Investigating and Verifying 3rd Generation Wireless Terminal Behaviour with the RTD (Rapid Test Designer)

Chris Foreman, Louise Mitchell

[Summary] Anritsu has developed a tool that provides simple creation of 3rd Generation Wireless Network protocols to analyse the behaviour of Terminals. The tool uses the popular MD8480 W-CDMA Signalling Tester and takes care of control commands required as well as creating lower level messages according to 3GPP (3rd Generation Partnership Project) specifications to provide an environment to develop and observe terminal behaviour in the laboratory.

1 Introduction

1.1 How will 3G (3rd Generation wireless) succeed?

Convergence, novel applications and reliable service are essential to generate revenue and satisfy user expectations in today's world of ubiquitous information transfer. So far 3rd generation wireless solutions have failed to generate much enthusiasm with the general public and since SMS (Short Message Service), the next application to captivate the markets is still awaited. In order for any "killer application" to be a success it needs to be reliable, easily integrated into terminals and certain to work at any time, anywhere on a variety of networks.

Already the wireless industry is looking at 4th Generation: and 3.5G systems are being deployed now as they are perceived to be needed for success. The race for data bandwidth is seen as essential to attract users. This may be true, but until applications are reliable and users find them simple to use, only the techno aware will be willing to try to use them.

Network Operators have already spent vast amounts, buying licences, rolling out their infrastructure and marketing their "lifestyle messages", so having to invest money in test equipment comes fairly low down their list of priorities. Testing has traditionally focussed on network measurement and optimisation equipment. However some network operators have chosen to invest in terminal performance measurements to ensure the behaviour of their UEs (User Equipment) is optimised for their networks and guarantee a better chance of success.

Since DoCoMo introduced its service in 2001, 3G phones have gained in popularity and now account for almost 50% of the mobile phones used in Japan. Users may not be aware of the complexity of 3G, however they do appreciate the attractive services that 3G offers. Higher bandwidth and packet switched access has provided fast efficient access to data, which in turn has provided exciting new applications like e-mail access, video calls, streaming data and TV: many which have not been fully exploited outside of Japan yet. In Japan and Korea, wireless subscribers have adopted many of the applications provided by 3G terminals and this is expected to be replicated in other parts of the world as terminals become more mature.

1.2 The Challenge of 3G

UMTS (Universal Mobile Telecommunication System) is a significantly more technologically complex and challenging handset technology than previous 2G (2nd Generation) systems as shown in Fig. 1. User expectations are also far higher than they were 10 years ago and they now expect terminals to work correctly at any time. With convergence and the compelling range of 3G applications and services available the

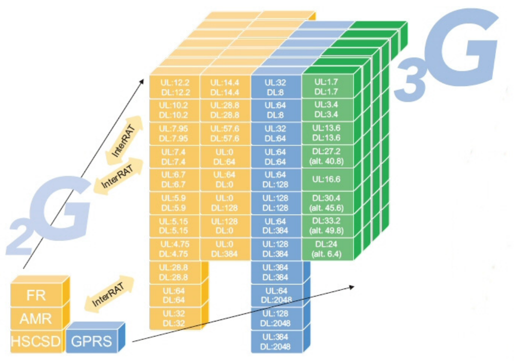

Fig. 1 Increase in complexity from 2G to 3G.

volume of test permutations required is enormous.

Some of the protocols in 3G may be based on GSM (Global System for Mobiles), but the complexity has become many orders of magnitude more complex. 2G was a largely voice based network with added data communication that evolved over many years. With 3G there are many different circuit and packet switched radio bearers and dynamic combinations of applications using data and voice which will be expected to exist in a future wireless world.

Many organizations have underestimated the complex and varied signalling that occurs when multiple applications run on a 3G network especially when it is being integrated onto an existing 2G network as is the case in Europe.

1.3 How does conformance testing help?

Although 3G terminals are developed to the 3GPP standards and need to go through conformance testing process, there are few conformance tests that attempt to simulate real world conditions or applications.

