



THE ETSI TEST DESCRIPTION LANGUAGE BRIEF INTRODUCTION

Philip Makedonski (University of Göttingen)

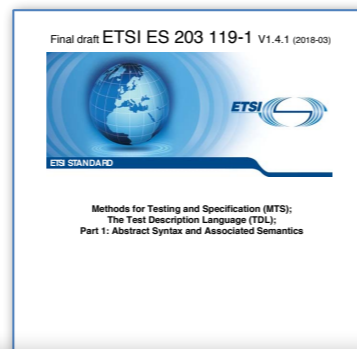
Martti Käärrik (Elviour OU)

Overview

What is TDL?

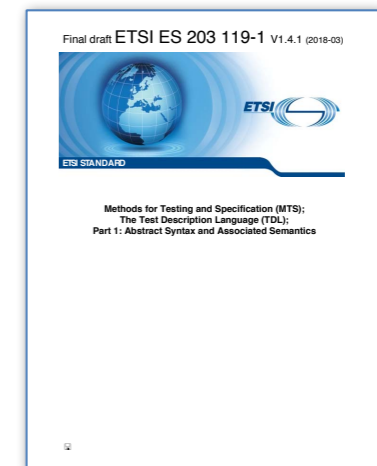
- Test Description Language
 - Design, documentation, and representation of formalised test descriptions
 - Scenario-based approach
- Standardised at ETSI by TC MTS
 - STF 454 (2013)
 - STF 476 (2014)
 - STF 492 (2015-2016)
 - STF 522 (2017)

X



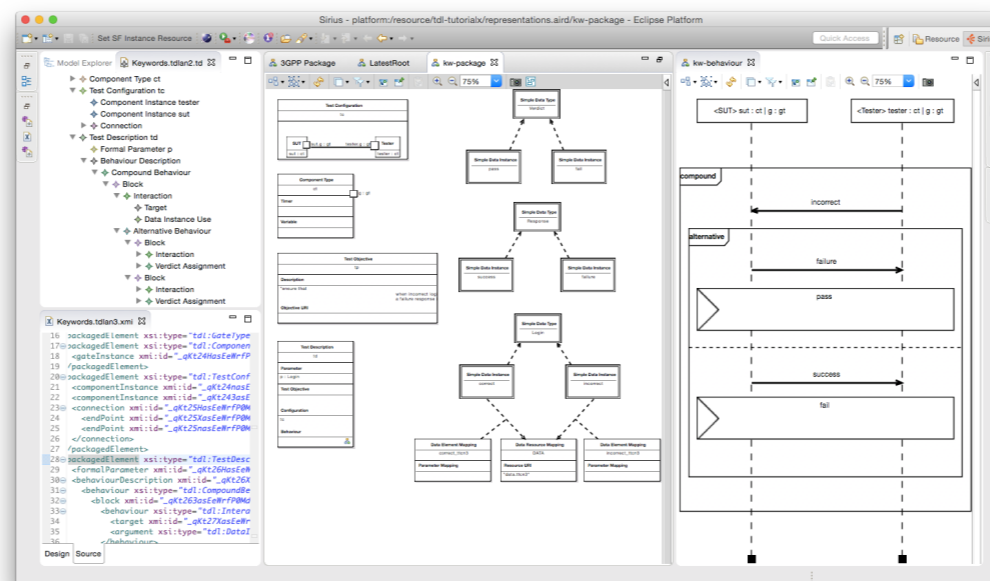
First steps

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



© ETSI 2018. All rights reserved

The TDL Open Source Project



X

© ETSI 2017. All rights reserved

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 (2015-2016)
 - STF 522 (2017)

Final draft ETSI ES 203 119-1 V1.4.1 (2018-03)



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



What is TDL?

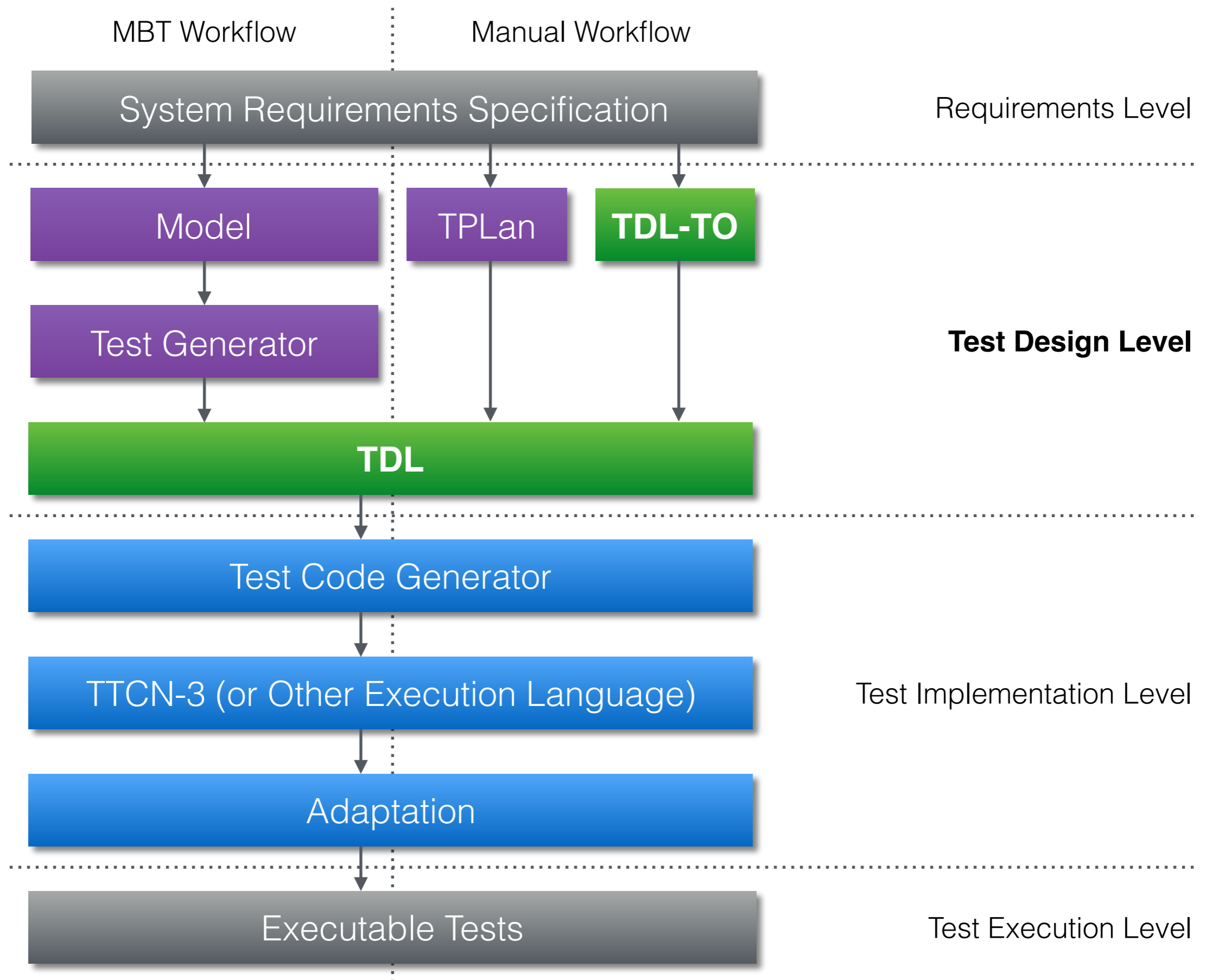
- Design, documentation, representation?
 - ease development and review
 - improve productivity and quality
 - both industry and standardisation
 - reduce implementation details

Final draft ETSI ES 203 119-1 V1.4.1 (2018-03)



**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 V1.4.1 (2018-03)



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



Final draft ETSI ES 203 119-1 V1.4.1 (2018-03)



**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 V1.4.1 (2018-03)




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 V1.4.1 (2018-03)




ETSI STANDARD

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

Part 1: MM Meta-Model and Semantics

Final draft ETSI ES 203 119-2 V1.3.1 (2018-03)




ETSI STANDARD

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

Part 2: GR Graphical Syntax

Final draft ETSI ES 203 119-3 V1.3.1 (2018-03)




ETSI STANDARD

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

Part 3: XF Exchange Format

Final draft ETSI ES 203 119-4 V1.3.1 (2018-03)




ETSI STANDARD

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

Part 4: TO Structured Test Objectives

Final draft ETSI ES 203 119-5 V1.1.1 (2018-03)




ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 5: UML profile for TDL

Part 5: UML Profile for TDL

Final draft ETSI ES 203 119-6 V1.1.1 (2018-04)




ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 6: Mapping to TTCN-3

Part 6: Mapping to TTCN-3

Final draft ETSI ES 203 119-7 V1.1.1 (2018-03)

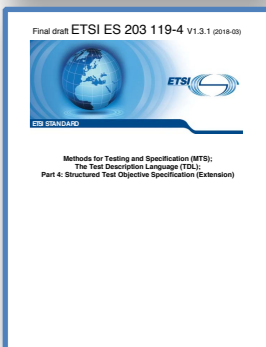
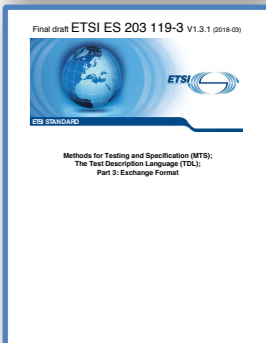
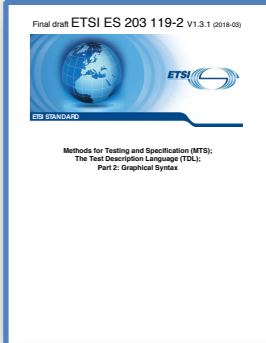
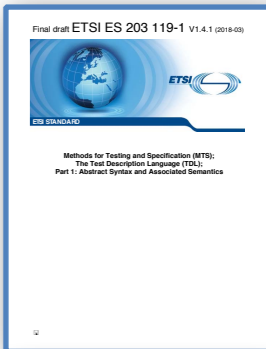
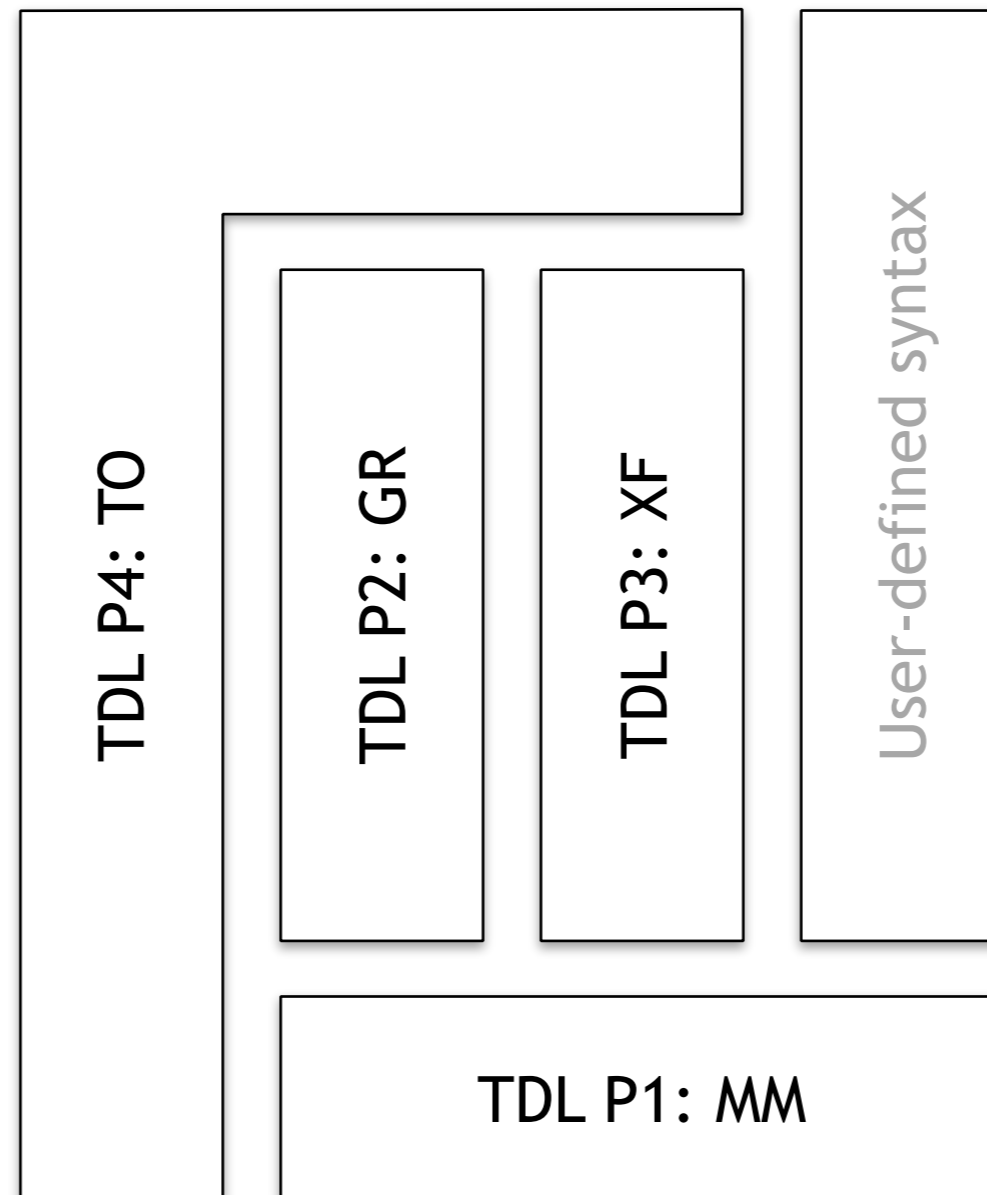


