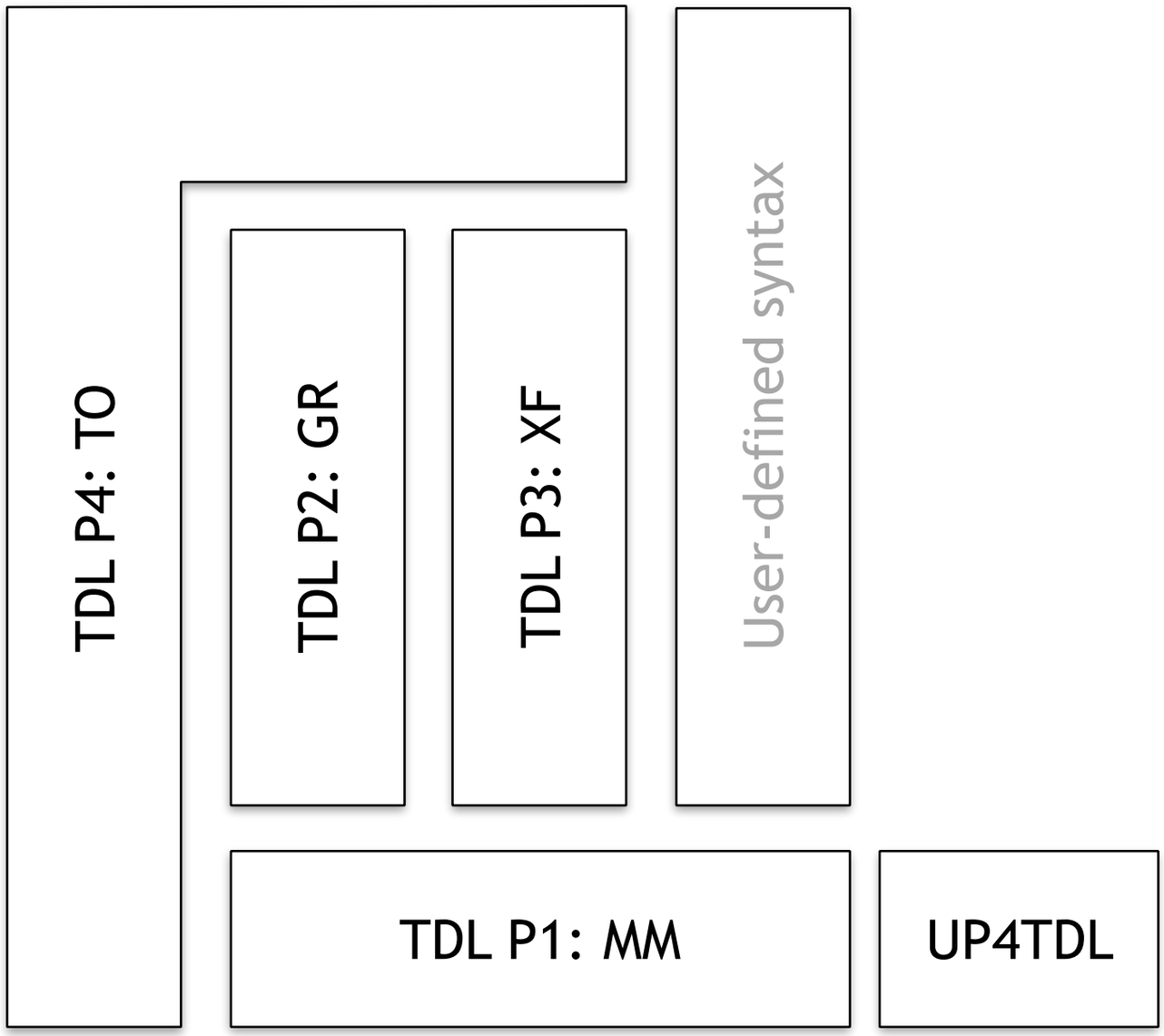
- Test Description Language
  - Design, documentation, and representation of formal test descriptions
  - Scenario-based approach
- Standardised at ETSI by TC MTS
  - STF 454 (2013)
  - STF 476 (2014)
  - STF 492 (2015)

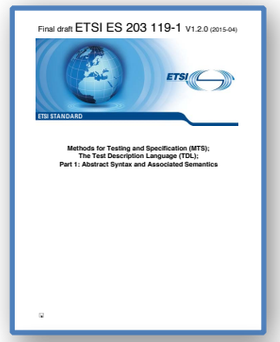

Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

ETSI STANDARD

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

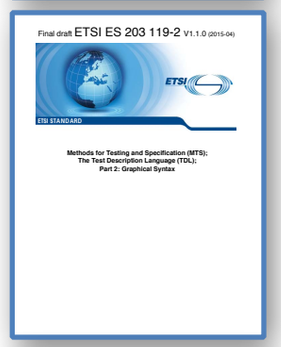
What is TDL?




Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

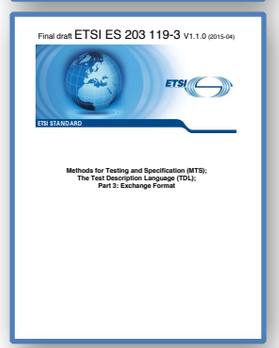
ETSI STANDARD

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


Final draft ETSI ES 203 119-2 V1.1.0 (2015-04)

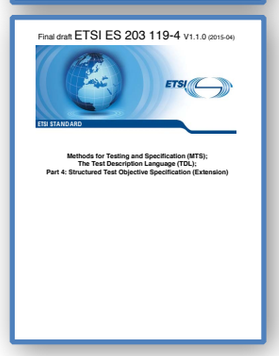
ETSI STANDARD

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


Final draft ETSI ES 203 119-3 V1.1.0 (2015-04)

ETSI STANDARD

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


Final draft ETSI ES 203 119-4 V1.1.0 (2015-04)

ETSI STANDARD

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

UML Profile 4 TDL, but why?

“to enable its interoperability with and application in UML-based working environments and model-based testing approaches”

Examples of UML Base *approaches* :

- Use UML Based Tool such Papyrus to Edit «UP4TDL» models, Moka to animate a TDL TestDescription
- Use TDL on SysML model
- Use PhiSystem profile for CPS
- Use UTP to model tests

UML Profile 4 TDL, but why?

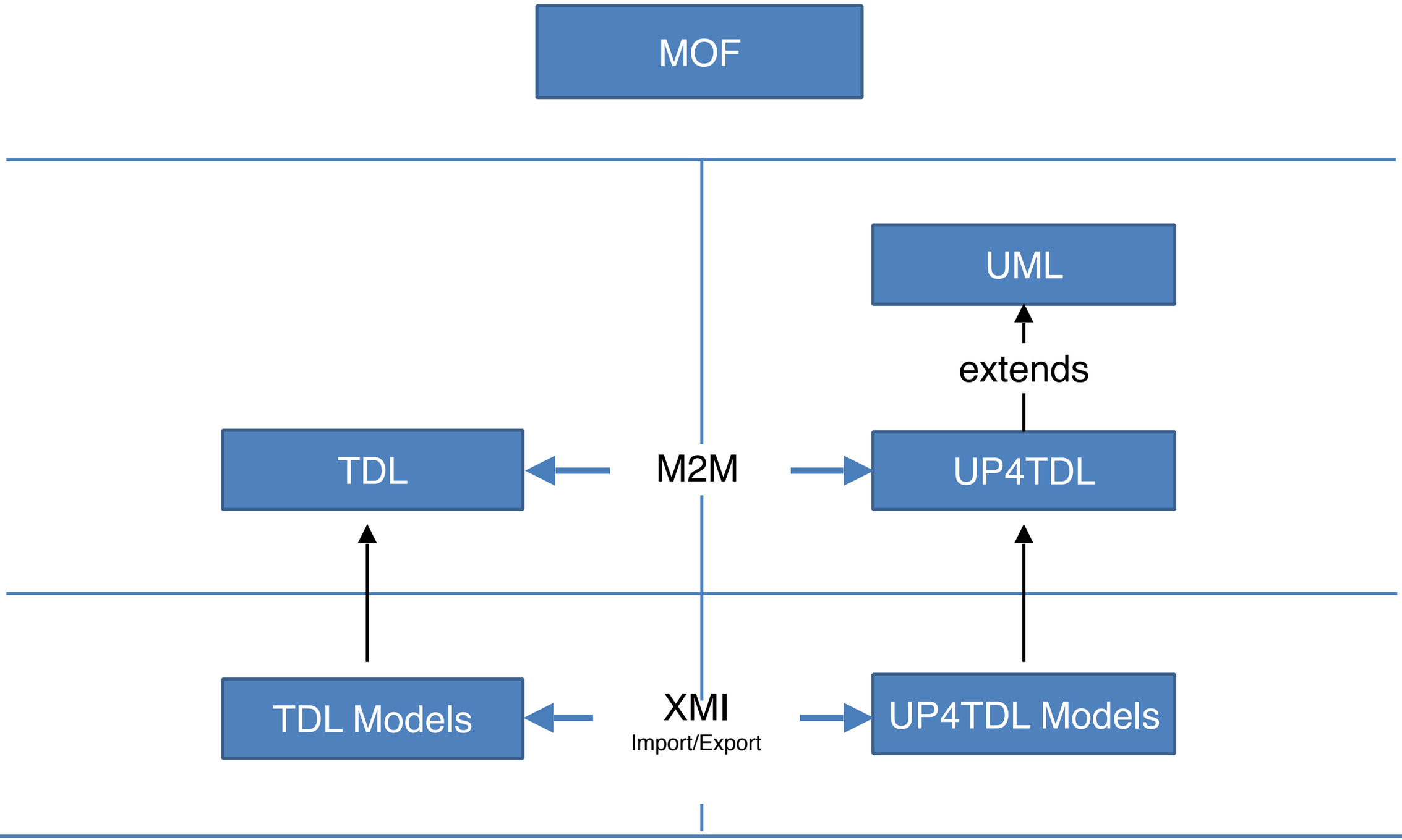
- Define a domain specific terminology, i.e. a domain specific notation instead of the plain UML2 notation.

- Complete/specialize the UML2 semantics for dealing with:
  UML Semantics Variation Points,
  For clarifying ambiguous definition,
  For specializing an existing semantics aspect of UML2.