In theory this means that UEs passing these tests will perform correctly on any 3GPP compliant network anywhere in the world - but only to a point. Conformance tests are prescriptive and carried out with a set of idealised parameters. Little testing of user plane or application testing is covered. They are also written in a specialised language – TTCN (Tree and Tabular Combined Notation) – that many engineers and most network operators are unfamiliar with.

Current networks are based on a mixture of Release 99 and Release 5 and in the future will migrate to later 3GPP Releases. There are also many features that may not be available on networks today, making “live” testing impossible.

The only solution is extensive protocol simulation and testing in all phases of the UE development process.

1.4 What solutions are available?

There are now network simulation packages based on protocol development test systems. They provide a simple but fairly rudimentary way of simulating network conditions. Most of these systems still provide a non-user friendly interface, using TTCN or C based language: this may be acceptable for teams that have built up a knowledge of 3G, but network operators and new entrants to the market may not be comfortable with the 3GPP protocols and will be unable to produce simulations that test specific performance areas that they will be measured against.

Because there is also an enormous variety in the way that information is transmitted depending on network configuration, traffic and the QoS (Quality of Service) offered, the only way to ensure that applications are robust is to perform many tests simulating all these variants. Although CT (Conformance Testing) tests may prove adherence to a specification, they may not catch scenarios that occur in the field, or prove correct operation under unusual conditions.

1.5 Regression testing

It is quite normal for UE development teams to retest their designs, for example, even after going through type approval to ensure that minor variations to their designs do not affect earlier results. Also as the networks evolve and the specifications change, regression testing is important. The variety and combinations possible in 3G make it more critical than ever to provide an efficient test solution and even to test beyond the standard conformance test limits to ensure correct performance.

1.6 The perfect answer

With the volume of tests required, the scarcity of skills and finding people that have a combination of programming and 3GPP protocol knowledge, a solution that gave a fast and efficient method of test creation was needed. Anritsu determined that a tool with building blocks extracted straight from the 3GPP specifications could be combined with a graphical method of test creation to provide the right tool. The ability to rapidly design tests without the need to learn a specific programming language like TTCN and also to guide test authors through the creation process with basic programming blocks and an expert system was the catalyst for the RTD or Rapid Test Designer.

2 The RTD (Rapid Test Designer)

2.1 A graphical environment

The RTD uses a simple and unique graphical interface and a procedure library built up of building blocks to provide a rapid test development tool that has been derived from a proven test system used by all the world's major terminal developers. See Fig. 2. It allows the creation of sophisticated variants and network simulations and has the potential to provide an accurate simulation of a network or a competitive network, so that UE behaviour can be tested and examined.

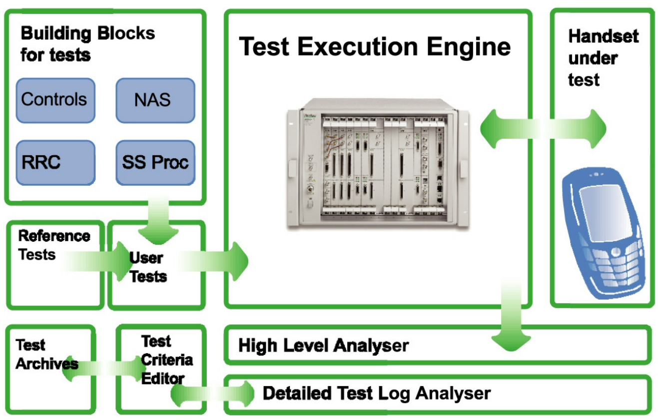

Fig. 2 Building blocks within the RTD

The MD8480 was introduced at the start of 3rd generation development and has evolved to provide HSDPA and other new technologies. It provides up to 4 W-CDMA and 1 GSM cell in a single unit and can be combined with further units to create sophisticated Network models.

2.2 How we developed the RTD

The RTD is a graphical tool that does not rely on special languages and provides a simple way to test many of the features of a UMTS terminal behaviour as shown in Fig. 3. It is easy to create and modify tests and the graphical interface provides a pictorial view that makes sharing test details and test purpose simpler than traditional methods. It also takes care of the system simulator commands and because Anritsu has developed the hardware, complexities that may be difficult to understand by 3rd parties are developed and maintained as part of the tool. Libraries of Test scenarios have been created for the RTD allowing new users to step in at a relatively advanced level.

