



TRex

An Eclipse-Based Tool for
TTCN-3 Editing, Refactoring and Metrics

Paul Baker¹, Dominic Evans¹,
Jens Grabowski², Helmut Neukirchen², Benjamin Zeiss²

¹Motorola Labs,
Viabes Industrial Estate, Basingstoke, UK

²Software Engineering for Distributed Systems Group,
University of Göttingen

Outline

- **Motivation**
- Metrics
- Refactoring
- TRex Tool
- Summary / Outlook

Motivation

- Huge legacy test suites at Motorola:
 - Migration to TTCN-3
 - Automatic conversion of a UMTS test suite:
 - 60,000 lines of TTCN-3 code
 - Hard to **read, use, re-use, and maintain.**
- Current TTCN-3 tools:
 - Editors, Compiler, Test-Execution
 - But: **No support for improving test suites!**

Approach

- Assess test suites,
 - Detect issues,
- } → **Metrics,
further Analysis**
- Restructure test suites. → **Refactoring**

Outline

- Motivation
- **Metrics**
- Refactoring
- TRex Tool
- Summary / Outlook

Metrics

“You cannot control what you cannot measure.”

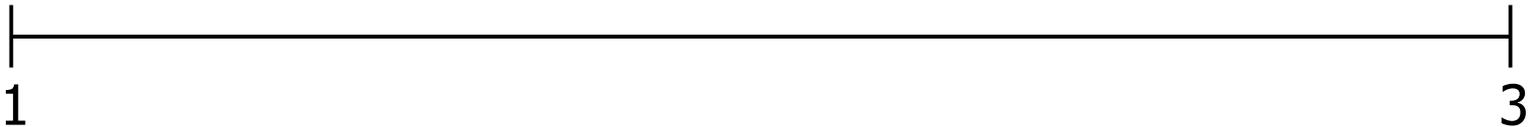
De Marco: *Controlling software projects*.
Yourdon Press, 1982

- Software metrics:
 - Size metrics,
 - Complexity metrics,
 - Object-oriented metrics.

TTCN-3 Metrics: Template Coupling

- Measure dependency between test data and test behaviour.
 - High coupling: Changing test data requires changing test behaviour
 - Low coupling: Changing test data requires no changes to behaviour.

Low template coupling **High template coupling**


$$\text{Template Coupling} := \frac{\sum_{i=1}^n \text{score}(\text{stmt}(i))}{n}$$

$$\text{score}(\text{stmt}(i)) := \begin{cases} 1, & \text{if stmt}(i) \text{ references a template without parameters.} \\ 2, & \text{if stmt}(i) \text{ references a template with parameters.} \\ 3, & \text{if stmt}(i) \text{ uses an inline template.} \end{cases}$$

Rule-Based Issue Detection

- Metrics-based:
 - *Number of references to a template = 0*
⇒ Remove template
 - *Number of references to a template = 1*
⇒ Consider inlining the template
- Further analysis:
 - Identical actual parameter value
⇒ Inline template parameter
 - Several templates only differ at the same element
⇒ Parameterize template

Outline

- Motivation
- Metrics
- **Refactoring**
- TRex Tool
- Summary / Outlook

Refactoring: Definition

„a change made to the internal structure of software to make it easier to understand and cheaper to modify without changing its observable behavior“

Fowler, M.: Refactoring – Improving the Design of Existing Code. Addison-Wesley (1999)

TTCN-3 Refactoring Catalog

- 28 refactorings from Fowler applicable to TTCN-3,
- 20 refactorings specific for TTCN-3.
- Catalog structure: Refactorings for improving
 - Test behaviour (20 refactorings):
 - *Extract Altstep*,
 - ...
 - Overall test suite structure (22 refactorings):
 - *Extract Module*,
 - ...
 - Data description (6 refactorings):
 - *Inline Template Parameter*,
 - ...