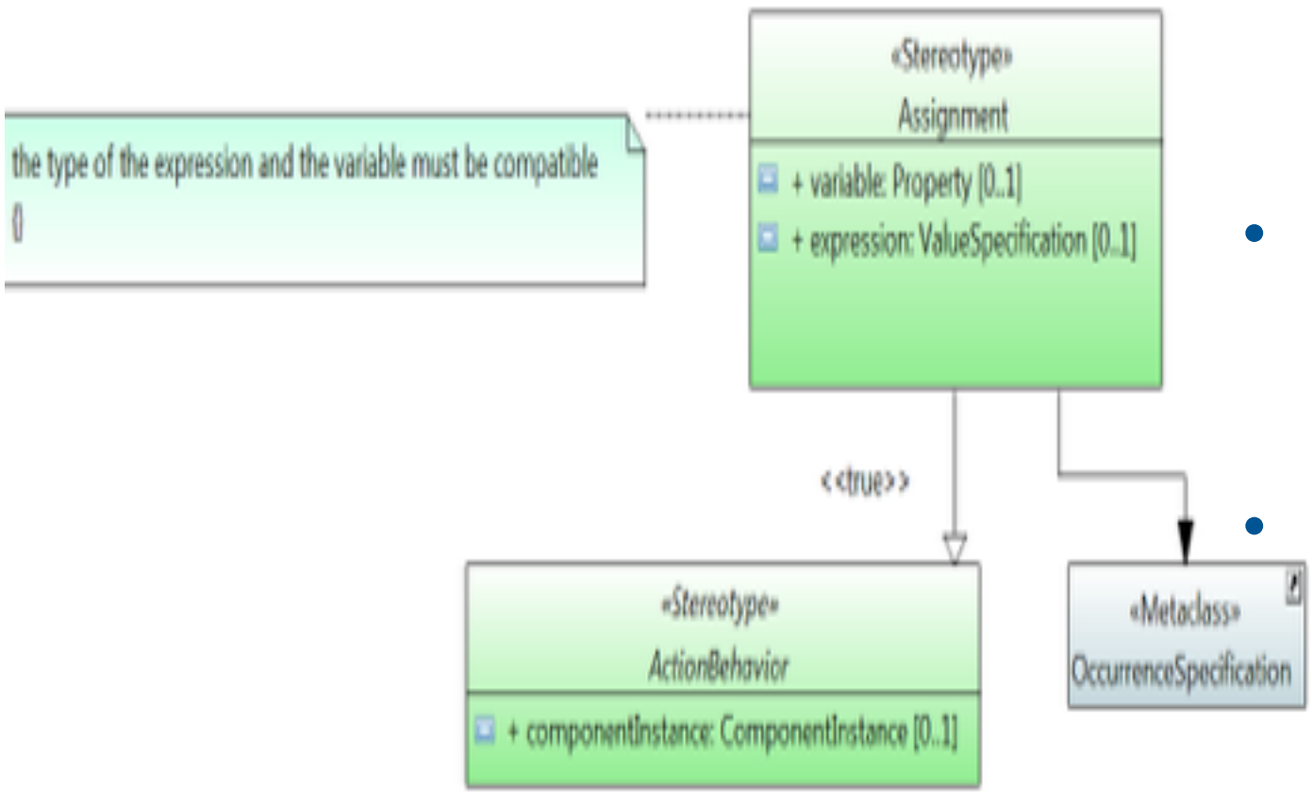
- Define usage constraints of the UML2 in order to drive/limit its usage
  e.g. for defining a domain specific methodology limiting the scope of UML.

- Define new meta-information for annotating a model for a given purpose
  e.g. for code generation purpose, for enabling model-based analysis such as quality performance analysis, etc.

UP4TDL, TDL or UML?



Extension Block: The stereotype



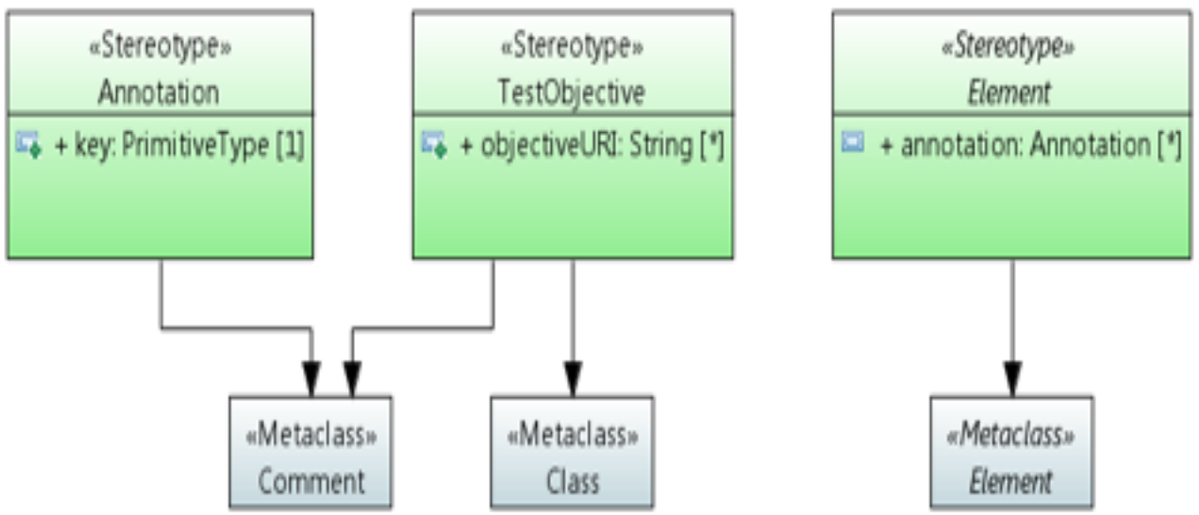
- **Extension**
  - Associations, Specializations, constraining, add properties (also called TaggedValues)
- **Example of property**
  - TestDescription holds the TestConfiguration it refers to.
- **Example of constraint**
  - Block : all MessageOccurence related to the same message shall be consecutive
  - TestDescription shall have a TestConfiguration
  - ComponentInstanceBinding : same/ compatible ComponentType

Foundation Package

Profile Definition

Concept Overview:

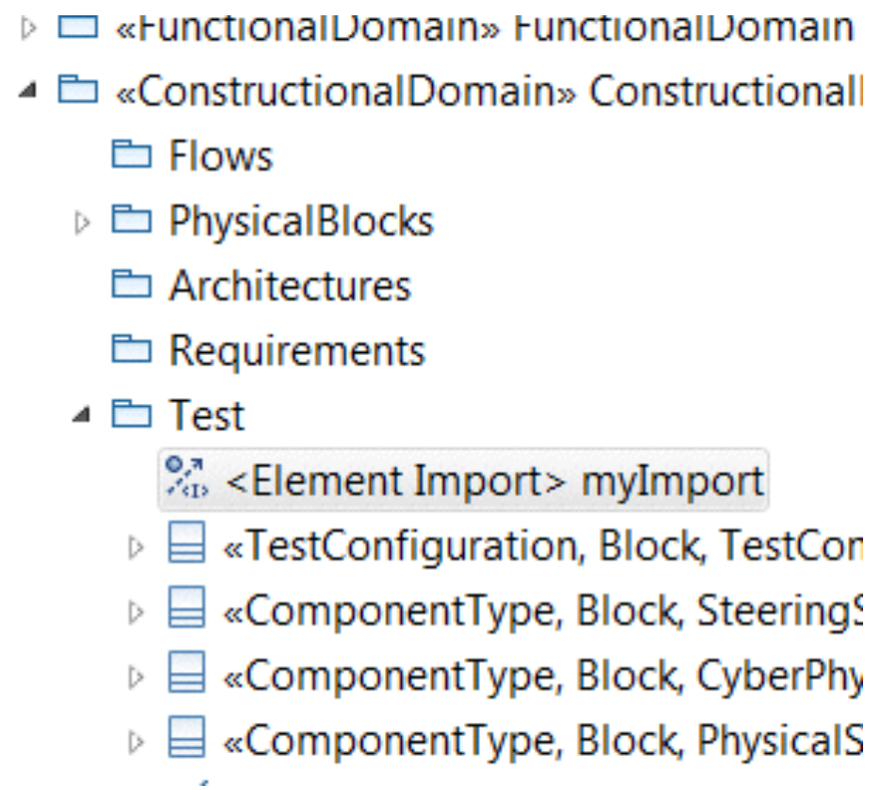
- Most concepts already in UML
- Main additions : Annotation, TestObjective



Profile Application

Functionality overview:

- Use UML “Packaging”
- Add Annotation & TestObjectives

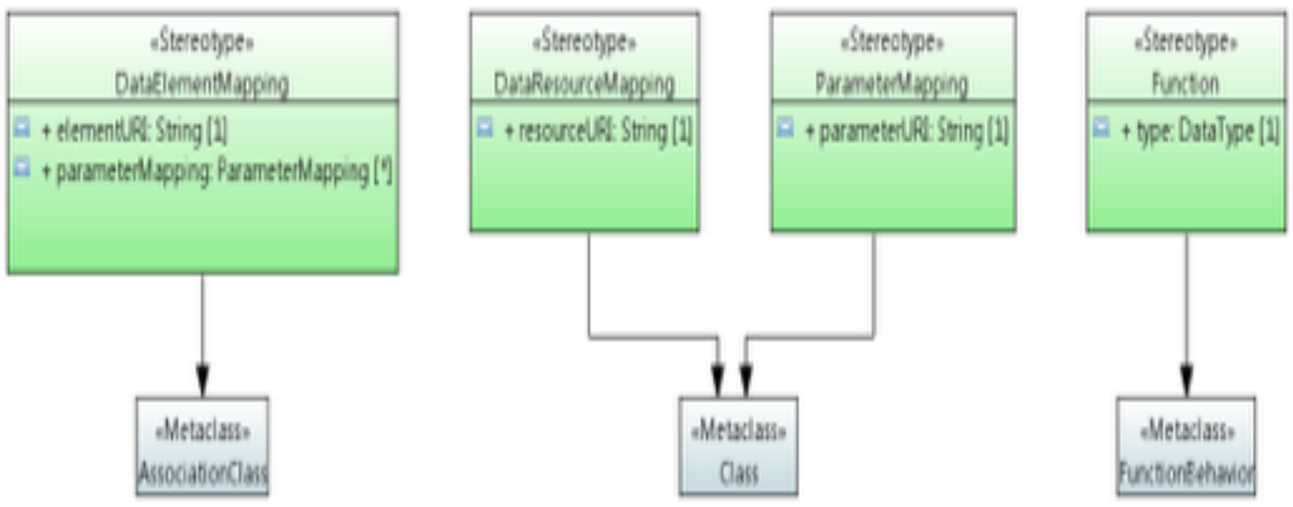


Data Definition Package

Profile Definition

Concept Overview:

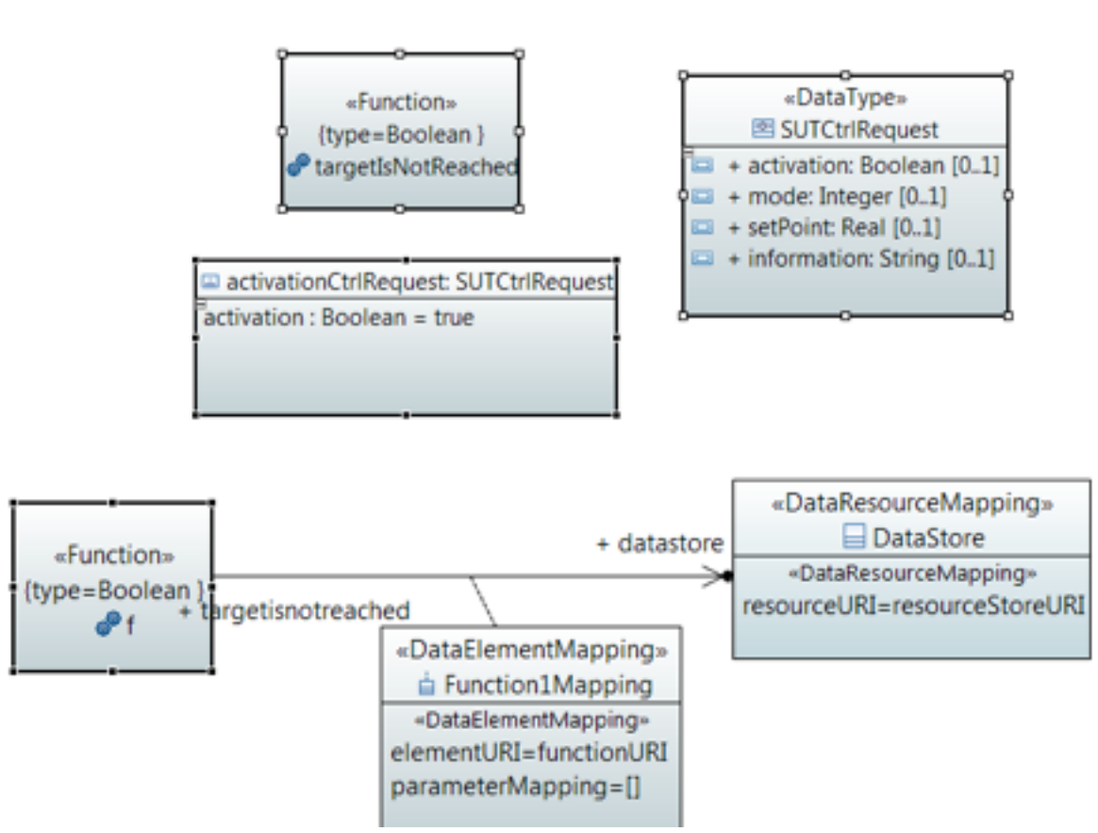
- Rely mostly on concepts present in the UML Class Diagram
- Main additions : DataMapping Concepts & Function