The RTD graphical environment allows the user to drag and drop functional blocks into a logical sequence, without having to consider underlying complexity which may not be directly important to the test. Logically available follow-on procedures are highlighted to the user, and in addition to static test analysis, this ensures any invalid message sequencing is picked up before run-time.

Each block may then be parameterised, either using proven catalogue entries for that procedure type, or modified by the user for custom test behaviour. This means for example, that it's possible to create an unlimited number of simulated system configurations, using only a single "Configuration System" procedure block.

Procedure parameterisation feeds directly into the underlying system model which is dynamically maintained throughout the life of the test, allowing the use of user-defined, and system supplied variables, thus lessening the likelihood of human error in test creation.

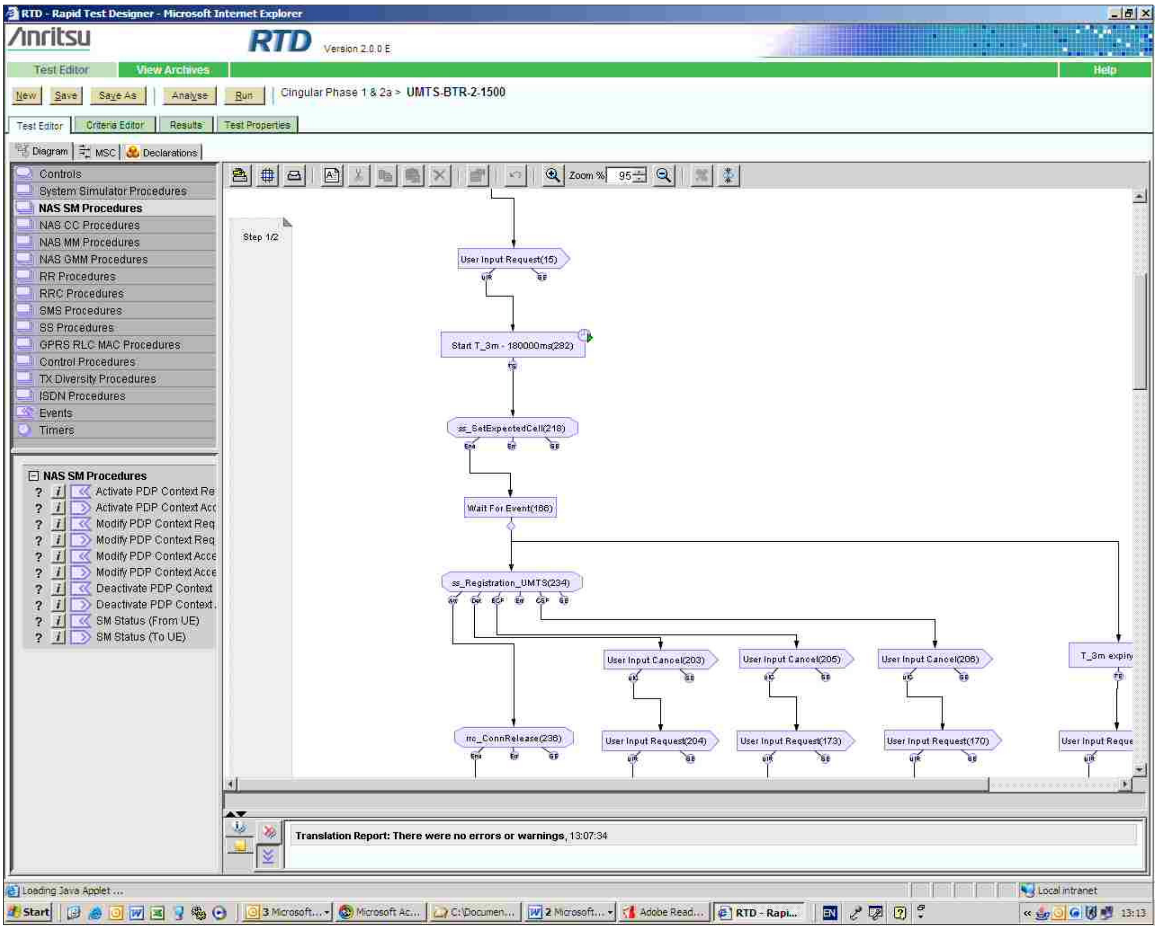

Fig. 3 The Graphical Test Creation Pane of RTD

This is not just the RTD equivalent of cumbersome pixit files, since even complex data types can be assigned to variables which may be used wherever valid throughout the parameterisation of the test, and even used in test flow decision.

2.3 Test script guidance

Test script creation is guided and checked in many areas. Before adding a procedural block, the tool advises if the block is valid, inadvisable forbidden or indeterminate. Fig. 4 shows this sample that the only valid blocks to connect to the node are “configure” or “shutdown” the system.

Underlying message and IE type definitions are imported into the system from ASN.1 provided by 3GPP specifications.

