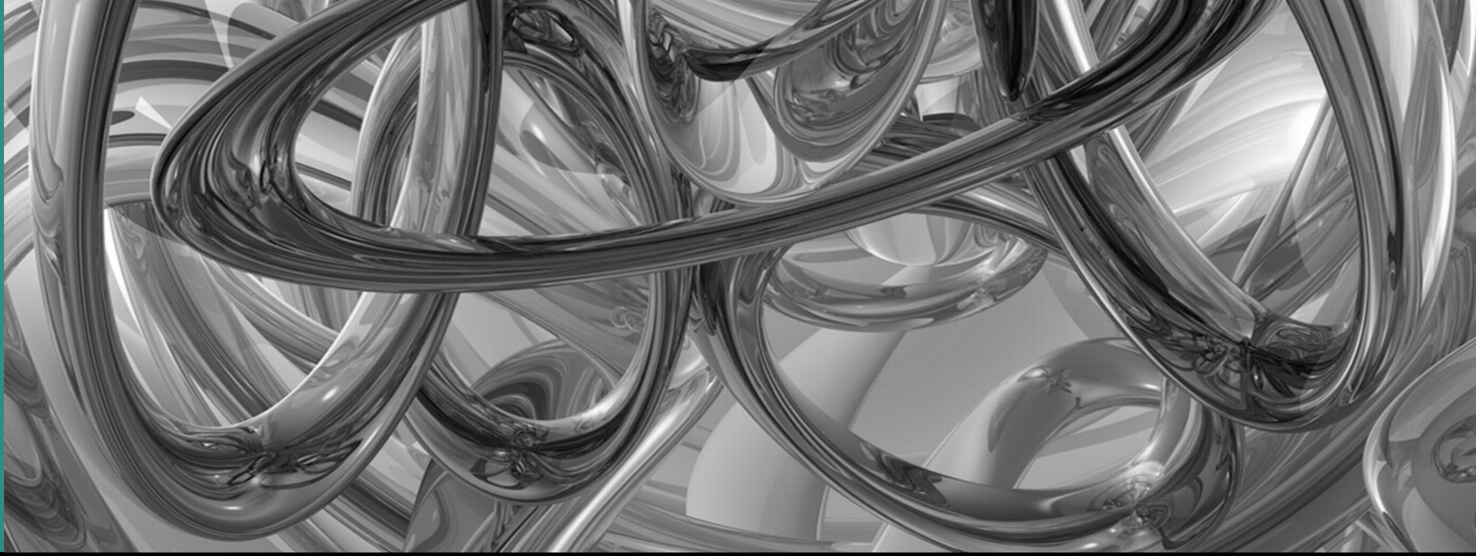


Using TTCN for Radio Conformance Test Systems

<u>Javier Poncela-González</u>, Juan Gómez-Salvador,
Carlos Valero-Roldán, Unai Fernández-Plazaola
Universidad de Málaga, Spain
Dept. Ingeniería de Comunicaciones

{javier, unai}@ic.uma.es

Characteristics of the Environment

- **Development of Test Systems for Wireless Devices**
- **Integration of widely varied technical areas**
  - Protocol
  - Hardware
  - Signal processing
  - RF
  - Reports
- **Human team > 10 people**
  - Skills/Knowledge in different areas
- **Low number of sales**
  - High cost of equipment
  - A reduction on the development cost provides an important competitive advantage

Contents

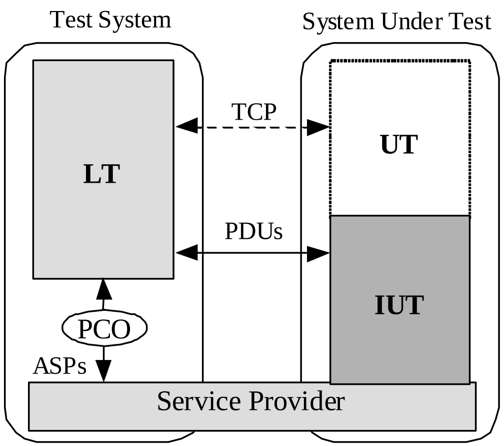
- **Introduction**
  - Radio Conformance Testing
- **Test Systems Architecture**
  - Design Principles
  - Protocol Test Systems
- **Adaptation for Radio Test Systems**
  - Architecture
  - Modules
    - Abstract Test Suite
    - Lower Subsystem
- **UMTS Radio Tests**
  - Example of Implementation
- **Ongoing Work**
  - Interconnection Matrix
- **Conclusions**

Contents

- **Introduction**
  - Radio Conformance Testing
- **Test Systems Architecture**
  - Design Principles
  - Protocol Test Systems
- **Adaptation for Radio Test Systems**
  - Architecture
  - Modules
    - Abstract Test Suite
    - Lower Subsystem
- **UMTS Radio Tests**
  - Example of Implementation
- **Ongoing Work**
  - Interconnection Matrix
- **Conclusions**

Conformance Testing Methodology

- **Defined in the early 90's**
- **The Implementation Under Test (IUT) is seen as a black-box**
  - The IUT can be stand-alone or embedded in a System Under Test (SUT)
- **Test Methods:**
  - Local, distributed, coordinated, remote
    - Type depends on:
      - Location of Upper Tester and
      - If a PCO is used above the IUT
  - Embedded / Non-Embedded:
    - Upper interface of the IUT accessible or not
- **Types of tests:**
  - Basic interconnection
  - Capability
  - Behaviour
- **Test verdicts:**
  - pass (success), fail (failure), inconc (inconclusive)
  - error



