Wireless Pers Commun (2010) 53:409–429
DOI 10.1007/s11277-010-9954-2

# Advancing Radio Testing Methodology via Formal Notations

**J. Poncela · J. T. Entrambasaguas · M. C. Aguayo-Torres**

Published online: 16 March 2010


**Abstract** Mandatory verification of the correct operation of equipment in public environments will increase the complexity of test equipment due to the high number of new vendors in the foreseen scenarios. Protocol and radio conformance tests are usually thought to be two separated fields, which makes that the test system design process is different for both cases. However, the only lack of radio test cases is the formalization of tests using a formal test notation. In this paper, it is shown how the design process can be merged for both protocol and radio test systems by modelling radio tests in TTCN. In this case, the same architecture and development tools can be used in the whole testing process, and a reduction of costs and time can be achieved. Abstraction of the instrumentation equipment and standardization of a control interface are required.

**Keywords** Conformance testing · Radio tests · Protocol tests · TTCN

## 1 Introduction

The new mobile devices that are foreseen in the mid- and long-term will allow us to access our social, information and work environments from any place. Society has assumed the paradigm of total connectivity: any time, any place, any service. For some time now, the design and production centres have been shifting from the traditional niches towards emerging countries, and this will provoke that a myriad of vendors will compete for, and a big part of them achieve, the user's approval. Vendors will tend to offer more varied functionalities in a market that is mainly defined by novelty, visual attractiveness and a cost-performance rate continuously decreasing.

In this scenario, the need to verify the correct operation of equipment in public environments is greater than ever. When a user acquires a communication equipment it is signing a contract with the vendor in which, in exchange for monetary compensation, it receives a

J. Poncela (✉) · J. T. Entrambasaguas · M. C. Aguayo-Torres
Department of Ingeniería de Comunicaciones, University of Malaga, 29071 Malaga, Spain
e-mail: javier@ic.uma.es

device that allows accessing the incorporeal data world, with the implicit understanding that the equipment not only properly performs its function, but that it also fulfils the regulations mandated in the telecommunications area. In order to vouch for the truthfulness of these statements, the vendor must carry out different checks along the design process in order to verify their equipment. Besides, selling a product requires, according to current legislation, an official certification. It guarantees that the equipment can correctly operate without disturbing the operation of other systems, and thus requires the equipment to undergo several static and dynamic tests.

Commercial communications equipment are defined as open systems where no particular implementation of their functionality is imposed, but where the external interfaces and the behaviour they must provide are specified. It is not until the end of the eighties when a testing methodology for open systems universally accepted is defined: the OSI Conformance Testing Methodology. This methodology formalizes the whole test process from the test definition stage up to the provision of a legal certification, including different test architectures and the notation that must be used for modelling the tests.

The certification of equipment requires undergoing several protocol and physical layer tests. In the case of mobile communications systems a radio access is used, thus they are called radio conformance tests. Traditionally, protocol and radio conformance testing have been considered distant worlds, as engineers who work in them often have no training in both fields. While protocol tests are targeted to verifying that the sequences of messages exchanged between two entities are performed in the correct order and with the proper syntax, the goal of radio tests is to certify compliance with aspects such as radioelectric compatibility in transmission and reception.

A characteristic difference between protocol and radio tests is the fact that the latter require specific instrumentation capable of performing certain radioelectric measurements on the air interface. Examples of this instrumentation equipment are: signal generators, modulators, oscilloscopes, spectrum analyzers, etc. This equipment will typically have a fairly high cost, and, since several equipment are usually required, this means that radio Test Systems are notably more expensive than protocol Test Systems ([9,21]).

However, most of the concepts are shared between both types of tests. Radio tests are standardized using a similar methodology to protocol tests, which involves the same set of documents ([13–15]), with the only exception that the testing methodology only gets to describe the scenarios and the expected behaviour of the equipment at the level of natural language. It lacks, then, the formalization of the Abstract Test Suite in the widely accepted test notation TTCN (*Testing and Test Control Notation*).

Within the standardizing bodies, during the nineties, several attempts were made at formalizing radio tests, but they were discarded upon not being able to formally model all the characteristics that this kind of tests possesses. Nevertheless, an analysis of the available radio test specifications leads to think that, if perhaps a complete formalization is not possible, it is undoubtable that a majority of the interactions and measurements that these tests carry out can be modelled.

In this paper we describe a methodology for the design of radio Test Systems that brings this field close to the level achieved by protocol tests.[1] The basic principle lies in the assimilation between protocol and radio tests, considering the latter as sequences of messages exchanged between the Test Case and the Instrumentation equipment.

First, the characteristics of the radio Test Systems are explained. Afterwards, the principles that guide the design are presented. Next, it is described how to adapt the elements of

[1] Some of the underlying ideas, based on the work carried out here, have also been presented in [18].

**Table 1** ITU standards for the test methodology

| Standard | Description |
|---|---|
| X.290 | Offers an overview of the test process and defines related concepts. |
| X.291 | States a set of requirements for the specification of Abstract Test Suites, providing some guidelines about their elaboration and the selection of the most appropriate Test Method. |
| X.292 | Defines the TTCN test notation (version 2). |
| X.293 | Issues about the test execution process. |
| X.294 | Describes the relations between the test laboratory and its customer and indicates the set of documents to use. |
| X.295 | Extends the test methodology to protocol profiles, although the ideas appear in a distributed manner among several of the other standards. |
| X.296 | Specifies the requirements with regard to the design and use of the conformance proformas. |

the protocol Test Systems architecture, and it is shown how Instrumentation can be modelled abstractly; the interface between the Instrumentation and the Test Case is also described. Finally, an implementation example is shown.

## 2 OSI Conformance Testing Methodology

The work for the definition of a test methodology started in 1984 under heading "Conformance Testing Methodology and Framework" (CTMF). At that time there was already copious knowledge and practical experience ([5,17,20,23,25]) as a result from several years of research. The outcome of this activity was standard ISO 9646 [12].

The OSI Conformance Testing Methodology tackles the issue of tests carried out by third parties. The black box philosophy underlying the OSI methodology is shown in the statement that "only the external behaviour of the system is retained for the definition of standards for real open systems" [11]. The aim of this methodology is allowing the comparability and wide acceptance of the results obtained in any test laboratory, both by vendors and customers. Although its origin lies in the OSI world (an example of its use is described in [4]), its application is not restricted to the OSI reference model, being applicable to any system or communications entity.

ITU (*International Telecommunications Union*) adopted the methodology (Table 1) focusing it on the certification of telecommunications systems. Since the beginning of the nineties, with GSM, ETSI, as standardization body at the European level, standardized, following this methodology, the certification process for equipment that were sold within its scope. Within ETSI, committee MTS (*Methods for Testing and Specification*) has been responsible for leading and promoting the test methodology ([7][2], [19]). The Conformance Testing Methodology has been successfully applied to wireless communications systems (GSM, DECT, Bluetooth, GPRS, UMTS, Hyperlan, WiMAX), as well as protocols of the TCP/IP family (IPv6, Mobile IP, SIP).