Once imported into the system, the definitions are used to ensure only valid options are given to the user and semantically correct message generation always occurs. Fig. 5 is showing that the measurement identity is limited to 1 to 16.

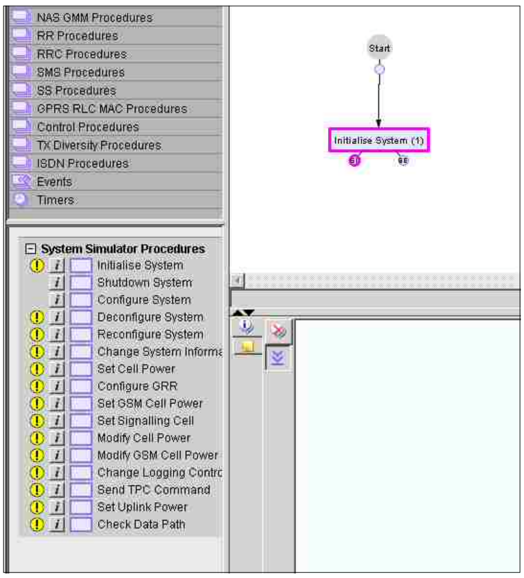

Fig. 4 Test script guidance.

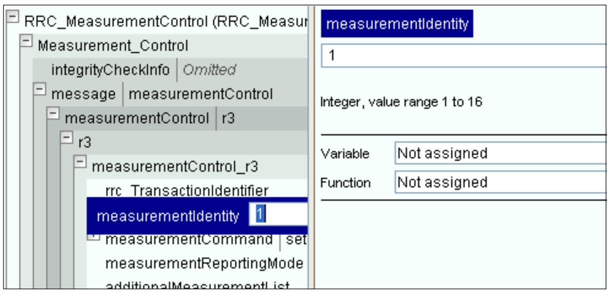

Fig. 5 Value limitation in parameterisation

In addition to this, however, in the RTD many of the predefined SRB/RAB (Signalling Radio Bearer/Radio Access Bearer) definitions available are based on the Reference Radio Bearer Configurations given in TS 34.108 (6.10). All are extensible and it's possible to define customised or new user-defined SRB/RABs. Using the simple procedure parameterisation feature, it is then possible to specify, by reference only, either a system- or user-defined radio bearer configuration.

The RTD supports concurrent 3GPP versions within the procedure library. The requirement within the test for conformance to a specific specification release may be designated by parameter, forcing the dynamic generation of the correct message for the specified release.

The system is dynamically monitored throughout the test and the underlying model updated accordingly, allowing for example, both absolute and relative power handling. More importantly, this underlying model provides essential state checking, and will halt the test when attempting the impossible, e.g. attempting invalid state transitions, or removing a non-existent radio link - either of which would damage the integrity of the system. See Fig. 6 for the test report and showing the reason.