Remote Test Method

Radio Conformance Tests

- **Verify that the transmission and reception radio characteristics meet the specifications**
  - Transmission power
  - Modulation
  - Synchronization
  - Out-of-band emissions
- **Documents**
  - Test specifications:
    - Test Purposes (TSS&TP)
    - ~~Test Suites (ATS)~~
  - Manufacturer:
    - Equipment characteristics (ICS, IXIT)
  - Laboratory:
    - Certificates (CTR)
- **Test procedures**
  - In natural language
  - Ambiguity, coding errors, validation efforts, …

5.9.4.2 Procedure

1) Set and send continuously Up power control commands to the UE u level.

2) Measure the power of the transmitted signal with a measurement filt Measurements with an offset from the carrier centre frequency betw 30 kHz measurement filter. Measurements with an offset from the ce 12 MHz shall use 1 MHz measurement bandwidth and the result may kHz or narrower filter measurements. The characteristic of the filter spectrum analyzer filter). The centre frequency of the filter shall be s Table 5.9.2. The measured power shall be recorded for each step.

3) Measure the wanted output power according to Annex B.

4) Calculate the ratio of the power 2) with respect to 3) in dBc.

5.9.5 Test requirements

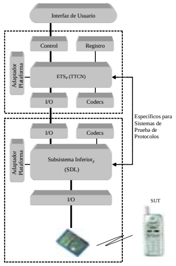
The result of 5.9.4.2 step 4) shall fulfil the requirements of Table 5.9.2.

Table 5.9.2: Spectrum Emission Mask

Frequency offset from carrier Δf	Minimum requirement
2.5 - 3.5 MHz	−33.5 − 15*(Δf − 2.5) dBc
3.5 - 7.5 MHz	−33.5 − 1*(Δf − 3.5) dBc
7.5 - 8.5 MHz	−37.5 − 10*(Δf − 7.5) dBc
8.5 - 12.5 MHz	−47.5 dBc
* The first and last measurement position with a 30 kHz filter is 2	
** The first and last measurement position with a 1 MHz filter is 4 rule, the resolution bandwidth of the measuring equipment shou bandwidth. To improve measurement accuracy, sensitivity and effic can be different from the measurement bandwidth. When the resol the measurement bandwidth, the result should be integrated over	

Contents

- Introduction
  - Radio Conformance Testing
- Test Systems Architecture
  - Design Principles
  - Protocol Test Systems
- Adaptation for Radio Test Systems
  - Architecture
  - Modules
    - Abstract Test Suite
    - Lower Subsystem
- UMTS Radio Tests
  - Example of Implementation
- Ongoing Work
  - Interconnection Matrix
- Conclusions



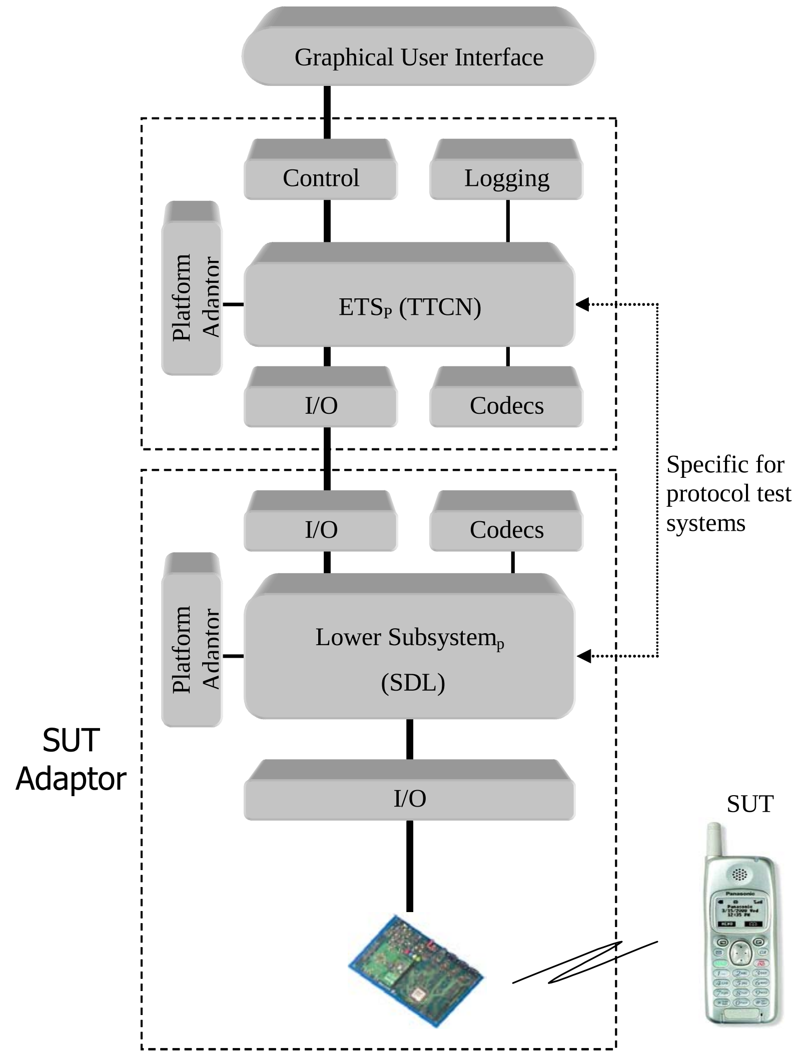
Design Principles

- **System architecture**
  - Same architecture for all types of tests
    - Use the same platform for protocol and radio tests
  - Modularity
    - Reuse of the test platform
- **Languages**
  - Use of languages standardized by ITU
  - Use the same notation to model protocol and radio test cases
    - Some manufacturers use HP-VEE/CVI/Labview, however, these are instrumentation control languages, not testing languages