Conformance tests provide a high guarantee that the equipment will properly interoperate once deployed, without disrupting the operation of the rest of equipment, either belonging to the same communications system or not. It is said that a system conforms if it meets the requirements (static and dynamic) applicable in its communication with other systems [13]. This testing approach is particularly targeted for equipment that operate in public networks

[2] This committee has replaced the original Protocol and Testing Competence Centre (PTCC).

that require governmental authorization for their installation, such as the fixed and wireless telephony and data public networks, but it is also valid for any other scope in which there is a wish to carry out tests on equipment or communications systems, either open or proprietary.

In this methodology, a Test System is a reference equipment used to verify the behaviour of an implementation. In order to do so, it emulates the behaviour of the rest of the communications system, being additionally capable of provoking abnormal situations, and it possesses the capacity to observe and measure parameters relevant for the tests. Its complexity is proportional to that of the communications system itself, besides it contains extra functionality in order to create abnormal behaviour scenarios.

When specifying the tests, the chosen solution was defining, for each entity, a set of tests (ATC–Abstract Test Cases) called Abstract Test Suite (ATS) [13]. For executing an ATS, it must be transformed into an Executable Test Suite (ETS), an executable version of the abstract model of the tests (the ATS), appropriately adapted to the execution platform.

To model the tests, notation TTCN (*Testing and Test Control Notation*), which has been specifically designed for this purpose, is used. There are two versions of this notation: TTCN-2 [13] and TTCN-3 [6]. It provides basic constructions for sending stimuli and receiving responses, comparing patterns and generating verdicts. In the latest version, apart from the classical tabular format, it offers a graphical format similar to MSC diagrams and a textual format similar to languages such as Java.

When applying for an equipment certification, vendors are required to declare the functionality of the implementation, known as the Implementation Under Test (IUT). From these statements, test laboratories are responsible for selecting the appropriate set of Test Cases. As a result a report is generated granting or denying the certification based on the verdicts obtained in the different test cases.

An Abstract Test Method (ATM) describes how to test the Implementation Under Test with a level of abstraction independent of any test means, but detailed enough as to allow the specification of Abstract Test Cases [8]. Specifically, an Abstract Test Method defines the test configuration, the interfaces through which the Test System can observe and influence the behaviour of the IUT and mechanisms for coordinating the test execution. In an Abstract Test Method there exist the elements shown in Fig. 1; a description of them is given in standard [13].

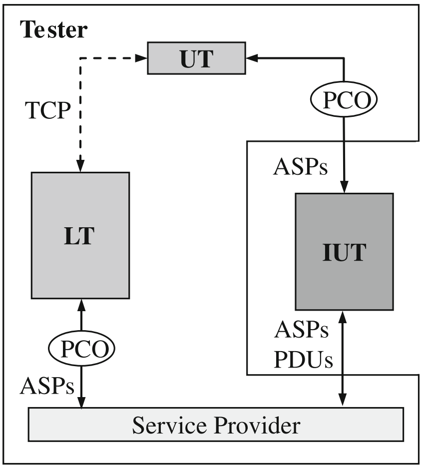


**Fig. 1** Conceptual scheme of the test architecture

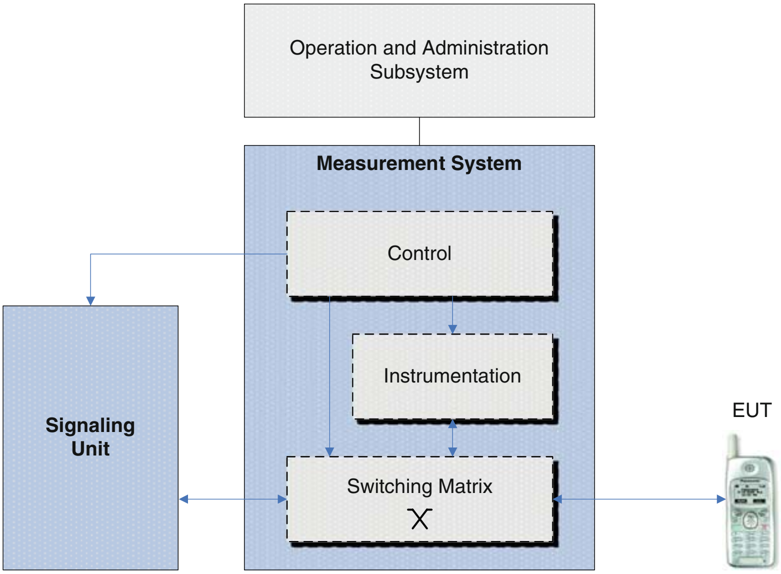


**Fig. 2** Elements of a radio test system

## 3 Overview of Radio Test Systems

Radio tests are electrical tests at the air interface, which determine whether the EUT (Equipment Under Test)[3] transmits and receives within the limits (frequency, power,…) set in the System Specifications. In order to carry out the test purpose, the Test System must take the EUT into a specific state, in which the appropriate measurement is performed. Tests are usually carried out in a conducted mode, so that the EUT is not influenced by issues external to the test scenario. Among typical tests are power measurements, bandwidth use, carrier frequency deviation, modulation quality, etc.

A radio Test System is composed of three elements (Fig. 2). The first one is the Administration and Operation Subsystem, which allows an operator to interacting with the Test System. The operator interface is equivalent to that provided in protocol Test Systems: selection and execution of test cases, test parameters edition, result visualization, test reports generation, etc. It must also include a module for graphically presenting the measurements, both in the time and frequency domains. Figure 3 shows an example of the operator interface used in a commercial system.

The second element is the Measurement System. It is responsible for the test realization and for carrying out the electrical measurements. From a conceptual point of view a radioelectric measurement (voltage, current, frequency,…) in a radio test is equivalent to interpreting the sequence of bits that form one frame in a protocol test. The Measurement System basically consists of instrumentation, remotely controlled, capable of carrying out the required measurements, and a control module, which coordinates the realization of such measurements and the creation of the radioelectric scenario.

The instrumentation equipment used includes spectral analyzers, oscilloscopes, signal generators, etc. The tests can also need additional circuitry such as filters and amplifiers. The

[3] In radio tests, the Implementation Under Test (IUT) is called Equipment Under Test (EUT).