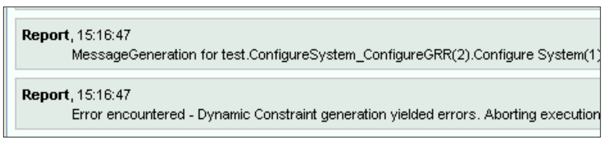

Fig. 6 Comprehensive on-screen error checking

2.4 Simple test variants

The RTD provides a simple way to create tests and variants. For example, this can be achieved by simply changing the RRC (Radio Resource Control) state parameterisation of a number of the procedures, which will change the control flow through the test, or selecting alternate SRB/RAB parameterisation to generate different outgoing signalling and control messages.

2.5 No need to understand how to program hardware

All underlying hardware control is achieved automatically - determined by the parameter requirements of a procedure block, e.g. which channels are configured, how may RABs are set up, and whether the terminal equipment port is to be configured and connected.

2.6 TTCN can be included

There is no need to use or write TTCN, (although it is possible to use existing routines if required). Instead, the appropriate underlying TTCN sequence is executed at run-time, dynamically updated to match parameter requirements of a procedure block.

The system is also extensible, in that where functionality has not yet been implemented, it is possible to place configurable procedure blocks which point to the required code (in the users' own built .dll (Dynamic Linked Library). The appropriate TTCN sequence is then injected at run-time.

3 Developing a test campaign

3.1 How to define test requirements

As companies aim for unique reasons to achieve subscriber billing, and hence revenue the needs and wants of different organizations will model the type of test coverage required. Some will be defined by previous experience, yet again the sort of issues that are likely to need checking may not have been anticipated.

Interoperability tests already exist in field trials and it is possible to simulate a number of these using the RTD. In fact, once a basic Network model is created two distinct paths are possible.

1. Variants of different Networks can be created simply by applying a catalog to the tests.
2. Simple changes can be made to each test to create variants.

3.2 Network model

The network model is a collection of pre-defined cells with different properties that are utilized in various configurations within tests to simulate different environments. These environments provide intra-frequency, inter-frequency and inter-RAT (Radio Access Technology) capability combined with multiple PLMN (Public Land Mobile Network) availability. The network model is used to create in-house test libraries and allow users to create their own test libraries based on the network model or a customized version of it. The network model is available in the RTD in several of the 3GPP defined frequency bands.

One big advantage of using the network model is the way that it can be used for applying a change across an entire test suite. This is particularly useful where new functions need to be checked ahead of deployment. Also where a new network feature is changed for example where a new parameter needs changing globally.

3.3 Analysis

The RTD has a built in tool that allows the path of a test to be compared against specific criteria. This means that a test is created and depending on the path chosen by the terminal, criteria can be defined by the test creator. The criteria would normally be chosen during test creation, but can be changed in the future and applied immediately to results without the need to re-run tests. This is particularly useful for regression testing and where changes to a specification are applied in the future.

The user simply chooses a path and applies criteria and comments to the path. Any number of paths can be chosen, making it possible to pre-judge behaviour without the need for any specific knowledge of the 3GPP specifications or reviewing complex logs.

3.4 Usability issues

In addition to the straightforward creation, analyse and execution of tests, the RTD tool aids the rapid understanding of tests created by other developers. There are a number of mechanisms in place which effectively allow a test to self-document.

Firstly, on-screen comments can be attached to a specific procedure, or scoped to include a whole sequence of procedures, improving usability when the run-time user has no experience of the test. See Fig. 7.

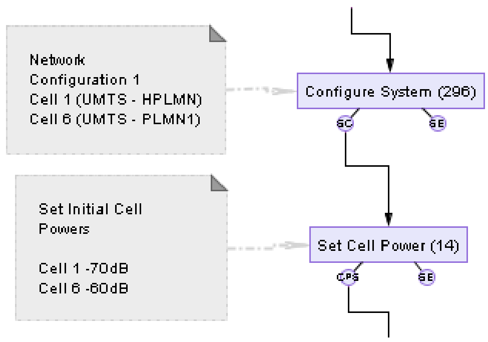

Fig. 7 Individual procedure comments

Procedure blocks can be placed on-screen in a way that is meaningful and adds context to the test; with functionally related procedures visually grouped together. See Fig. 8.