Requirements Capture	**Specification and Implementation**
MSC URN	SDL eODL CHILL UML
Testing	**Data Types**
TTCN	ASN.1

Architecture of Protocol Test Systems

- **User Interface**
- **Executable Test Suite**
  - Test control
  - Platform adaptation
  - Codecs
  - Input/Output
- **SUT Adaptor**
  - Platform adaptation
  - Codecs
  - Input/Output
  - Communication with SUT

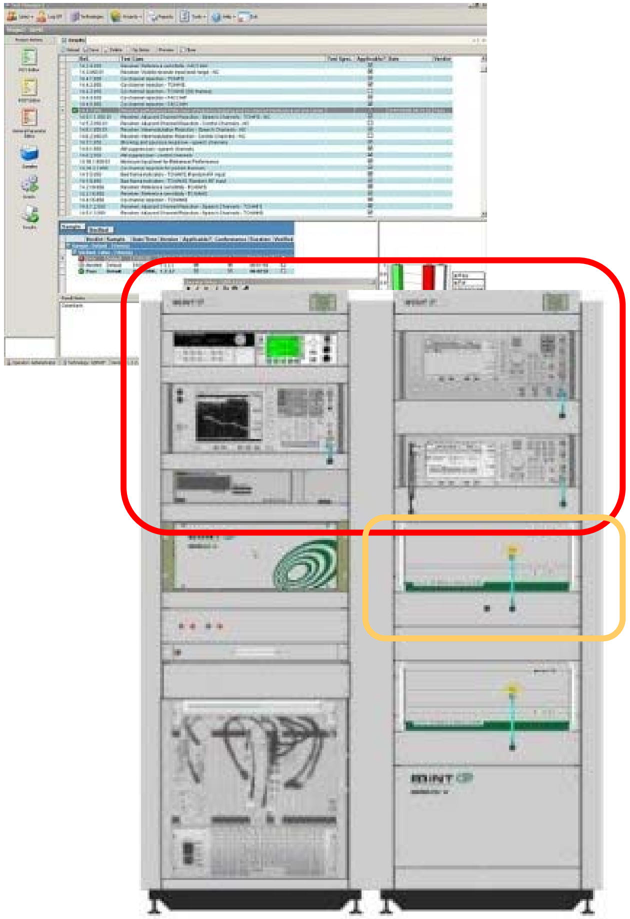


Contents

- **Introduction**
  - Radio Conformance Testing
- **Test Systems Architecture**
  - Design Principles
  - Protocol Test Systems
- **Adaptation for Radio Test Systems**
  - Architecture
  - Modules
    - Abstract Test Suite
    - Lower Subsystem
- **UMTS Radio Tests**
  - Example of Implementation
- **Ongoing Work**
  - Interconnection Matrix
- **Conclusions**

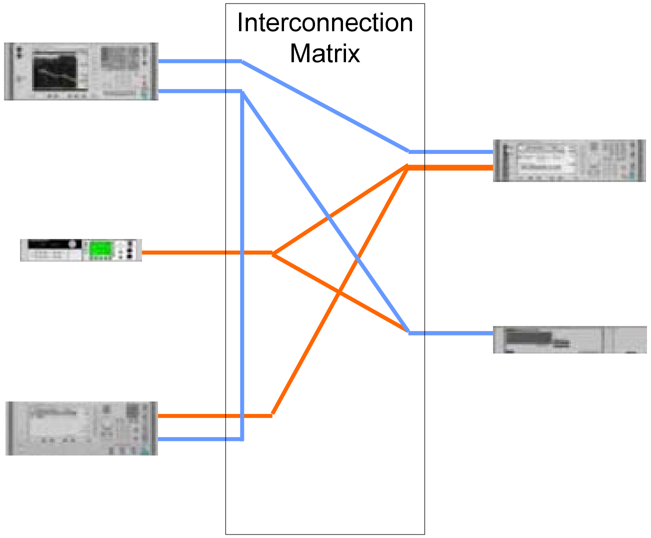
Elements of a Radio Test System

- **User interface**
  - Test control
  - Execution logging
  - Report generation
  - Graphical results
- **Test case libraries**
  - Test behaviour
  - Possibly a user API
- **Instrumentation**
  - Measurement instrumentation
  - Interconnection matrix
- **Signalling unit**
  - Set the EUT into the appropriate state to carry out the test



