Welcome to the World of Standards

ETSI

World Class Standards

THE ETSI TEST DESCRIPTION LANGUAGE (TDL)

Results from the ETSI project STF 454

A. Ulrich, G. Adamis, F. Kristoffersen, Ph. Makedonski, M.-F. Wendland, A. Wiles

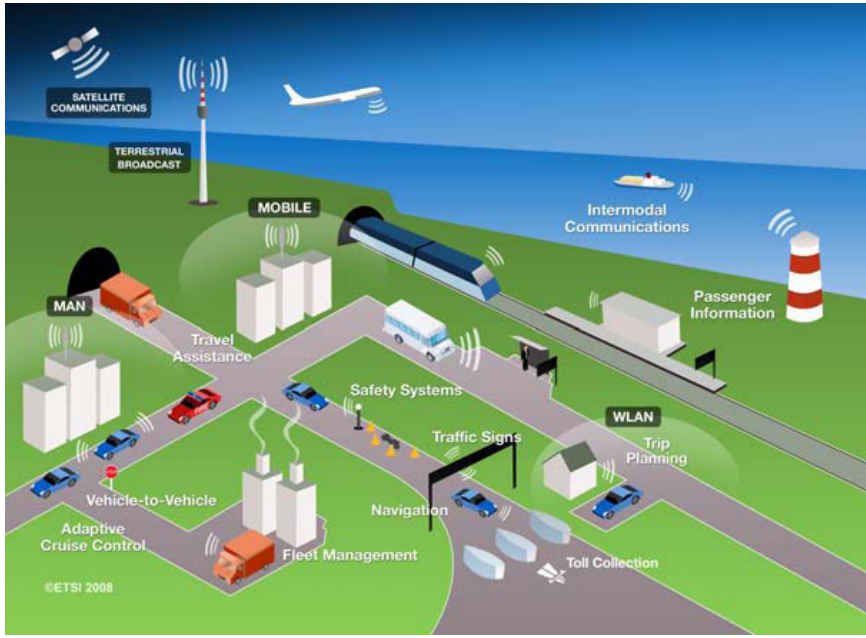
Outline

- Motivation and introduction
- Design principles
- An example
- Conclusions, next steps

MOTIVATION AND INTRODUCTION

Validating Complex Systems

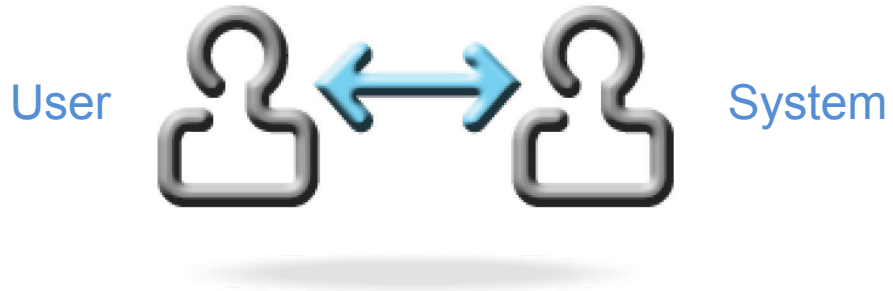
- Engineered systems become more and more complex
  - Complex design (→system of systems)
  - Complex behavior (→real-time)
  - Complex data (→big data)
- Validation and testing need to cope with complexity
  - Proper modeling techniques
  - Proper test automation
  - Proper fault analysis techniques



Intelligent Transport Systems © ETSI

Software Development Turns Agile

- An agile process follows different approaches
  - Story/feature driven modeling
  - Test driven development, etc.
- Leads to scenario-based approach in testing
  - Describe a scenario of interacting with the system
  - Define test objectives from requirements and connect them to scenarios
  - Derive tests from scenarios and automate them



TDL Addresses Needs from Practice

TDL for testing reactive distributed real-time systems

- Provides common black-box testing concepts
- Adjustable to domain-specific needs
- Supporting agile testing process