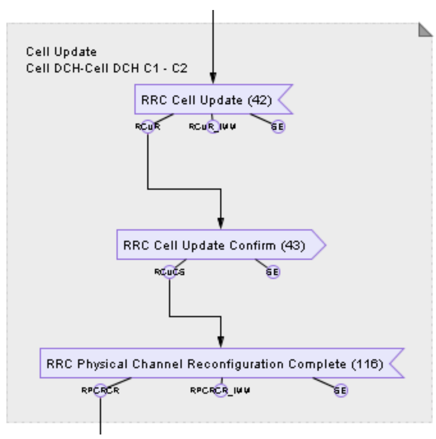

Fig. 8 Blocks of functionality with comments

Furthermore, the test creation pane can be zoomed in and out to give the small section of functionality being assessed some context within what may be a very large test case.

Finally, using in the test properties pane, sample is shown in Fig. 9, a general description of the test can be supplied by the creator, or if using a pre-defined library, the information provided here can offer valuable information regarding test purpose and test behaviour.

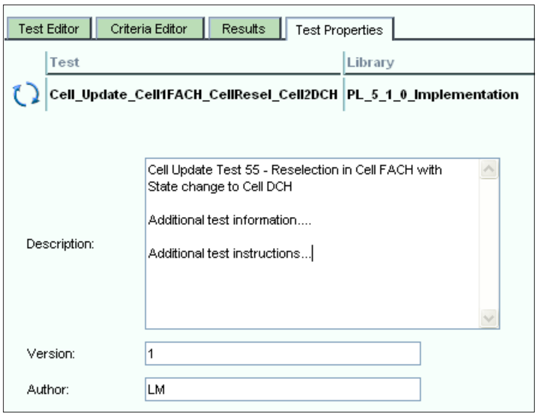

Fig. 9 Test properties information

3.5 What about existing tests?

With each new RTD version, comes an updated procedure library, providing extended functionality including, most recently, HSDPA. An upgrade process is in place to ensure that no test written using a previous version of the tool will be invalidated, or display run-time behavioural differences.

Additional functionality equates to new parameters becoming available to a procedure. When the test is loaded, it will be upgraded to include these, whilst maintaining backward compatibility. For example, prior to tool support for both Release 5 and Release 99 message generation, it wasn't necessary for the creator to specify the required release. The upgrade process therefore adds the release parameter to relevant procedures, but omits a value – causing default Release 99 behaviour.

By providing this simple upgrade mechanism, a test can very quickly be amended for the new Release 5 message generation, simply by specifying the new parameter value. See Fig. 10 for release version handling.

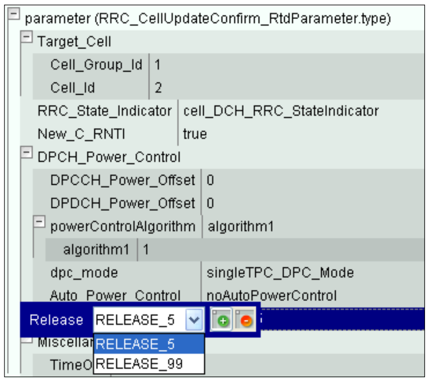

Fig. 10 Release version handling

The same is true for pre-defined RABs. As new SRB/RAB definitions are added to the RTD, for example those supporting HSDPA, these too become available to an upgraded test through user parameterisation. See Fig. 11.