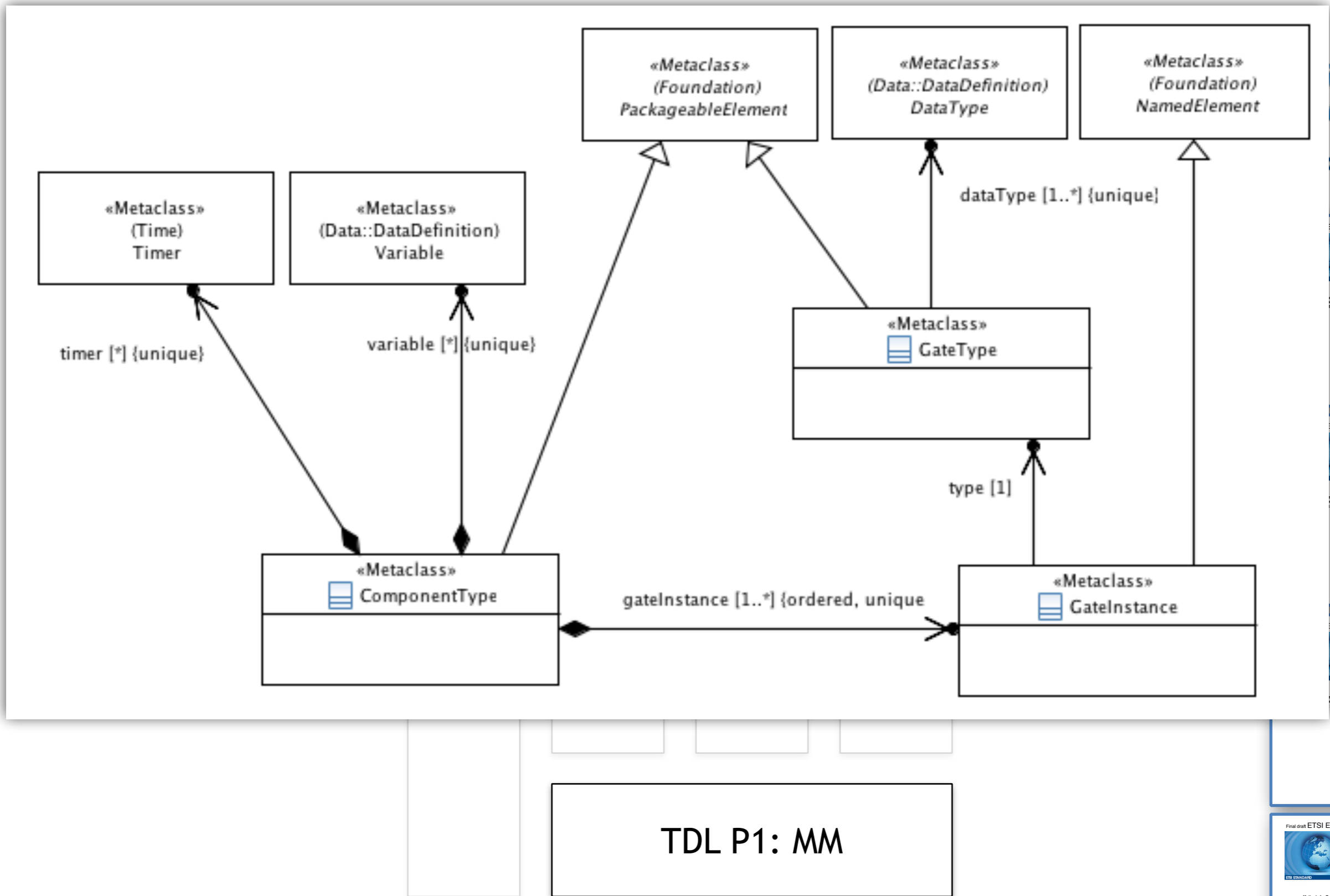
ETSI STANDARD

Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 7: Extended Test Configurations

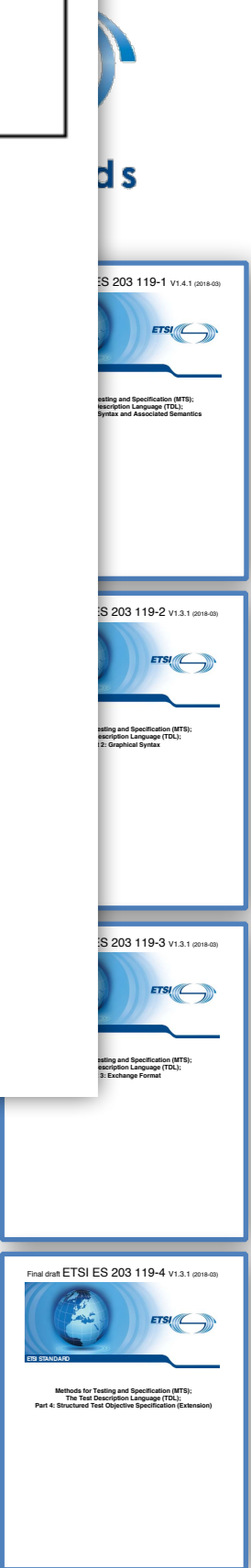
Part 7: Extended Test Configurations

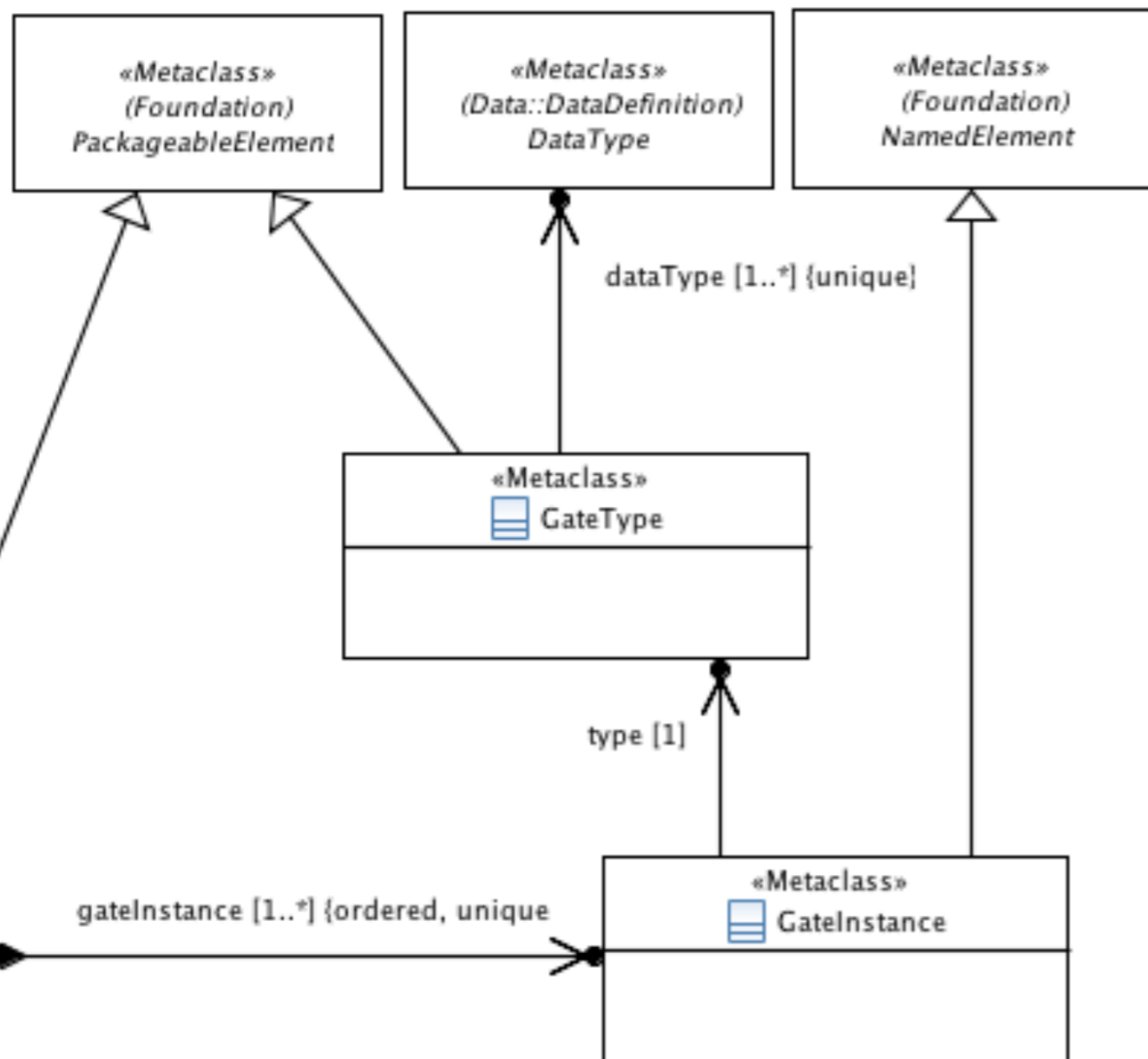
What is TDL?





TDL P1: MM





Semantics

A 'GateType' represents a type of communication points, called 'ComponentInstance's. A 'GateType' specifies the 'DataType's to be used in both directions.

Generalization

- PackageableElement

Properties

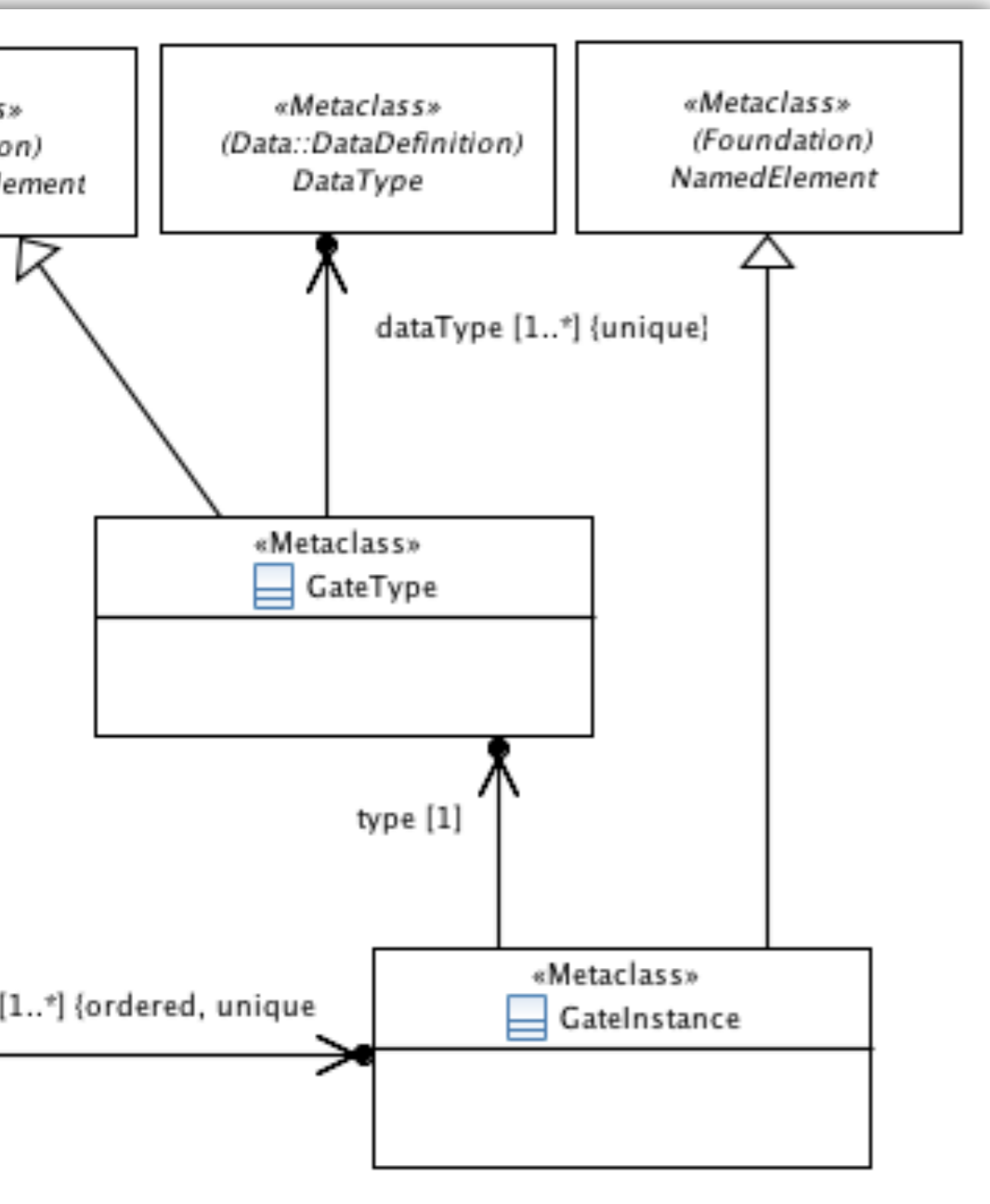
- dataType: DataType [1..*] {unique}
The 'DataType's that can be exchanged via 'GateInstance's shall adhere to the 'DataType's that are allowed to be exchanged via 'GateType's.

Constraints

There are no constraints specified.

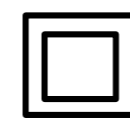
TDL P1: MM





6.4.2 GateType

Concrete Graphical Notation



GATYPENAMELABEL

Data Type: DATATYPELISTLABEL

Formal Description

context GateType

GATYPENAMELABEL ::= self.name

DATATYPELISTLABEL ::= self.dataType.name->separator(',')

Comments

No comments.