TDL is standardized

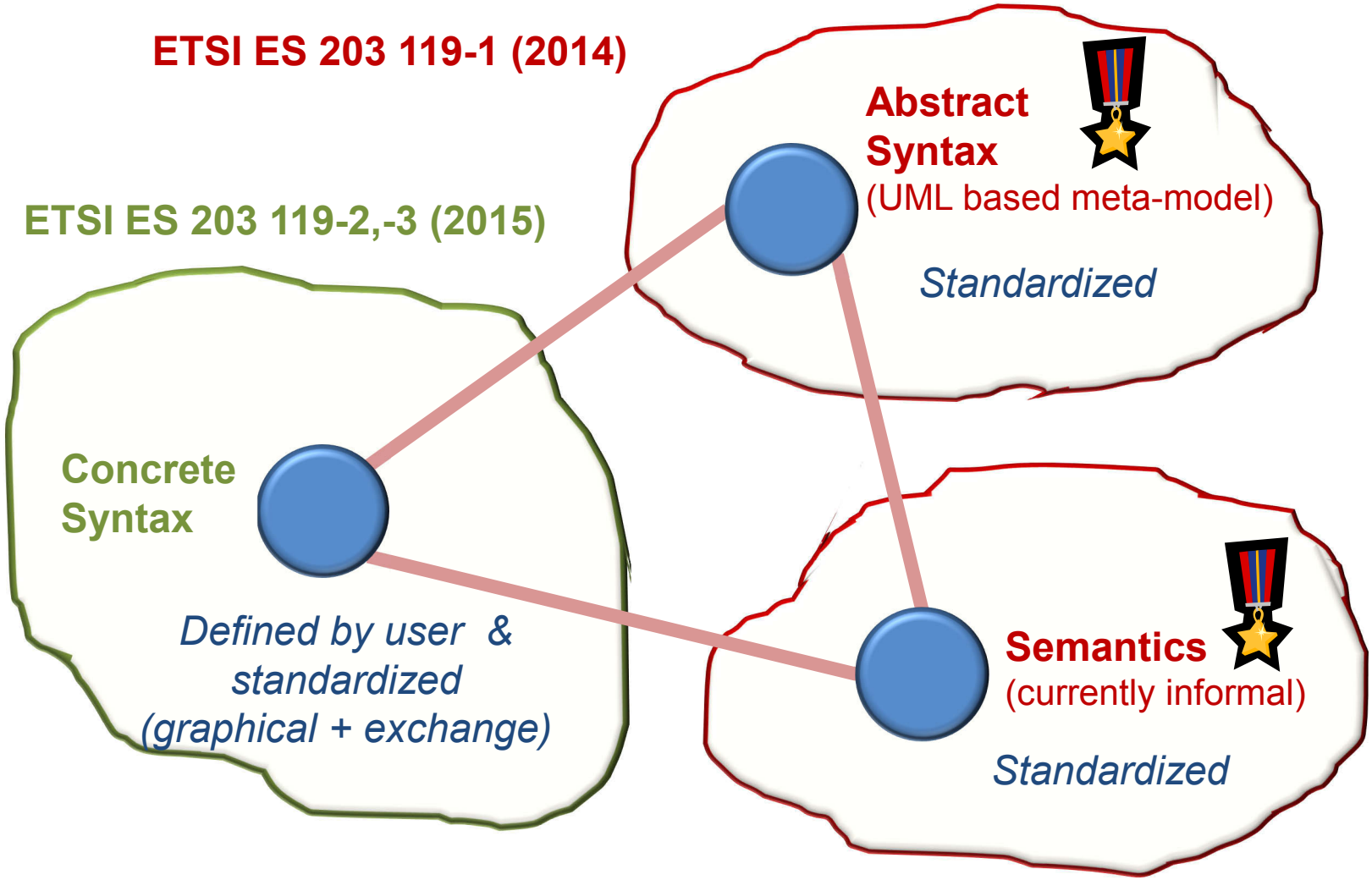
- Clear semantics
- Interoperability of tools and test specifications
- Maintained and kept updated with user needs

TDL use cases

- Manual specification of tests for functional/conformance/interoperability testing
- Representing tests from other sources, e.g. output from MBT test generators
- Documentation of tests