Profile Application

Functionality overview:

- Declare Types, Instance, Actions and Functions
- Map those Elements to a Resource

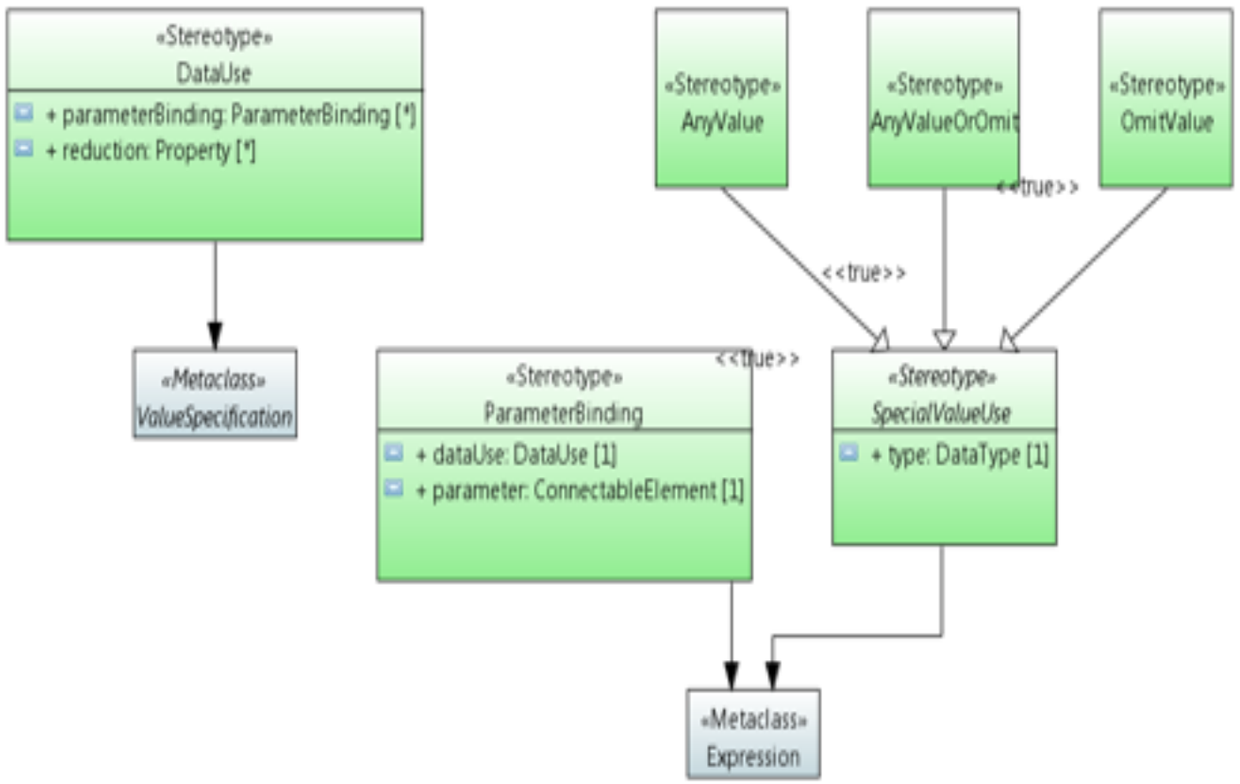


DataUse Package

Profile Definition

Concept Overview:

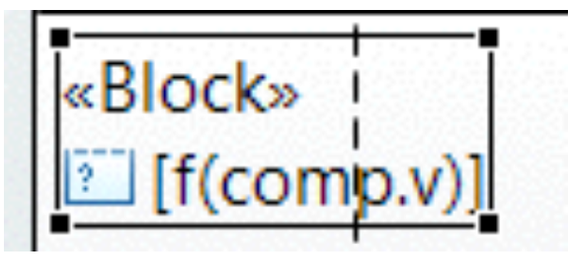
- Mostly extend UML::Expression
- aim: build a Data Use grammar



Profile Application

Functionality overview:

- Call functions, use variable, use Instance in:
  - Guards
  - Arguments of AtomicBehaviours (e.g. Interaction)

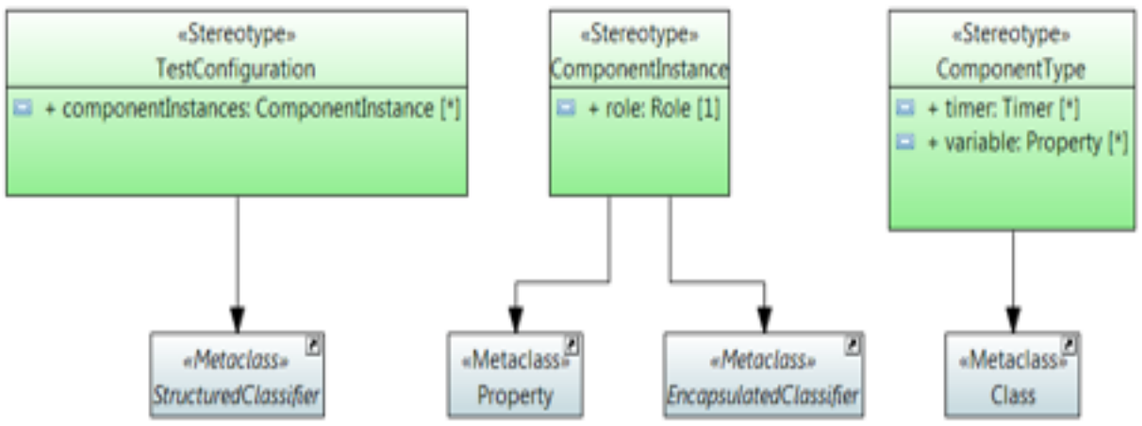


TestConfiguration Package

Profile Definition

Concept Overview:

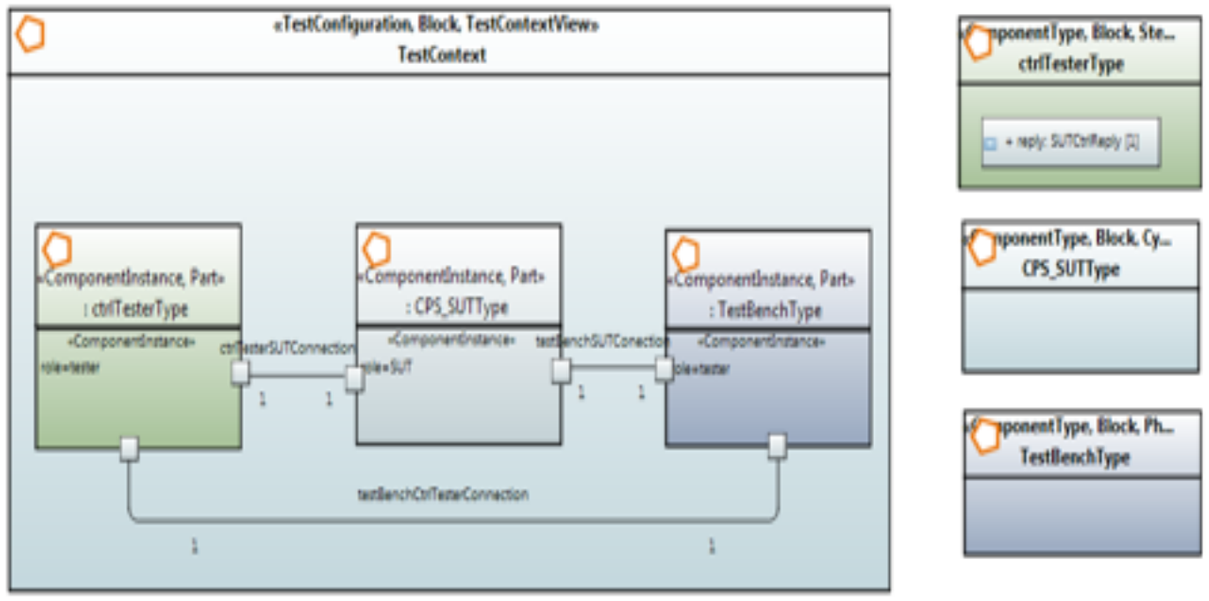
- Rely on concepts involved in CompositeStructure Diagram
- Allows combination with SysML



Profile Application

Functionality overview:

- Edit TestConfiguration
  - Add ComponentInstance
  - Declare ComponentType
  - Add Gates & Connections

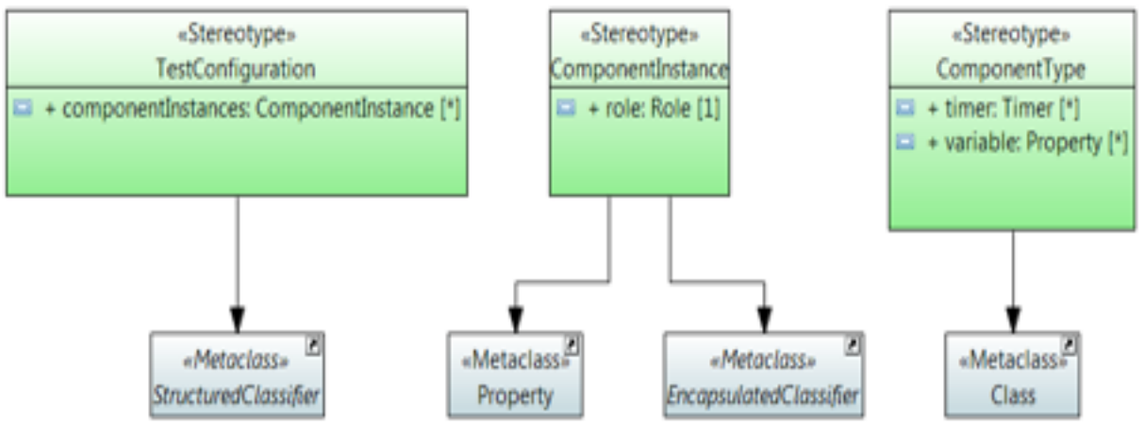


CombinedBehaviour Package

Profile Definition

Concept Overview:

- Stereotype extending CombinedFragment
- A Block Stereotype that extend InteractionOperand