Refactoring

- Refactoring Format:
 - Name, Summary, Motivation, Mechanic, Example
- Example (Inline Template Parameter):

```
module ExampleModule {  
  template ExampleType exampleTemplate(charstring addressParameter) := {  
    ipv6 := false,  
    ipAddress := addressParameter "127.0.0.1"  
  }  
  
  testcase exampleTestCase() runs on ExampleComponent {  
    pt.send(exampleTemplate("127.0.0.1"));  
    pt.receive(exampleTemplate("127.0.0.1"));  
  }  
}
```

Outline

- Motivation
- Metrics
- Refactoring
- **TRex Tool**
- Summary / Outlook

TRex Features (1/2)

- TTCN-3 core notation editing
 - Syntax Highlighting, Code Completion, Outline View, Code Formatter, Text Hover, Open Declaration, Reference Finder
- Compiler Integration
 - Telelogic Tau G2/Tester
- Metrics
- Rule-Based Refactoring Suggestions
- Refactorings:
 - Rename
 - Inline Template
 - Inline Template Parameter
 - Merge Template

TRex Features (2/2)

Metric	Total	References
Number of altsteps	8	
Number of components		
Number of functions	5	
Number of others	11	
Number of ports		
Number of templates	32	
proto_templates.ttcn	32	
bsc1 : stim_bsc_t		19
sccp_id_100 : sccp_id_t		14
address0 : address_t		2
address1 : address_t		2
bsc0 : stim_bsc_t		2
call_req_ack1 : CallReqAck_t		2
CallReq_t1 : CallReq_t		1
ResReq_t1 : ResReq_t		1

Quick Fix

Select the fix for "Template is never referenced. Consider removing."

Select a fix:

remove template

Problems:

Problem	Location
<input checked="" type="checkbox"/> dataDefinition.ttcn3	line 24
<input checked="" type="checkbox"/> dataDefinition.ttcn3	line 20

Select All
Deselect All
Find Similar Problems

OK Cancel

0 errors, 6 warnings, 0 infos (Filter matched 6 of 56 items)

Description	Resource	Location
TRex Merging Rules (3 items)		
⚠ This and 2 templates 'dataNullHello, dataZeroHi' could be parametrised on field: 'payload'.	dataDefinition.ttcn3	line 32
⚠ This and template 'dataZeroHi' could be parametrised on field: 'seqNo'.	dataDefinition.ttcn3	line 42
⚠ This is a duplicate of template 'dataNullHello'.	dataDefinition.ttcn3	line 32
TRex Never Referenced Rule (2 items)		
⚠ Template is never referenced. Consider removing.	dataDefinition.ttcn3	line 20
⚠ Template is never referenced. Consider removing.	dataDefinition.ttcn3	line 24
TRex Referenced Once Rule (1 item)		
⚠ Template referenced only once. Consider inlining.	dataDefinition.ttcn3	line 37

Outline

- Motivation
- Metrics
- Refactoring
- TRex Tool
- **Summary / Outlook**

Summary

- Means for improving TTCN-3 test suites
 - Metrics
 - Rule-based issue detection
 - Refactoring
- TRex:
 - IDE for TTCN-3
 - Automatic assessment and restructuring

Outlook

- Current work
 - Enhanced editing functionality
 - Metrics based on control-flow and call-graphs
 - Identification of useful metrics boundary values
- Outlook
 - Implementation of more refactorings
 - Pattern-based issue detection
 - Bi-simulation to prove equivalence of test suites

TRex Availability

www.trex.informatik.uni-goettingen.de

Welcome to the TRex Website

The comprehensive test of modern communication systems leads to large and complex test suites which have to be maintained throughout the system life-cycle. Experience with those written in the standardised Testing and Test Control Notation (TTCN-3) has shown that the maintenance of test suites is a non-trivial task and its burden can be reduced with appropriate tool support. To this aim, we have developed the TRex tool, published as open-source Eclipse plugin under the Eclipse Public License, which provides IDE functionality for the TTCN-3 core notation and supports the assessment and automatic restructuring of TTCN-3 test suites by providing suitable metrics and refactorings.

We invite interested Java programmers to contribute as well to the development of TRex.

2006/05/16 - TRex issues tracked with JIRA

Atlassian supports our efforts by contributing their bug tracking and issue tracking application, JIRA, to our project.

- Open Source
- Eclipse Public License
- Available Now!
- Join the project!

- Thank you for your attention!



Please visit us at the tool exhibition!

www.trex.informatik.uni-goettingen.de