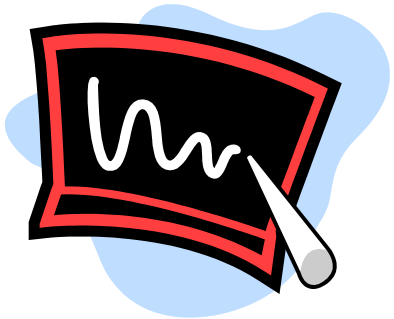
DESIGN PRINCIPLES

ETSI Standards on TDL Design

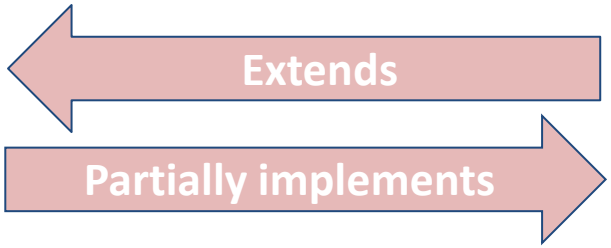


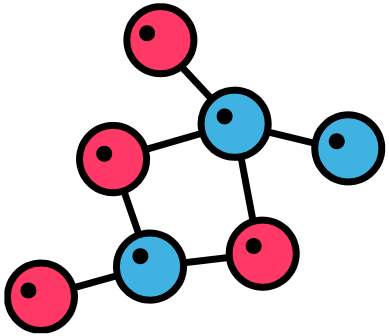
TDL is Adjustable by User

- Concrete syntax may cover only parts of the meta-model
- Meta-model can be extended by a user if need arises
- User extensions of the meta-model can be subjected to further TDL standardization and maintenance