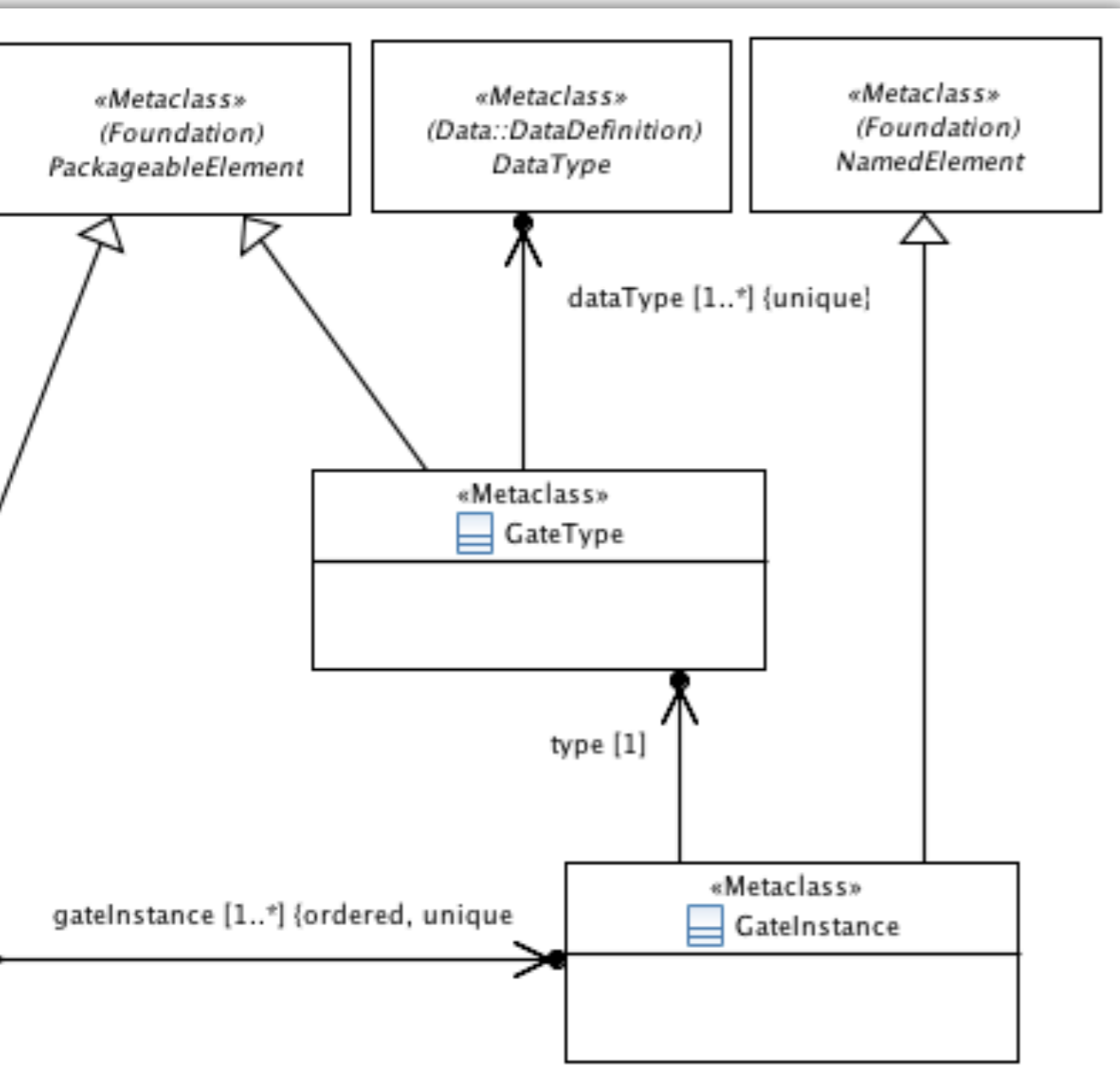
Radio

Data Type: Message, Signal

TDL P2: GR

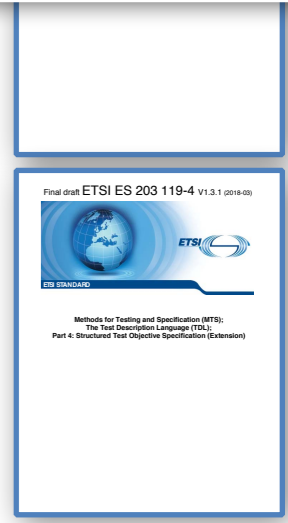
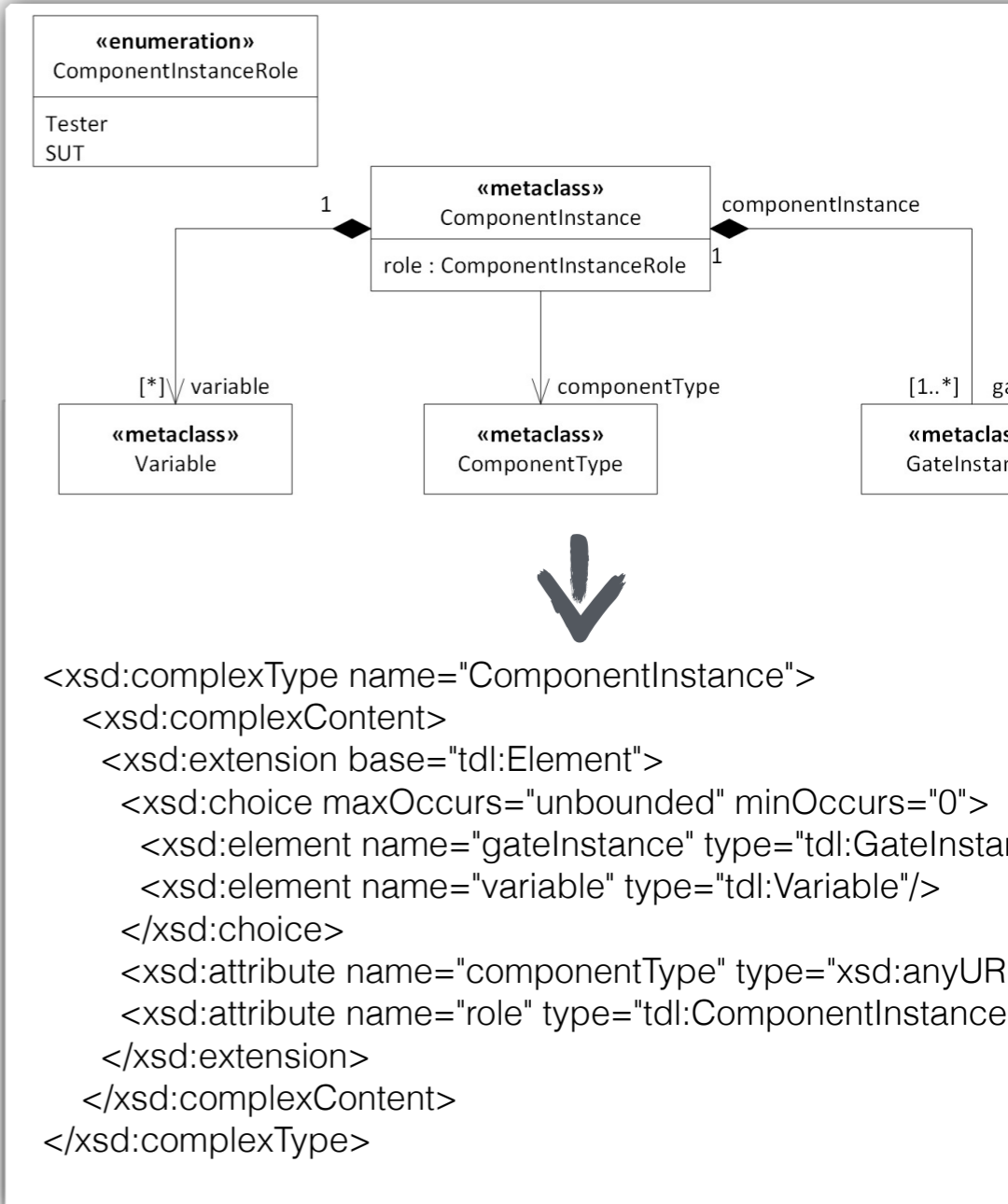
TDL P1: MM

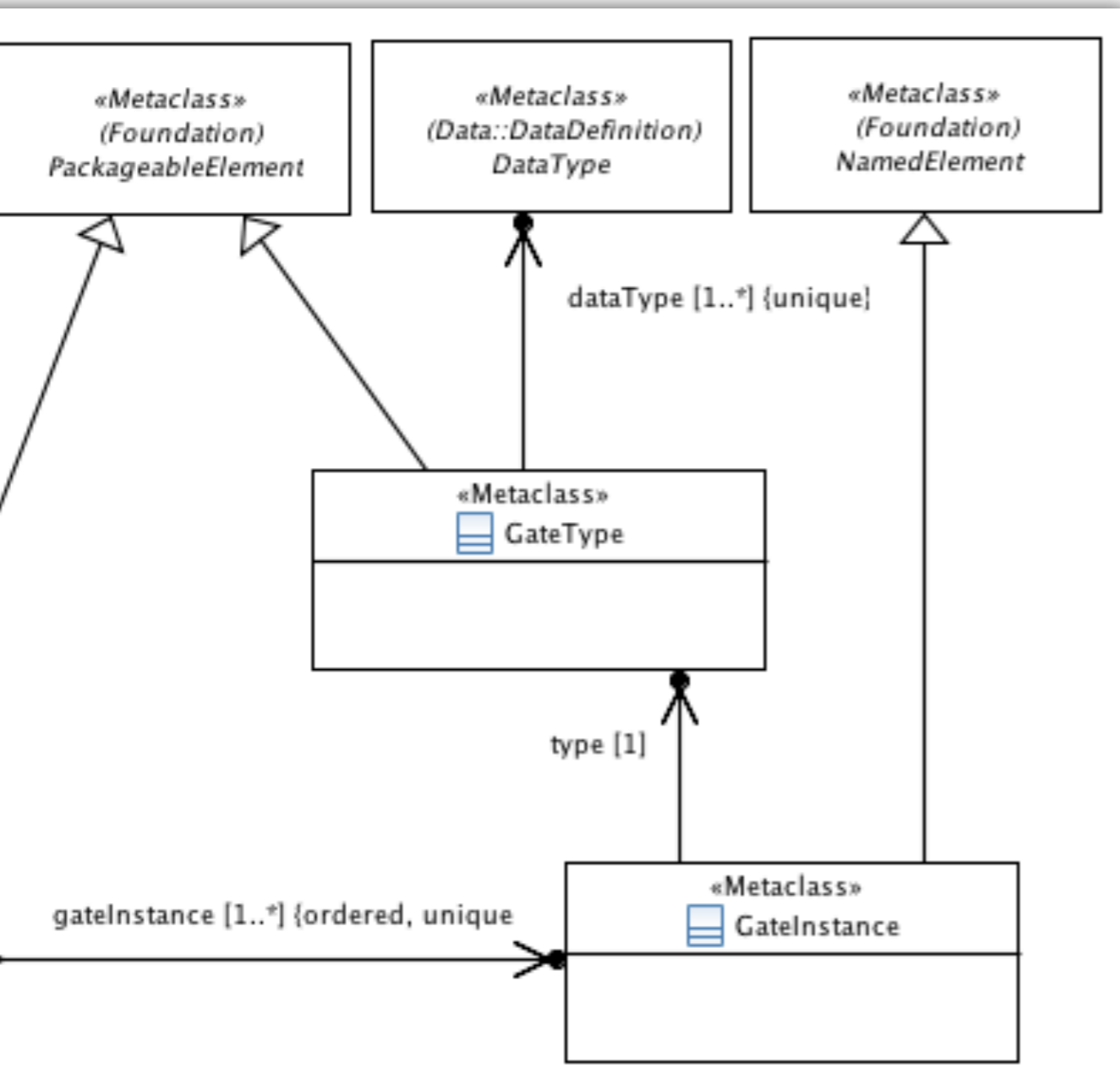




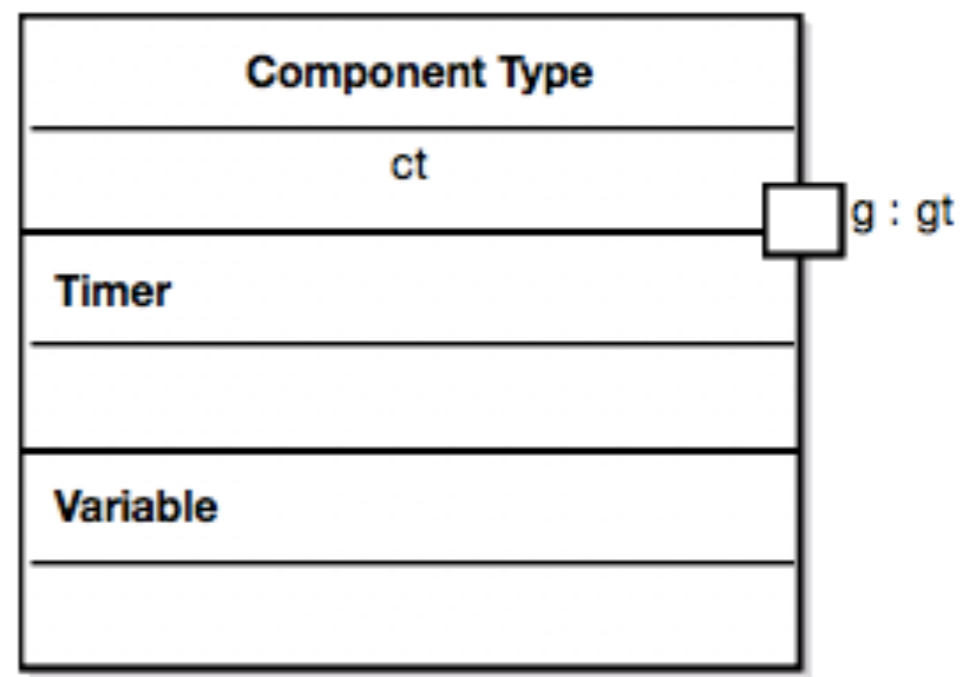
TDL P3: XF

TDL P1: MM





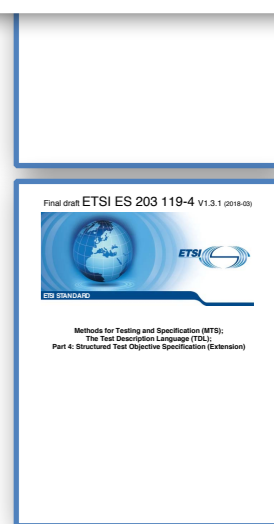
TDL P3: XF



```

<packagedElement xsi:type="tdl:ComponentType"
  <gateInstance xmi:id="_qKt24HasEeWrFP0MdfQNp
</packagedElement>
  
```

TDL P1: MM



What is TDL?