Profile Application

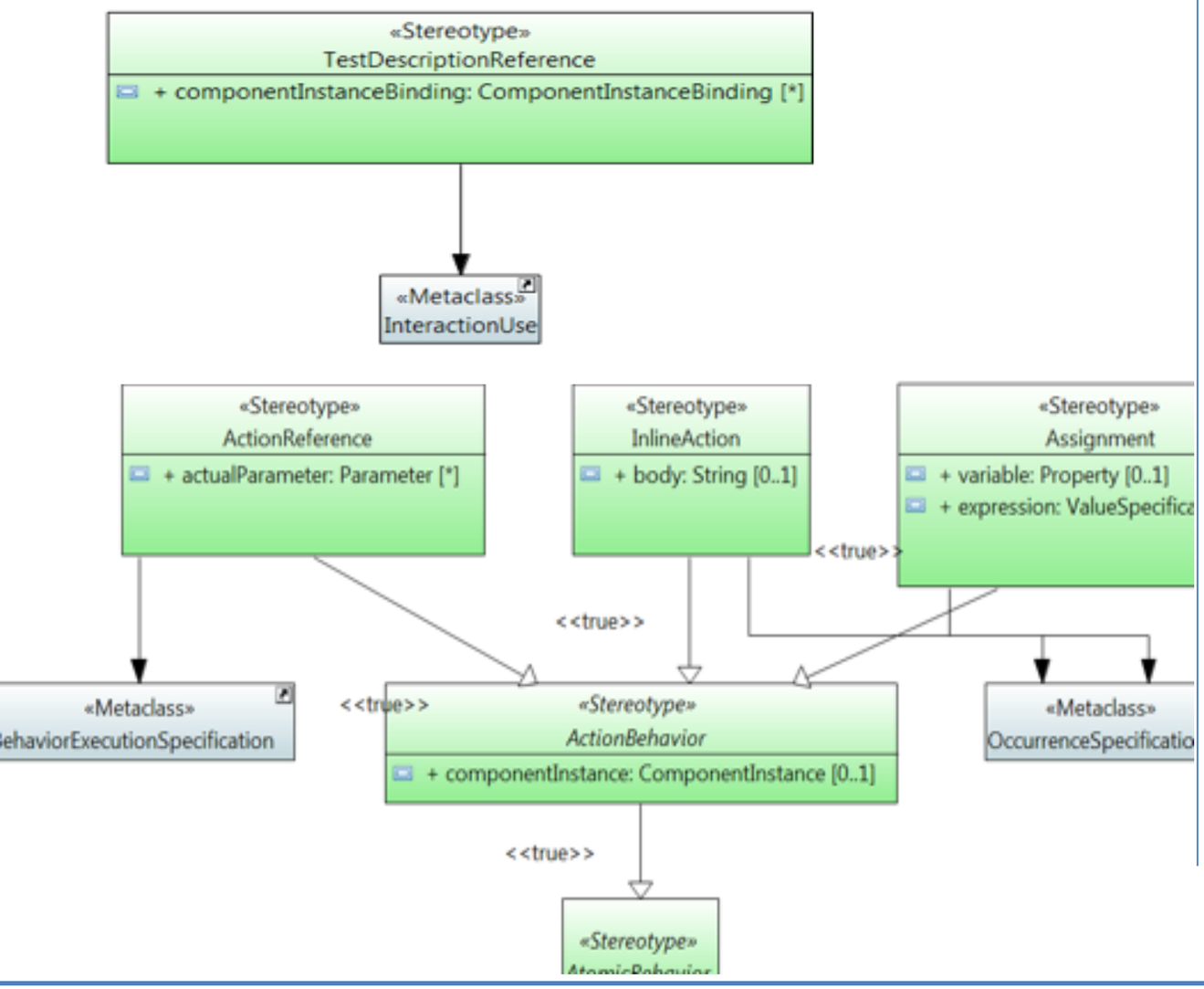
Functionality overview:

- Create CombinedBehaviour & add blocks to these behaviours

AtomicBehaviour Package : Actions

Profile Definition

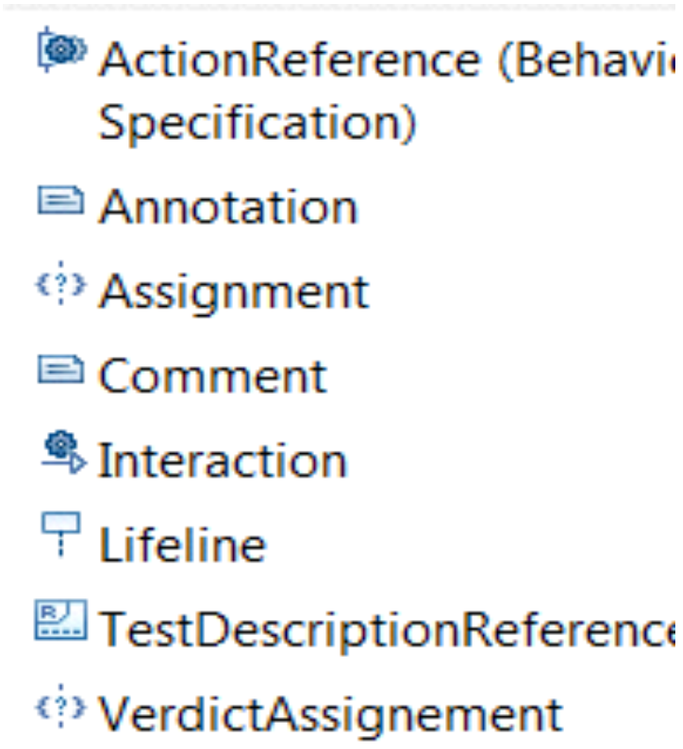
Concept Overview:



Profile Application

Functionality overview:

- Add/Edit Actions, Verdict Assignment, TestDescriptionReference to a TestDescription

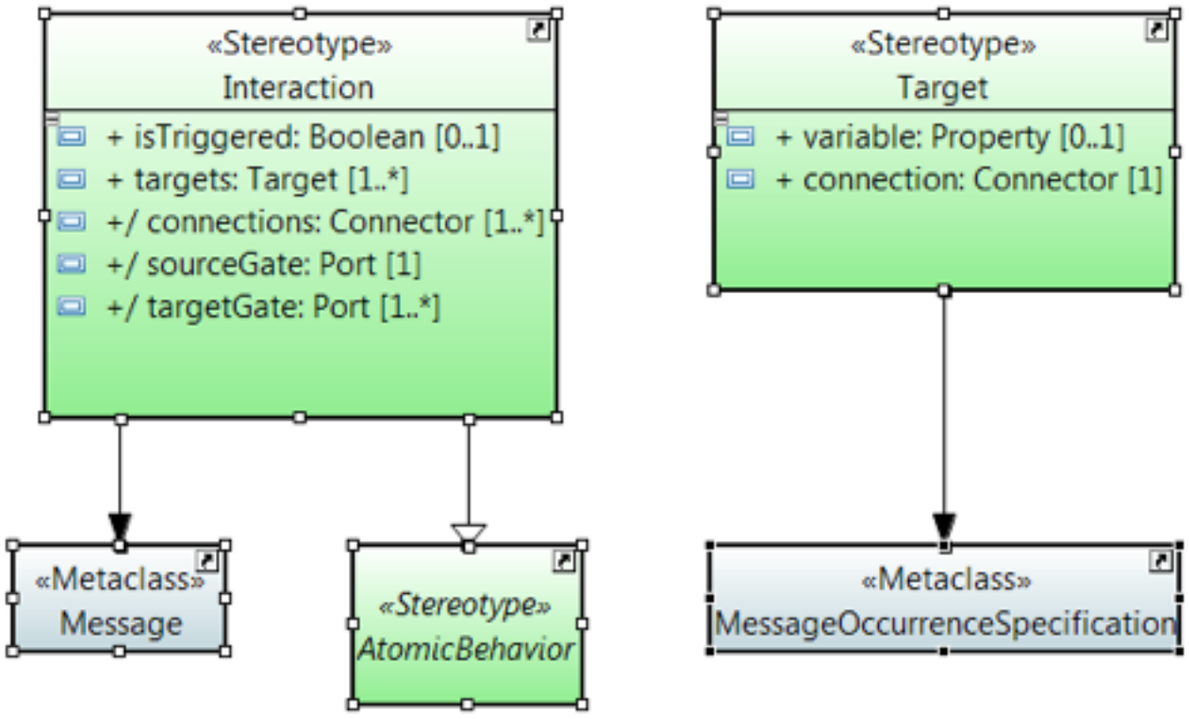


AtomicBehaviour Package : Interaction

Profile Definition

Concept Overview:

- Extension of OccurrenceSpecification



Profile Application

Functionality overview:

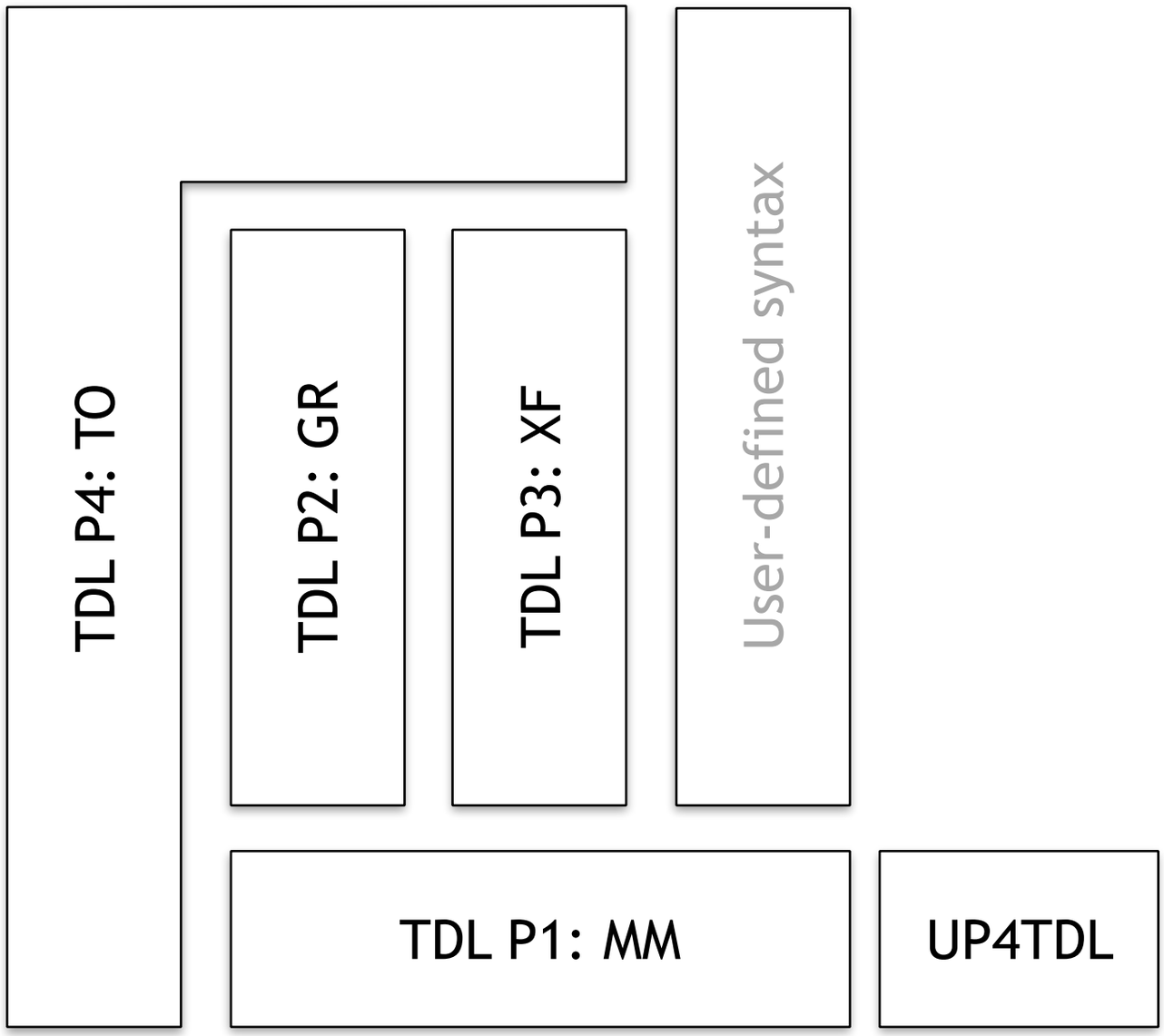
- Add Interaction, edit its argument

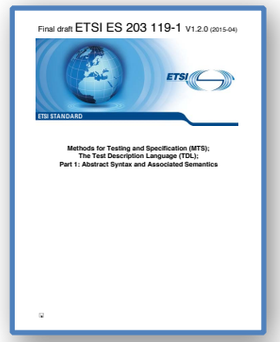
Editing UP4TDL Models

Papyrus-Based Editor:

- Planned :
  - Automatic model validation
  - Xtext editor For DataUse
  - Extended Wizards for TDL projects & TDL diagrams
  - Implementation of the TDL graphical syntax
- OpenSource Eclipse Editor for UMLExtension of 3 UML Diagrams :
  - Composite Structure Diagram for TestConfiguration
  - Sequence Diagram for TestDescription
  - Class Diagram for Data Declaration

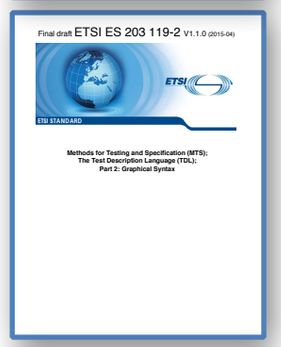
What is TDL?




Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)

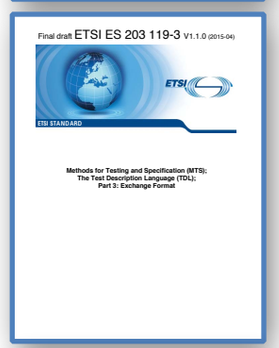
ETSI STANDARD

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


Final draft ETSI ES 203 119-2 V1.1.0 (2015-04)

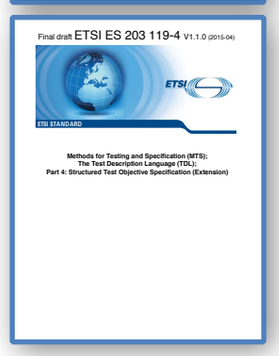
ETSI STANDARD

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


Final draft ETSI ES 203 119-3 V1.1.0 (2015-04)

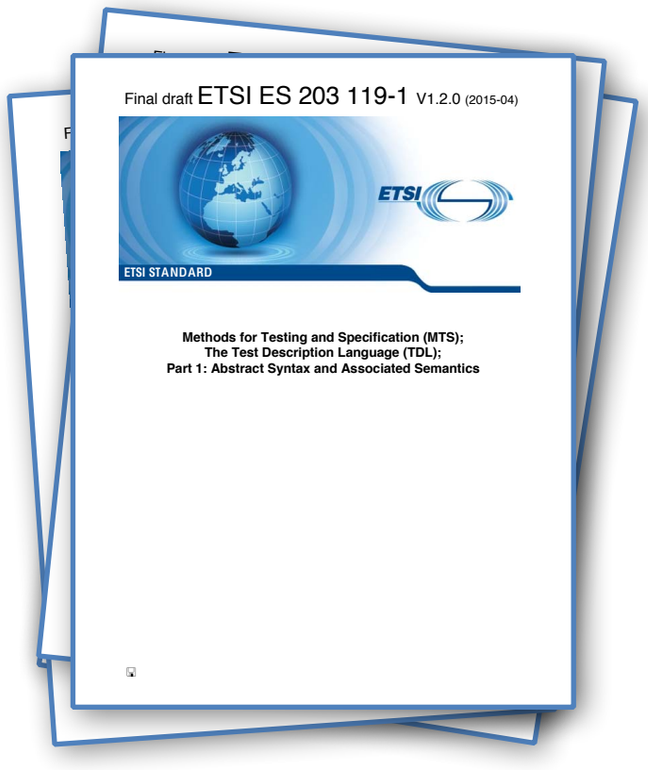
ETSI STANDARD

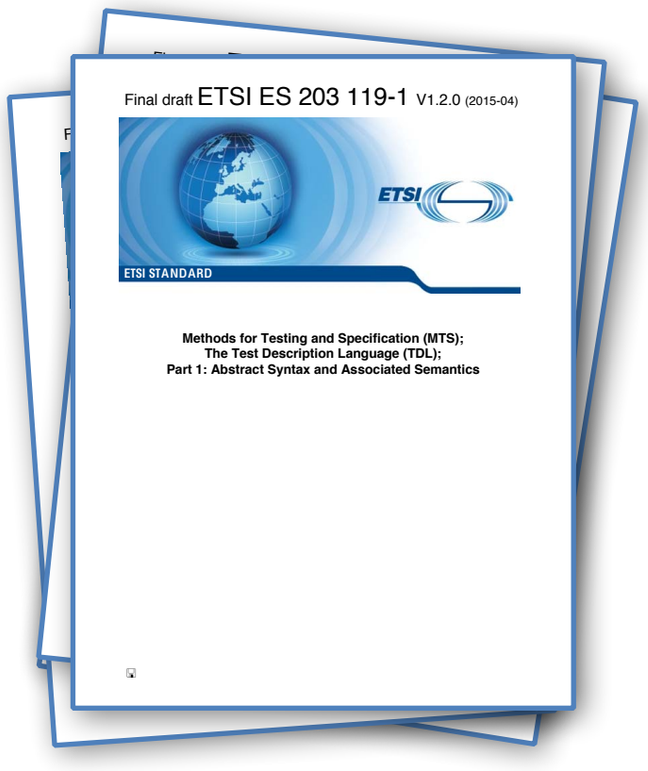
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 3: Exchange Format


Final draft ETSI ES 203 119-4 V1.1.0 (2015-04)

ETSI STANDARD

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


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)
ETSI
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics


Final draft ETSI ES 203 119-1 V1.2.0 (2015-04)
ETSI STANDARD
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 1: Abstract Syntax and Associated Semantics

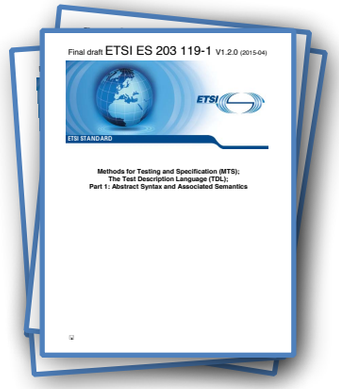
Where does TDL fit in?

Keyword-Driven Testing

MBT

Representation

Generation



Rail

Standards

Visualisation

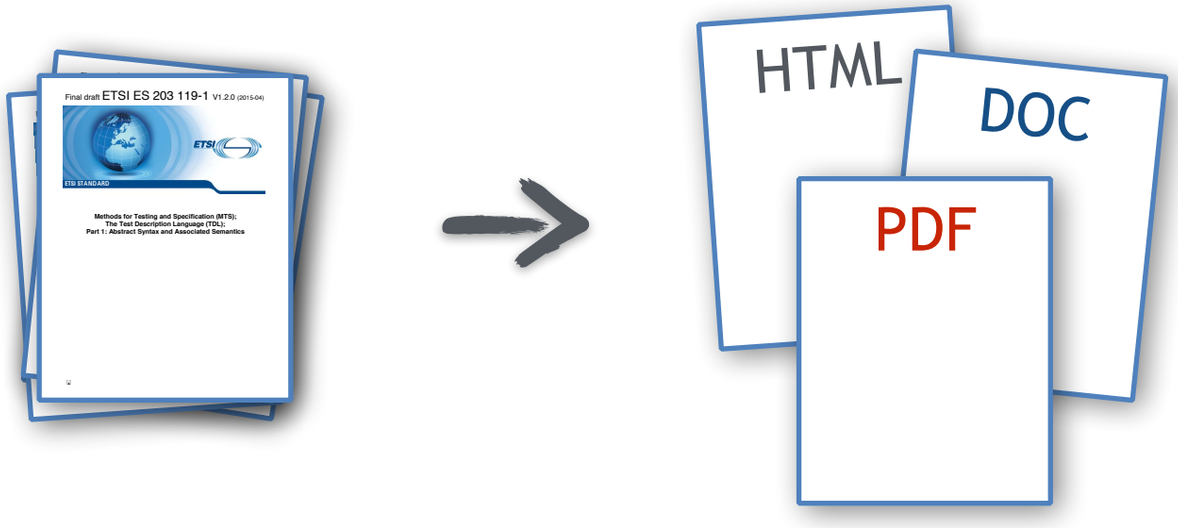
Documentation

ITS

Interoperability

Conformance

Where does TDL fit in?



Documentation

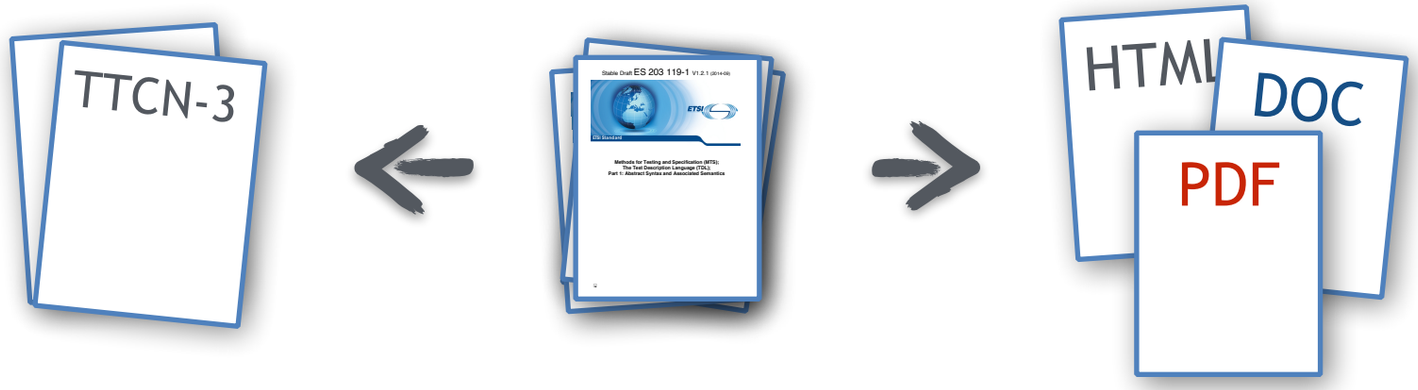
Where does TDL fit in?