User-defined TDL concrete syntax



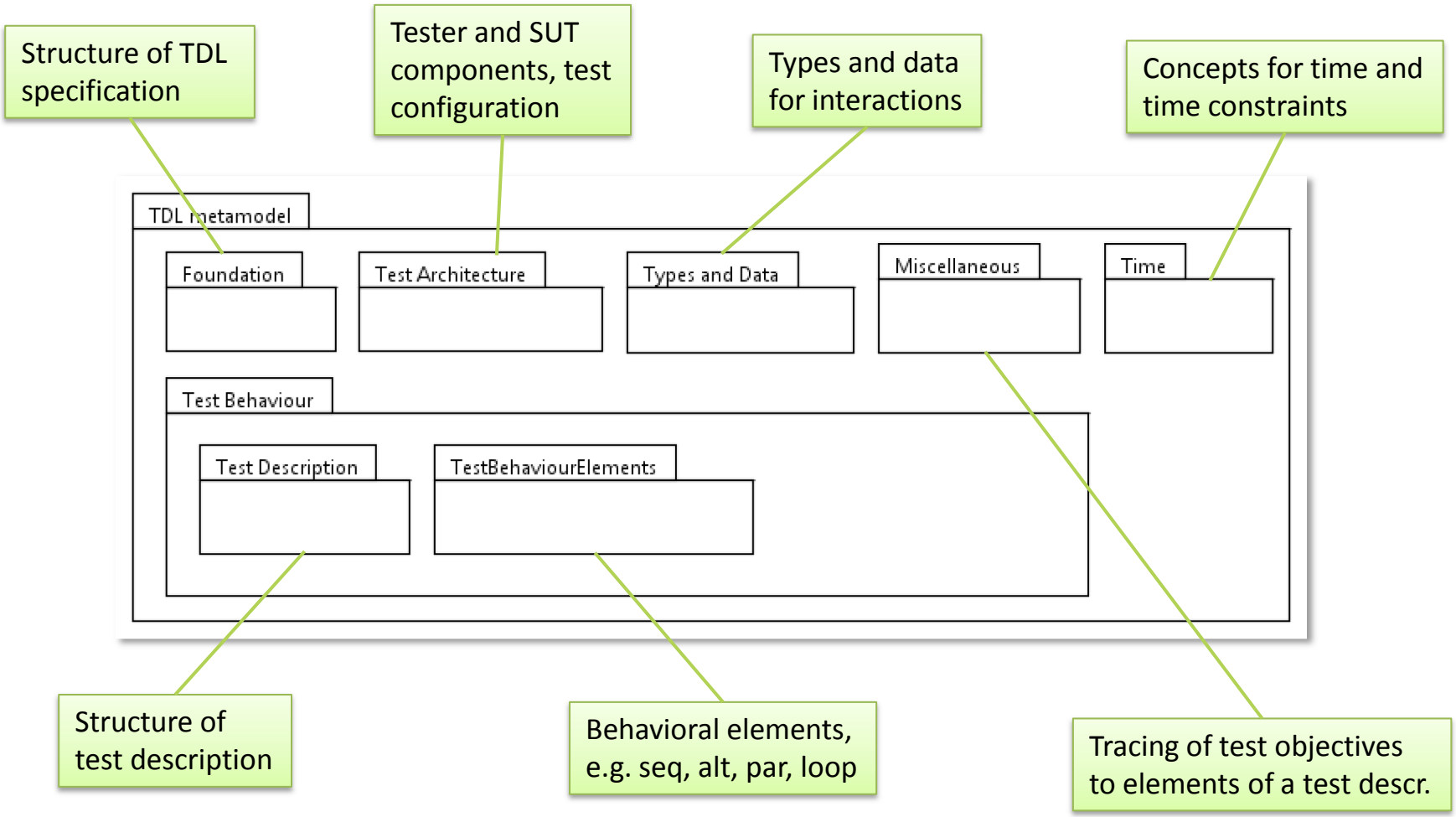


TDL meta-model with user extensions

Key elements of a TDL specification

- Test configuration
  - Set of interacting **components** in the roles **Tester** or **SUT**
- Test description
  - Represents the **expected** foreseeable (passing) behaviour, i.e. any deviation is a fail
  - Expresses a test in terms of **interactions** of data exchanged between tester and SUT components
  - Interactions are **totally ordered**, i.e. they are implicitly synchronized among components
- Test data
  - Represented as abstract **name tuples**