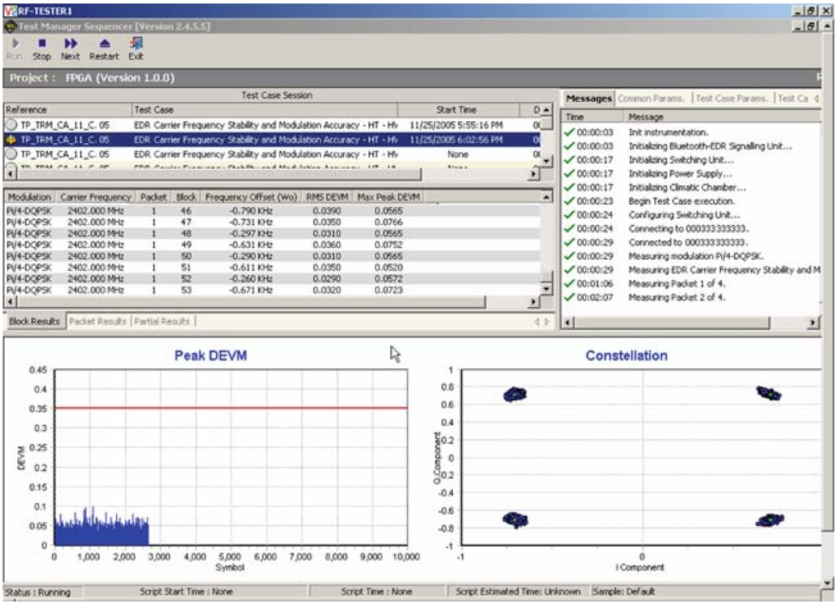


**Fig. 3** Operator interface of the commercial radio test system BITE [3]

connections among the different equipment depend on the test being executed. Both these connections as well as the connections of the required circuitry are realized via a switching matrix, which creates the path that signals must traverse for each test.[4] The control of the instrumentation can be performed manually or through remote control interfaces such as RS232, VXI (*VME eXtensions for Instrumentation*), GPIB (*General Purpose Interface Bus*) o USB (*Universal Serial Bus*).

The third element is named Signalling Unit and it is responsible for taking the EUT into the state required for each test; this element is guided by the control module of the Measurement System. In the case of protocol Test Systems, these two elements merge, as the same element that carries out the signalling is responsible for "measuring" the received PDUs, even though these measurements consist on the reception and interpretation of frames.

The Test Specifications contain only the Test Structure and Test Purpose (TSS and TP) document. That is, the Abstract Test Suite in TTCN notation is not generated. This forces each test system vendor to develop the radio tests with the languages and notations they consider more adequate. Common alternatives [2] are instrumentation control languages, for example LabWindows/CVI (*C Virtual Instrument*) or HPVEE (*Hewlett-Packard Visual Engineering Environment*), as well as general purpose languages, such as C/C++.

## 4 Design Principles

The current design approach of radio Test Systems has several weaknesses. The most obvious are:

- Lack of formality of radio tests. As each vendor implements tests from a specification in natural language, the validation of such implementations requires a higher effort than

[4] For example, attenuation or reflection paths.

in the case of protocol tests, because the chance of an error is higher as there is a longer distance between the natural language specification and the implementation than between the latter and the Abstract Test Suites.
- The design environment and languages used are different depending on whether radio tests or protocol tests are implemented, with the ensuing additional learning and integration effort both with the other development tools and the vendor manufacturing process.

To overcome these limitations we propose to use the following principles as an integral part of the design process for these Test Systems:

- Use the same architecture as for protocol Test Systems. In this way, it is possible to offer the Test System operator the same interface independently of the kind of test (protocol or radio) that it is being executed. So the elements that help integrating the various Subsystems are reused. This architecture is only valid for the Measurement System.
- Formalize radio tests using the same notation as for protocol tests. TTCN is a notation designed for the modelling of tests, because of which it provides native constructions that other languages lack. If this notation were used to build an ATS that were included in the standard Test Specifications, the quality of the Test Cases would be increased. For example, the modelling of these tests would only need to be validated once (the ATS) and not as many as different vendors there are.
- Use the same development environments. Using the same notation as for protocol tests allows reusing the same design environment, and thus the interfaces offered by code generated from the tests are the same. This makes it possible to reuse not only other components, such as the Administration and Operation Subsystem, but also the automatic interface generators used in the design of protocol Test Systems. The engineer does not need to learn a new development environment.

Overall, being able to reuse experience, tools and languages, achieves a cost reduction and an improvement of the design process. In the following sections it is described how to apply these principles.

## 5 Architecture of Test Systems

### 5.1 Protocol Test Systems

An architecture for the design of protocol Test Systems is described in [22]. This architecture has been used in commercial Test Systems for Bluetooth and UMTS. At a high abstraction level, the architecture contains the Subsystems shown in Fig. 4. The functionality of each of these Subsystems is as follows:

- Administration and Operation Subsystem: Allows the operator to controlling the test execution and managing the results. It is an element external to the Test Method.
- Test Subsystem: Includes the tests and all elements required for their execution and the generation of a verdict. It represents what it is called the Test Executable in TTCN-3.
- Lower Subsystem: Contains the protocols and elements needed for communicating the Test Subsystem with the System Under Test. It performs an equivalent function to that of the System Under Test Adaptor [6]. It encompasses the part of the service provider that it is within the Test System.

Figure 5 shows a mid-level detail of the architecture. Both the Test Subsystem and the Protocols Module are modelled in an abstract language, that is, independent of the platform. These

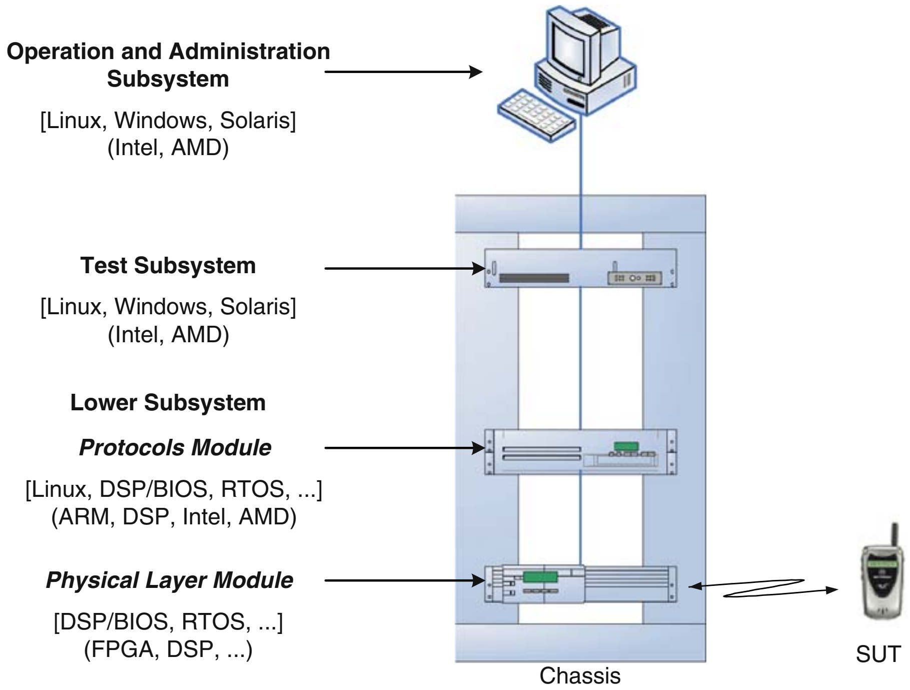


**Fig. 4** Flexible distribution of subsystems among physical platforms

models are the Abstract Test Suite (in TTCN) and the Protocol Abstract Model (in SDL [16] Both of them need a set of libraries that interpret the language semantics, at compilation time, and schedule the sequence of actions to carry out. Because of this, the structure of both subsystems includes a module for the abstract model and another module, called Engine, which corresponds to the libraries that interpret the semantics.