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

TDL P4: TO

TP Id	TESTOBJECTIVENAMELABEL
Test Objective	DESCRIPTIONLABEL
Reference	URIOFOBJECTIVELABEL
Config Id	<CONFIGLABEL>
PICS Selection	<PICSSELECTIONLABEL>

Initial Conditions

INITIALCONDITIONSLABEL

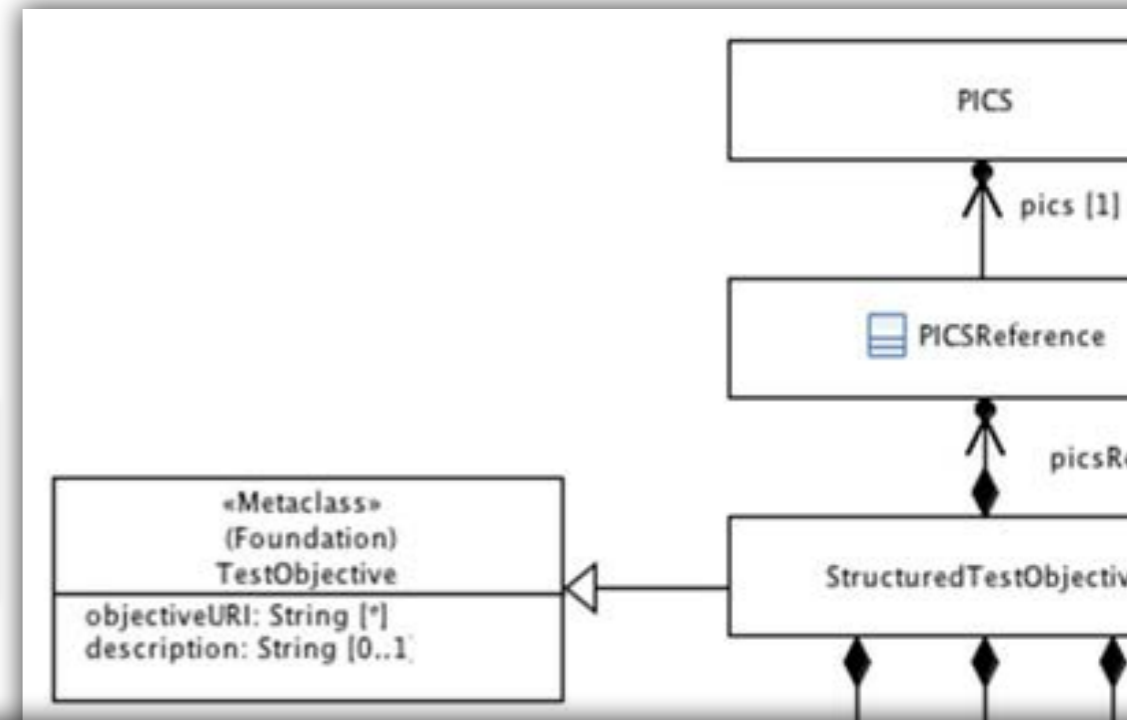
Expected Behaviour

EXPECTEDBEHAVIOURLABEL

Final Conditions

FINALCONDITIONSLABEL

+ repetitions: Value [0..1]
+ interval: Value [0..1]



First steps

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

Final draft ETSI ES 203 119-1 V1.4.1 (2018-03)



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



First steps

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



101 Final draft ETSI ES 203 119-1 V1.4.1 (2018-03)

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

Final draft ETSI ES 203 119-2 V1.3.1 (2018-03)



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

First steps

- 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

First steps: Data

```
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]";  
};
```

Test Design



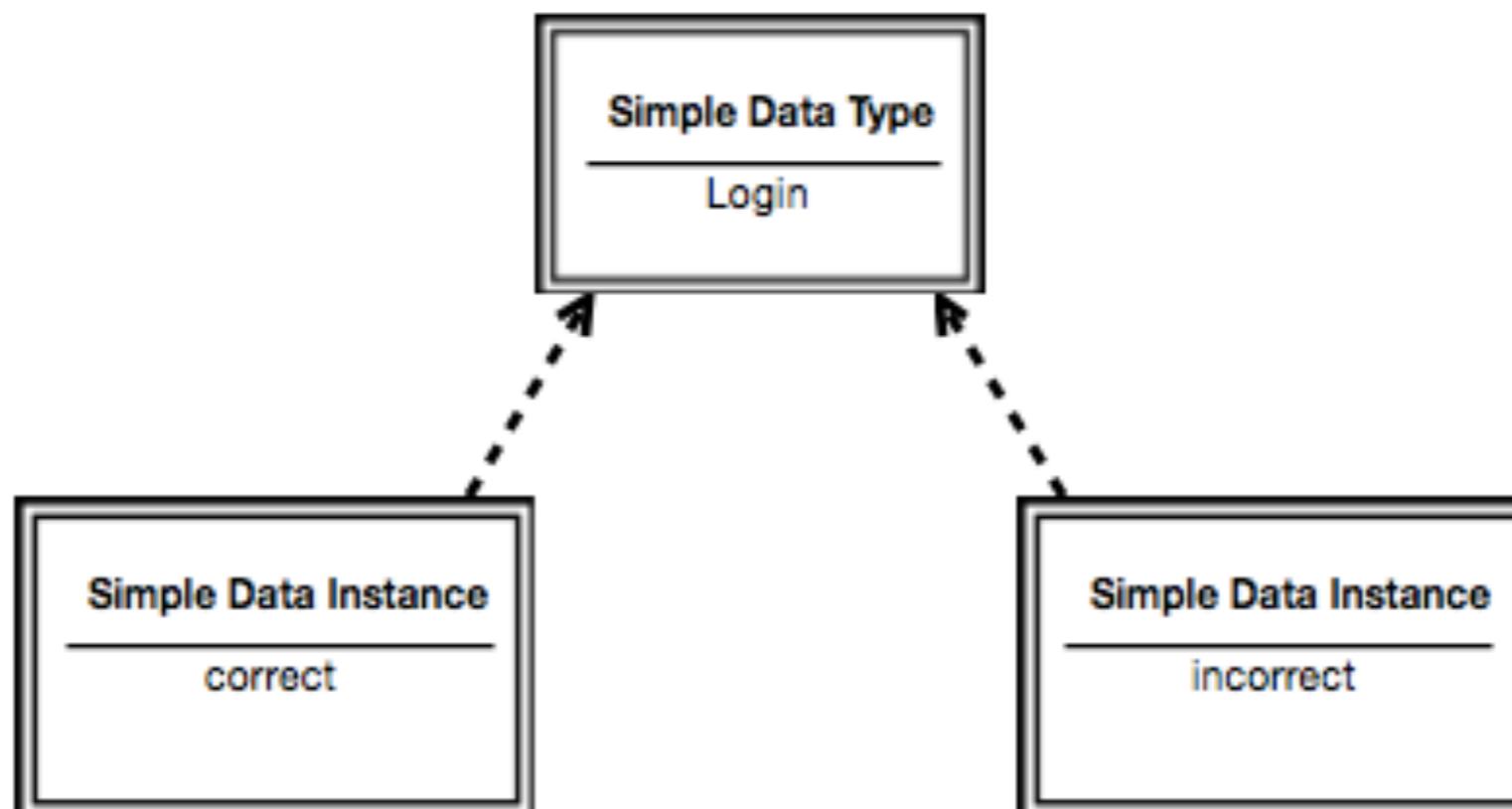
Test Implementation

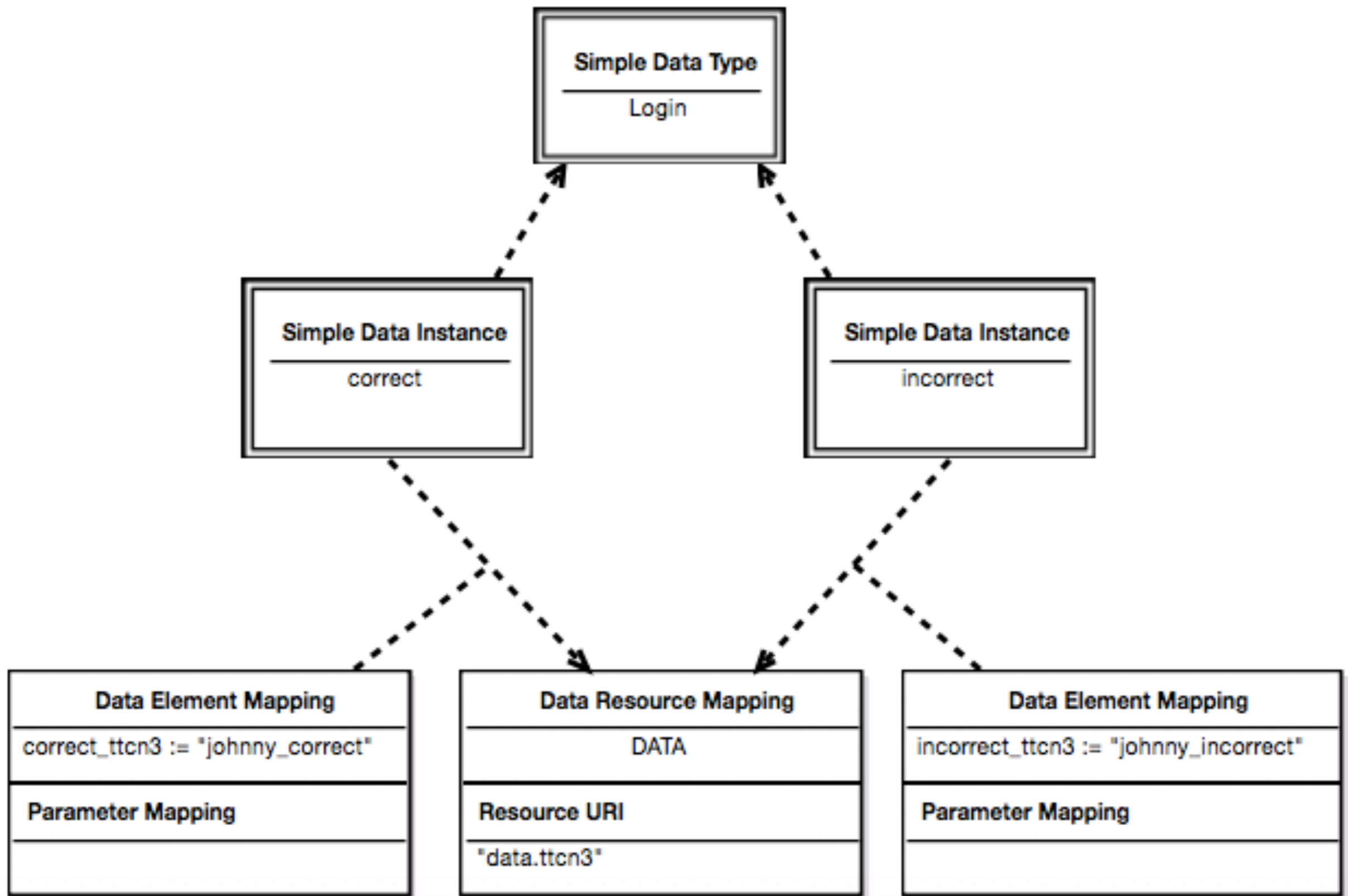


First steps: Data

```
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;
```





First steps



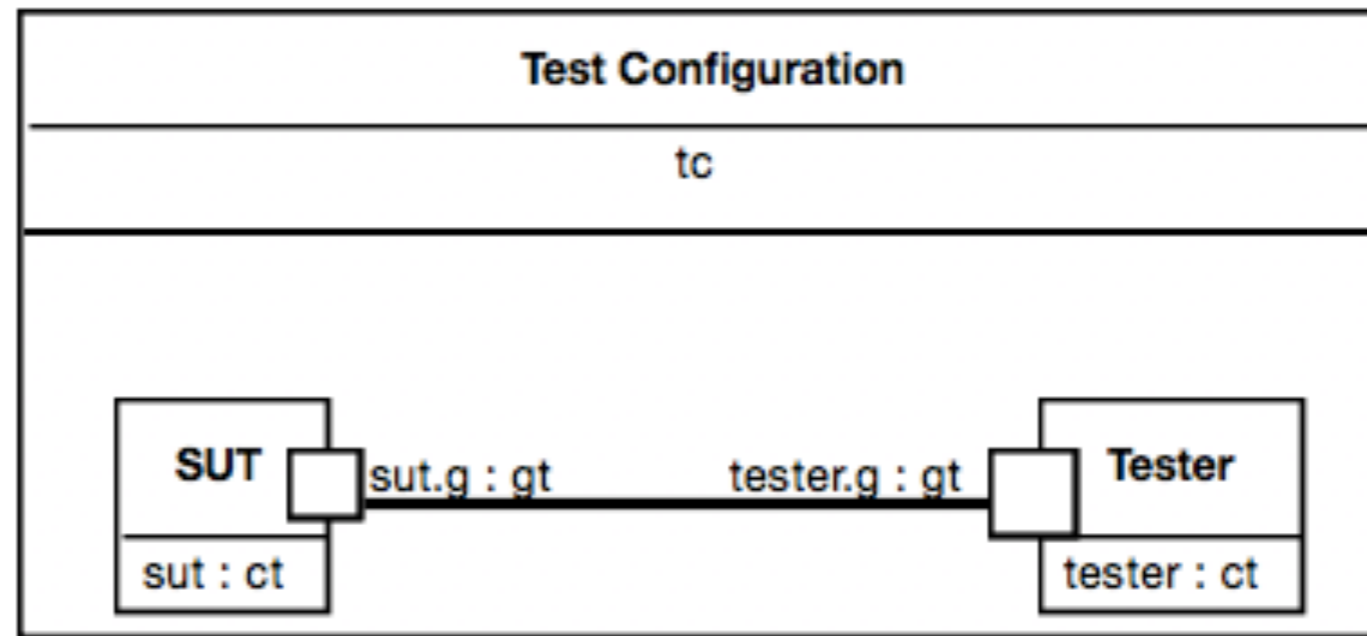
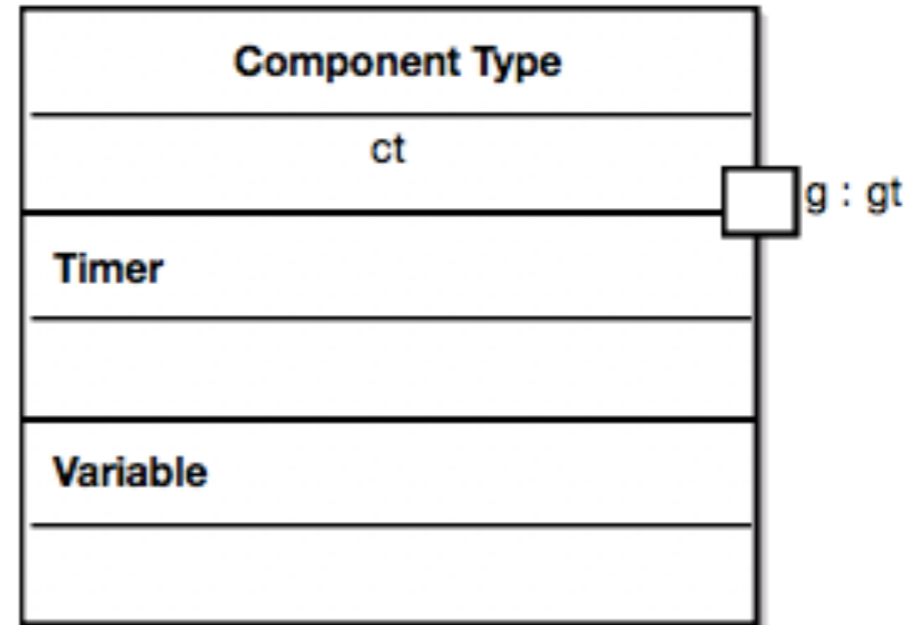
- Test configuration
 - typed components and gates
 - timers and variables
 - connections among gates
 - component roles

