

Development and Validation of TISpan Interworking Tests using TTCN-3

**Michael Gläser¹, Sebastian Müller²,
Axel Rennoch³, Peter Schmitting²**

1: Probus GmbH (D), 2: FSCOM, (F), 3: Fraunhofer FOKUS (D)

Contents

1. Introduction
 - ETSI STF 297/301 (IETF SIP <-> ITU-T ISUP)
2. ATS development
 - Preparations: Conventions & Configuration
 - Reuse of SIP Conformance test code
 - Design of ISUP test sequences and templates
3. Test Execution
 - Test Adapter & Tools
 - Validation & Test Campaigns
4. Fault categories
5. Project management
6. Conclusions and Outlook

Motivation

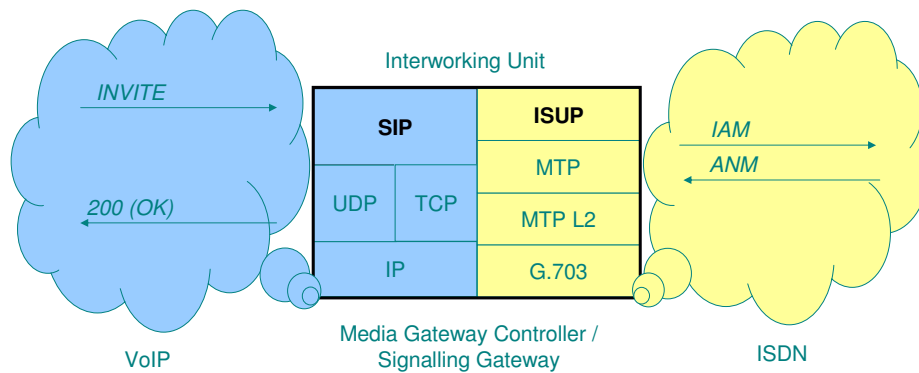
- TTCN-3 programs are
 - like other software products
 - subject of a development process (incl. validation)

- knowledge of the test language notations is not sufficient

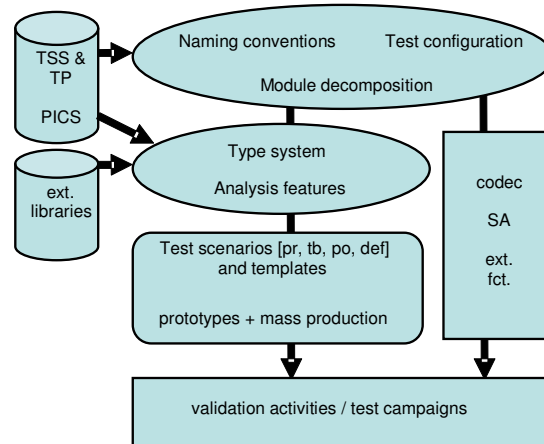
- experiences with larger projects need to be collected & exchanged

SUT protocol stack

- Signalling Gateway between “old” ISDN telephony and IP-based Networks
- SIP over UDP/TCP
- ISDN user part over SS7



TTCN-3 test development process

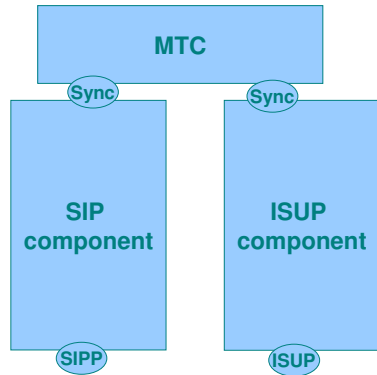


ATS requirements (selection)

1. Top-down design.
2. A uniquely defined testing architecture and test method.
3. Uniform TTCN-3 style and naming conventions.
4. TTCN-3 is human-readability.
5. TTCN-3 specification is compilable and maintainable.
6. Test cases shall be designed to be easily adaptable, upwards compatible with future releases of the base protocols.
7. The test declarations, data structures and data values shall be largely reusable.
8. Modularity and modular working method.
9. Minimizing the requirements on the implementation of test adapter (e.g. external functions).
10. Giving enough design freedom to the test equipment manufacturers.