Documentat

- 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 }
            }
```

7.2.2.3.2 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 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**

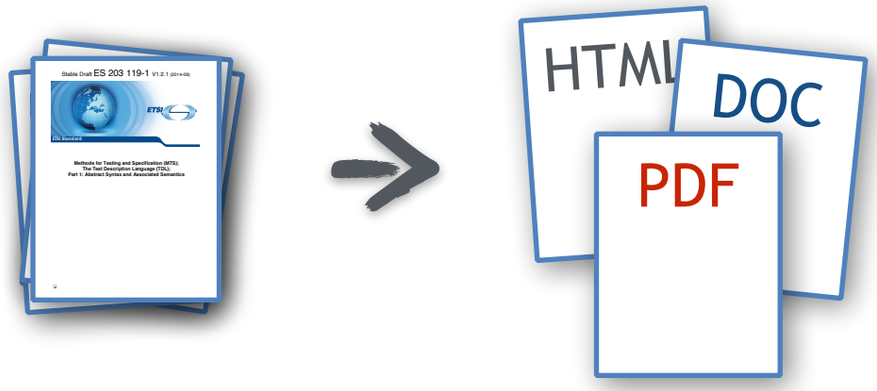
From ETSI TS 186 011-2 V3.1.1 (2011-06):

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



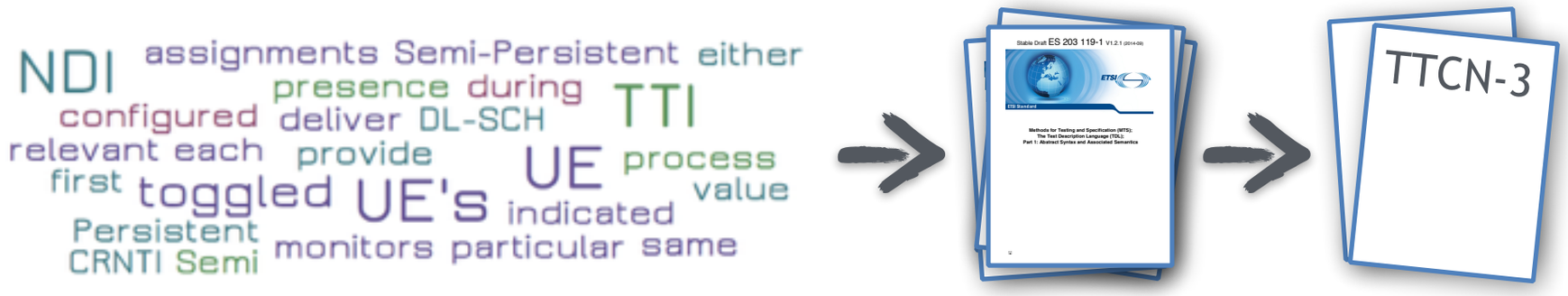
From ETSI TS 102 868-2 V1.1.1 (2011-03):

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	

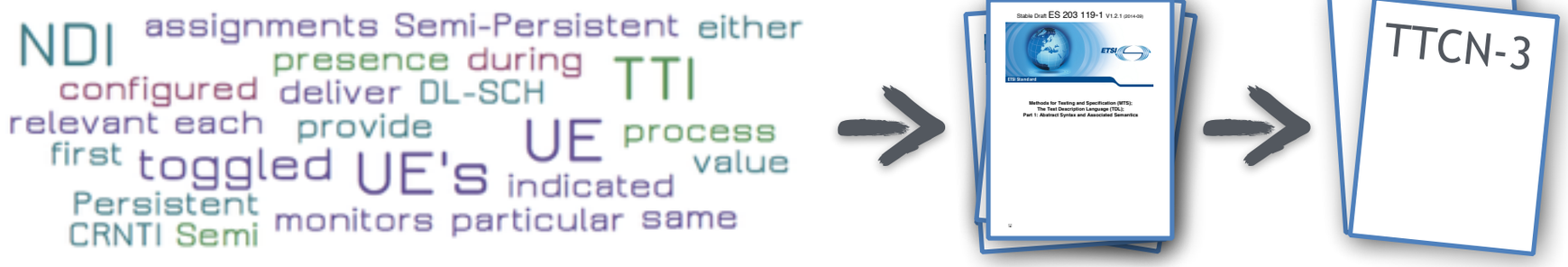
with {
the IUT being in the "initial state" and
the IUT having sent a valid CAM message
containing DoorOpen TaggedValue
}	

Expected behaviour	

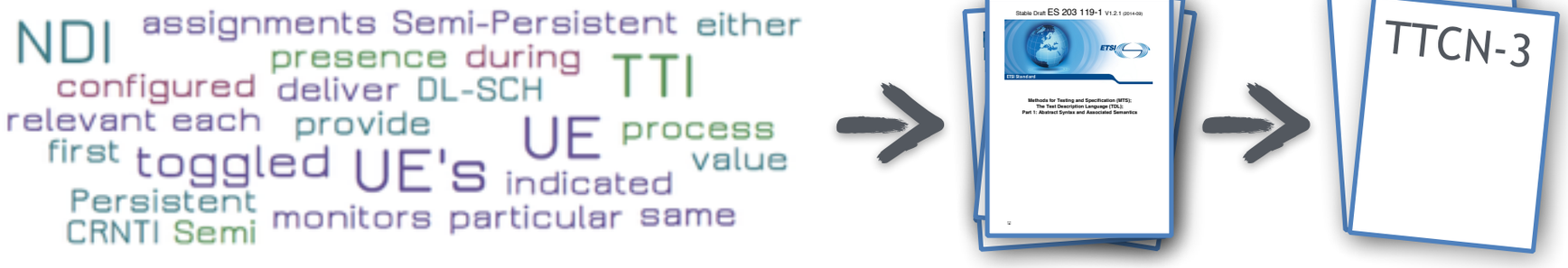
ensure that {
when {
the door is closed
}
then {
the IUT sends CAM messages
containing DoorOpen TaggedValue during the 30s following the door closing event
}
}	



- 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

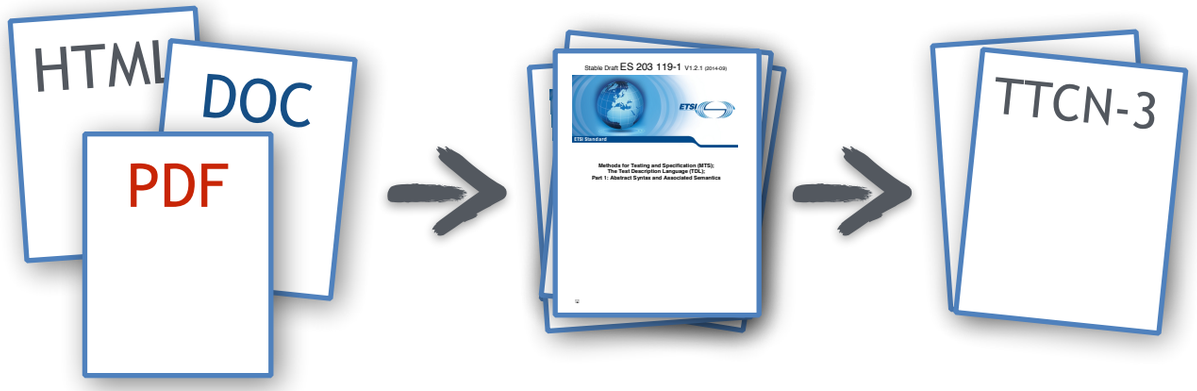


- Challenges
  - Generated test sequences
    - proprietary format - not accessible, tool-specific integrations to requirements management, test planning
    - straight to executable code - loss of meta-data, difficult parameterisation
  - Mapping between abstract (symbolic) and real test system interface
    - 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)

- 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

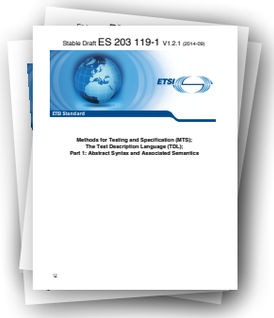
Where does TDL fit in?

Requirements
User stories

Manual test derivation
in Syntax #1

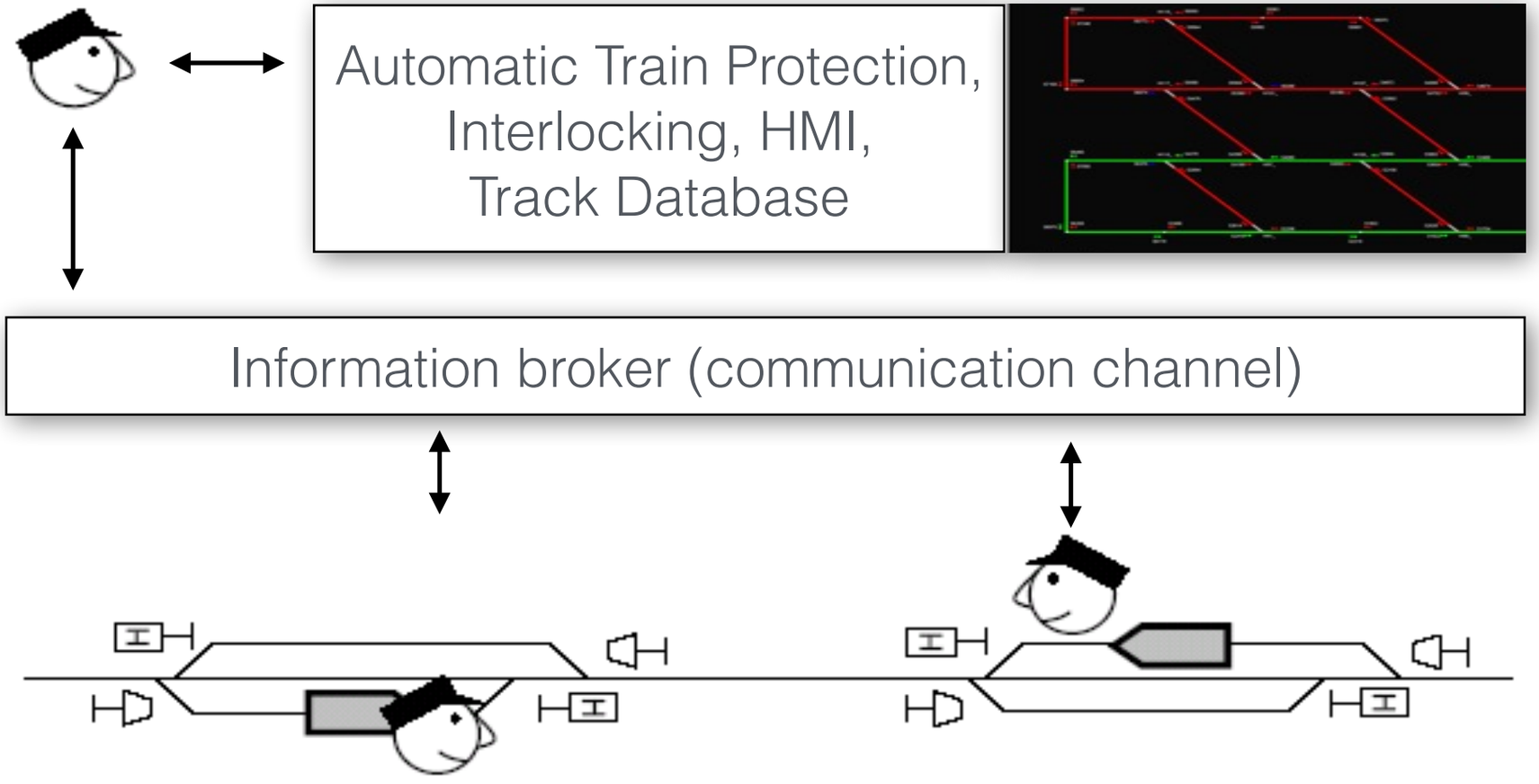
System models

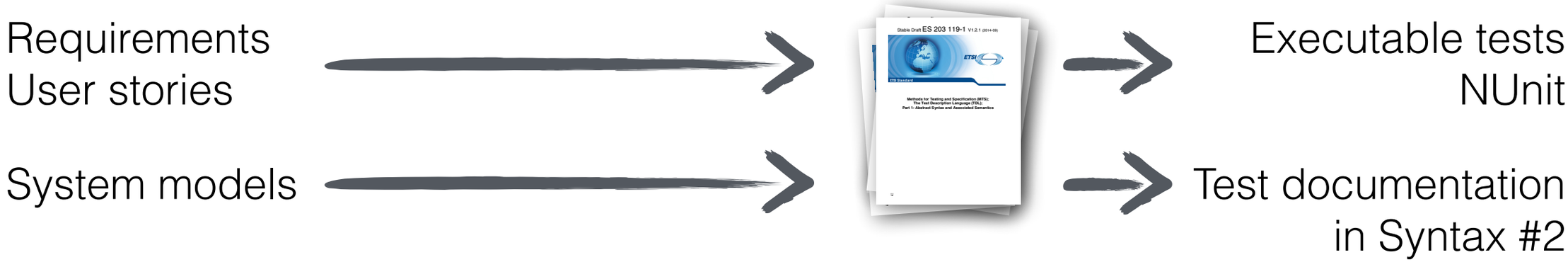
Automatic test generation



Executable tests
NUnit

Test documentation
in Syntax #2

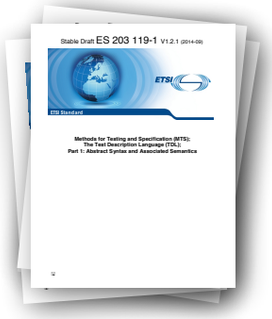




- Context
  - Testing communication between independent rail sub-systems
- Challenges
  - High-level concurrency and non-determinism
  - Multiple aspects over the whole system - safety, real-time, functionality
  - Different development techniques for different components