For their execution, it is necessary to adapt the Abstract Models and their Engines to the chosen platform. The required elements have been grouped in two modules, which have been called Adapter Module, which enclose the abstract model and allows access to services typical of an operating system (timers, input/output, memory management, concurrency,…), and Management Module, which provide high level services (codecs, control, execution logging,…).

The interface offered by the Test System to the System Under Test can be a software or an electrical interface. For mobile communications systems it is an interface at the physical layer, because of which the Lower Subsystem includes functionality from Layer 3 protocols down to the signal processing a transmitter or receiver must perform at the physical layer. Due to it, this Subsystem has been structured in two modules: Protocols Module and Physical Layer Module. The first one is closer to the telematics concept of protocol (sending and receiving messages, asynchronous, request-response), while the second one relates more to signal processing (modulation, reception and transmission filters, time synchronization, frequency domain,…). The separation of functionalities among them depends on the requirements on velocity and time synchronization, as in the Protocols Module response times can be longer. This frontier is usually found at the Medium Access Layer, placing this level's functionalities in either module according to their characteristics.

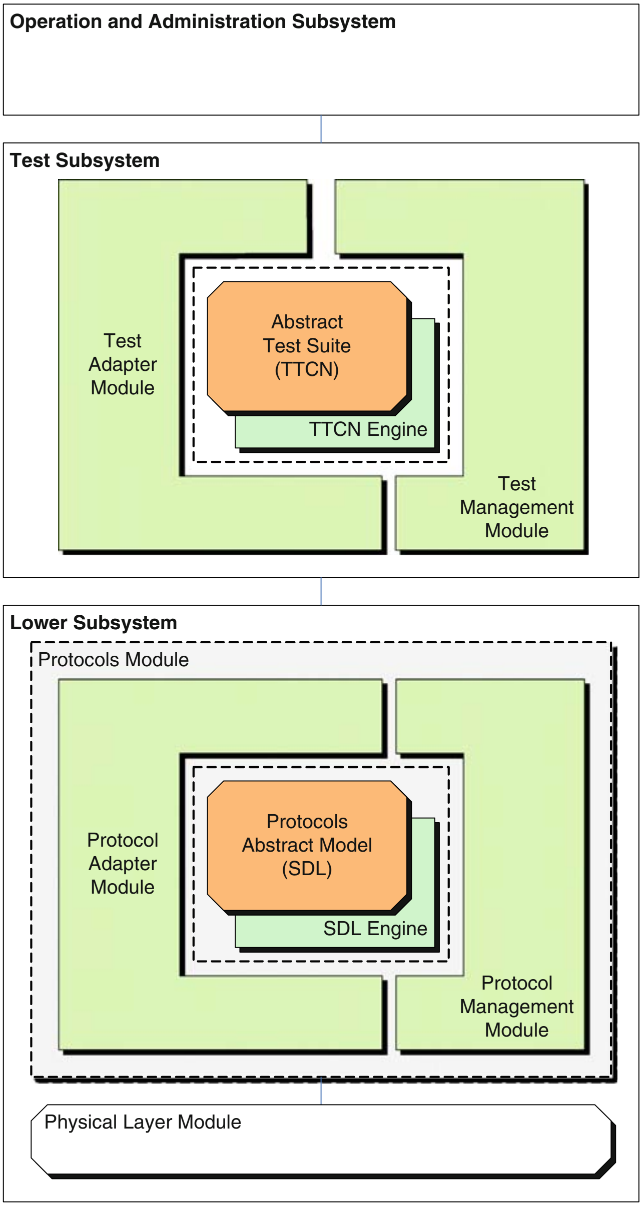


**Fig. 5** Mid-level abstract scheme for the architecture

The structure of the architecture is generic, and can be used for certifying any implementation of any technology, loading the appropriate subsystems. Each subsystem is modelled as an independent entity (two in the case of the Lower Subsystem), enabling being assigned to different platforms. For example, as the Lower Subsystem contains the Physical Layer, it will pose tighter time restrictions for processing packets and generating the responses. In this case, an option would be using a platform with a board specifically allotted to executing this subsystem. On the other hand, the Operation and Administration Subsystem does not

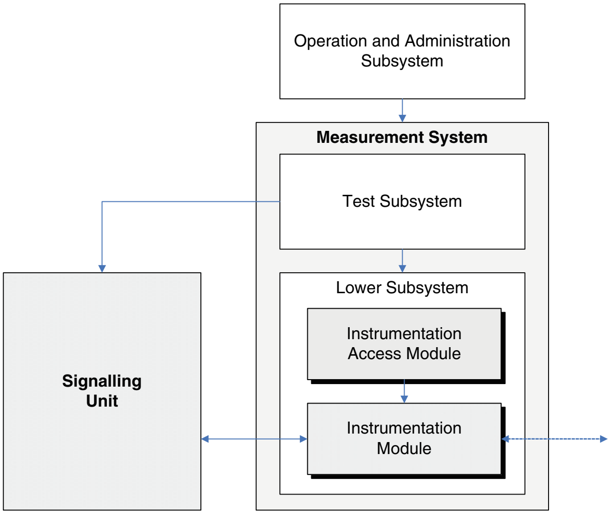


**Fig. 6** Radio test system architecture

need to physically be placed next to the Test Subsystem[5], enabling an operator to remotely control the tests execution; for example, if anechoic or temperature-controlled chambers are used. Thereby, the architecture makes available an assignment of subsystems onto physical platforms as the one shown in Fig. 4.

### 5.2 Radio Test Systems

Figure 6 shows how the described architecture can be used for radio Test Systems. The different components are described next.

As previously noted, the Operation and Administration Subsystem offers an equivalent functionality to that included in protocol Test Systems. The difference is that visualization of results is achieved through graphical representation of measurements, instead of via message sequences. This subsystem has been separated from the Measurement System in order to emphasize the fact that the Test Subsystem and the Lower Subsystem are the ones who carry out the tests, in a similar way as for protocol Test Systems.

The Measurement System is composed of the Test Subsystem and the Lower Subsystem. The purpose of both subsystems is similar to that in protocol Test Systems.

- The Test Subsystem controls de test execution and carries out the sequence of interactions (stimuli and events) needed; it controls the Signalling Unit as well as the Instrumentation. When TTCN is used for modelling the test cases, the structure of this Subsystem does not vary with regard to protocol Test Systems.
- The Lower Subsystem provides access to the EUT air interface. It contains two modules:
  - Instrumentation Access Module: it is responsible for adapting the communication between the Instrumentation and the Test Subsystem. It plays the role of the Protocols Module.
  - Instrumentation Module: contains all instrumentation equipment needed for the tests plus the switching matrix.

[5] This subsystem itself could also be distributed among several physical elements.