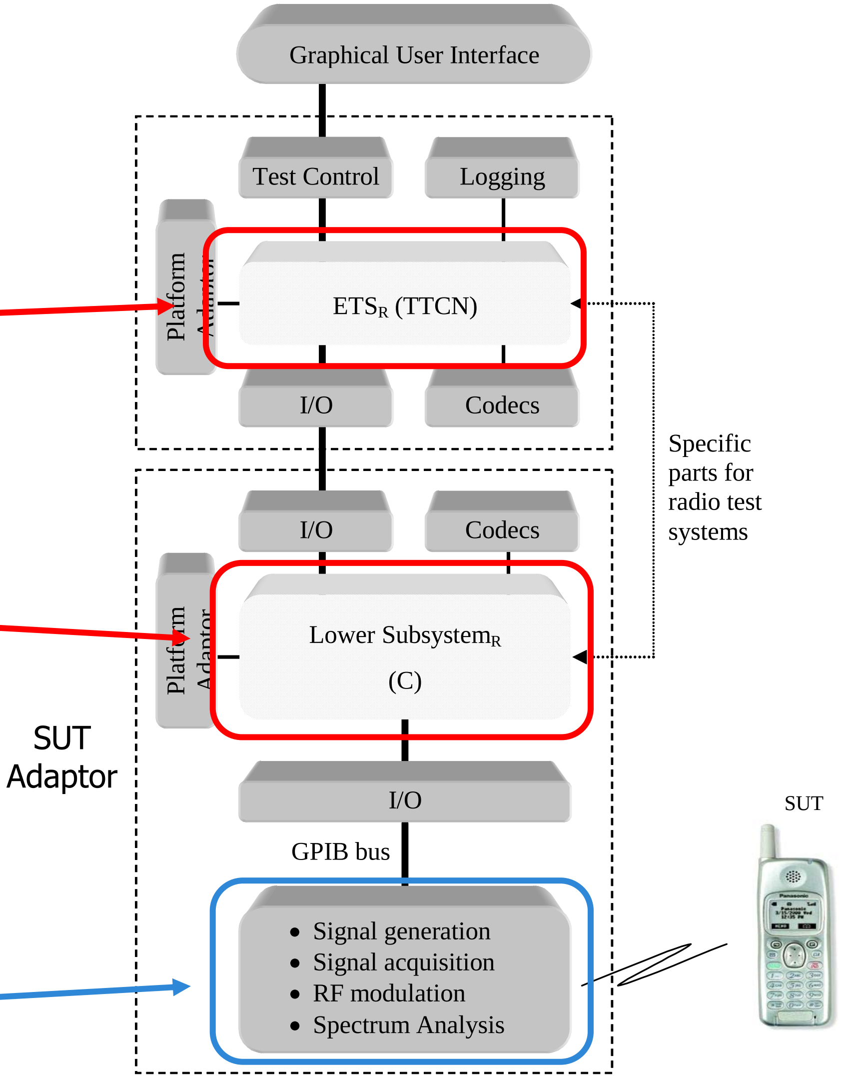
Instrumentation

- **Automated, remote control**
  - RS232, VXI, GPIB, LAN, USB, …
- **Measurement instrumentation**
  - Such as Spectrum analyzers, oscilloscopes, signal generators, power meters …
  - Represents the biggest cost in the test system
  - Trend: Implement as many functions as possible in software in order to reduce instrumentation costs
- **Interconnection matrix**
  - Connects the inputs/outputs of measurement equipment as required by the test case



Architecture for Radio Test Systems

- **User Interface:**
  - Graphical visualization of results (ej: masks, spectrum, …)
- **Executable Test Suites**
  - Only change the tests themselves, not the other elements
- **Lower Subsystem**
  - Communication with instrumentation
    - Control via GPIB
  - Generic component
- **Instrumentation**



Abstract Test Suites

- **Formalization of test sequences**
  - Notation: TTCN
    - No ambiguities
    - Same notation as for protocols
- **Test method**
  - Remote method
  - Lack of a test control protocol standardized in the radio test specs
    - ➔ Manual control of the EUT
- **Signal processing**
  - Library of functions
    - Measurement: BER (Bit Error Rate), BLER (BLock Error Rate), EVM (Error Vector Module)
    - Processing: Demodulation, bit synchronization, filters, bandwidth calculation, …
      - Can be seen as software instrumentation
    - Many functions can be made common to different systems if properly designed and parameterized (ej: GSM, Bluetooth, UMTS, …)

Interface with SUT Adaptor

- **TTCN-2**
  - Communication via Messages
  - They cover all required functionality
  - Each message represents one or more actions, depending on characteristics of the instrumentation
  - Confirmation (positive or negative) is required for all messages
    - Timeout if not received ➔ Verdict **INCONC**