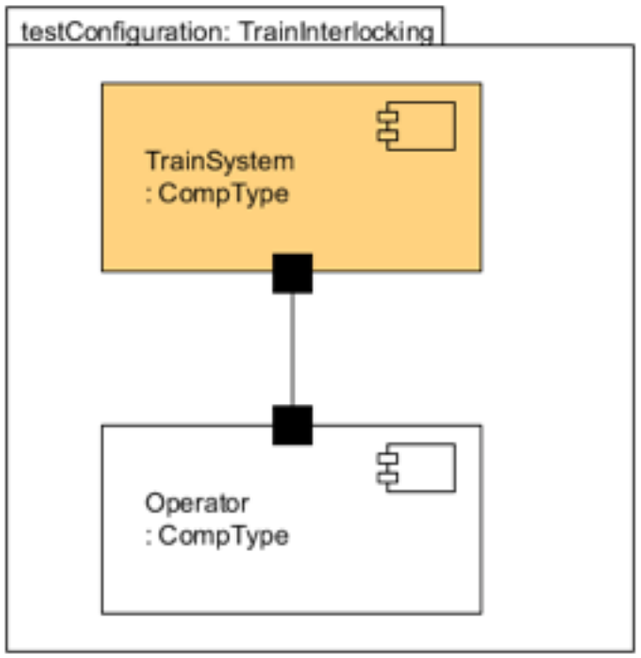
Requirements
User stories

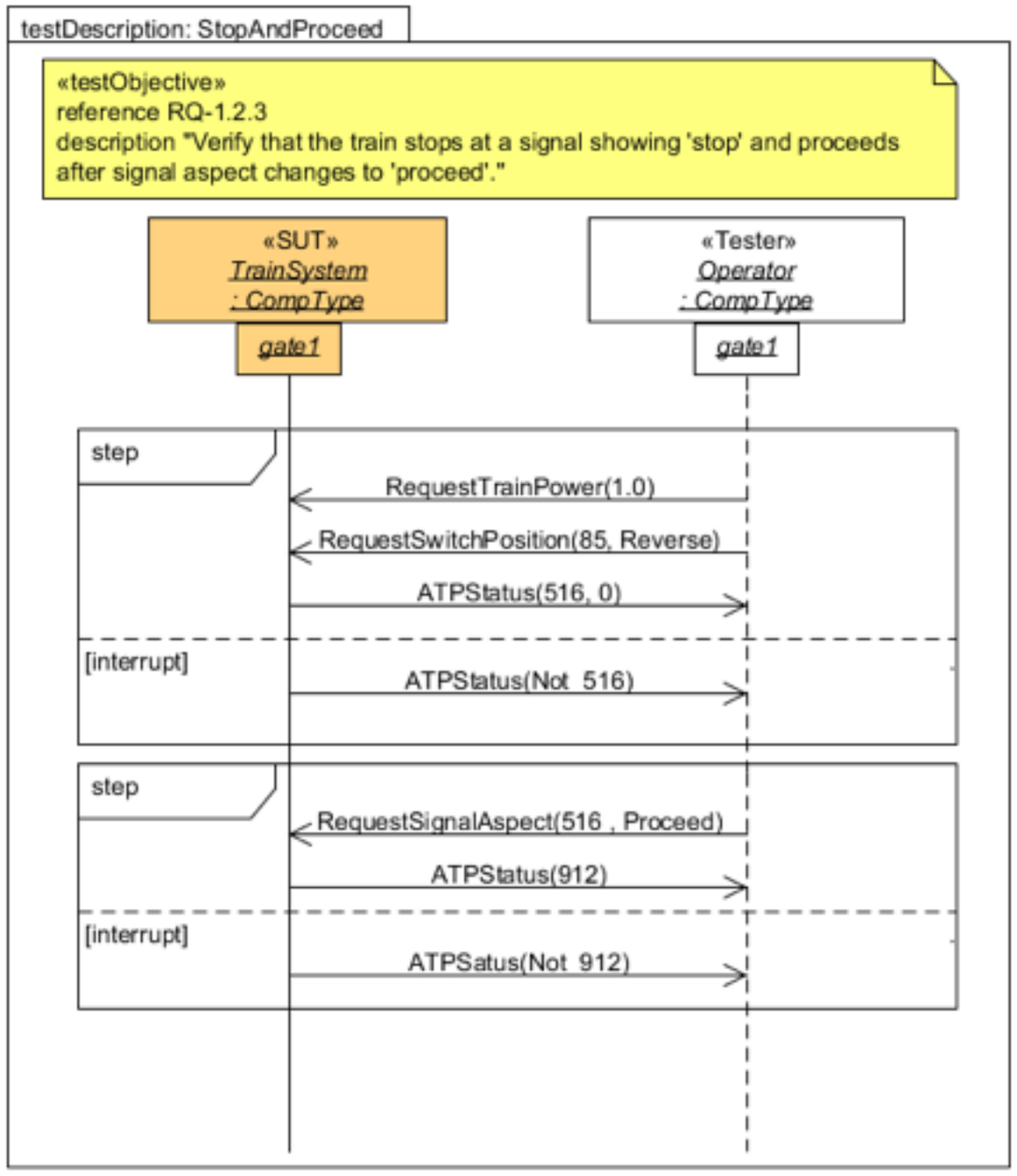
System models



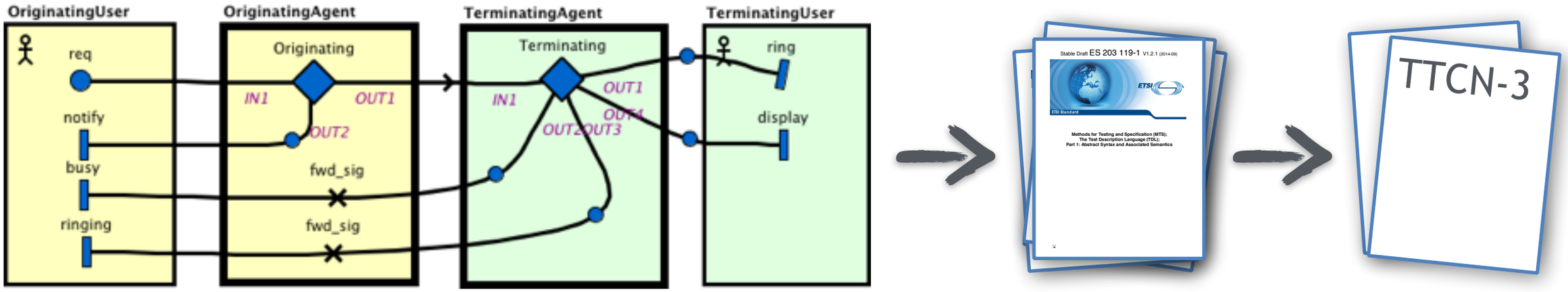
Executable tests
NUnit

Test documentation
in Syntax #2

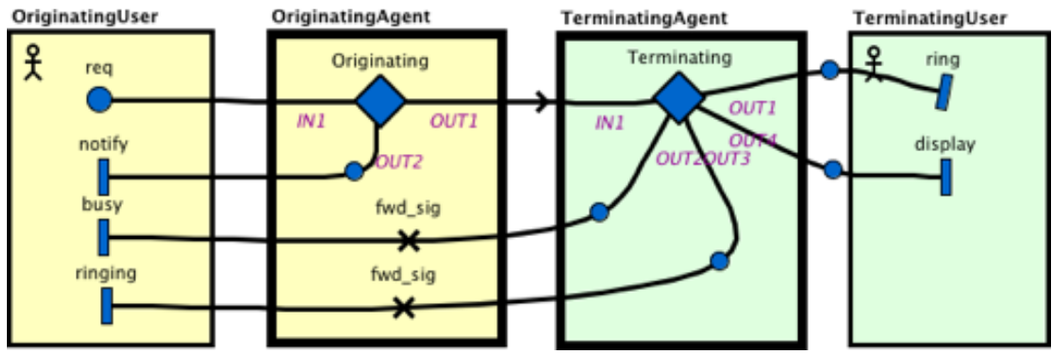


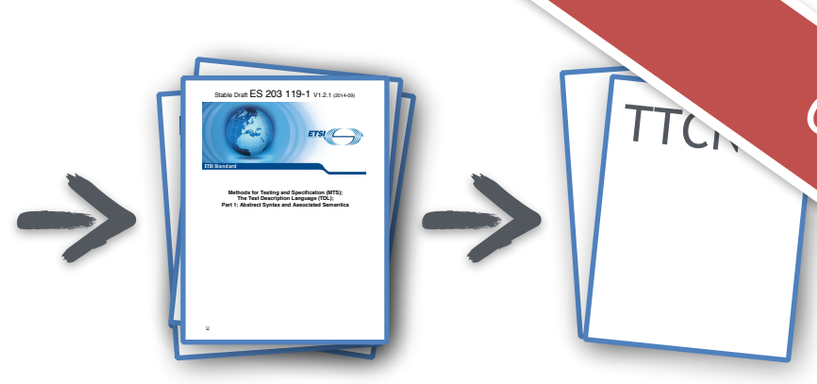


Where does TDL fit in?

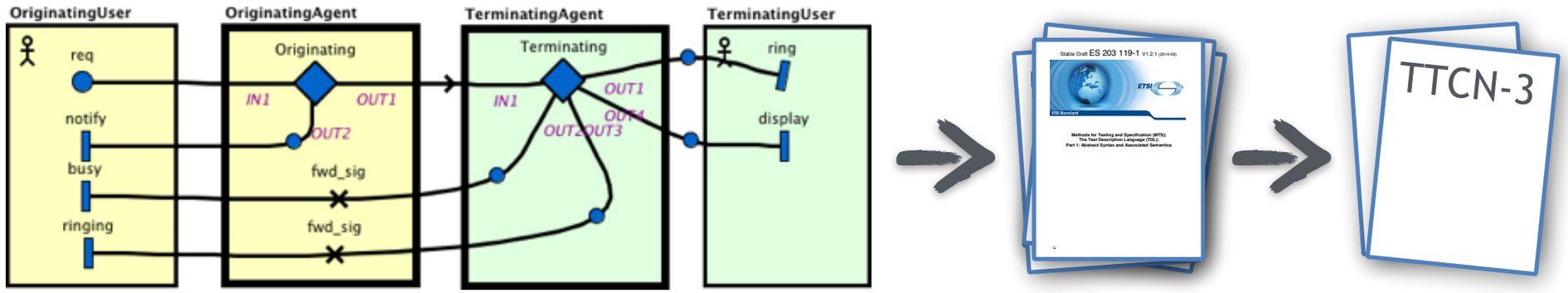


- User Requirements Notation (URN)
  - Elicitation, analysis, specification, and validation of requirements
  - Complementary views - goals (GRL) and scenarios (UCM)
  - ITU-T Recommendation Z.151 (10/12)

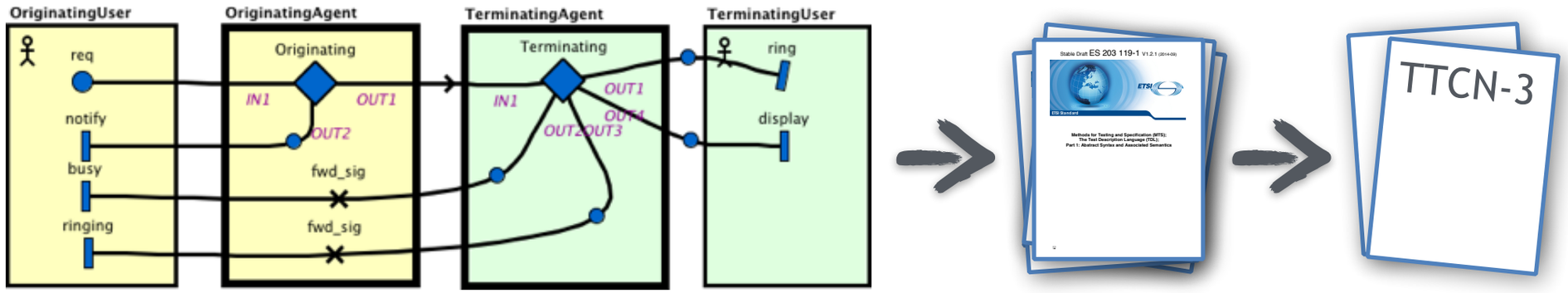




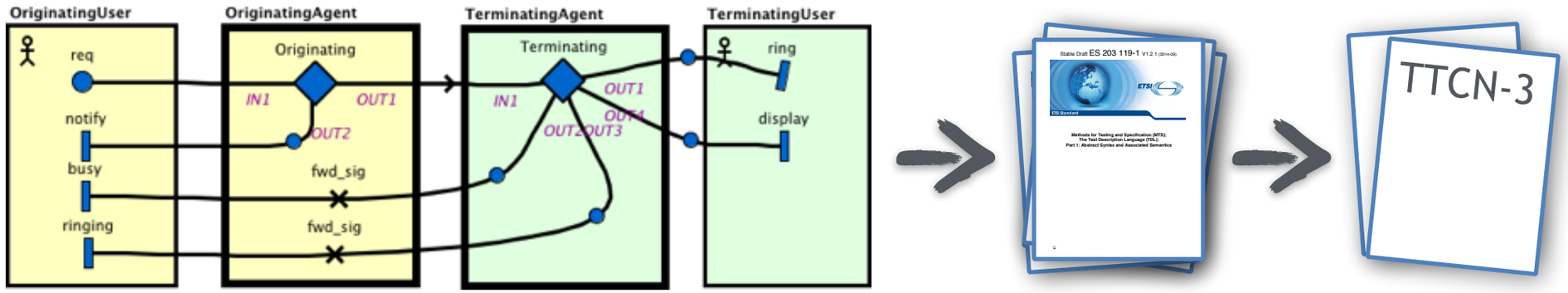
- Context
  - Test systems for cockpit systems and avionics solutions
  - Alternative means for
    - standards-based and model-based test generation and test automation
    - replace proprietary solutions
  - Transformation from high-level requirements and scenarios in UCM to TDL
  - Transformation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

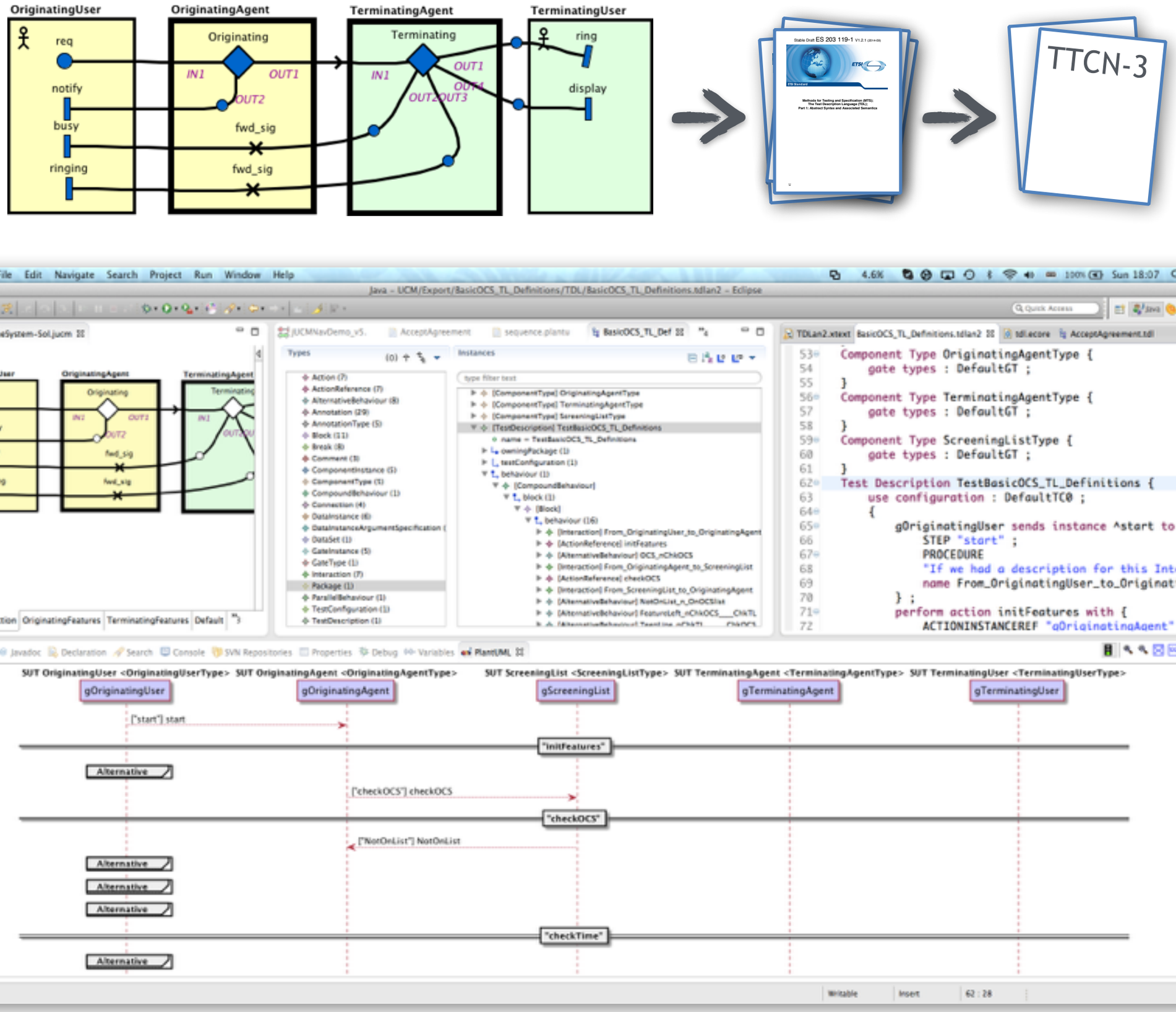


- Motivation
  - Tree-like structure of tests
    - TDL/TTCN-3 reflect this, existing transformations from UCM to e.g. MSC/UML do not
  - UCMs do not include much data information
    - appropriate stage to add data for executable test cases (UCM/TDL/TTCN-3/other)?
  - Peculiarities of the domain
    - 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

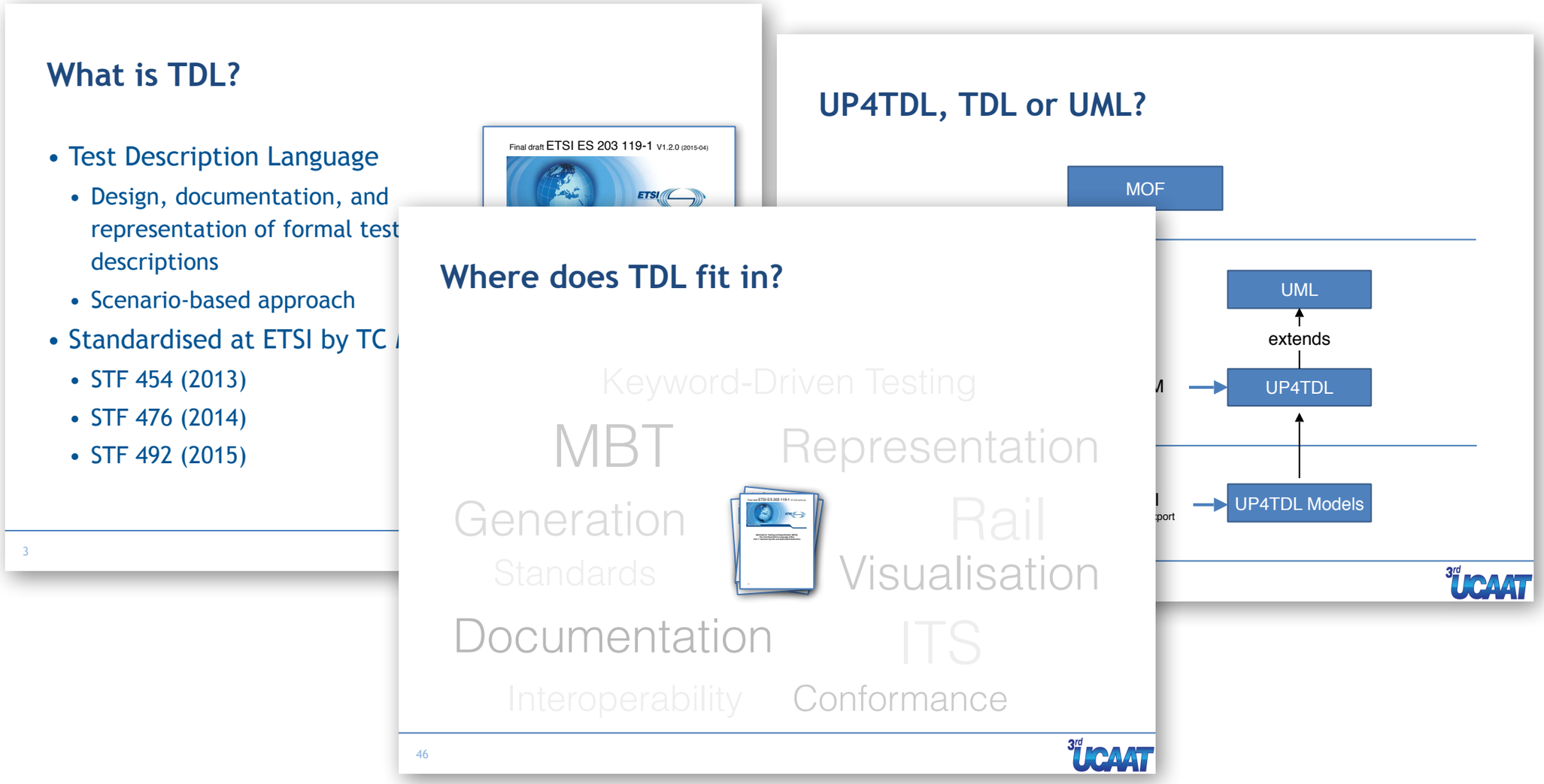


http://jucmnav.softwareengineering.ca/ucm/bin/view/ProjetSEG/ExportTdlUserGuide

Concluding remarks

- New technology, growing rapidly
- Open-source reference implementation under way
  - 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?

STF 492 - https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx
TDL - tdl.etsi.org

What would you want to see in TDL?

ETSI's Bug Tracker

Logged in as: *makedonski* (Philip Makedonski - manager) *13-09-2014 22:00 IST* Project: TDL TDL Switch

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

