

Develop a Reliability Test in TTCN-3

Wu Ji, Xu Luo, Liu Chao
Beihang University

Who are we?

- We are from SEI in Beihang University
 - The weightlifting of 2008 Olympic Game would happen there
- Our main efforts are on software testing, process improvement, MDA
- MDT and TTCN-3
 - support by China high-tech program
 - Collaborate with FOKUS and Testing Tech.

Outline

- Motivation
- System architecture
- Test suite generation
- Test execution
- Observations
- Summary

Motivation

- TTCN-3 is announced to be applicable for
 - Conformance test
 - Interoperability test
 - Functional test
 - Reliability test
 - Load test
 - Performance test
- reliability is important in many fields, potential users from these fields in China come up with the question
 - How to implement the reliability test in TTCN-3? (Is it possible?)
- Less effort has been given on reliability test in TTCN-3 community

Motivation

□ What is Reliability Test?

- Statistically sample test cases according to usage profile
- Evaluate reliability based on the <#fail, time_point>

□ Differences with functionality test

- Focus on the combinations of usages, i.e. invocations of SUT features
- Test cases must be representative of user profile

Motivation--Issues

□ Test case

- How to *get* the large volume of test cases?

□ Test

- • Test case generation from MCUM
- • Test data management in datapool
- • Collect failure record from trace log

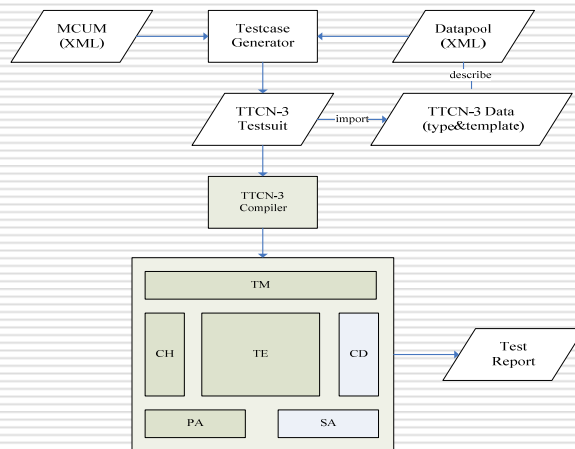
□ Test result

- How to collect failure records with time point?

Outline

- Motivation
- System architecture
- Test suite generation
- Test execution
- Observations
- Summary

System Architecture



Outline

- Motivation
- System architecture
- Test suite generation
- Test execution
- Observations
- Summary

TTCN-3 Test suite generation

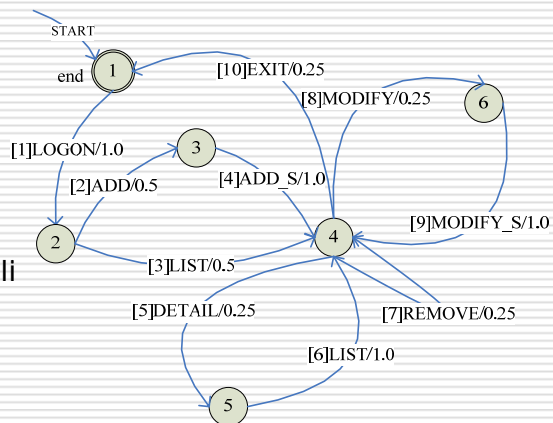
- Process
 - Sample usage scenario from model
 - Transfer into TTCN-3 test case
 - Compose TTCN-3 Executable (TE)
- SUT Usage Model
- Datapool Design
- Generate TTCN-3 test case

SUT Usage Model

- SUT
 - Web-based Project Management System
- Features to be tested
 - Logon & Exit
 - Add/Delete/Modify a project
 - Return project detail information
 - Search projects
 - List all projects

SUT Usage Model

- SUT MCUM (Markov Chain Usage Model)
 - Node --- SUT state
 - Line --- User stimuli
 - [Start & End state]



Datapool Design

- What is *datapool* ?
 - A *pool* to hold the data for Atom Action
- Atom action
 - Is a *transition* in MCUM
 - Is a stimuli/response pair from state A to state B
 - Is an interaction between the user and the SUT
 - Is a TTCN-3 '*send & receive*' pair
- Datapool & Atom action
 - Each *atom operation* has a type ('LOGON', 'EXIT'...)
 - *Datapool* is accessed by the tag identical to the *atom action* type