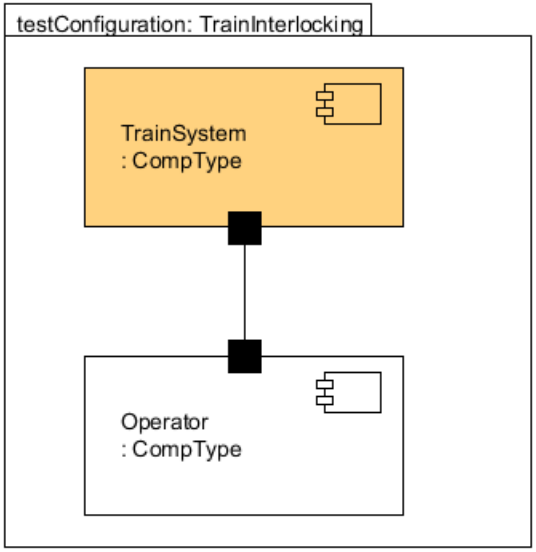
TDL Meta-Model Overview



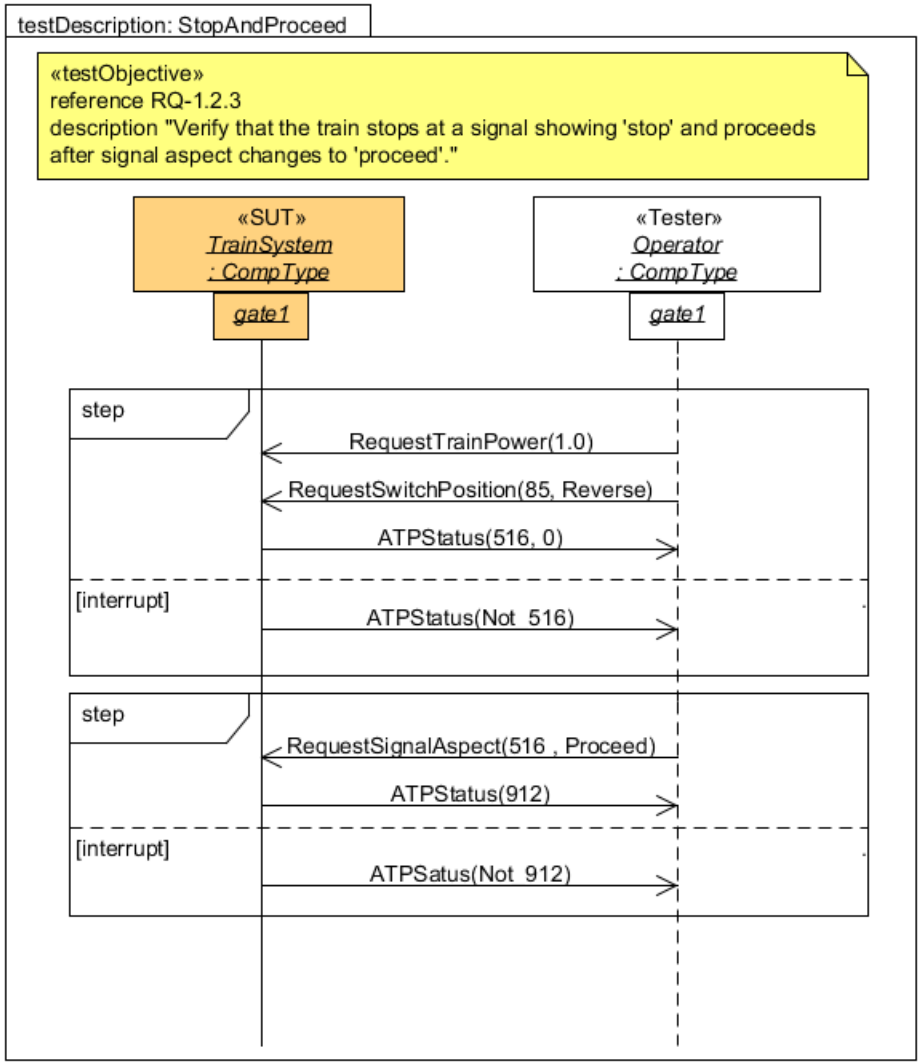
AN EXAMPLE

Example: Scenario on a Rail Interlocking System (Siemens, MBAT)