ATS component ports & configuration



most important definitions

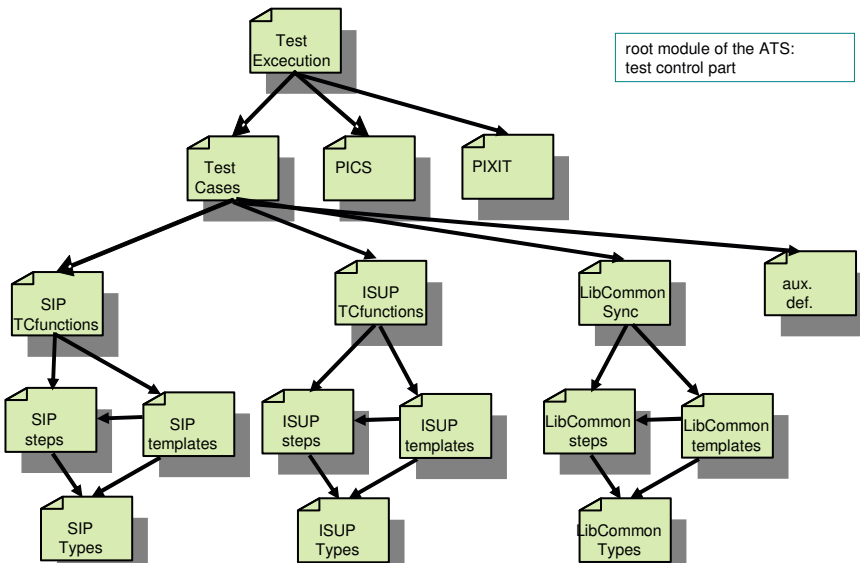
- identification of ports and types
- use of parallel test components?

basic principles:

- independent behaviour
- reuse of SIP definitions
- reuse of IPv6 synchr. lib
- synchronization via MTC
- message-based ports



TTCN-3 modules imports - overview





Naming conventions

- no rules for identifiers defined within TTCN-3 standards
- existing rules for the SIP part
- existing rules from the IPv6 synchronization lib
- no renaming for available definitions, to keep existing knowledge/adapters
- all new definitions will use the prefixes approach used in the IPv6 ATS
- in praxis prefixes and suffixes are often used, e.g.:
 - *f_mystep, f_Sip_TC_101_001*
 - *v_myvariable, c_myconst*
 - *m_mytemplate, mw_mytemplate*
 - ...
- follow your own rules suitable for your project, depending on existing standards etc.



LibCommon Sync

- source STF 276 (IPv6 project)
- TTCN-3 framework
- set of TTCN-3 functions to e.g.
 - start and control parallel components,
 - exchange synchronization signals between components
- set of charstring constants for synchronization points: „preambleDone“, „sync1“, „sync2“, ... , „testbodyDone“
- predefined timers e.g. to avoid deadlocks at synchronization points



SIP related part

- reuse / benefit of STF270 work (snapshot Sep. 2005)
- initial idea: import of existing definitions
- problem:
 - some definitions need to be extended:
SIP component, data structures (sets, records etc.)
- problem: ETSI SIP ATS has been changed end of 2005
- insufficient SDP type definition (one „charstring“ only):
needs to be analysed, composed/modified



ISUP related part

- no preconditions
- simple design:
a pair of functions for sending/receiving each message type:
f_send_ACM, f_awaiting_ACM
- systematic approach for test data
 - generation of stereotype template definitions for all messages and parameters using PIXIT entries for outgoing and incoming values
 - manual refinements
- use of microsoft word macros for automatic production of TTCN-3 code for templates and test behaviour skeletons



TTCN-3 behaviour generation

IN

TP101009		SIP	SUT	ISUP
	INVITE	>	>	IAM
	180 Ringing	>	>	ACM
	200 OK INVITE	>	>	ANM
		Conversation	>	
	BYE	>	>	REL
	200 OK BYE	>	>	RLC

OUT

```
f_awaiting_IAM(mw_IAM_USI(mw_USI_Std(mw_USIContents_Std(?,
c_USI_userInfoL1_ALaw)) ifpresent));

/*Send default ACM message*/
f_send_ACM(m_ACM_MSG_Def, true);

/*Send default ANM message*/
f_send_ANM(m_ANM_MSG_Def, true);

// Check Conversation
f_IsupBicc_CheckConversation();

/*Await default REL message*/
f_awaiting_REL(mw_REL_MSG_Def);

/*Send default RLC message*/
f_send_RLC(m_RLC_MSG_Def, true);
```