8/30/2007

T3UC 2007

13

Datapool Design

```
<?xml version="1.0" ?>
- <datapool description="" xmlns:xsi="http://www.w3c.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="Datapool.xsd">
- <stimuli name="LOGON">
+ <choice>
- <choice>
  <send template="LOGON_2_Req" /> } equivalences
  <recv template="LOGON_2_Res" />
</choice>
</stimuli>
+ <stimuli name="LIST">
+ <stimuli name="DETAIL">
+ <stimuli name="EXIT">
+ <stimuli name="ADD">
+ <stimuli name="ADD_SAVE">
+ <stimuli name="MODIFY">
+ <stimuli name="MODIFY_SAVE">
+ <stimuli name="REMOVE">
</datapool>
```

8/30/2007

T3UC 2007

14

Generate TTCN-3 testsuite

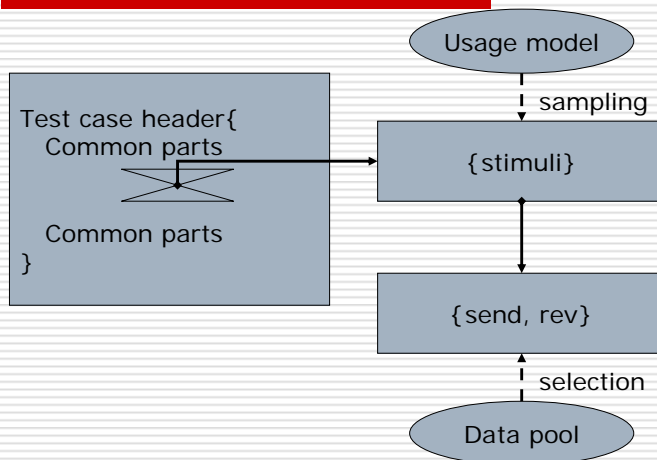
- Goal
 - Generate TTCN-3 testscript based on the MCUM
- Input
 - MCUM.xml
 - Datapool.xml
 - Map stimuli data tag to ttcn-3 template
 - Other parameter
 - TestcaseCount
- Output
 - ReliabilityTest.ttcn3

8/30/2007

T3UC 2007

15

Approach overview



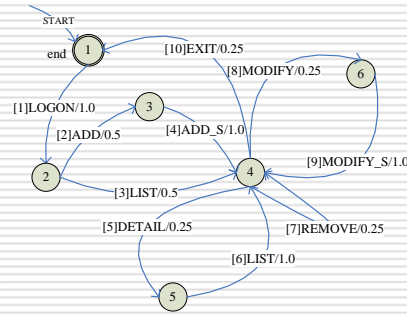
8/30/2007

T3UC 2007

16

Generate TTCN-3 *testcases*

- A *testcase* is a *travel* from the start-state to the end-state
- *Travel* means statistical sampling
- Each state may have several stimuli choices (lead to different states)
- Which to choose depends on the stimuli distribution and the sampling strategy



TRAVEL1 – TC1

[1]LOGON
[3]LIST
[10]EXIT

TRAVEL2 – TC2

[1]LOGON
[2]ADD
[4]ADD_S
[10]EXIT

Generated TTCN-3 Test case

```
function MyFunc(in Request req, in
Response res) runs on mtcType {
  mtcPort.send (req);
  localtime.start;
  alt {
    [] mtcPort.receive (res) {
      localtime.stop;
      setverdict (pass);
    }
    [] mtcPort.receive {
      localtime.stop;
      setverdict (fail);
    }
  }
  [] localtime.timeout {
    setverdict (fail);
  }
}
```

```
testcase tc_x() runs on mtcType system systemType
{
  map (mtc: mtcPort, system: systemPort);
  MyFunc(LOGON_1_Req, LOGON_1_Res);
  if(getverdict == fail) {
    self.stop;
  }
  MyFunc(ADD_1_Req, ADD_1_Res);
  MyFunc(ADD_SAVE_1_Req, ADD_SAVE_1_Res);
  MyFunc(DETAIL_1_Req, DETAIL_1_Res);
  MyFunc(LIST_1_Req, LIST_1_Res);
  MyFunc(REMOVE_1_Req, REMOVE_1_Res);
  MyFunc(DETAIL_1_Req, DETAIL_1_Res);
  MyFunc(LIST_1_Req, LIST_1_Res);
  MyFunc(EXIT_1_Req, EXIT_1_Res);
  unmap (mtc: mtcPort, system: systemPort);
}
```

Deal with Test data

- Two kinds of test template
- template to send
 - Request: URL
 - *Format: {protocol://host:port/path?param}*
- template to match the receive
 - Response: expected web page
 - matching point: *table&element*
 - *Specific to the SUT*