First steps: Configuration

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;  
}
```



First steps



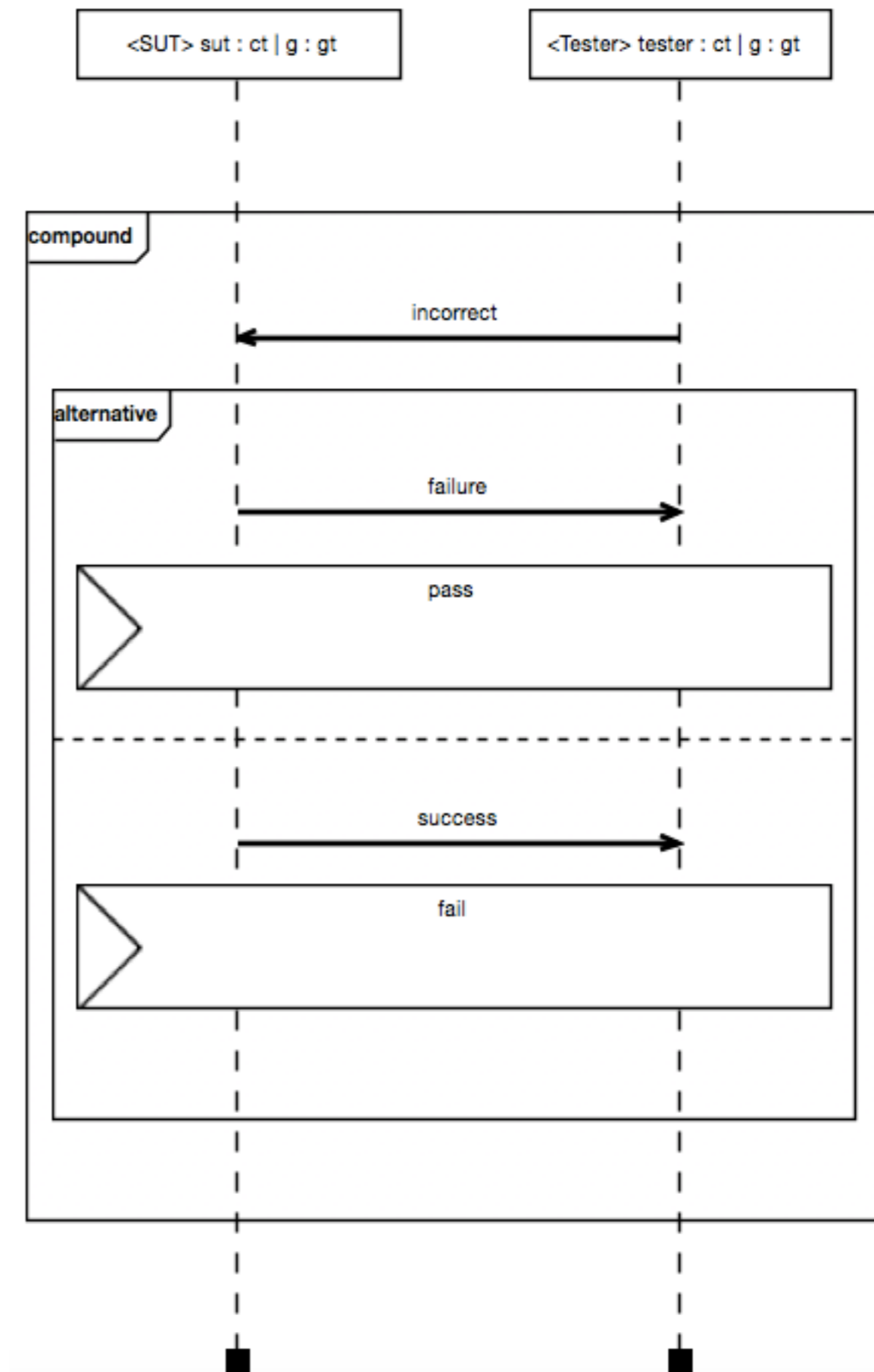
- Test behaviour
 - defines expected behaviour
 - failure upon deviations by default
 - actions and interactions
 - alternative, parallel, iterative, conditional
 - defaulting, interrupting, breaking

First steps: Behaviour

```
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;
    };
  }
```



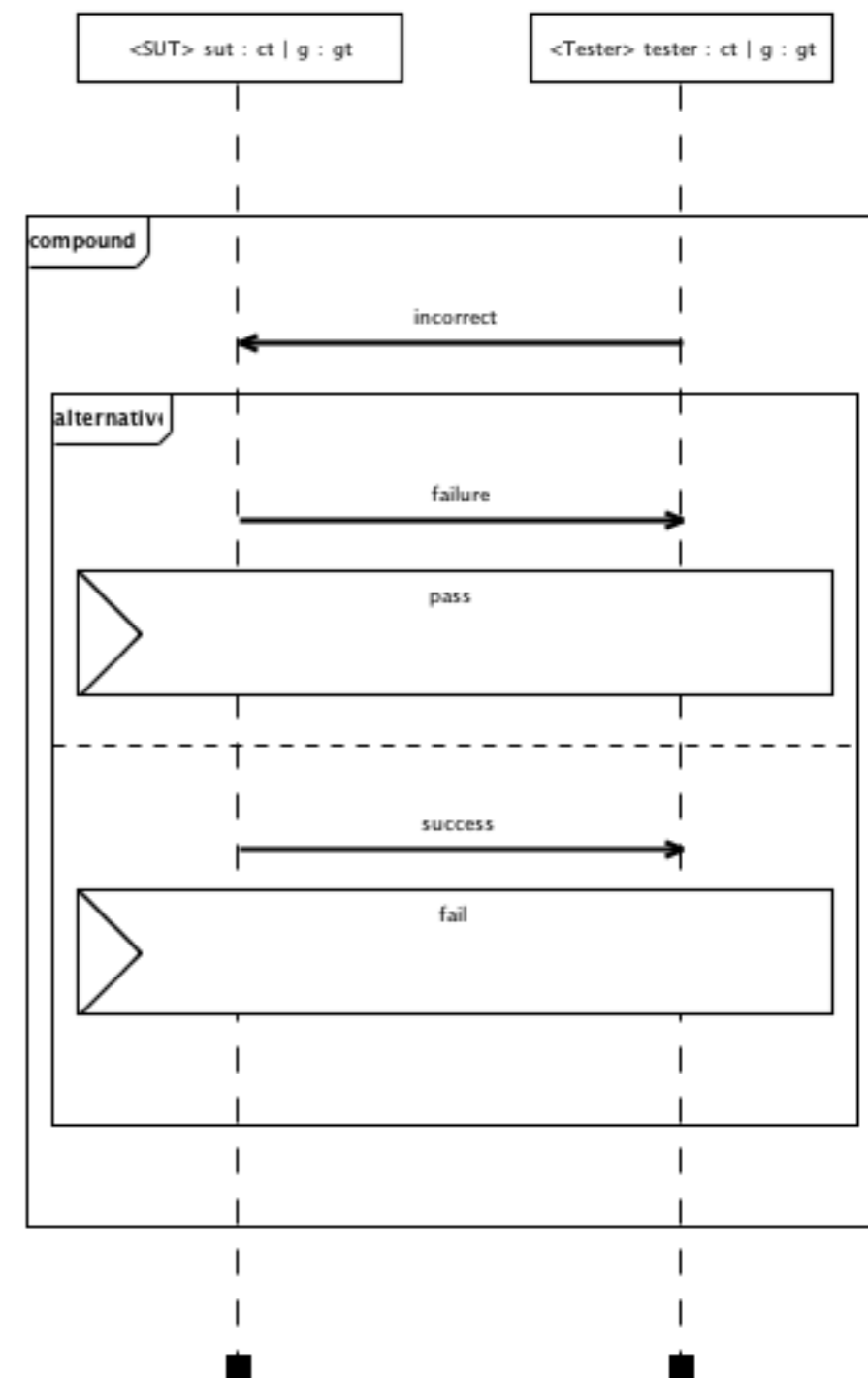
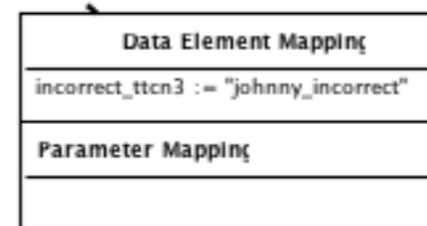
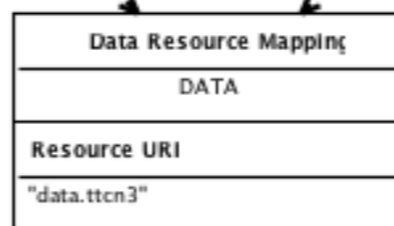
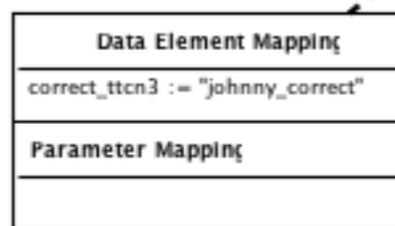
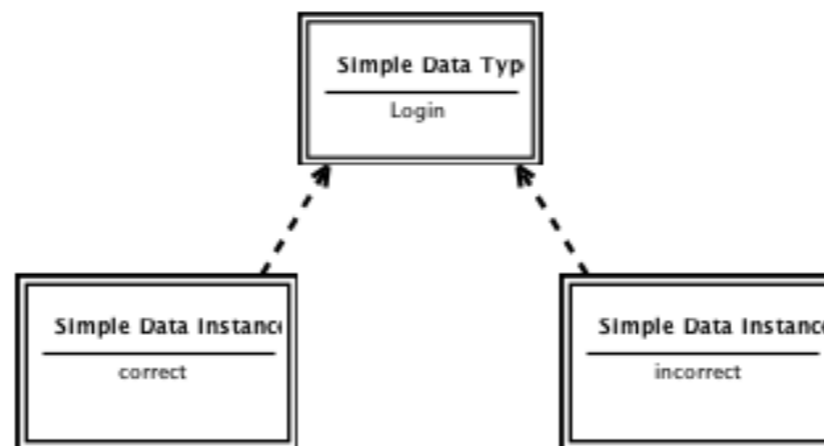
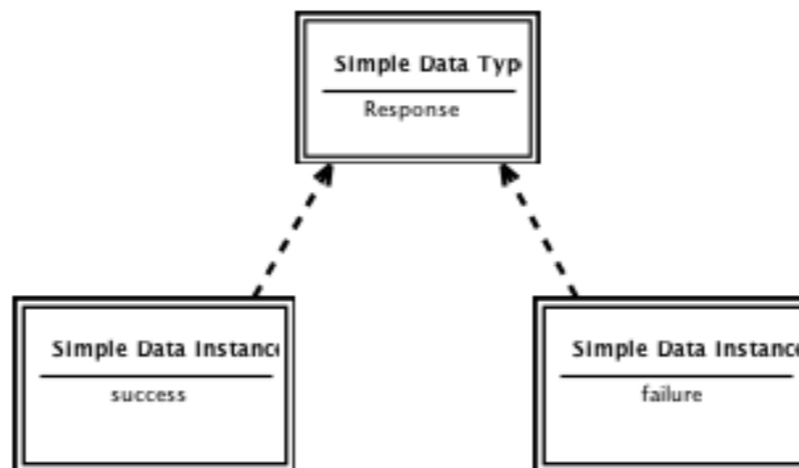
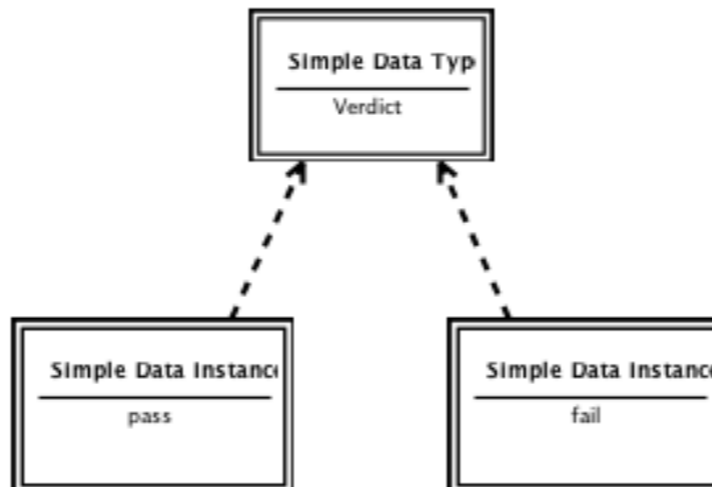
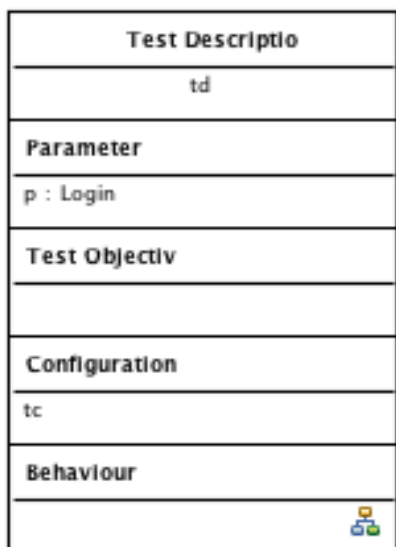
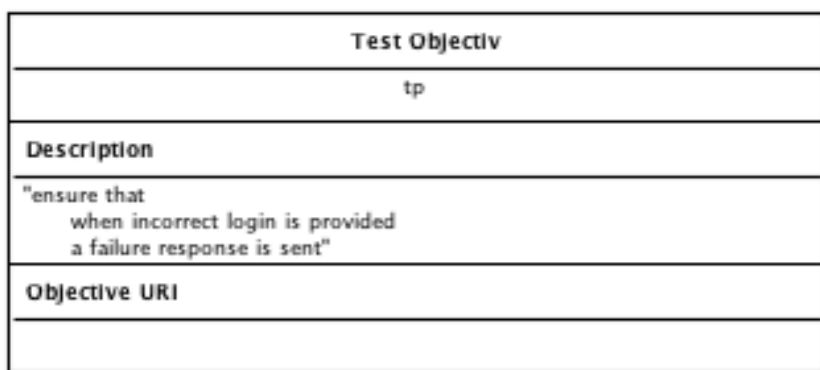
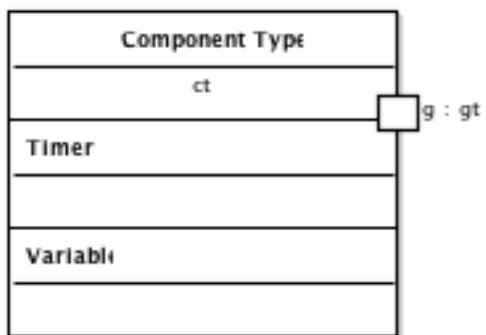
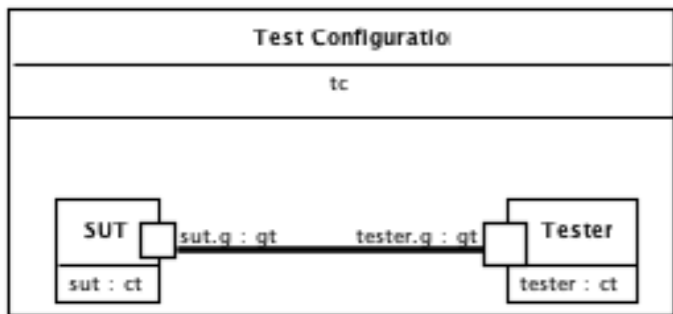
First steps



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

First steps: Objectives