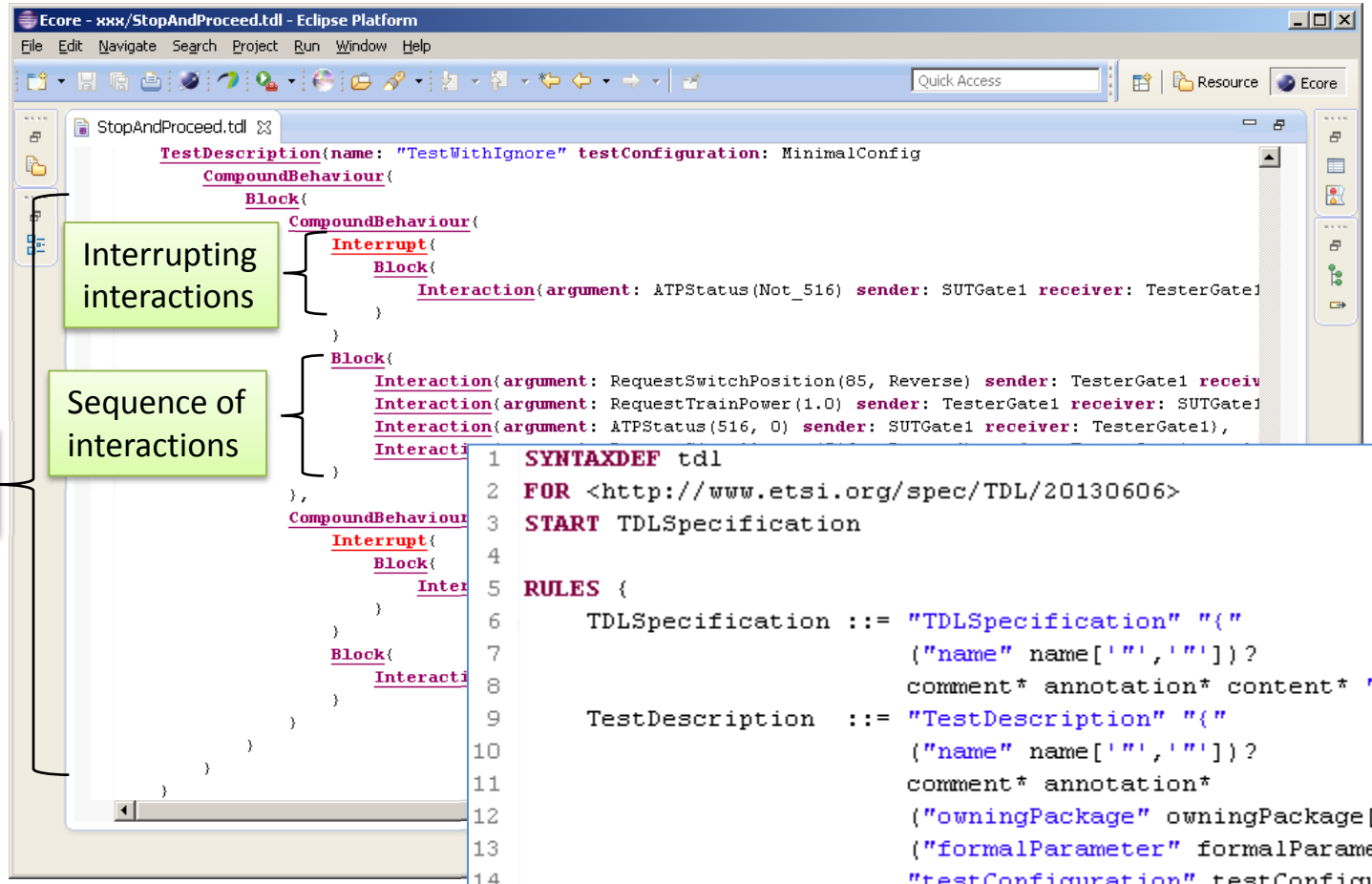
Test Configuration



Test Description



Generated Editor for Textual TDL Specifications (EMFText)




```
1  SYNTAXDEF tdl
2  FOR <http://www.etsi.org/spec/TDL/20130606>
3  START TDLSpecification
4
5  RULES {
6      TDLSpecification ::= "TDLSpecification" "{"
7                           ("name" name['"','"'])?
8                           comment* annotation* content* "}";
9      TestDescription  ::= "TestDescription" "{"
10                          ("name" name['"','"'])?
11                          comment* annotation*
12                          ("owningPackage" owningPackage[])?
13                          ("formalParameter" formalParameter[])*
14                          "testConfiguration" testConfiguration[]
15                          ("testObjective" testObjective[])*
16                          behaviour timeConstraint* "}";
17
```

(Partial) TDL Concrete Syntax

CONCLUSIONS, NEXT STEPS

Conclusions, Next Steps

- TDL meta-model is available currently as an ETSI draft standard
  - Further validation of the meta-model necessary
  - Final draft for publication planned for January 2014
- Next steps
  - Design of concrete syntaxes (graphical + exchange format)
  - Getting tool support: editors, analyzers, test generators
  - Further refinement of the TDL meta-model
  - Extend TDL to support test automation
    - Extensions to ensure executability
    - Composition of test descriptions → User story models

Team & Acknowledgement

- **Team**

Andreas Ulrich, Siemens AG, andreas.ulrich@siemens.com (STF lead)

Gusztáv Adamis, Ericsson, Gusztav.Adamis@ericsson.com

Finn Kristoffersen, Cinderella, finn@cinderella.dk

Philip Makedonski, U Göttingen, makedonski@informatik.uni-goettingen.de

Marc-Florian Wendland, FhG FOKUS, marc-florian.wendland@fokus.fraunhofer.de

Anthony Wiles, ETSI CTI, Anthony.Wiles@ETSI.ORG

- **Acknowledgement**

A. Ulrich, Siemens AG acknowledges partial funding of this activity from the ARTEMIS Joint Undertaking, grant agreement no. 269335 (MBAT) and the German BMBF.