Outline

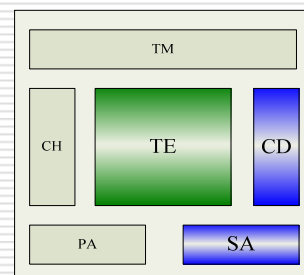
- Motivation
- System architecture
- Test suite generation
- **Test execution**
- Observations
- Summary

Test execution

- Test System
- Test Result
- Problems

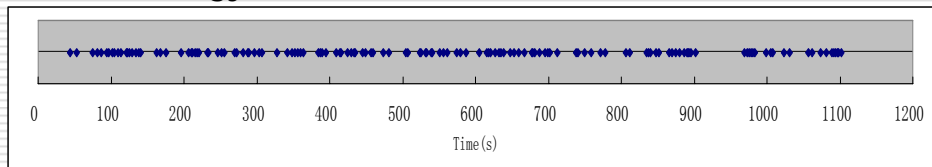
Test System

- TE:
ReliabilityTest.ttcn3, Datapool_
type.ttcn3,
Datapool_template.ttcn3
- TestAdapter
 - Deal with the http connection
 - Send & receive data via TCP/IP
- Codec
 - Deal with the web page
 - Parsing the html page
 - extracting the *table&element*



Test Result

- ❑ Generate & Run 500 TTCN-3 testcases (25 minutes)
- ❑ We got 357 passed and 143 failed: disasters?!
 - Mostly are no-response failures
- ❑ But, in functionality test, no failure!
- ❑ The usage profile based stimuli combination strategy works!



8/30/2007

T3UC 2007

23

Outline

- ❑ Motivation
- ❑ System architecture
- ❑ Test suite generation
- ❑ Test execution
- ❑ **Observations**
- ❑ Summary

8/30/2007

T3UC 2007

24

Observations

- Based on our solution and the development, we observe
 - TTCN-3 has deficiencies to manage and select test templates
 - TTCN-3 has deficiencies to support test cases generation
 - Test time control and access is difficult
 - Log analysis is difficult
 - Codec development is difficult

Test template Management and Selection

- Template coding and selection
 - manual way is hard and boring for a large volume of templates requirements
 - may cause unconscious errors
- We actually do (beyond TTCN-3 language)
 - design a datapool to manage test templates
 - develop an algorithm to implement test data selection → test case
- We need in fact
 - template organization mechanism (*group* is not enough)
 - test partition management
 - allow for *random choose* templates to test case

Test Case generation

- Auto-generation of test case is a must
- We actually do (also beyond TTCN-3)
 - Design the usage model
 - Design a sampling algorithm to get usage scenarios
 - Generate TTCN-3 test case code from usage scenario
- We need in fact
 - A mechanism to support test case generation from models

8/30/2007

T3UC 2007

27

Test time Control and Access

- Time is a key factor in reliability test
 - Control *simulated user* access frequency
 - Too frequent access would be blocked for some web applications
 - Collect the time point that SUT failed
- We actually do
 - By timer operations to delay test cases one by one
 - Record failure time point from test system trace log
- We need in fact
 - to flexibly parameterize the time delay btw test cases
 - to record failure time point automatically

8/30/2007

T3UC 2007

28

Log Analysis

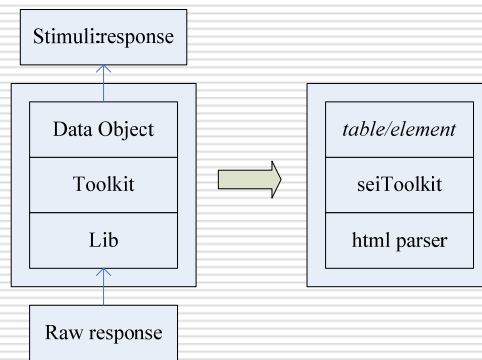
- ❑ TTCN-3 support tool provides all the test trace data, but very limited log analysis feature
- ❑ User needs a stand alone log analysis tool
 - Filter log according to test case, test component, port, etc
 - Associate log records

Codec Development

- ❑ Web-based SUT
- ❑ Codec is the most complex and difficult part in this experiment
 - HTML: markup language, large size, full of noise data
- ❑ TTCN-3 community does not deliver specific abilities for Web-based test

Codec Development

□ Codec architecture & our implementation



Outline

- Motivation
- System architecture
- Test suite generation
- Test execution
- Observations
- **Summary**

Summary

- Reliability test in TTCN-3 is feasible, but, MUST pay enough attentions on
 - Test case generation
 - Template organization and selection
 - Test time control and access
 - Log analysis
 - Codec development

Summary

- Test case generation and template organization are common requirements for many test
 - Should we propose CRs to ETSI?
- Thanks for the attention, any questions?