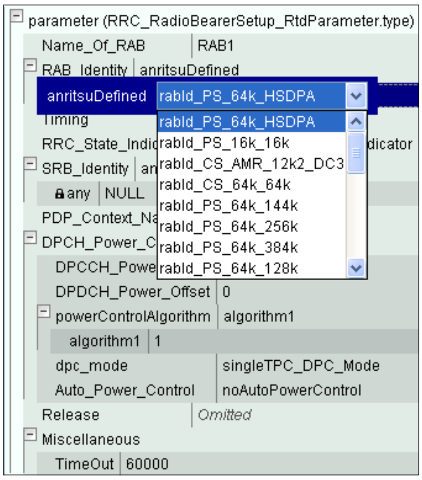

Fig. 11 Additional RAB definitions available

3.6 Test automation

Using the automation facility provided by the RTD, it is possible to fully automate tests (where the user equipment allows) using the AT (Asynchronous Terminal) command set, to reduce and even remove altogether the user interaction required to run a test. If even a single test is run in this way, the likelihood of human error is reduced. From a larger perspective, it means that whole archives of tests can be run, without even having to first load them. The RTD even provides a 'Run' option from the archive selection screen. See Fig. 12.

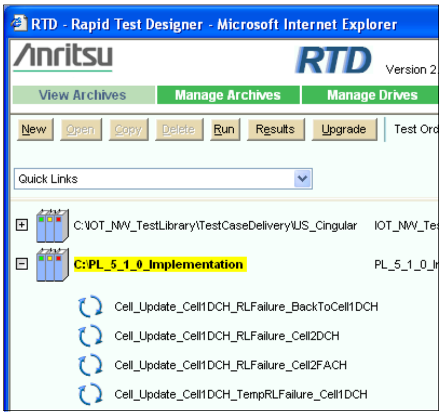

Fig. 12 Running full archives from archive view

This added functionality is especially useful for the vast regression and verification testing requirements of the 3G mobile telecommunications industry.

Just like any other procedure provided by the RTD, an AT command can be placed within the test sequence, as shown in Fig. 13, and parameterised with the actual command string required.

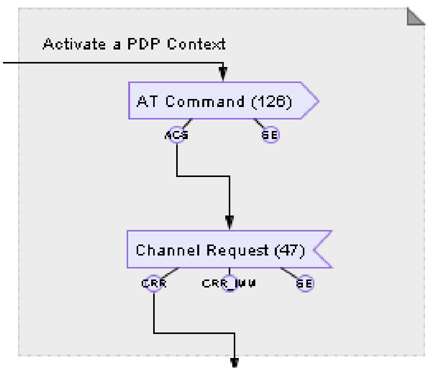

Fig. 13 In-line AT Commands

The RTD doesn't just allow the rapid generation and testing of functionality, it has been created with a strong focus on the reuse of generic functionality. Two value added extras include the ability to define compound procedure and user defined catalogues.

3.7 Compound procedures

Using the RTD it is possible to abstract-away whole sections of test sequence behaviour into 'compound procedures'. A compound procedure is given a user defined name and appears much like any other RTD procedure within the test as shown in Fig. 14. This allows them to be linked, wherever valid to other procedure blocks.

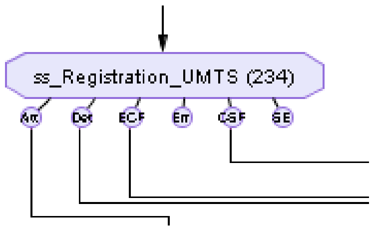

Fig. 14 Compound procedure with user defined name and outcomes

Unlike a standard procedure block, which can be opened to reveal its parameters, a compound procedure can be opened to reveal its component parts – simple procedure blocks, which can be formatted in much the same way as the test itself. Compound procedure outcomes can be defined and used as link points in the outer test.

Test behaviour that is not essential to the ultimate purpose of the test (pre- post-amble, for example), can be hidden away, to better visually represent the test function and further simply the test for the user.

Moreover, when particular set of procedures is commonly reused, it can be highlighted and archived into a compound procedure, stored separate to the test file, and imported into any test where the functionality is required. For example, a user may define a generic registration procedure, covering all expected cases of UMTS registration.

Likewise, where a particular parameter configuration is used commonly this can be stored in a user defined catalog file and referred to, just like any RTD defined catalog entry for a procedure as shown in Fig. 15. Any test which imports this user defined catalog can use the new entry to parameterise similar procedures.