```
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;
    }
  }
}
```



First steps

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

Final draft ETSI ES 203 119-4 V1.3.1 (2018-03)



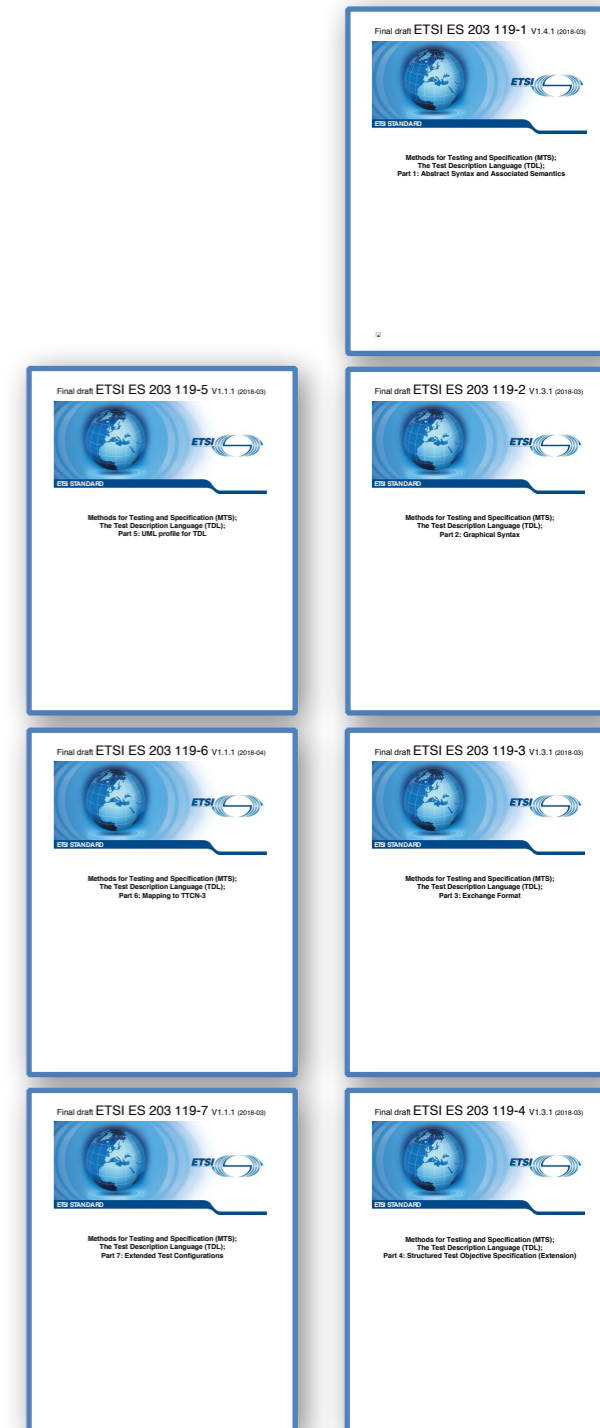
Methods for Testing and Specification (MTS);
The Test Description Language (TDL);
Part 4: Structured Test Objective Specification (Extension)

First steps: Structured Objectives

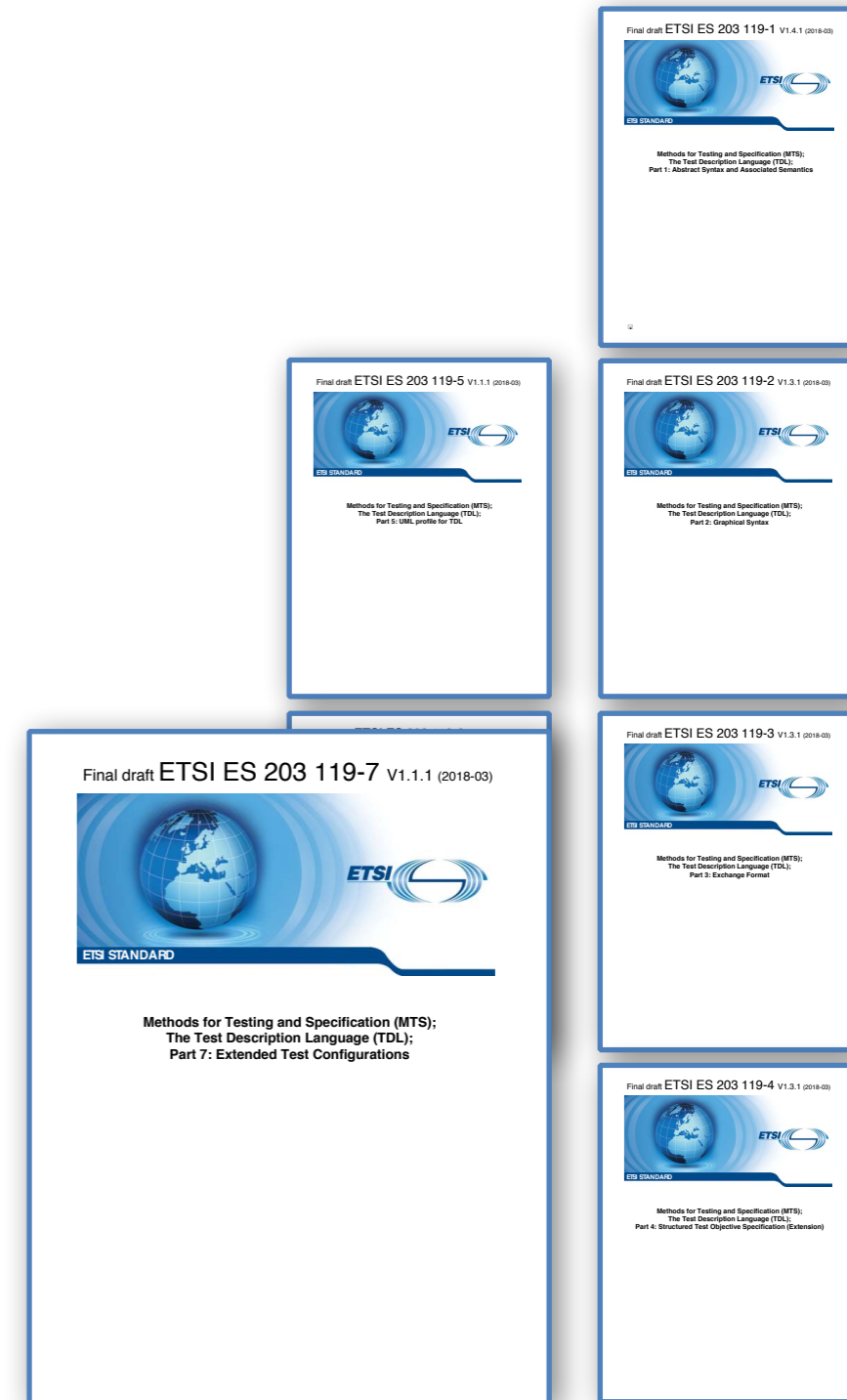
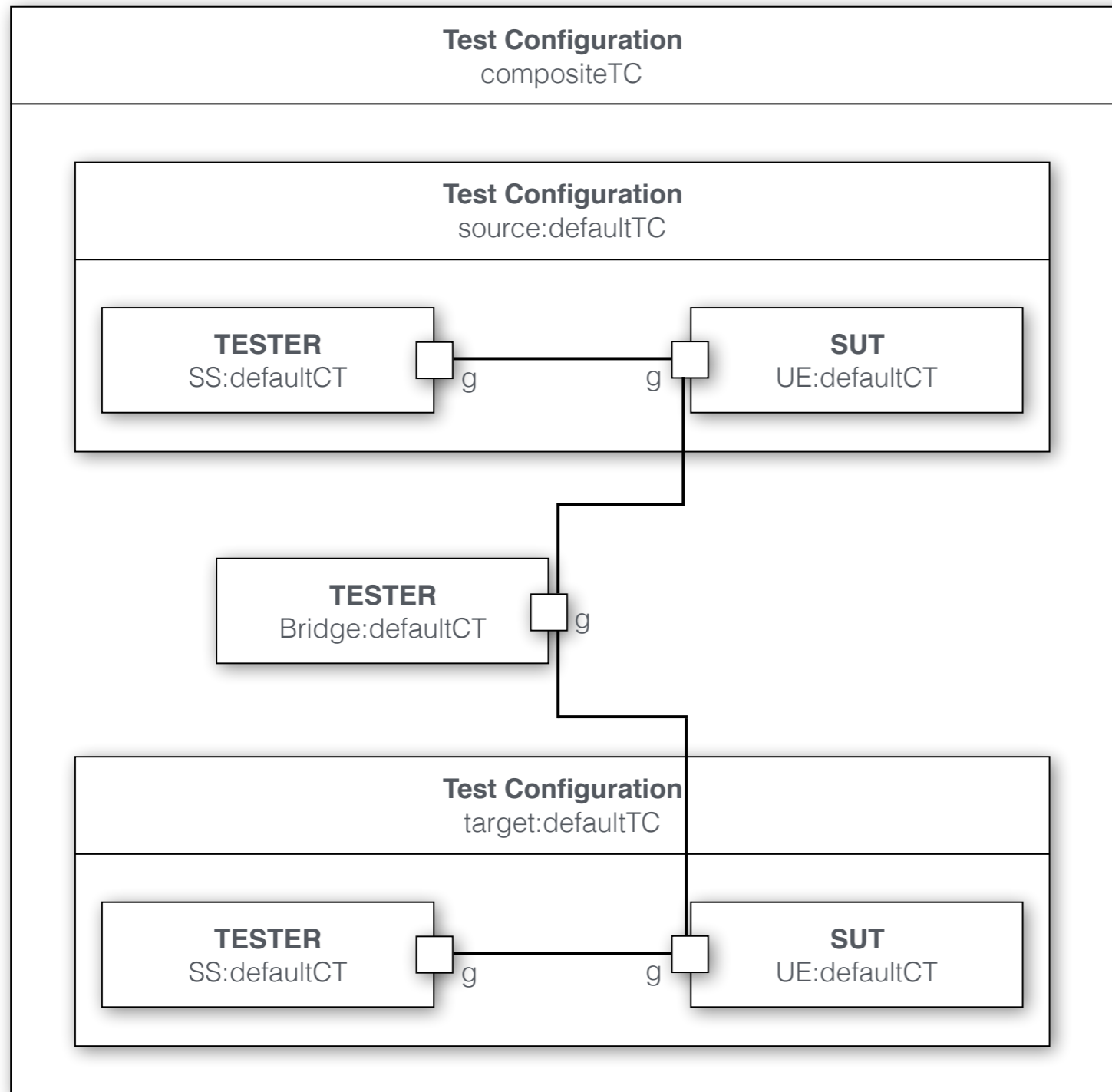
```
Test Purpose {
  TP Id "TP/CAM/INA/DOP/BV/02"
  Test objective "Checks that CAM message includes
                 DoorOpen information 30s after closed"
  Reference "TS 102 637-2 [1], clauses 7.1 and 7.2"
  PICS Selection PICS_PUBTRANSVEH
  Initial conditions
  with {
    the IUT entity having reached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