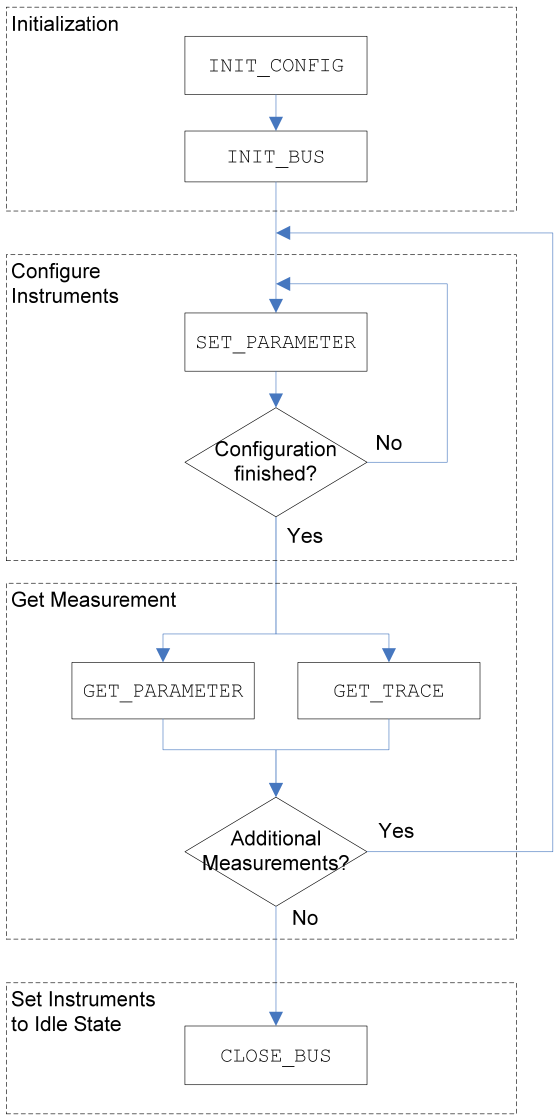
- **TTCN-3**
  - Communication via Messages and Procedures
  - Same characteristics as for TTCN-2

TTCN-2
`INIT_CONFIG`
`INIT_BUS`
`SET_PARAMETER`
`GET_PARAMETER`
`GET_TRACE`
`CLOSE_BUS`

TTCN-3
Messages
`SET_PARAMETER`
`GET_PARAMETER`
`GET_TRACE`
Procedures
`INIT_CONFIG`
`INIT_BUS`
`CLOSE_BUS`

Test Suite Interface

- **Initialization**
  - **INIT_CONFIG**: Load configuration of the instrumentation
  - **INIT_BUS**: Initialize instrumentation
- **Behaviour**
  - **SET_PARAMETER**: Configure measurements
  - **GET_PARAMETER**: Get instantaneous value of a measurement
  - **GET_TRACE**: Get a measurement with several points
- **Termination**
  - **CLOSE_BUS**: Set the instrumentation into idle state (GPIB status: local)



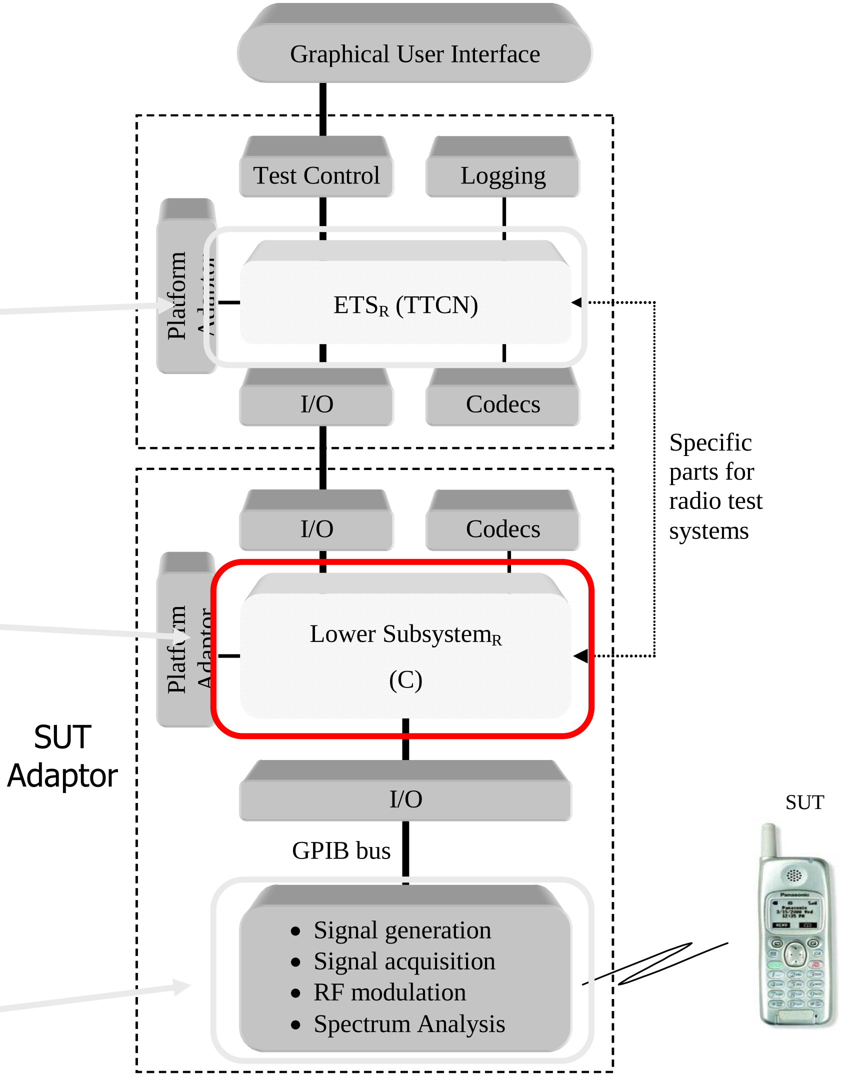
Instrumentation Commands

- Commands represent abstract actions that must be performed by or on the instrumentation

