IMPLEMENTING THE STANDARDISED MAPPING OF TDL TO TTCN-3

Philip Makedonski (University of Göttingen)
Jens Grabowski (University of Göttingen)
Overview

Background

- Test Description Language
  - Design, documentation, representation of formalised test descriptions
  - Scenario-based approach

- Testing and Test Control Notation
  - Specification and implementation of all kinds of black-box tests
  - Component-based approach

Why?

- Highly requested
  - brought up every time the mapping is discussed
- More comprehensive standard validation
- Better communication possibilities
- Easier to use
- Easier to understand
- Easier to maintain

Experiences

- Standard specification makes things easier!
  - many hard decisions have already been made
  - focus on realisation instead (not always straightforward)
- Lower level (text-based) specification challenging
  - besides BNF, no official meta-model for TTCN-3 available
  - approximated meta-model enables use of available tools
- Open-source availability big boost
  - view and modify internals when necessary
  - benefits from broader upstream ecosystem

Implementation: Text-based vs Model-based

- High-level, non-linear, traceable
Background

• Test Description Language
  • Design, documentation, representation of formalised test descriptions
  • Scenario-based approach

• Testing and Test Control Notation
  • Specification and implementation of all kinds of black-box tests
  • Component-based approach
Background

• 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)
Why?

• Highly requested
  • brought up every time the mapping is discussed

• More comprehensive standard validation
  • built on top of initial proof-of-concept prototype
  • wider application of the mapping, address corner cases

• Collect and report on experiences
  • implementation and validation of the standard
  • application of model-based technologies
  • evolving the standard
Implementation: Text-based

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.\( g \);
}

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

- Linear, complex, limited, messy
Implementation: Text-based vs Model-based

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;
}

- High-level, non-linear, traceable
Model-based Mapping

- Work with higher level structural representation
  - target structure rather complex
  - syntactical details derived automatically
  - non-linear approach for stepwise enrichment
  - traceability and references to equivalent constructs
  - structural validation already during transformation

- But:
  - standard described with text-based mapping
  - no official meta-model for TTCN-3
Tooling

- Eclipse + EMF - modelling platform
- Xtext - textual mapping for models
- Epsilon / ETL - model-to-model transformation
- MoDisco - tree-based model editing (optional)
- Sirius - graphical model editing (optional)

- TOP - EMF-based TDL tools
- TRex v2 / t3tools v2 - EMF-based TTCN-3 tools
- Custom automation tools
Tooling

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;
}
Tooling
Tooling
Tooling
Experiences

- Standard specification makes things easier!
  - many hard decisions have already been made
  - focus on realisation instead (not always straightforward)
- Lower level (text-based) specification challenging
  - besides BNF, no official meta-model for TTCN-3 available
  - approximated meta-model enables use of available tools
- Open-source availability big boost
  - view and modify internals when necessary
  - benefits from broader upstream ecosystem
Experiences

• Model-based approach
  • focus on essential parts - transformation logic
  • building blocks provided by the platform
  • convenient stepwise enrichment

• Custom tooling to streamline repetitive tasks
  • translation between different formats
    • expected TTCN-3 to model (for analysis)
    • TDLan to model
    • model to model transformations
    • model to TTCN-3
Conclusion

• Current status
  • ~80% of specification covered
  • still a prototype
  • open for contributions

• Future work
  • further refinement towards 100% specification coverage
  • comprehensive testing and evaluation
  • TTCN-3 to TDL?