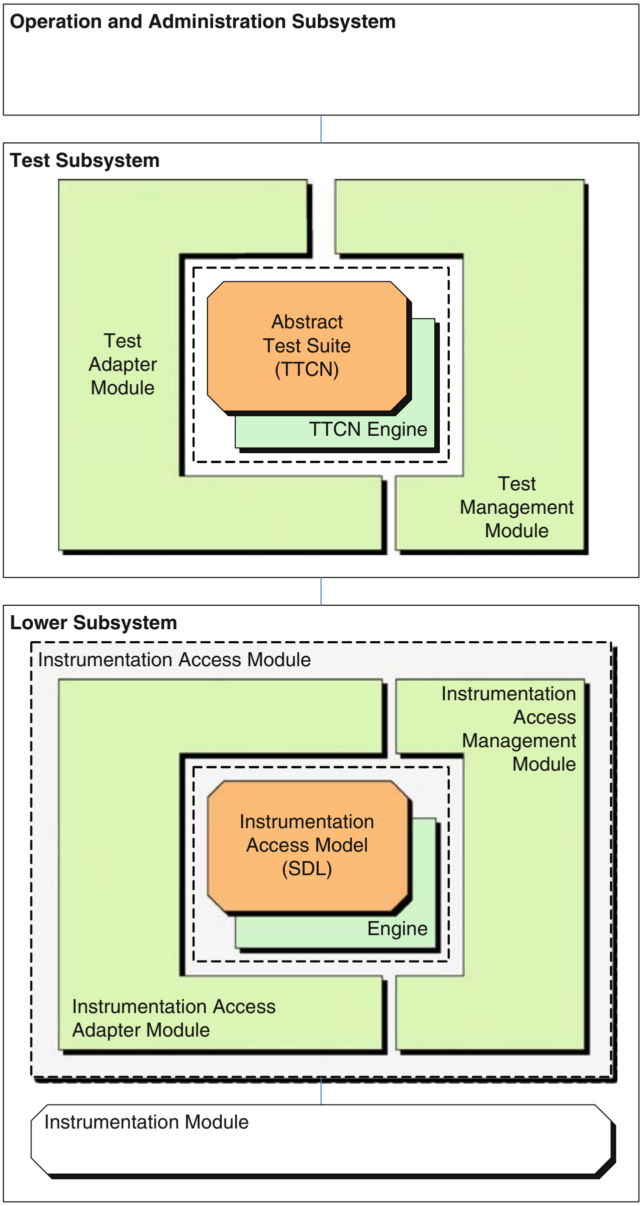


**Fig. 7** Specific components for radio test systems

If tests are modelled in TTCN, it is possible to reuse most of the components of the protocol Test Systems architecture. Some differences can be observed between both architectures (Fig. 7). The Instrumentation Access Model substitutes the Protocols Abstract Model, as now it only plays the role of intermediary between the Test Subsystem and the Instrumentation. No specific language has been proposed for its modelling. In the same way, the Engine denotes the set of libraries that implement the semantics of the modelling language. The

Instrumentation Adapter and Management Access Modules are the same ones as those used in protocol Test Systems; their name has been changed to reflect the use of Instrumentation.

The codec responsible for communicating the Test Subsystem and the Lower Subsystem needs not be implemented, as it is the same one already used in protocol tests, although it must be adapted to the set of primitives and messages used in the interface between both subsystems. However, the codec used in the communication between the Instrumentation Access Module and the Instrumentation Module must be built because it must be adapted to the logical interface offered by the instrumentation equipment; this codec can be reused among different radio Test Systems. So it happens with the Input/Output Management element at the lower frontier of the Instrumentation Access Module. This element hides the physical interface offered by the instrumentation equipment; thus, it can also be reused.

The Signalling Unit is an element that can be considered a self-contained entity. In order to automate the test executions, it must provide an interface that allows its control from the Test Subsystem. It realizes all protocol layers for the signalling between the Test Subsystem and the EUT, to set this into the appropriate state for each test.

The Signalling Unit can be built using the protocol Test Suites and their corresponding Lower Subsystem. The Test Cases already contain the test steps required to place the EUT in most, if not all, the initial states of the radio tests. Superfluous code can be removed and initialization and closing sequences extracted, thus generating a new protocol Test Subsystem. The selection and execution of the appropriate sequences would be controlled from the radio Test Subsystem through a specific interface; its implementation is relatively simple.

## 6 Radio Abstract Test Suites

A radio Test Case is composed of the same stages as protocol Test Cases (preamble, body and postamble), having each of them the same meaning. Radio Test Cases have a more linear appearance than protocol Test Cases, as once the EUT is in the expected state, the measurement is carried out and the test finishes; the possible alternative communication branches with the EUT that occur in protocol tests are hidden in the Signalling Unit.

In a radio Test Case, from a high level view, the following actions are performed (Fig. 8):

1. Creating the test scenario. It encompasses the following aspects:
   a. Configuring the connections between the instrumentation equipment, the Signalling Unit and the EUT.
   b. Controlling the Signalling Unit.
2. Configuring the measurement to carry out.
3. Performing the measurement.
4. Generating a verdict.

A measurement can be characterized by:

- Type of measurement,
- Parameters associated to the measurement type and
- Connections (electrical configuration) between elements of the Test System.

As the Test Cases are abstract, for them to faithfully reflect the Test Purpose, it is necessary to model in an abstract way the concept of measurement. This abstraction implies two things:

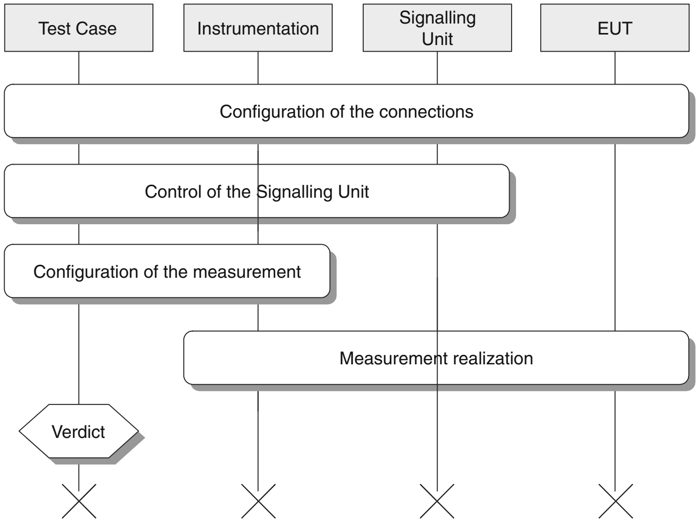


**Fig. 8** Typical sequence of actions in a radio test case.[6]

the abstract definition of the instrumentation equipment, in order to specify both the semantics of the measurement as well as its electrical configuration, and the communication with the other elements of the Test System, in order to set the parameters of the measurement to carry out.

The signalling process that a radioelectrical test may require aside from the configuration and realization of the measurement, as for example calculating the deviation of the carrier frequency, is not modelled in TTCN, as this notation is not suitable for developing mathematical algorithms. These algorithms are modelled in external functions, which are called from the Test Cases.

The next sections describe how it is possible to include and use an abstract model of the Instrumentation in the Test Cases.

### 6.1 Modelling of the Instrumentation

To abstractly model the Instrumentation, the concept of Virtual Instrument (VI) is defined. A Virtual Instrument is an element capable of performing one or several measurements, where each one can be configured by zero or more parameters, and which possesses one or more input and/or output connectors.

