CAR/MEM/VTT

Refactoring and Converting a TTCN-2 Test Suite

TTCN-3 User Conference 2005

CONNECTING PEOPLE

Business Card

- Thomas Deiß
- Nokia Research Center Bochum
- Principle Scientist
  - TTCN-2, TTCN-3, SDL, ASN.1, C
  - Test System Development
- thomas.deiss@nokia.com

CAR/MEM/VTT

Background Situation

- TTCN-2 used successfully
- TTCN-3 appearing as new language
- How to convert TTCN-2 test suites to TTCN-3 to preserve investment?

Agenda

- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

Agenda

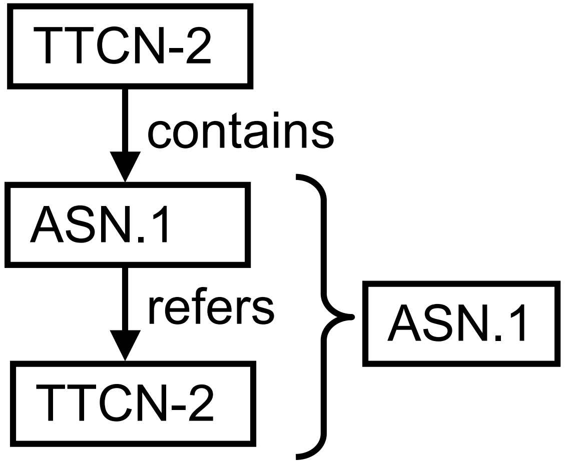
- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

Why Conversion ?

- **_not_** because TTCN-3 is a nice language
- New features added, increased expressiveness
  - Procedure-based communication
  - Dynamic Configurations
  - Extensible, XYZ to TTCN-3 mapping
- Increased maintainability
  - Improved support for modularity
- Align with technology for new test systems (TTCN-3)
- Mandatory: Significantly less effort for conversion than for development from scratch

Success Story

- Conversion of the Bluetooth SPP test suite
  - Approx. 43 kloc of TTCN-2
  - Approx. 30 kloc of TTCN-3
- Almost automated conversion
  - ASN.1 types in the TTCN-2 test suite
  - Refer again TTCN-2 types
  - Cannot be converted to TTCN-3
- About 2 hours needed to convert refered to TTCN-2 types to ASN.1
  - About 15 minutes when repeated



- (Almost) automated conversion is possible

Agenda

- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

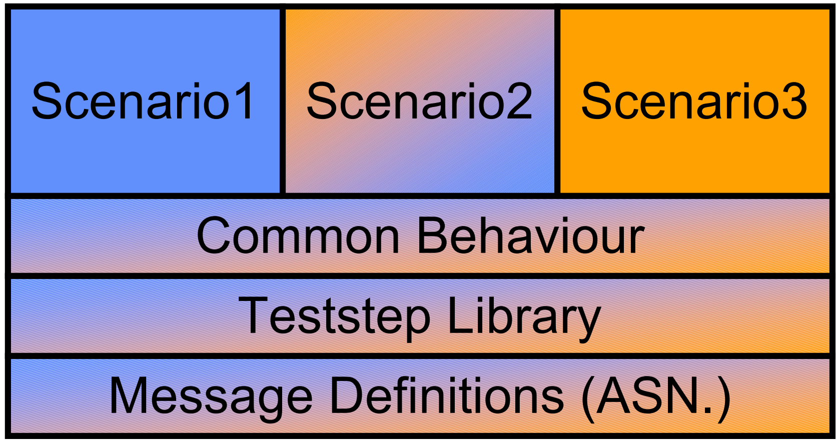
Test Suite to be Converted (I)

- SUT: UMTS Network Element (NE)
- TTCN-2 test suite
  - Continously extended
  - Error corrected in existing parts
- ASN.1 message definitions
  - Evolve together with standards
- System Under Test Adapter
  - Several different protocol stacks
  - Several layers per protocol stack
  - Evolve together with standards

- Test System evolves while being converted

Test Suite to be Converted (II)

- Test System is large
  - Approx. 2500 test cases
  - Approx. 1.1 mloc / 31MB TTCN-2 files
  - Approx. 120 kloc ASN.1 message definitions
- Just 5 files, resulting in 3 test suites
  - Developed at different sites



Scenario1

Scenario2

Scenario3

Common Behaviour

Teststep Library

Message Definitions (ASN.)