Test case sample

```
testcase TC_101_001 (inout CSeq loc_CSeq_s)
runs on ServerSyncComp
system TestAdapter {
  //Variables
  var SipComponent v_sipComponent ;
  var IsupBiccComponent v_isupBiccComponent ;
  f_cf00Up(v_sipComponent, v_isupBiccComponent);

  //Start
  f_IncCSeq(loc_CSeq_s);
  v_sipComponent.start(f_Sip_TC_101_001(loc_CSeq_s));
  v_isupBiccComponent.start(f_IsupBicc_TC_101_001());

  // synchronize both PTCs on 3 synchronization points
  f_serverSync2ClientsAndStop({c_prDone, c_sync1, c_tbDone});

  f_cfTwoPtcDown(v_sipComponent, v_isupBiccComponent);

} // end TC_101_001
```

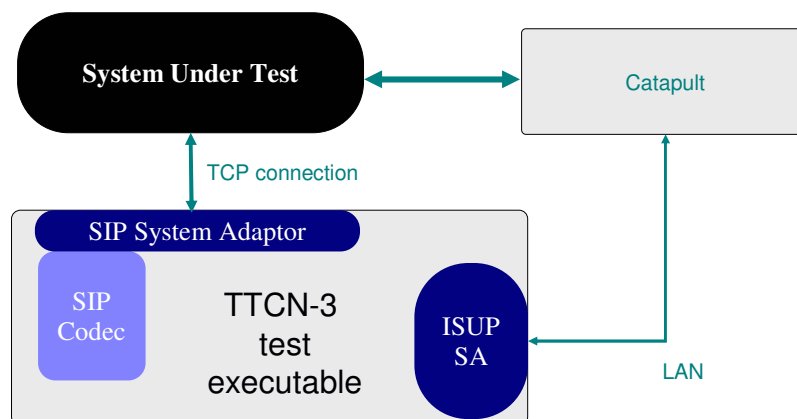


System Adapter & CoDec

- reuse of the SIP Codec
- High investigation for the production of the adapter
- SA has to satisfy assumptions made in the TTCN-3 spec: external functions (e.g. on ISUP/BICC distinction)



Test Validation Environment





Fault categories

ATS:

- unexpected messages due to incomplete test purpose description
- SIP messages retransmissions (due to test system late answer): extension of default
- incorrect component synchronization (e.g. missing in one component)
- change of TTCN-3 semantic (str2oct -> char2oct)

SA/Codec

- incomplete message types:
e.g. incorrect SIP message ignored by SUT

SUT

- missing feature implementation



TTCN-3 tools

various Test development environments:

- comfortable editor:
syntax support (highlight, formatting), navigator, outline
- use of different compilers at ETSI helps to identify problems
(e.g. differences on warning/error rating)

TestingTech Test execution management:

due to environment at validation lab

- + data and graphical sequence result analysis
- long compilation time due to big ATS size (up to 15 min!)
- test case parameter use need implementations within SA



Project management

Test suite volume: 357 test cases

ATS development:

- STF297: 98 mandays
- up to four persons in parallel

ATS validation:

- STF301: 50 mandays
- 19 lab days involving three persons



Project experiences

- Test purpose description quality is key issue
 - corrections reported to ETSI TISPAN WG6
- System Adapter is an essential working item
 - SA validation requires ATS (parallel process)
- high time pressure due to limited resources
 - some quality impact!
- never trust an ATS without validation activities
- transparent and suitable tests campaign, but high investigations



Conclusions & proposals

- complete TTCN-3 development process
- reuse of synchronization library but limited reuse of SIP conf. ATS
- abstract test suites are (frameworks) that could be extended by the customers

possible improvements:

- core language: overloading of TTCN-3 functions
- tools:
 - detect/remove (un)used definitions
 - advanced tools features to support manual development steps: structured template creation and maintenance, naming conventions etc.
 - debugger to avoid "log" statements!



Future outlook

STF 306:

- Interworking SIP ./ ISDN test suite (basic call, suppl. service partly)
- development and validation
- 4 persons, 150 man days
- May - Nov 2006

- continuation planned in 2007 (complete TP list)

improvements

- reformation of "old" SIP ATS conventions
- refined SDP type system



Martin Brandt (ETSI TISPAN WG6)
Andrej Ciglic (Iskratel Ltd.) & Team
Antony Wiles (ETSI PTCC)

Thank you !