The formalization of the Instrumentation starts by defining the set of Virtual Instruments that can be commonly accepted in radioelectrical tests. The Test Suites will make use of the Virtual Instruments present in this list. The switching matrix can be seen as a set of electrical paths, each of which can be modelled as a Virtual Instrument.[7] In order to model a Virtual Instrument, the following characteristics must be specified:

[6] When a line crosses over one of the boxes, it means that entity does not take part in the activity.

[7] Another option is modelling the switching matrix as a special Virtual Instrument. This solution is less general, as the functionality of the Virtual Instrument would depend on the Test Suite because the matrix is different for each Test System, and could not be modelled as a generic Virtual Instrument. When building the Test System, the physical implementation of the switching matrix could not be performed automatically.

1. Electrical ports (connections) and direction (input or output).
2. Types of measurements it can perform (semantics).
3. Parameters that can be configured for each type of measurement (configuration).

It is necessary to enumerate the possible types of measurements as well as the set of configurable parameters for each Virtual Instrument, and these listings must be accepted by test designers and Test System vendors. A possible modelling of the Instrumentation is that defined by SCPI (Standard Commands for Programmable Instrumentation) [24], which specifies a set of commands for instrumentation control.

The next issue is where to define the Virtual Instruments. This definition can be made in the document where the Test Method is specified, through a full specification of each Virtual Instrument or via references to some other standard where such Virtual Instruments have been earlier specified. This definition would be made in natural language. A Virtual Instrument may be, for example, a power meter, and its configurable parameters might be the frequency band to measure, the type of measurement (average power, peak power,…), the duration of the measurement, etc. The most satisfactory option would be being able to model these Instruments within the TTCN module itself, but the current TTCN semantics does not allow for this mechanism; some comments are later given on how this alternative could be put into practice.

Before performing a test, the configuration of the measurement must be setup, that is, connecting the Instrumentation, the Signalling Unit and the EUT in the proper way. These connections can be realized directly or, if additional circuitry is needed, through the switching matrix. The connections to implement can be indicated within each Test Case. Given that their implementation requires some kind of communication with the Lower Subsystem, these requirements can be modelled as a primitive that names the ports to be connected (see Sect. 6.2).

### *6.1.1 Modelling via Test Parallel Components*

An interesting alternative for formally modelling the Virtual Instruments within the Test Suite is using Parallel Test Components (PTC). Each Virtual Instrument would be modelled as a PTC, where the ports of the Virtual Instrument would be defined as PCOs. This would allow creating, connecting and destroying them using native TTCN constructions (`create, start, stop, kill` y `connect`). Upon receiving a message from the Test Case, the PTC would forward such message towards the Virtual Instrumentation; this makes necessary defining a new PCO in the PTC for the communication with the Instrumentation, aside from its own ports.

For this alternative to be implementable, the TTCN semantics must be modified, as it does not allows that a Test Component is realized by an entity external to the Test Suite. Besides, the connection between PCOs of different Test Components must be done externally, which would require to consider this issue in the generated code, invoking external mechanisms. An option would be extending the semantics of the Parallel Test Components, for example by using a reserved keyword such as extern, that would denote that the Component is implemented outside the Test Suite. The changes needed in the code generators would be minimum. It would only be necessary to identify whether the PTC is external or not, and, in the former case, forward all operations on the PTC and all its communications to an external element, the instrumentation equipment. It is, as shown, a small change but it would allow using all the TTCN modelling power to formalizing the radio tests, as it would avoid having to define the Virtual Instruments externally to the Test Suite.

**Table 2** Communication primitives with the instrumentation access module

| Primitive | Parameters | Description |
|---|---|---|
| INIT_INSTRUMENT_REQ | (INSTRUMENT id, ListPORTS lp) | Activates an specific Virtual Instrument |
| CONNECT_REQ | (INSTRUMENT id1, PORT port1, INSTRUMENT id2, PORT port2) | Connects two different Virtual Instruments |
| SET_PARAMETER_REQ | (INSTRUMENT id, INSTPAR par, PARVAL val) | Configures the Virtual Instrument |
| GET_PARAMETER_REQ | (INSTRUMENT id, INSTCOM com, COMVAL val) | Requests that the Virtual Instrument executes the indicated command |
| STOP_INSTRUMENT_REQ | (INSTRUMENT id) | Returns the Virtual Instrument to its idle state |

### 6.2 Interface with the Instrumentation

The formalization of the radio tests includes modelling their interactions with the instrumentation equipment via an interface of abstract primitives. The information carried by these primitives are abstract messages; the Lower Subsystem is responsible for translating these messages into the appropriate command sequences for the specific instrumentation equipment included in the Test System.

This interface uses five primitives. All of them have been defined as confirmed, in order to quickly detect any errors. The communication with the Lower Subsystem is realized via one or more PCOs. Table 2 summarises the primitives defined. Figure 9 shows how these primitives are used in the logical flow of the Test Case. The parameters of each primitive indicate the command which is requested.

## 7 Example of a Radio Test Case

This section describes the implementation of the UMTS Test Case TRM/09 [1], which verifies that the EUT emission meets the limits of the spectral emission mask. To do so, the output power is measured at different frequencies and compared with the power level of reference. This Test Case needs a spectrum analyzer; model FSIQ26 has been used. The tests are carried out on an uplink reference channel type DPCH (Dedicated Physical CHannel) with a user data rate of 12.2kbps and an uplink bitrate at the physical interface of 60kbps. The call must be set up as mandated by the generic procedure and the configuration shown in Section E.3.1 in [1].

First, the implementation of the Instrumentation Access Module is presented. Then, the test procedure sequence and the use of the abstract interface primitives are shown.

### 7.1 Instrumentation Access Module

The Instrumentation Access Module is responsible for implementing the communication between the Test Subsystem and the Instrumentation (Fig. 7). As the Test Suites do not set the use of any specific instrumentation equipment, the Instrumentation Access Module must hide the characteristics which are particular of the instrumentation equipment in use. This forces it to:

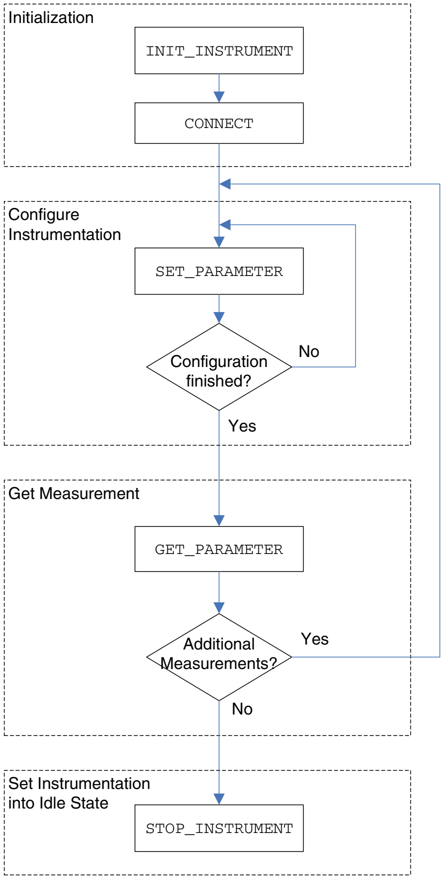


**Fig. 9** Logical flow for the interface primitives

- Manage the physical interface with the Instrumentation, adapting it to the interface expected by the Test Subsystem.
- Translate Test Subsystem commands into Instrumentation specific commands and representing them in the correct format.
- Hide differences between instrumentation equipment from different vendors.