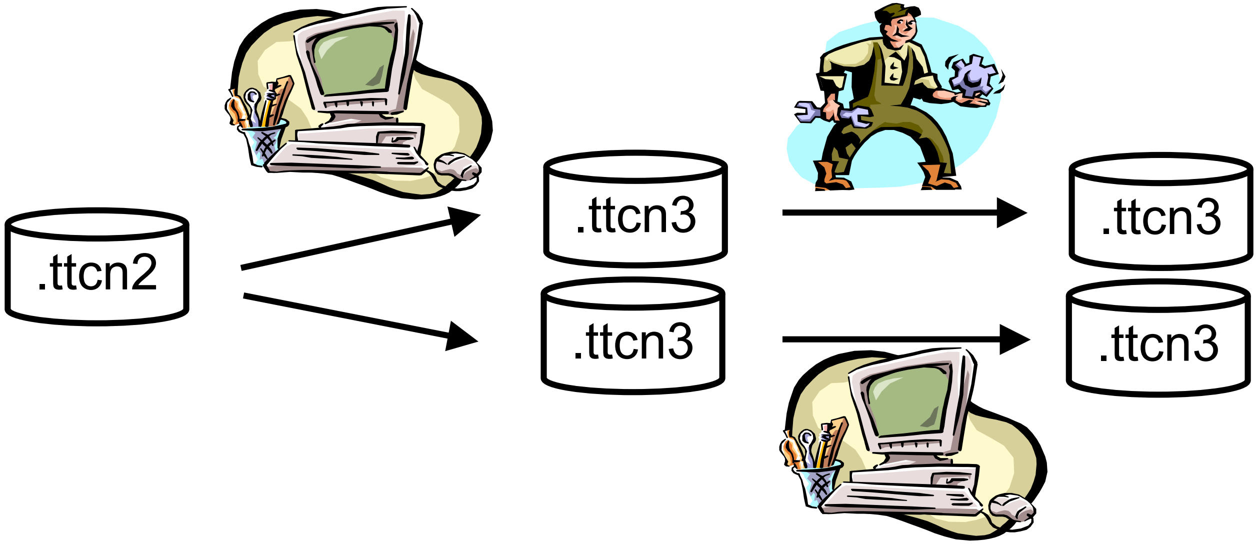
Goal of Conversion

- Increase maintainability
  - Better modularity support in TTCN-3
  - TTCN-2 tool used does not support modularity
- Increased expressiveness of TTCN-3 for complicated test scenarios

- SUT is expected to be developed further
  - Effort worth spent

Approach

- No automated conversion
- Maintainability increasement requires refactoring
  - Automated conversion cannot improve structure of the code
  - Convert by a tool, then reassemble manually



- Refactoring requires additional validation

CAR/MEM/VTT

Agenda

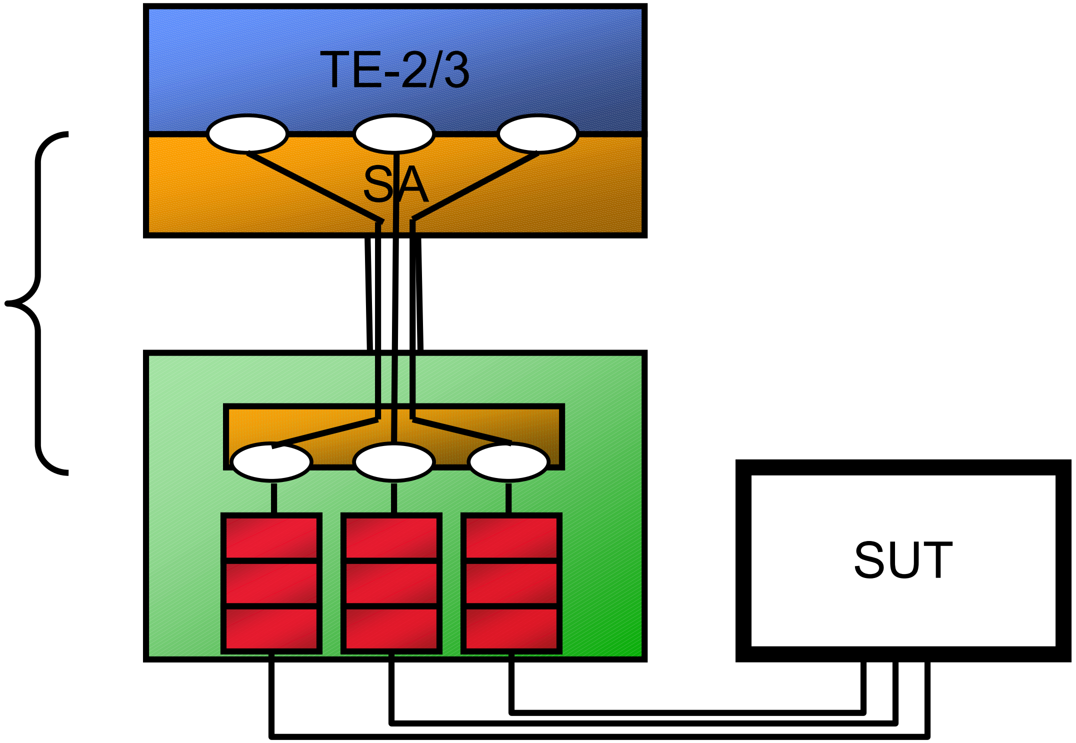
- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