- Part 1: New features
 - collections, procedures
 - local ordering option
 - UML Profile moved to Part 5
- Part 6: Mapping to TTCN-3
 - compatibility and consistency
 - reuse tools and assets
- Part 7: Extended Configurations
 - instantiate existing configurations
 - reuse and extend

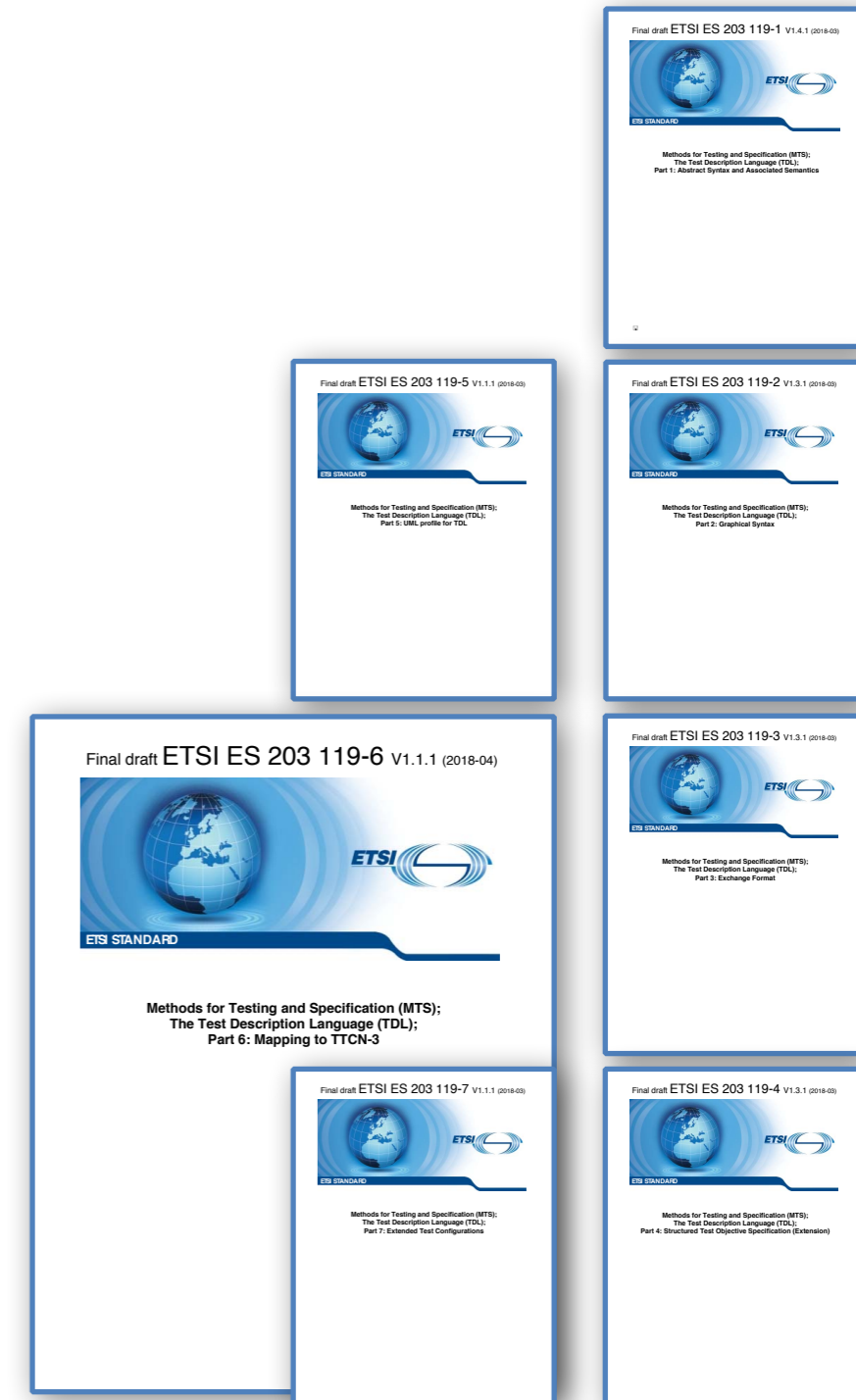


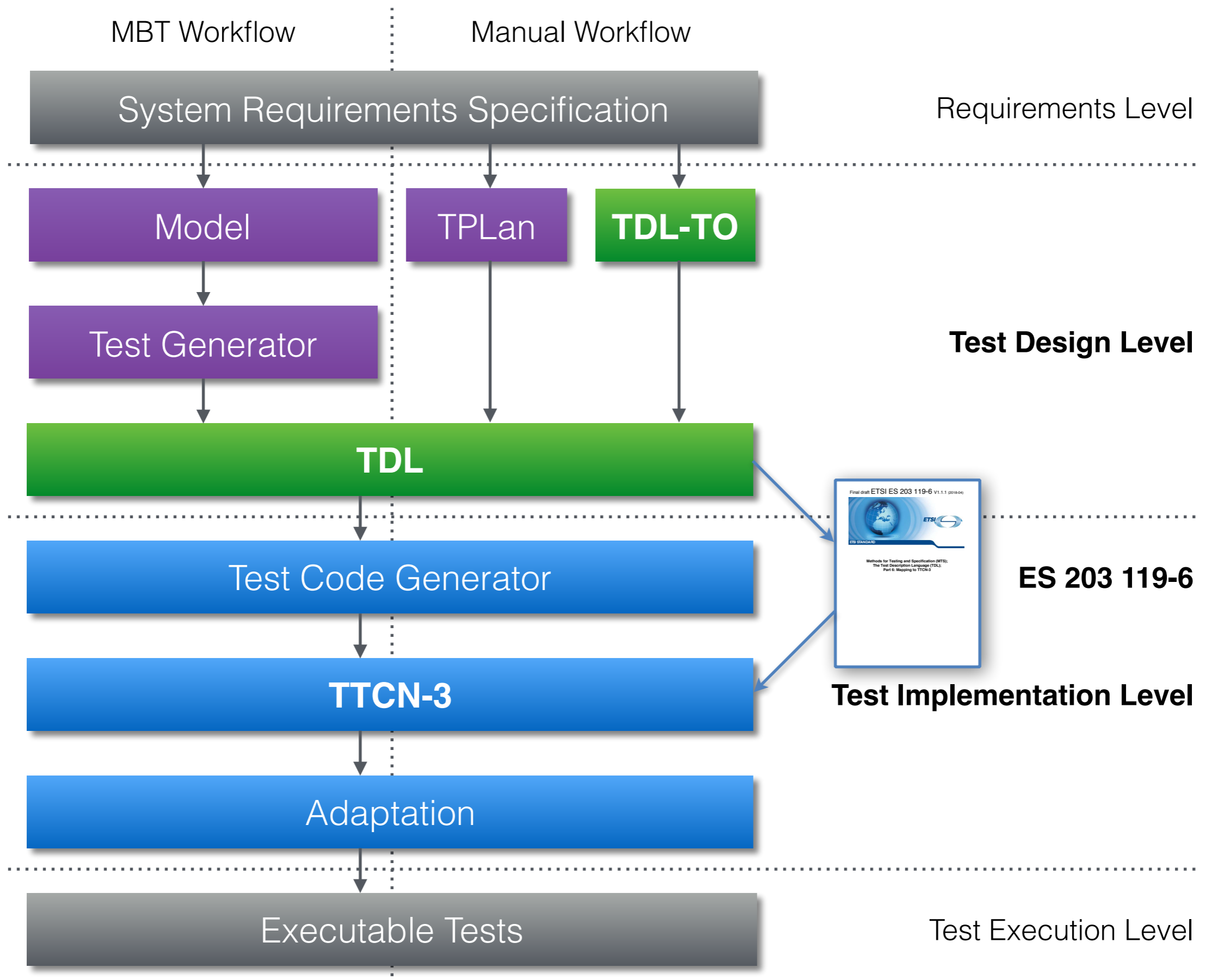
What is new in TDL?



Mapping TDL to TTCN-3

- Establish a connection between TDL and TTCN-3
 - generation of executable tests from test descriptions
 - standardised, ensuring compatibility and consistency
 - re-use existing tools and frameworks for test execution
 - re-use existing TTCN-3 assets (data, behaviour)





The TDL Open Source Project



- New technology, growing rapidly
- TDL open source project for essential tool support
 - 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

The TDL Open Source Project



- Meta-model implementation and validation (Part 1)
- Textual editors (Annex B of Parts 1 and 4)
- Graphical editor (Part 2)
- Translation between representations (Part 3)
- TTCN-3 generation (Part 6, currently ongoing)

The TDL Open Source Project

Sirius - platform:/resource/tcl-tutorialx/representations.aird/kw-package - Eclipse Platform

Model Explorer: Keywords.tdlan2.td

- Component Type ct
 - Test Configuration tc
 - Component Instance tester
 - Component Instance sut
 - Connection
 - Test Description td
 - Formal Parameter p
 - Behaviour Description
 - Compound Behaviour
 - Block
 - Interaction
 - Target
 - Data Instance Use
 - Alternative Behaviour
 - Block
 - Interaction
 - Verdict Assignment
 - Block
 - Interaction
 - Verdict Assignment

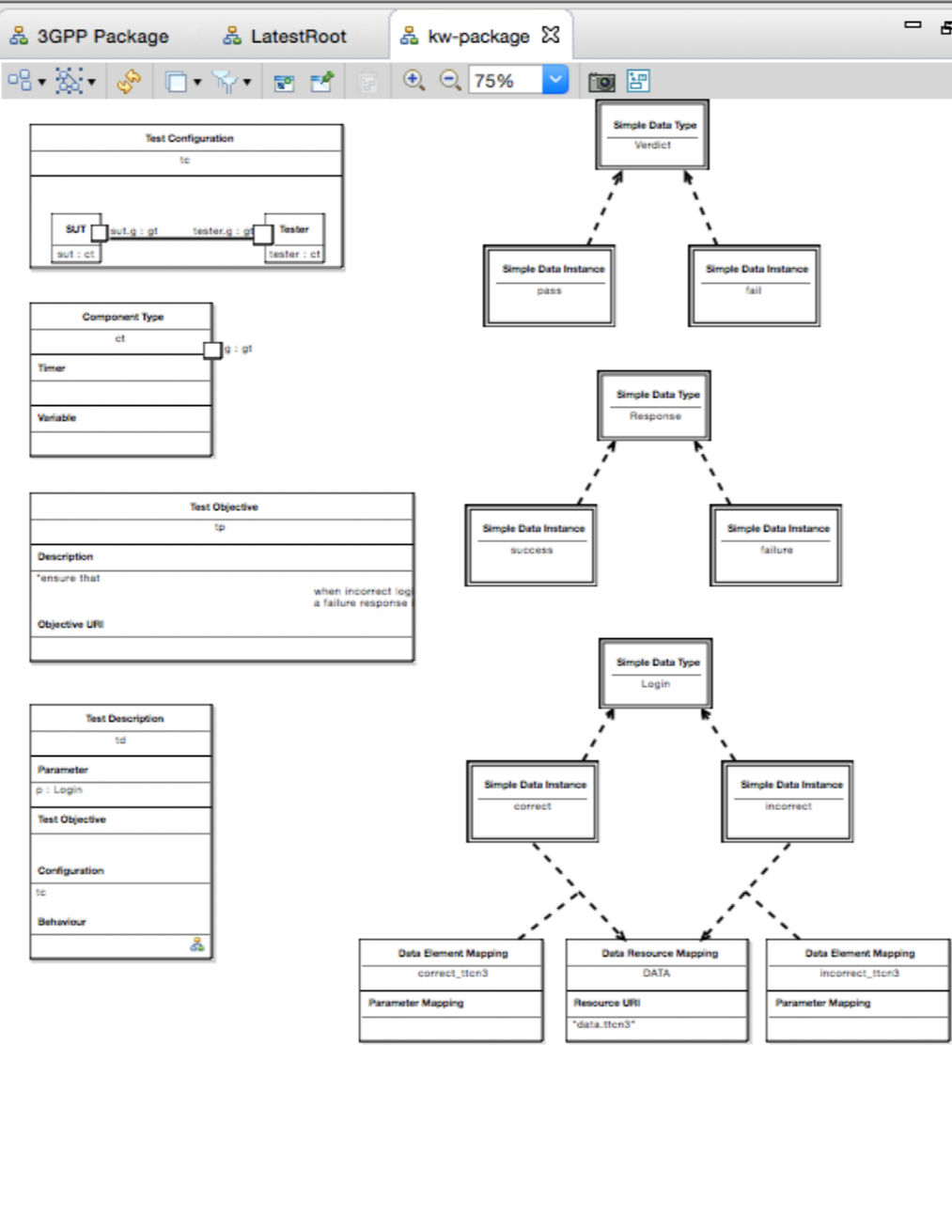
Keywords.tdlan3.xmi

```

16 packagedElement xsi:type="tdl:GateType
17 packagedElement xsi:type="tdl:Componen
18 <gateInstance xmi:id="_qKt24HasEeWrfP
19 /packagedElement>
20 packagedElement xsi:type="tdl:TestConf
21 <componentInstance xmi:id="_qKt24nasE
22 <componentInstance xmi:id="_qKt243asE
23 <connection xmi:id="_qKt25HasEeWrfP0M
24 <endPoint xmi:id="_qKt25XasEeWrfP0M
25 <endPoint xmi:id="_qKt25nasEeWrfP0M
26 </connection>
27 /packagedElement>
28 packagedElement xsi:type="tdl:TestDesc
29 <formalParameter xmi:id="_qKt26HasEeW
30 <behaviourDescription xmi:id="_qKt26X
31 <behaviour xsi:type="tdl:CompoundBe
32 <block xmi:id="_qKt263asEeWrfP0Md
33 <behaviour xsi:type="tdl:Intera
34 <target xmi:id="_qKt27XasEeWr
35 <argument xsi:type="tdl:DataI
36 </behaviour>
    
```

