Dependency and Guideline Analysis for TTCN-3

Steffen Herbold, Philip Makedonski,
Jens Grabowski, Kathrin Becker,
Stefan Kirchner, Benjamin Zeiss

Georg-August-Universität Göttingen, Germany

Outline

- Motivation
- Dependency Analysis
- Guideline Analysis
- Tools
- Summary & Outlook

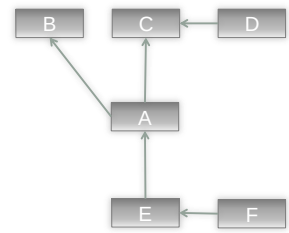
Motivation

- Increasing test suite complexity
- More than 200.000 LOC for next-generation test suites
  - Maintainability?
- Early dev. version of the ETSI 3GPP LTE/SAE test suite:
  - Approx. 20.000 LOC
  - 411 imports
  - 3361 references
  - 2457 external references
  - 904 internal references

→ **Coupling?**

- Enforcing guidelines
  - prevents mistakes
  - reduces effort for maintenance and deliveries

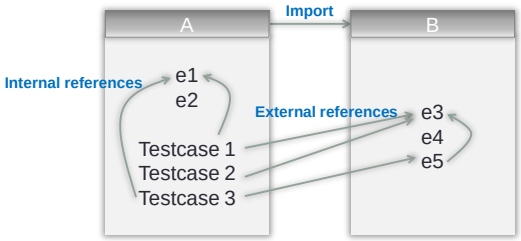
Motivation: Module Dependencies



- How does a local change affect the rest of the test suite?
- Are there any superfluous imports?
- What elements are affected by an element freeze?
- Is a module a library?
- Is a module element public, private, or deprecated?

t3tools TTCN-3 5

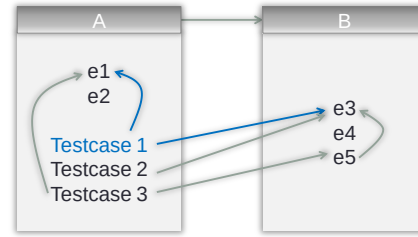
Motivation: Module Dependencies



t3tools TTCN-3 6

Dependency Analysis

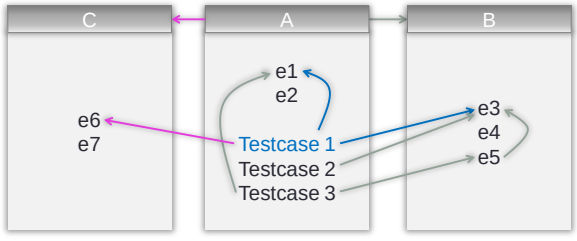
- Local change:



t3tools TTCN-3 7

Dependency Analysis

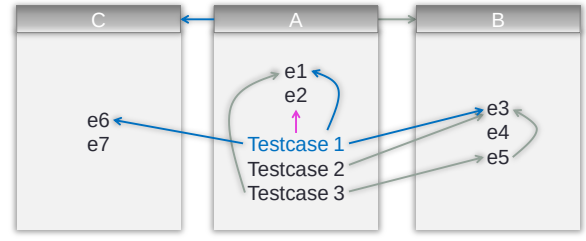
- Local change:
  - Addition of new external dependencies, higher coupling



t3tools TTCN-3 8

Dependency Analysis

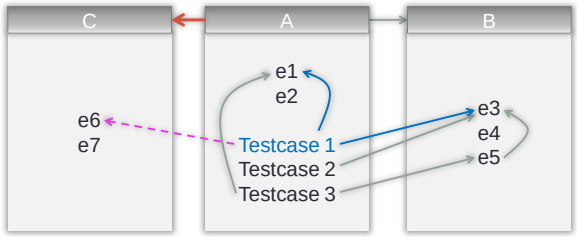
- Local change:
  - Addition of new internal dependencies, stronger cohesion



t3tools TTCN-3 9

Dependency Analysis

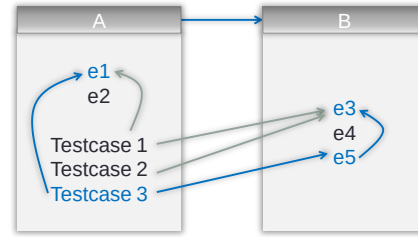
- Local change:
  - Removal of dependencies, less coupling, superfluous imports



t3tools TTCN-3 10

Dependency Analysis

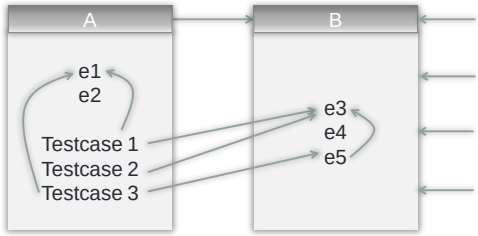
- Element freeze:
  - Testcase 3 is frozen, all dependencies must not be changed anymore



t3tools TTCN-3 11

Dependency Analysis

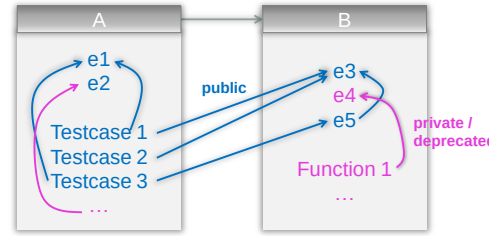
- Primitive library:
  - No further imports, only incoming dependencies, no test cases
- Non-primitive library:
  - No test cases, *mostly* incoming dependencies



t3tools TTCN-3 12

Dependency Analysis

- Public / private / deprecated elements:



Guideline Analysis

- Guidelines are a ***constructive QA measure*** to prevent mistakes or quality problems.
- Guideline analysis is an ***analytical QA measure*** to continuously enforce guidelines during the development.
- Examples:
  - Naming conventions
  - Test data structuring
  - Style conventions
  - Modularization rules

Guideline Analysis: Naming Conventions

- Examples:
  - Test case numbering:
    - **TC_COR_0009_47_ND**
  - Non-Default altstep prefix:
    - **a**_receiveSetup()
  - Default altstep prefixes:
    - **d**_receiveSetup()
- Implications:
  - Better understandability

Guideline Analysis: Test Data Structuring

- Examples:
  - Grouping of related definitions
  - Alphabetic ordering of types within groups
  - Order and placement of local definitions
- Implications:
  - Improved locality ➔ Better understandability, Better maintainability

Guideline Analysis: Style Conventions

- Examples:
  - Formatting style
  - Nesting of alt-statements
  - Depth of stacked template modifications
- Implications:
  - Better understandability
  - Better maintainability
  - Better reusability

Guideline Analysis: Modularization Rules

- Examples:
  - Modules names imply their content
    - TypesAndValues, Templates, …
  - Standard-Imports must exist
    - LibCommonDefs, …
- Other modularization concepts exist!
- Implications:
  - Better locality ➔ Better understandability
  - Bundling of elements that belong together

Dependency / Guideline Relationships

- Dependencies promote quality attributes:
  - Bad quality affects dependent modules
  - High Fan-In ➔ Big Impact on Quality
  - Determination of modules with high risk
- Guidelines may involve dependencies:
  - No unused imports
  - Standard imports must exist
  - Over-specific runs on clause

Tools

- T3Q
  - Static guideline checking
- T3D
  - HTML Documentation Generator (Javadoc-like)
- T3Pendency
  - Test-Suite Dependency Analysis
- Open-Source
  - Eclipse Public License (EPL)
  - Based on the TRex infrastructure
  - TTCN-3 v4.1.1 support
  - Cross-platform
  - Command-line tools, scheduled execution possible

T3Q – TTCN-3 Guideline Checker

- Fine-grained XML configuration with project profiles
- Approx. 30 guideline checks implemented
  - Naming conventions
  - Log format must match a regular expression
  - No unused definitions on module level
  - Templates module must contain only template definitions
  - No unused imports
  - No "all" keyword in port type definitions
  - No label or goto statements
  - …
- Code formatting
- Basic size metrics (LOC, No. of test cases,…)

T3D – TTCN-3 Documentation Generator

- XML representation of module definition dependencies
- Generation of different switchable views using XSLT:
  - Main view
    - TTCN-3 listings with cross-links
  - Testcase view
    - Dependencies between test cases and module parameters
  - Import view
    - Import relationships
  - Documentation as HTML
- Customizable look & feel

T3D – Main View

T3D – Import View

imports	Modules	imported by
configuration all; types all; functions all; Indirect dependencies: modulepars	modulepars import3 configuration **testcases** main types templates import2 functions import4 templates-0 import1 ExampleModule	main all;

T3D – Module Parameter View

Index / module_with_moduleParameters / par1 - Module Parameter/Testcase View

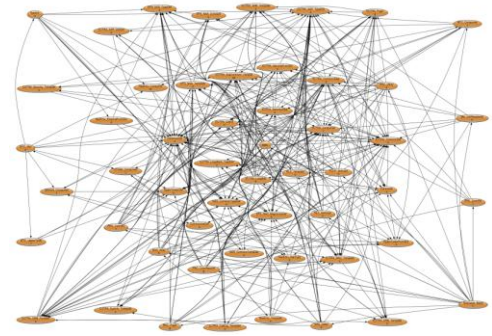
par1

Testcase	Path
testcase1	>> par1
testcase1_2	>> par1
testcaseAll	>> par1
testcasef1	>> function1 >> par1
testcaseloop1_1	>> function1 >> par1
testcaseloop1_2	>> testcaseloop1_1 >> function1 >> par1
testcaseloop1_3	>> testcaseloop1_2 >> testcaseloop1_1 >> function1 >> par1
testcasetcf1	>> testcasef1 >> function1 >> par1

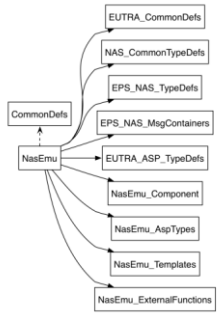
T3Pendency – TTCN-3 Dependency Analyzer

- Calculation of dependency metrics:
  - Number of Imports / Number of superfluous Imports
  - Number of modules that reference a given module (Fan-In)
  - Number of modules referenced by a given module (Fan-Out)
  - Number of internal / external references
- Can be determined at the level of:
  - Modules
  - Module definitions
- Public / private suggestions
- Graphviz visualization

T3Pendency – TTCN-3 Dependency Analyzer



T3Pendency – TTCN-3 Dependency Analyzer



Summary & Outlook

- Summary:
  - Dependency analysis
  - Guideline analysis
  - Relationships between Dependencies and Guidelines
  - T3Q, T3D, T3Pendency tools
- Outlook:
  - Freely available, open-source (EPL)
  - Download at http://t3tools.informatik.uni-goettingen.de
  - TRex for Refactoring and Metrics, IDE:
    - http://www.trex.informatik.uni-goettingen.de
  - More guideline checks, more features, but …
    - No commercial support → community-driven tool maintenance!

Contact

- Websites:
  - http://www.trex.informatik.uni-goettingen.de
  - http://t3tools.informatik.uni-goettingen.de
- E-Mail:
  - t3tools@informatik.uni-goettingen.de
- Acknowledgments:
  - ETSI CTI, STF 160