TTCN design considerations for LTE protocol tests

3GPP RAN5 #39

05 – 09 May 2008, Kansas City / USA

75 Years of Driving Innovation

ROHDE&SCHWARZ

TTCN Design Considerations for LTE Tests

The following areas should be considered during TTCN design for LTE tests:

- **Modularization**

  to ease parallel work, ATS maintenance and decrease compilation time during test case development

  à efficient debugging of early LTE TCs (09-2008 … 12-2009)

- **Configuration management**

  to save execution time for regression tests by reusing existing configuration of the previous scenario

  à test optimization in a more mature LTE phase (2010 …)

- **Combined procedures**

  to increase test coverage by combining several test scenarios from the conformance suite in one test case

  à Combined Procedures for End-to-End Testing (see R5-081302)

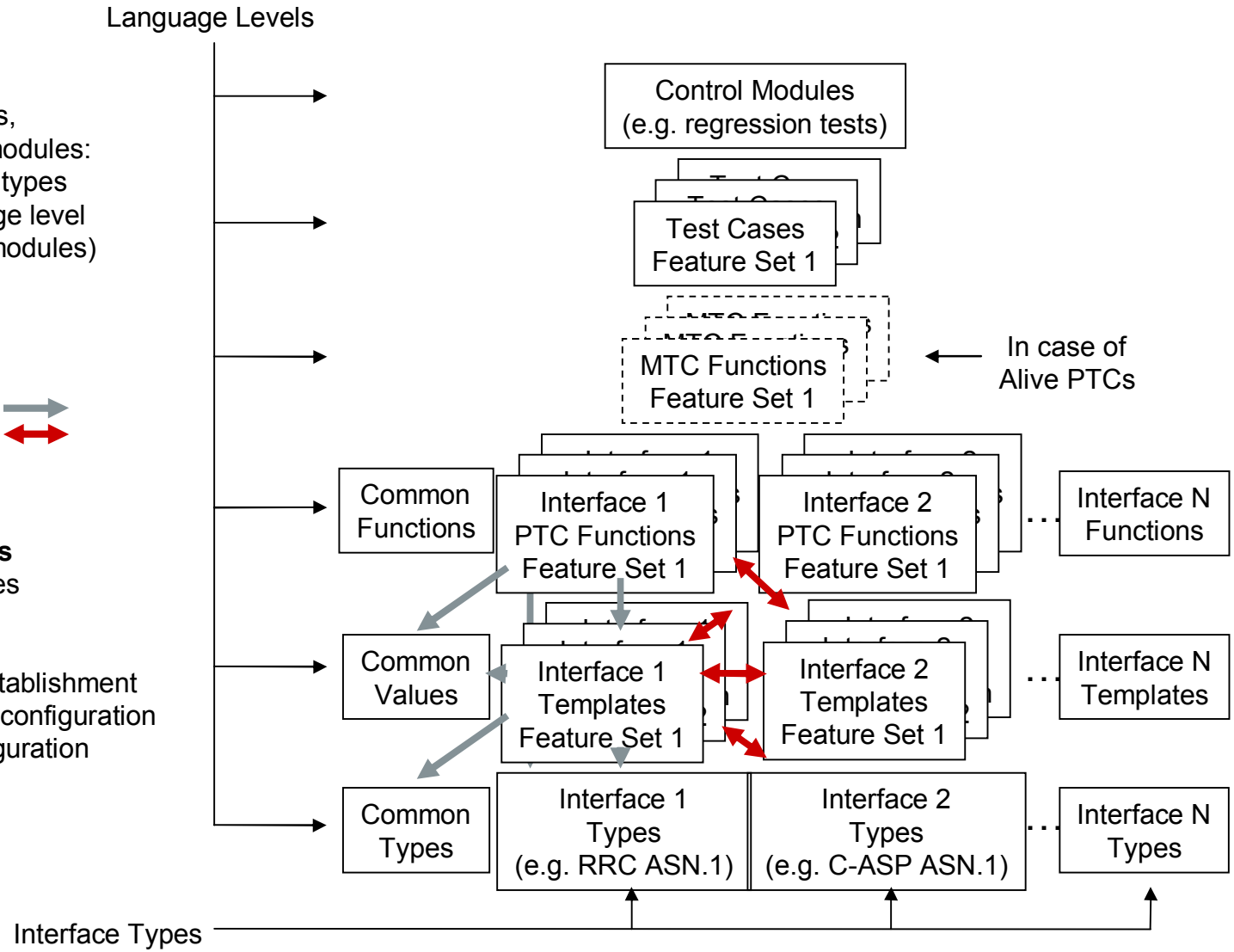
Modularization

Import rules
To avoid import circles,
no imports between modules:
- of different interface types
- on the same language level

(except for common modules)

Import examples
Allowed imports:
Not allowed imports:

Feature set examples
- Idle Mode Procedures
- L2 MAC
- L2 RLC
- RRC Connection Establishment
- RRC Connection Reconfiguration
- Measurement Configuration



Configuration Management

Example for a simple solution to support configuration management (… no extra MMC-TF160 maintenance costs)

```
// A PIXIT parameter to control whether initial configuration is needed in the Test Case
modulepar boolean mp_ConfigRequired; // Default value is “true” for conformance testing
…
// Only one additional If statement is required when setting the initial configuration in the Test Case Body
If (mp_ConfigRequired) {
   f_PerformConfiguration();
}
…
// Grouping test cases with the same configuration together into one control part
control {
   var verdicttype v_Verdict; // stores verdicts for failure handling
   v_Verdict := execute(TC_SetInitialConfigurationOnly()); // Extra test case to set initial configuration
   if (v_Verdict == pass) {v_Verdict := execute(TC_1());}
   if (v_Verdict != pass) {v_Verdict := execute(TC_SetInitialConfigurationOnly());}
   …
   if (v_Verdict == pass) { v_Verdict := execute(TC_n());}
} // The PIXIT value is set to “false” when executing such a control part
// In that way, several control parts can be specified as building blocks for regression testing
```

Combined Procedures

```
// Conformance test case X – e.g. Compressed mode
testcase TC_X() runs on MTC_CT system System_CT {
   f_Preamble();
   f_Body_X();
   f_Postamble();
}
```

```
// Conformance test case Y – e.g. Inter-RAT handover
testcase TC_Y() runs on MTC_CT system System_CT {
   f_Preamble();
   f_Body_Y();
   f_Postamble();
}
```

```
// Test case for combined procedure
testcase TC_XY() runs on MTC_CT system System_CT {
   f_Preamble();
   f_Body_X();
   f_Body_Y();
   f_Postamble();
}
```

```
function f_Preamble() runs on MTC_CT {
// Creating alive PTCs
   PTC_A.create alive;
   PTC_B.create alive;
   …
}
```

```
function f_Body() runs on MTC_CT {
// Starting the PTCs
   PTC_A.start(f_PTC_A());
   PTC_B.start(f_PTC_B());
   …
   all component.done;
}
```

```
function f_Postamble() runs on MTC_CT {
// cleanup
   …
   all component.kill;
}
```