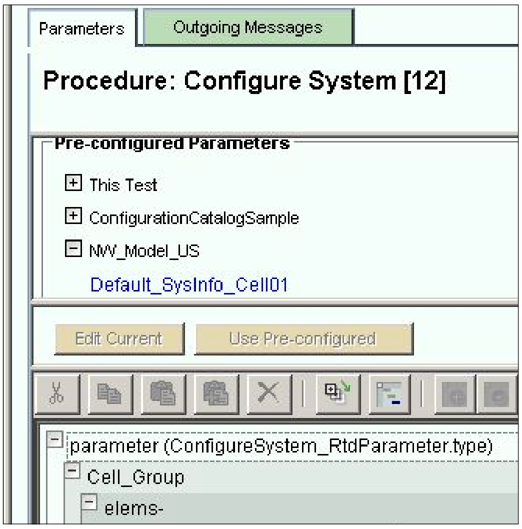

Fig. 15 Selecting user defined catalog entries

4 Using the RTD in practice

4.1 Roaming and cell reselection

As well as different network infrastructure suppliers, there are also a variety of ways to transport information depending on the quality of service required. This means that there will be a multitude of different ways that applications will be delivered to users. Even a simple voice call may be assigned a bearer that may change during the call due to traffic density or handover from 3G to 2G.

Network operators will further demand that the algorithms in the UEs give them as much opportunity to gain revenue from the subscriber, and to thus exercise them in ways that provoke a favourable “misdirection”.

For example, when arriving in a new country, network operators will want to ensure that UEs attach to their preferred network and not to a competitor’s. So reselection and reselection tests are essential to simulate environments where there are competing networks.

A network simulation is required that provides controlled functions similar to those found on real networks. This is the only reliable way to ensure international roaming gives customers the user experience they expect and deserve.

The combination of more than one MD8480C provides the tool with the ability to generate up to 8 W-CDMA cells and 2 GSM/GPRS cells. This can allow multiple cells scenarios that closely represent handovers, cell reselection and roaming with competitive networks present. This can provide valuable information to protect the real network from incorrect protocol exchanges that may potentially interfere with other users or worse still bring it down. It can also be used for the preparation of new services before they are deployed on the live network.

3G application developers will also want to ensure that their application running on a given network (don’t forget in the battle to win customers with ‘killer’ applications many applications will be network unique) aren’t susceptible to being accessed and thus copied by rivals.

4.2 Engineering resource challenges

In order to perform many different types of network scenario simulations, TTCN would be far too cumbersome. By making use of the RTD, hardware that has been used to design, develop and prove 3G/2G UEs, allow testers unfamiliar with TTCN to create scenarios in minutes rather than days. The growing volume of test scenarios can then be effectively managed and the capability and inherent security of the 3GPP spec fully exploited.

Even with the RTD, there are tough choices to be made to keep the volume of testing to a manageable level. By using the Criteria Editor carefully and creating tests that cover a specific purpose, the volume of test and analysis is reduced and some degree of priority can be given.

UE defects may be prioritised into different categories.

The most serious faults such as those causing loss of revenue or serious faults may prevent terminals from being deployed.

Less serious faults may include those that have a graceful recovery or are not considered serious enough to cause user annoyance or loss of revenue.

Beyond that, there are faults that may occur and are corrected in future releases.

Finally, terminals already deployed may cause issues on certain networks. The RTD provides an elegant tool to simulate the different conditions in a laboratory environment.

5 Conclusion

The RTD provides a highly productive tool for engineers that need to produce good test coverage for 3rd Generation UEs without the need for complex programming languages or a deep knowledge of the 3GPP protocols.

Authors:

Chris Foreman (Mobile Communications Test and Measurement Europe (MCE)), Product Marketing Manager

Chris has been involved in mobile communications for most of his career. He has been involved with test solutions for the early cellular systems and has seen the ratio of RF (Radio Frequency) to Protocol testing change dramatically to the current systems that depends almost totally on signalling to achieve their goals.

Louise Mitchell (MCE), Senior Software Development Engineer

Louise has been involved in developing the procedural blocks that are an essential part of the RTD since its inception over three years ago.