3GPP Package LatestRoot kw-package

kw-behaviour



UML Diagrams:

- Test Configuration (tc)**: Includes SUT (sut: g: gt) and Tester (tester: g: gt).
- Component Type (ct)**: Includes Timer and Variable.
- Test Objective (tp)**: Description: "ensure that when incorrect log a failure response".
- Test Description (td)**: Includes Parameter (p: Login), Test Objective, Configuration (tc), and Behaviour.
- Simple Data Type Hierarchy**:
 - Verdict**: Simple Data Instance (pass, fail)
 - Response**: Simple Data Instance (success, failure)
 - Login**: Simple Data Instance (correct, incorrect)
- Data Element Mapping**:
 - correct_ttcn3 (Parameter Mapping)
 - DATA (Resource URI: "data.ttcn3")
 - incorrect_ttcn3 (Parameter Mapping)

Sequence Diagram (kw-behaviour):

- Participants: <SUT> sut : ct | g : gt and <Tester> tester : ct | g : gt
- Compound:
 - incorrect (from Tester to SUT)
 - alternative:
 - failure (from Tester to SUT)
 - pass (from SUT to Tester)
 - success (from Tester to SUT)
 - fail (from SUT to Tester)

The TDL Open Source Project

Sirius - platform:/resource/tdl-tutorialx/representations.aird/kw-package - Eclipse Platform

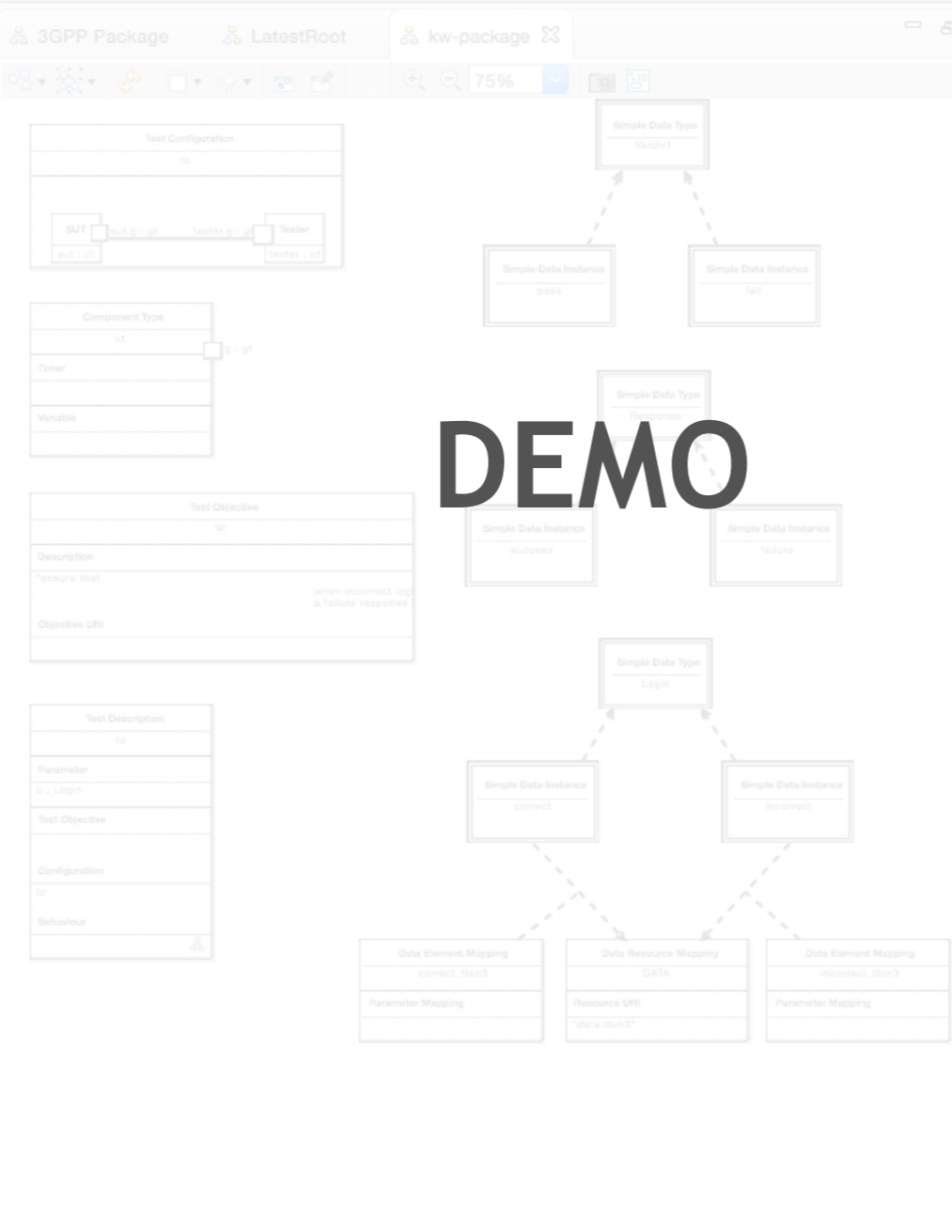
Model Explorer: Keywords.tdlan2.td

- Component Type ct
- Test Configuration tc
 - Component Instance tester
 - Component Instance sut
 - Connection
- Test Description td
 - Formal Parameter p
 - Behaviour Description
 - Compound Behaviour
 - Block
 - Interaction
 - Target
 - Data Instance Use
 - Alternative Behaviour
 - Block
 - Interaction
 - Verdict Assignment
 - Block
 - Interaction
 - Verdict Assignment

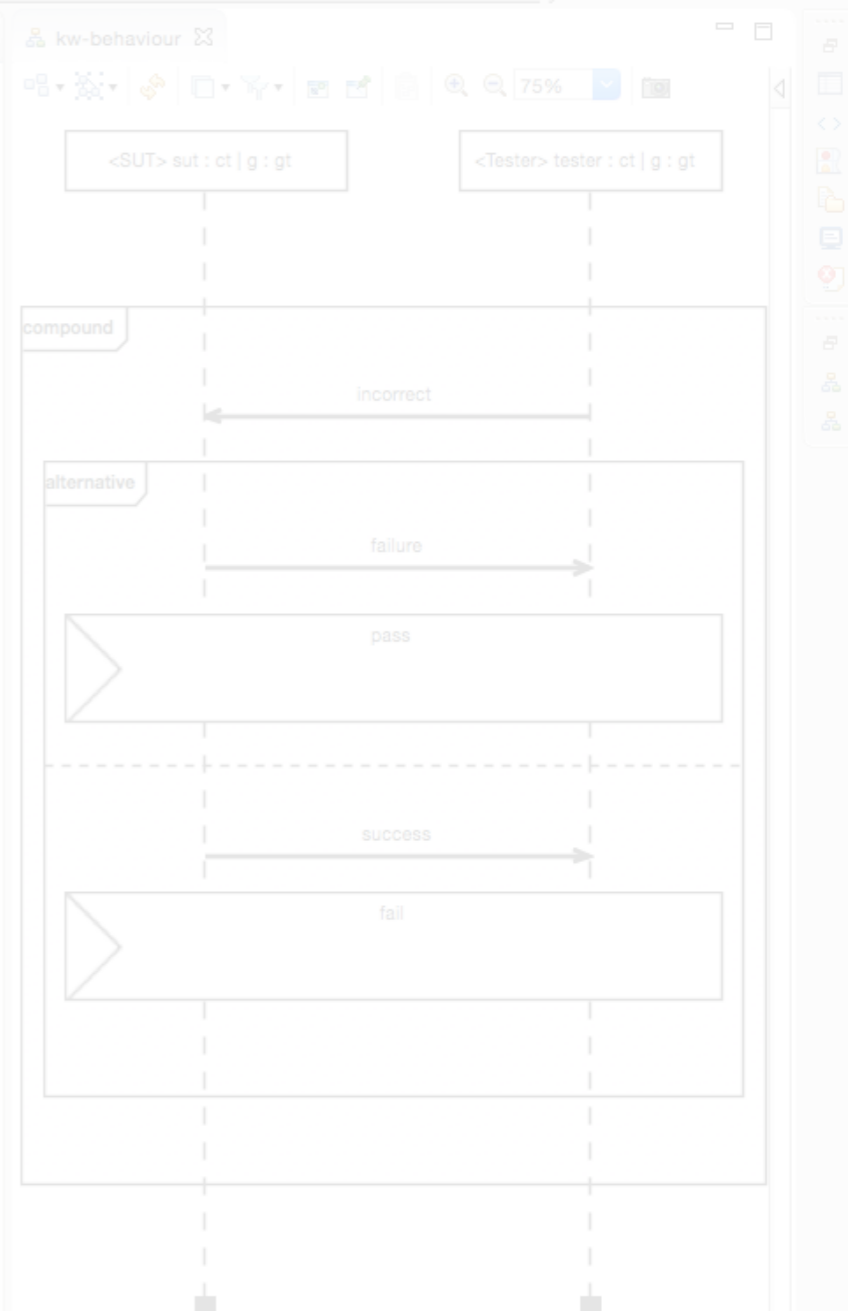
Keywords.tdlan3.xml

```
16 <packagedElement xsi:type="tdl:GateType"
17 >packagedElement xsi:type="tdl:Componen
18 <gateInstance xmi:id="_qKt24HasEeWrfP
19 /packagedElement>
20 <packagedElement xsi:type="tdl:TestConf
21 <componentInstance xmi:id="_qKt24nasE
22 <componentInstance xmi:id="_qKt243asE
23 <connection xmi:id="_qKt25HasEeWrfP0M
24 <endPoint xmi:id="_qKt25XasEeWrfP0M
25 <endPoint xmi:id="_qKt25nasEeWrfP0M
26 </connection>
27 /packagedElement>
28 <packagedElement xsi:type="tdl:TestDesc
29 <formalParameter xmi:id="_qKt26HasEeW
30 <behaviourDescription xmi:id="_qKt26X
31 <behaviour xsi:type="tdl:CompoundBe
32 <block xmi:id="_qKt263asEeWrfP0Md
33 <behaviour xsi:type="tdl:Intera
34 <target xmi:id="_qKt27XasEeWrf
35 <argument xsi:type="tdl:DataI
36 </behaviours>
```

3GPP Package LatestRoot kw-package



kw-behaviour



DEMO

Summary

What is TDL?

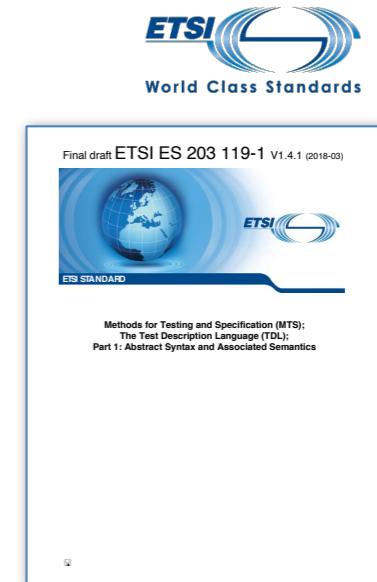
- Test Description Language
 - Design, documentation, and representation of formalised test descriptions
 - Scenario-based approach
- Standardised at ETSI by TC MTS
 - STF 454 (2013)
 - STF 476 (2014)
 - STF 492 (2015-2016)
 - STF 522 (2017)

X



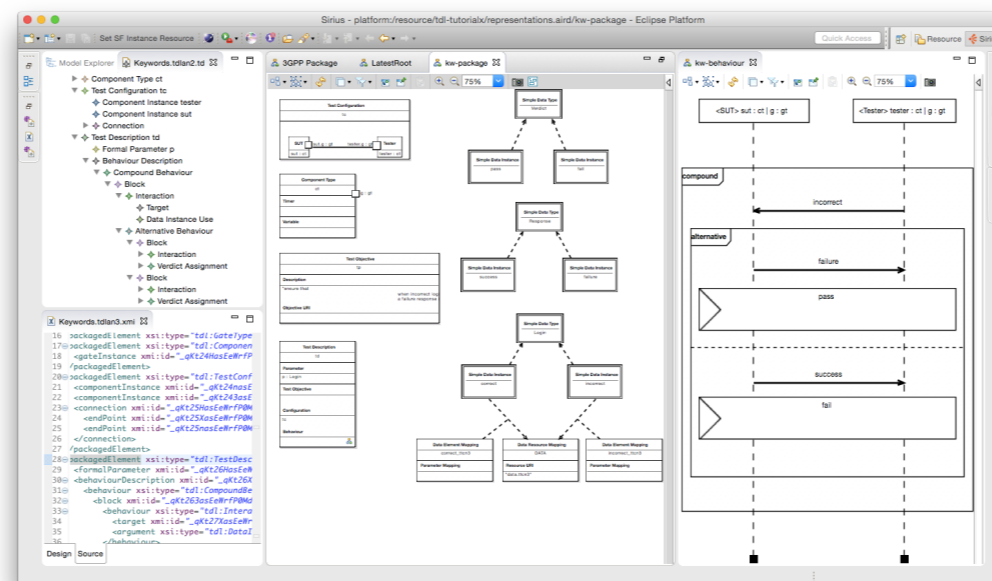
First steps

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



© ETSI 2018. All rights reserved

The TDL Open Source Project



X

© ETSI 2017. All rights reserved