The implementation made allows integrating instrumentation equipment from different vendors, even though they use a subset of the standard commands or even proprietary commands; it also takes into account the fact that the same command may have different meanings for each equipment. The configuration mechanism offers a great flexibility, as using different instrumentation equipment only requires slight alterations. Access to the Instrumentation has been performed through the GPIB bus [10] with SCPI commands.

The configuration of the Instrumentation Access Module is achieved via two level of information (Fig. 10). The first level shows the available equipment and information for its addressing. Each instrumentation equipment owns an additional configuration file (Fig. 11) where commands, and their parameters, used in the Test Suited are associated with the commands, and their parameters, used by the equipment. Each line defines the association for a command; those commands that require a response from the instrumentation (requests) end

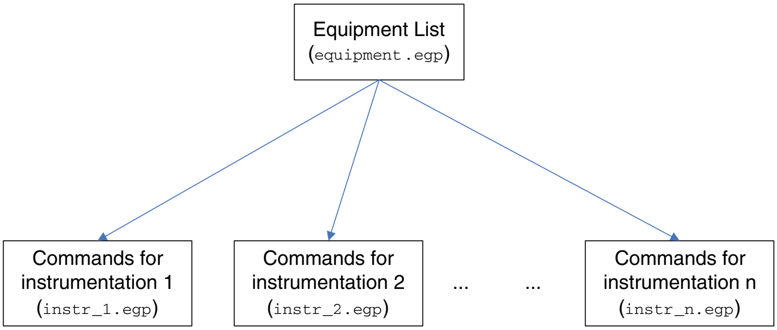


**Fig. 10** Relations among configuration information

```
# DEVICE: Spectrum Analyzer (specan.egp)

# Non-query commands
Reset                   *RST
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?
Detector?               SENS1:DET1:FUNC?
_Average                AVER
_Sample                 SAMP
_Rms                    RMS
```

**Fig. 11** Examples of command associations for spectrum analyzer

with character '?'. The parameters of each command appear in consecutive lines after the command and begin with character '_'.

### 7.2 Test Procedure

The test procedure carries out the following steps, where the use of the abstract interface primitives is shown:

1. Instrumentation equipment is initialized (`INIT_INSTRUMENT`) and measurement scenario is created (`CONNECT`).
2. Measurement filter in the spectrum analyzer is configured (`SET_PARAMETER`) with a 30 kHz bandwidth.
3. Peak power is measured (`GET_PARAMETER`).
4. Measurements are taken (`GET_PARAMETER`) of the power in band ±2,5 MHz around the carrier frequency, $f_c$, at intervals of 5 kHz. First the higher band frequencies are measured.
5. Afterwards, measurements for the lower band are taken.
6. Spectrum analyzer is configured (`SET_PARAMETER`) to use a bandwidth resolution of 50 kHz.
7. Measurements are taken (`GET_PARAMETER`) of the power in band ±12,5 MHz around the carrier frequency, $f_c$, at intervals of 5 kHz, except in the band previously used.

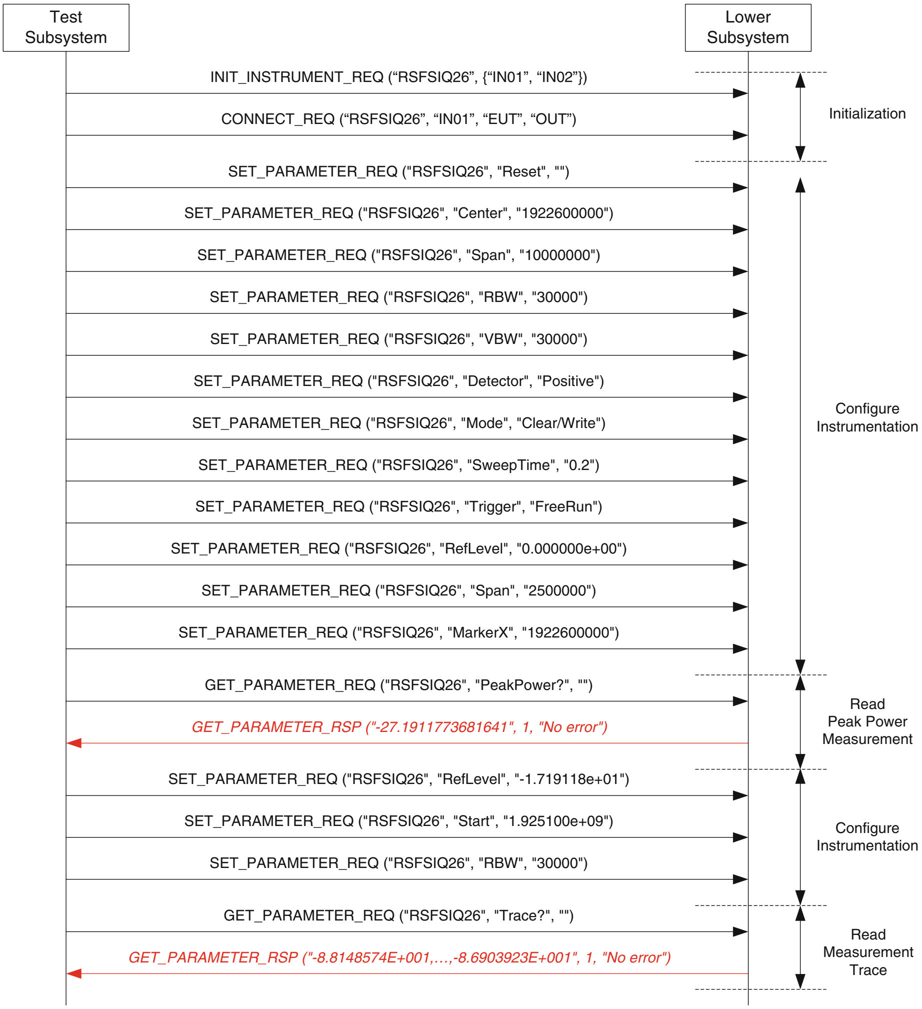


**Fig. 12** Sequence of messages for the test case until the first measurement is performed.[8]

8. Obtained measurements are compared with the reference emission mask and verdict is generated.
9. Test case ends (`STOP_INSTRUMENT`)

Figure 12 shows the sequence of messages exchanged between the Test Subsystem and the Lower Subsystem from the beginning of the test until the fourth step, where the first measurement is obtained. Figure 13 shows an example of the measurements obtained in this test; the dashed line gives the level of the reference power mask.

[8] Responses (messages SET_PARAMETER_RSP) have not been included in order no to unnecessarily clog the diagram.