Test System Conversion
SUT Adapter

Generic distributed SA

easy to use here

2 man weeks to use existing SA



- SUT Adapter has to be changed also
- Here: rather smoothly
- In general: has to be investigated case by case

Test System Conversion
Codec

- Messages are defined in ASN.1
- Encoded with standardized encoded (PER)
- TTCN-3 tools generate PER codecs
- Some few man weeks for specific types

Agenda

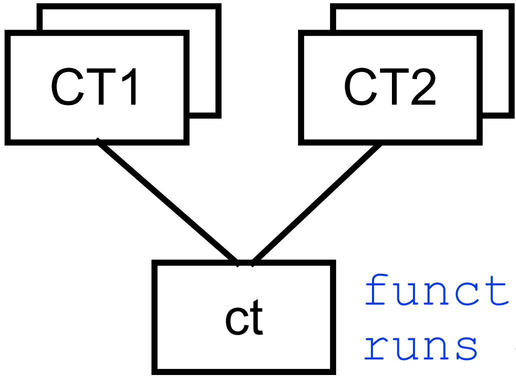
- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

TTCN-2 to TTCN-3 Conversion

- Conversion tool has been used
  - TTtwo2three by Testing Technologies
- Tool evaluated and improved in pilot phase
  - General improvements
  - Specific improvements for this test systems
- General requirements
  - Readable TTCN-3 code
  - Reasonable turnaround times

Converter Improvement I
Manually written Component Types

- General conversion by TTtwo2three
  - TTCN-3 for a single component type
  - Record type for test suite and test case variables
  - Auxiliary functions used to ensure TTCN-2 semantics
- Goal:
  - Use different component types
  - As specific as possible
  - Use compatibility of components
  - Use component variables instead of records to simplify code



```
function f()
runs on ct {
...}
```

- Manually written component types
- TTtwo2three adapted appropriately

Converter Improvement II
Improved Modularization

- For each protocol
  - Type definitions (ASN.1)
  - Templates, behaviour
- Own module for large test case
- Much more modules than in TTCN-2 test suite

- (Manually) define mapping of identifiers to modules
- Converter generates code in the correct module
  - Generate also the `import` statements

Converter Improvement III
TTCN-2 Idioms

- Get rid of TTCN-2 idioms
- Condition '1 == 1' used for syntactical reasons
-
```
if ( 1 == 1 ) {
  <statementlist>
}
```
- Check for occurrences of '1==1' on syntactical level
```
<statementlist>
```

Converter Improvement IV
Test Steps

- (TTCN-2) teststep ➔ (TTCN-3) altstep
- Teststeps can start with `send` statement

```
altstep alt_ () ... {
  [else] {
    pt.send (...) ...
```

```
function f_ () ... {
   pt.send (...) ...
```

- Superfluous `alt`
- Code hard to read
- Runtime penalty
  - Lots of `alt` statements
  - Lots of snapshots

- Empirical Result: 95% of teststeps as functions
- Handled by Converter
- Idea: Stephan Schulz

Agenda

- Introduction
- Test System to be Converted
- Test System Conversion
- TTCN-2 Test Suite Conversion
- Summary

CAR/MEM/VTT

Summary

- Conversion of large test suites is possible
  - Sometimes even almost automatically
- Refactoring implies validation
- Conversion tool(s) support is available
- Effort for adapting SA, codecs, ... needed

NOKIA

CONNECTING PEOPLE

Thank you for both your participation and for your attention