Command	*Command text*	*Command*	*Command text*
TSC_com_center	"Center"	**TSC_com_ndbdown**	"NdBDown"
TSC_com_detector	"Detector"	**TSC_com_peakpower**	"PeakPower"
TSC_com_DigStd	"DigitalStandards"	**TSC_com_reflevel**	"RefLevel"
TSC_com_gapsweep	"GapSweep"	**TSC_com_rbw**	"RBW"
TSC_com_gapsweeppretrig	"GapSweepPreTrig"	**TSC_com_reset**	"Reset"
TSC_com_get_peakpower	"PeakPower?"	**TSC_com_slope**	"Slope"
TSC_com_get_start	"Start?"	**TSC_com_span**	"Span"
TSC_com_get_stop	"Stop?"	**TSC_com_start**	"Start"
TSC_com_get_trace	"Trace?"	**TSC_com_stop**	"Stop"
TSC_com_ifbw	"IFBw"	**TSC_com_sweeptime**	"SweepTime"
TSC_com_markerX	"MarkerX"	**TSC_com_trigger**	"Trigger"
TSC_com_MeasResult	"MeasResult"	**TSC_com_triggerlevel**	"TriggerLevel"
TSC_com_mode	"Mode"	**TSC_com_vbw**	"VBW"

Architecture for Radio Test Systems

- **User Interface:**
  - Graphical visualization of results (ej: masks, spectrum, …)
- **Executable Test Suites**
  - Only change the tests themselves, not the other elements
- **Lower Subsystem**
  - Communication with instrumentation
  - Control via GPIB
  - It's still a generic component
- **Instrumentation**



Lower Subsystem

- **Generic for**
  - different radio test systems and
  - instrumentation of different manufacturers
- **Responsible for the communication between the Executable Test Suite and the Instrumentation**
  - Hides the physical characteristics and behaviour of the bus
    - Possible errors, delays, …
  - Communication Bus
    - GPIB (present in most, if not all, instrumentation)
- **Maps the messages (and procedures) of the Test Suite into GPIB commands**
  - Instrumentation of different manufacturers, proprietary commands
  - Subsets of standard GPIB commands
  - Same command may have different meanings in each equipment
  - → Solution: Configuration files
    - A change of instrumentation equipment only requires a change of configuration files

Configuration Files

- **Main**
  - Shows available equipment
  - Contains the GPIB configuration

```
# Configuration file for Test System
#id_equipo  GPIB_board  dir_GPIB  EOT_mode    file
Spec_An         0          20        2      specan.egp
Wave_Gen        0          29        2      wavgen.egp
```

- **Equipment**
  - Mapping between messages/procedures and GPIB commands for a specific equipment
    - ? : Requires answer
    - _ : Options

```
# DEVICE: Spectrum Analyzer (specan.egp)
# Generic command     Specific device Command
#         Non-query commands
Reset                 *RST
PeakPower             CALC1:MARK1:MAX
Span                  SENS1:FREQ:SPAN
Trigger               TRIG1:SEQ:SOUR
_FreeRun              IMM
_Line                 LINE
_RFPower              RFP
#         Query commands
PeakPower?            CALC1:MARK1:Y?
RefLevel?             DISP:WIND1:TRAC1:Y:SCAL:RLEV?
SweepTime?            SENS1:SWE:TIME?
Detector?             SENS1:DET1:FUNC?
_Average              AVER
_Sample               SAMP
_Rms                  RMS
ErrMsg?               SYST:ERR?
```

Interface (API)

- **The Lower Subsystem offers a generic API**
  - Can be used by any Radio Test System that uses GPIB instrumentation
- **Each message/procedure is handled by one subroutine**
  - 6 functions
    - Returned value: Error code
    - Parameter `err`: Error description
  - Written in C

```
int InitConfig (Instrument *instr, char *err)

int InitBus (Instrument *instr, char *err);

int SetParameter (Instrument *instr, char *id, char
     *com_gen, char *par_gen, char *err);

int GetParameter (Instrument *instr, char *id, char
     *com_gen, char *par_gen, char  *valor_dev_str,
     int l_valor_dev_str, char *err);

int GetTrace (Instrument *instr,char *id, char *tx,
     char *ty, char *err)

int CloseBus (Instrument *instr, char *err);
```

```
typedef struct
{
  char  id_instr[L_ID_INSTR];
  int   gpib_board;
  short dir_gpib;
  int   eot_mode;
  char  arch_instr[L_ARCH];
  Commands *coms;
  void  *sig;
} Instrument;
```