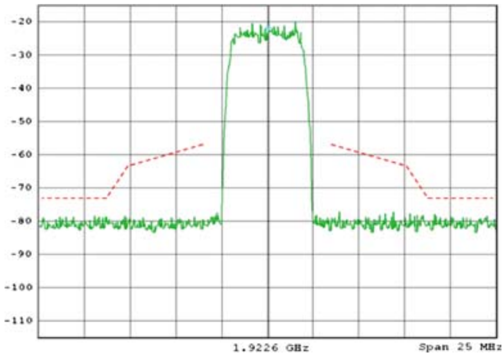


**Fig. 13** Graphical outcome of the Test Case (Spectral emission mask)

## 8 Conclusions

In this paper it has been shown that it is possible to design radio Test Systems using the same architecture as for protocol Test Systems. This assimilation allows using all already existing tools in the area of protocol tests and design elements common for both types of tests, such as those for execution control, report generation, test logging, etc., resulting in a decrease of the development costs. This architecture also provides for integrating instrumentation equipment from different vendors as well as the immediate substitution of any equipment by another with equivalent capabilities.

The use of TTCN for modelling the Test Cases helps increase their degree of formalization, avoiding ambiguities inherent to natural language descriptions nowadays used for the specification of radio tests. This would simplify the validation process for Test Systems, as the behaviour of the tests would already have been agreed by the standardization bodies. An abstract interface of primitives has been proposed for controlling the Instrumentation from the Test Suite. An example of how this abstract interface can be used has also been shown.

The formalization of radio tests would bring forward a simplification and a reduction of costs in the validation of test cases and, this would have a beneficial impact on the certification process.

## References

1. 3GPP, TS 34.121 (2003). 3rd Generation partnership project; Technical specification group terminals; Terminal conformance specification; Radio transmission and reception (FDD), v3.14.0.
2. Baños, J. (2002). Testing of bluetooth products in the industrial environment. In *28th Annual conference of the IEEE Industrial Electronics Society (IECON), Sevilla (España)*, 5–8 November.
3. Bluetooth Qualification Tester (BITE). (2003). *AT4 wireless*, from http://www.at4wireless.com/web_esp/sistemas/soluciones_ensayo/bite.
4. Bush, M., Rasmussen, K., & Wong, F. (1990). Conformance testing methodologies for OSI protocols. *AT and T Technical Journal, 69*(1), 84–100.
5. De Lapeyre, F. J., Woolaard, K. C., & Yvergnaux, S. (1988). The use of an ISO formal technique for conformance test specifications. *BT Telecom Technology Journal, 6*(1), 23–30.
6. ETSI, ES 201 873 (2003–2007). Methods for testing and specification (MTS); The testing and test control notation version 3, pp. v1.1.1–v3.2.1.
7. ETSI, *Comité ETSI (2004). Methods for testing and specification*, from http://portal.etsi.org/portal_common/home.asp?tbkey1=MTS.
8. ETSI, *Test specifications*, from http://portal.etsi.org/mbs/Testing/conformance/conformance.asp.

9. Gómez, J. A., Poncela, J., Fernández, U., Ruiz, F., & Entrambasaguas, J. T. (2001). *Arquitectura genérica para pruebas de radio y pruebas de protocolo*. In *XVI simposium nacional de la Unión científica internacional de radio (URSI'01)*, Villaviciosa de Odón (Madrid), 19–21 September.
10. IEEE, Std 488.1-1987 (1987). *IEEE standard digital interface for programmable instrumentation—Description*.
11. ISO 7498 (1989/97). *Information technology—Open systems interconnection—Basic reference model*.
12. ISO 9646 (1994/98). *Information technology—Open systems interconnection—Conformance testing methodology and framework*.
13. ITU-T, Recommendation X.290 (1995). *OSI conformance testing methodology and framework for protocol recommendations for ITU-T applications—General concepts*.
14. ITU-T, Recommendation X.292 (2002). *OSI conformance testing methodology and framework for protocol recommendations for ITU-T applications—The tree and tabular combined notation (TTCN)*.
15. ITU-T, Recommendation X.296 (1995). *OSI conformance testing methodology and framework for protocol recommendations for ITU-T applications—Implementation conformance statements*
16. ITU-T, Recommendation Z.100 (1992). *Specification and description language (SDL)*.
17. Linn, R. J. (1989). Conformance evaluation methodology and protocol testing. *IEEE Journal on Selected Areas in Communications, 7*(7), 1143–1158.
18. Lobato, S., Ruiz, E., Vigo, J. A., & Jiménez, J. M. (2008). *Generalised architecture for radio conformance testers using TTCN-3, The TTCN-3 user conference, Madrid*.
19. MTS. (1996). *Methods for testing and specification*.
20. Naik, K., & Sarikaya, B. (1992). Testing communication protocols. *IEEE Software, 9*(1), 27–37.
21. Poncela, J., Gómez, J., Valero, C., & Fernández, U. (2007). Using TTCN for radio conformance test systems. In *13th system design language forum (SDL Forum'07), París (Francia)*, 18–21 September.
22. Poncela, J. (2009). *Methods and tools for the design and verification of mobile communication systems*, Ph.D., University of Malaga.
23. Sarikaya, B. (1989). Conformance testing: Architectures and test sequences. *Computer Networks and ISDN Systems, 17*(2), 111–126.
24. SCPI Consortium (1999). *Standard commands for programmable instrumentation (SCPI)* (Vol. 1).
25. Zeng, H., Chanson, S. T., & Smith, B. R. (1989). On ferry clip approaches in protocol testing. *Computer Networks and ISDN Systems, 17*(2), 77–88.


## Author Biographies

**J. Poncela** received the M.Sc. degree in telecommunication engineering from the Polytechnic University of Madrid, Spain, in 1994 and the Ph.D. degree from the University of Málaga, Spain, in 2009. He worked in Alcatel Spacio before joining the University of Málaga at the Communication Engineering Department. He has actively collaborated with multinational companies (Nokia, AT4 wireless) on formal modeling and system testing in Bluetooth, UMTS and satellite systems. His actual research interests include methodologies for efficient development of complex communications systems, analysis of end-to-end QoS over heterogeneous networks and systems and models for the evaluation of QoE.

**J. T. Entrambasaguas** received the M.Sc. and Ph.D. degrees in Telecommunications Engineering from the Polytechnic University of Madrid (UPM), Spain, in 1975 and 1990, respectively. From 1975 to 1978, he was with E.T.S. Ingenieros de Telecomunicación, UPM, Madrid. From 1978 to 1993 he worked at Fujitsu-Spain R and D center, mainly in the development of packet switching systems, data transmission, and hardware/software technology for computer local area networks. In 1993, he joined the University of Málaga where he is now professor in the Department of Communications Engineering. His current research interests include digital signal processing, wireless access networks, power-line communications and methodologies for developing and testing complex communications systems.

**M. C. Aguayo-Torres** received the M.S. and Ph.D. degrees in Telecommunication Engineering from the University of Málaga, Spain, in 1994 and 2001, respectively. Currently, she is working at the Department of Communications Engineering, at the same university. Her main research interests include probabilistic QoS guarantees for wireless communications, end-to-end QoS over heterogeneous networks, cross-layer design, multiuser OFDM, and adaptive modulation and coding for fading channels.