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

Viewing Issues (1 - 10 / 10) [Print Reports] [CSV Export] [Excel Export]

	P	ID	#	Project	Severity	Status	Updated	Summary
		0006768	1	TDL meta-model	**major**	assigned (Andreas Ulrich)	01-08-2014	New MM element as the starting point of the Behaviour Description of a Test Description
		0006773		TDL	feature	assigned (Andreas Ulrich)	31-07-2014	Accessing DataProxy arguments
		0006765		TDL meta-model	**major**	assigned (Andreas Ulrich)	31-07-2014	Time Observation
		0006764	1	TDL meta-model	minor	assigned (Andreas Ulrich)	31-07-2014	Description of VerdictType shall be modified
		0006763		TDL meta-model	minor	resolved (Andreas Ulrich)	31-07-2014	Blocks of ParallelBehaviour should be able to declare Guards
		0006767	1	TDL	minor	resolved (Andreas Ulrich)	11-07-2014	Allow to reference test descriptions that run on a different test (sub-) configuration
		0006772		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Variable assignment from Interaction and ActionReference
		0006771		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Component variables
		0006770		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Named parameters
		0006769		TDL	feature	assigned (Andreas Ulrich)	10-06-2014	Move parameters from DataInstance to DataSet

Select All Move OK

new feedback acknowledged confirmed assigned resolved closed

MantisBT 1.2.14 [^]
Copyright © 2000 - 2014 MantisBT Team
denis.filatov@etsi.org

STF 492 - https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx
TDL - tdl.etsi.org

Getting started with TDL?

- Cook up your own tooling?
- Demos
  - CEA - “Embedding TDL into the UML environment”
  - Elvior - “Visualising generated tests with TDL”
  - MetaCase - “Custom representations and editors for TDL”
  - UG - “TDL in education with custom tooling”
- Visit us at the TDL booth for further information!

Applying TDL in Practice: A Hands-on Tutorial

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

STF 492 - https://portal.etsi.org//STF/STFs/STFHomePages/STF492.aspx