Contents

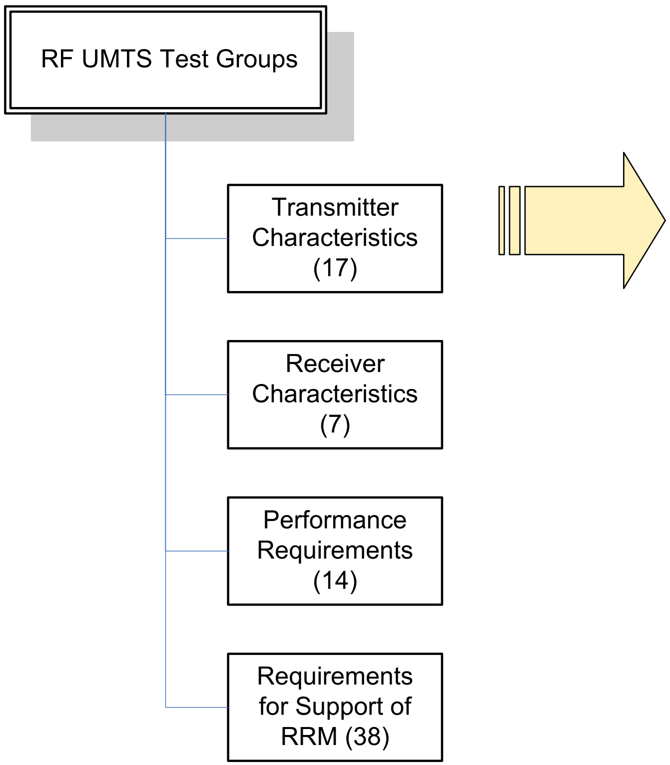
Radio Test System for UMTS

- **UMTS is a member of the 3G mobile communication systems family**
- **Multiple simultaneous connections of variable rate**
    - Up to 2 Mbit/s
  - Worldwide roaming, security, negotiated QoS, …
- **Access radio technology**
  - Direct Sequence CDMA (DS-CDMA ≡ WCDMA)
    - Bands of 5 MHz
    - Robustness against interferences
    - Spectral efficiency
    - Frequency reuse
  - Two modes
    - FDD: one uplink carrier and one downlink carrier ($\Delta = 5$ MHz)
    - TDD: uplink slots and downlink slots

Maximum Bitrate (kbps)	Mobile Maximum Speed (km/h)	Cell Type
144	500	Macrocell
384	120	Macrocell Microcell
2048	10	Microcell Picocell

Radio Tests for UMTS

- **3GPP TS 34.121**
- **71 radio tests classified in 4 groups**

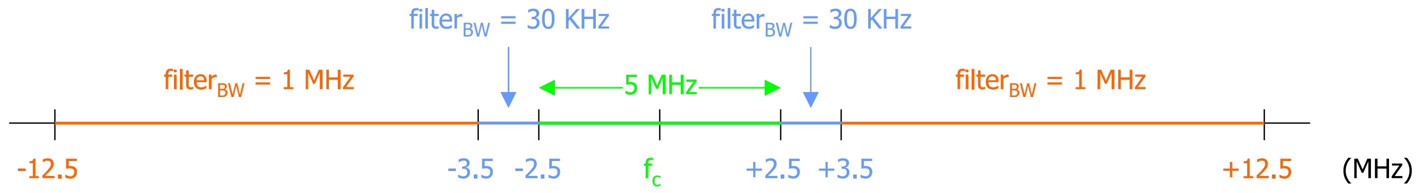


Transmission Tests

Maximum Output Power	**Change of TFC**
Frequency Error	**Power Setting in UL Compressed Mode**
Output Power Dynamics in the UL	**Occupied Bandwidth (OBW)**
Open Loop Power Control in the UL	**Spectrum Emission Mask**
Inner Loop Power Control in the UL	**Adjacent Channel Leakage Power Ratio (ACLR)**
Minimum Output Power	**Spurious Emissions**
Out-of-Synchronization Handling of Output Power	**Transmit Intermodulation**
Transmit ON/OFF Power	***Transmit Modulation***
Transmit OFF Power	**Error Vector Magnitude**
Transmit ON/OFF Time Mask	**Peak Code Domain Error**

Implementation Example - Description

- **Test Case (FDD)** ***Spectrum Emission Mask***
  - Verify that the power of the User Equipment (UE) out of band does not exceed the set thresholds
- **Initial conditions**
  - UE in loop mode after having established a call
- **Test procedure**
  - Send control command to increase transmission power until the UE is transmitting at maximum level
  - Measure transmitted power. The central frequency of the measurement filter is moved and the power is measured at each step.
    - The filter bandwidth depends of the distance to the carrier frequency
    - Calculate relation between measured power and reference mask
  - Measurement instrumentation: Spectrum analyzer (FSIQ26)



Implementation Example - Modelling

1) **Initialize the Test System**
   - Read configuration (`INIT_CONFIG`)
   - Set the equipment in remote mode (`INIT_BUS`)
2) **Configure instrumentation with a 30 kHz filter**
   - Set filter parameters (`SET_PARAMETER`)
3) **Measure in interval f_c ±3.5 MHz (steps of 5 kHz)**
   - Read measurements (`GET_TRACE`)
4) **Configure instrumentation with a 50 kHz filter**
5) **Measure in interval f_c ±12.5 MHz (steps of 25 kHz)**
6) **Compare measurements with the reference mask**
7) **Set verdict**
8) **Go back to idle state** (`CLOSE_BUS`)

Test Case Name	TC_TRM08_SpecEmissMask
Group	TRM/TC_TRM08/
Purpose	Verify that the spectral emission mask is met for different frequency variations of the carrier, both for high and low frequencies
Configuration	
Default	Check_T_global_trm08
Comments	
Selection Ref	TCS_TRM08
Description	Verify the spectral emisión mask for low and high frequencies

Nr	Label	Behaviour Description
1		START T_global_trm08
2		+Initialize_system
3		+Inic_an_esp_trm08_ftx_low
4		+Calc_SpecEmissMask_low
5		+EUT_ftx_high
6		+Inic_an_esp_trm08_ftx_high
7		+Calc_SpecEmissMask_high
8		+Check_res_trm08

Test Behaviour

Test Step Name	Calc_SpecEmissMask_low
Group	Calc_Steps/TC_TRM08/
Objective	
Default	Check_T_global_trm08
Comments	

Nr	Label	Behaviour Description	Constraints Ref
21		(TCV_frec_stop := TSO_RESTAR(TSC_fx_low,"2500000"))	
22		GPIB!SET_PARAMETER_REQ	Set_parameter_req(TSC_id, TSC_com_stop, TCV_frec_stop)
23		+Set_parameter_err_rsp	
24		START T_wait_1_s	
25		?TIMEOUT T_wait_1_s	
26		GPIB!GET_TRACE_REQ	Get_trace_req(TSC_id, TSC_com_get_pot, TSC_par_pot)
27		+Get_trace_err_rsp	
28		(TCV_arizq := TCV_par)	

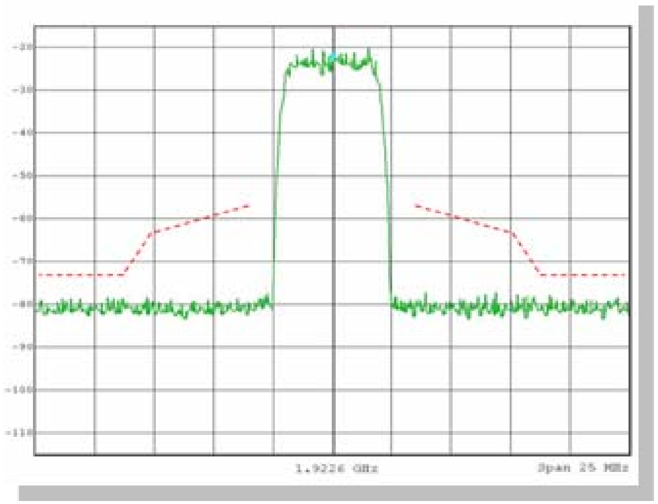
Measurement of the Spectral Emission

Implementation Example - Modelling

1) **Initialize the Test System**
   - Read configuration (`INIT_CONFIG`)
   - Set the equipment in remote mode (`INIT_BUS`)
2) **Configure instrumentation with a 30 kHz filter**
   - Set filter parameters (`SET_PARAMETER`)
3) **Measure in interval f_c ±3.5 MHz (steps of 5 kHz)**
   - Read measurements (`GET_TRACE`)
4) **Configure instrumentation with a 50 kHz filter**
5) **Measure in interval f_c ±12.5 MHz (steps of 25 kHz)**
6) **Compare measurements with the reference mask**
7) **Set verdict**
8) **Go back to idle state** (`CLOSE_BUS`)

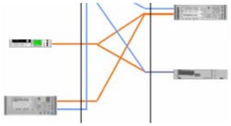
Test Case Name		TC_TRM08_SpecEmissMask
Group		TRM/TC_TRM08/
Purpose		Verify that the spectral emission mask is met for different frequency variations of the carrier, both for high and low frequencies
Configuration		
Default		Check_T_global_trm08
Comments		
Selection Ref		TCS_TRM08
Description		Verify the spectral emisión mask for low and high frequencies
Nr	**Label**	**Behaviour Description**
1		START T_global_trm08
2		+Initialize_system
3		+Inic_an_esp_trm08_ftx_low
4		+Calc_SpecEmissMask_low
5		+EUT_ftx_high
6		+Inic_an_esp_trm08_ftx_high
7		+Calc_SpecEmissMask_high
8		+Check_res_trm08

Test Behaviour



Contents

- Introduction
  - Radio Conformance Testing
- Test Systems Architecture
  - Design Principles
  - Protocol Test Systems
- Adaptation for Radio Test Systems
  - Architecture
  - Modules
    - Abstract Test Suite
    - Lower Subsystem
- UMTS Radio Tests
  - Example of Implementation
- Ongoing Work
  - Interconnection Matrix
- Conclusions



Interconnection Matrix – Ongoing Work

- **Connects inputs and outputs of instrumentation equipment as the test case requires**
- **Instrumentation is decomposed in generic instrumentation functions**
  - Each instrumentation function is modelled as a virtual equipment
    - Box with input and output connections
    - Well-defined abstract behaviour
      - Examples: Spectrum analyzer, Signal generator, Power meter, …
  - Implementation
    - Each instrumentation function can implemented in one or several real equipment
    - One or several virtual equipment maybe realized in one real equipment
- **Issues**
  - ➔How to make this decomposition so that instrumentation functions are really generic (and usable for different technologies under test)
  - ➔How to define inputs and outputs
  - ➔How to create the configuration
    - For example, with a new message: Matrix_Connect (virt02:A09, matrix:B05)

Conclusion

- **Common architecture for protocol and radio test systems**
  - Generic modules
  - Components are shared and, thus, development costs are reduced
    - Sharing includes also languages and the development process
- **For radio tests, a specific test notation, TTCN, is used instead of instrumentation control or general programming languages**
  - Improvement of radio test specifications
  - Ambiguity is eliminated from test procedure descriptions
  - One step is removed from the test system development process
    - Simplifies the validation of test systems
- **Integration of instrumentation from different manufacturers**
  - Communication via bus GPIB
  - Adaptation to the interface provided by each instrument
- **Demonstrated in radio test systems for Bluetooth and UMTS**

Thanks for your Attention !