Testing of 3G 1xEV-DV Stack – A Case Study

Ira Acharya and Hemendra Kumar Singh

Tata Consultancy Services, D-4, Sector-3, Noida-201301, India
{iraa,hemendra_singh}@delhi.tcs.co.in

Abstract. Due to immense competition in the market, mobile equipment vendors and service providers are faced with the challenge of delivering solutions early, way ahead of their competitors. Time-to-market pressures necessitate a carefully worked out test strategy for verifying and validating the correctness of mobile communication solutions. This paper outlines the various challenges that are faced in the protocol testing of mobile communication products. An evaluation of formal languages such as SDL and TTCN in the design, development and testing phases of such products is also included in the paper. The strategy employed in the testing of a 3G 1xEV-DV Base Station stack has been covered as a case study.

Keywords. Testing, conformance, simulation, test automation, test scripts, testing challenges, IUT, formalism, formal description techniques, 1xEV-DV, EV-DV, DV, protocol stack, 3G, reference implementation, SDL, TTCN, MSC, 3GPP2, queues, timers, task, behavior, LAC, MAC, RLP, Signaling, L2, L3, logging, video conferencing, packet data channel, PDCHCF, Message Integrity.

1 Introduction

With the increase in the sophistication and complexity associated with mobile computing, the challenges faced in the verification and validation of mobile communication components becomes manifold.

This paper describes briefly some of the challenges that are confronted in the testing of components of radio access networks in third generation networks. This paper discusses the challenges in the context of testing a 3G-reference stack for a 1xEV-DV Base Station. The stack development had been undertaken while the standards were still evolving, due to which the risk of having to undertake major rework was very high. As a direct consequence of this, the design, development and testing of the stack required a carefully worked out strategy to be able to cope with changing specifications.

2 Outlining the Challenge

The difficulties that were faced in the testing of the 3G stack solution, by virtue of it being an aggressively evolving technology are listed below:

- Non-availability of handsets or related equipment: One of the major problems encountered in testing of the built product was that handsets against which the de-

D. Hogrefe and A. Wiles (Eds.): TestCom 2003, LNCS 2644, pp. 20–32, 2003.

veloped product needed to be tested were not available. This was handled by use of the complementary protocol stack method, which is described in the subsequent sections of the paper.

- Conformance to specifications: Testing the implementation against its specification was a very challenging and critical task. The specifications are large and complex, which only increased the intricacies of the problem. TTCN was used for testing conformance of the stack.
- Platform independent protocol: The target segment for this kind of protocol development included mobile manufacturers who use their own custom hardware, therefore the best interest laid in providing a stack solution which would be portable across platforms. SDL was chosen as the design and development platform to handle this. The usage of an SDT provides a mechanism to generate code for most of the contemporary RTOS's prevalent in the industry today.
- Automated Testing Environment: To reduce the testing time, an automated testing environment was desired through the usage of tools. TTCN supporting tools and the SmarTEST tool developed in-house were used for automated testing.
- Evolving Specifications: The most challenging of all issues and unique when compared to the regular testing issues encountered was the non-availability of a frozen standard against which the stack components could be tested. There was the need to be able to cater to changing test conditions and changing Implementations Under Test (IUT). To deal with this issue SDL was used as the design and development platform, as it aids in the rapid incorporation of changes without impacting the rest of the system. SDL tools facilitate testing at the model level itself. They enable the testing of continuously changing systems.
- Time-to-market pressures: To handle the pressures associated with delivering an early solution owing to stiff competition in the market. This resulted in a reduced development lifecycle, and a reduced timeframe for the testing activity. This was again one of the factors, which influenced the decision of choosing SDL as the development platform.

3 The Role of Formalism

Formal Description Techniques (FDT) were used in the design, development and testing of the stack. FDT's are increasingly used in the industry while working with complex communication protocols [15]. In the tight schedules for completion of the work, they helped in guaranteeing syntactically and semantically un-ambiguous formal descriptions of the protocol as well as interoperable and compatible implementations of the protocol so that it can have value to the potential customer.

Specification and Description Language (SDL), [6] was used for the design of the system primarily because of the fact that the SDL system can be tested at an early stage in the design phase, as the built system is completely specified in SDL.

Another formal language introduced by the OSI standardization committees 'Tree and Tabular Combined Notation (TTCN)' was used for describing conformance test cases [7, 8]. Abstract Test Suites (ATS) were created using TTCN. The ATS were a description of tests to be executed for the system. The tests were described using a black-box model, i.e. only control and observe using the available external interfaces. The advantage of using TTCN for testing included the possibility of specifying constraints of complex data types, to react to alternative results of use cases and to define the expected results [9]. Interworking between SDL and TTCN is shown in Figure 1.

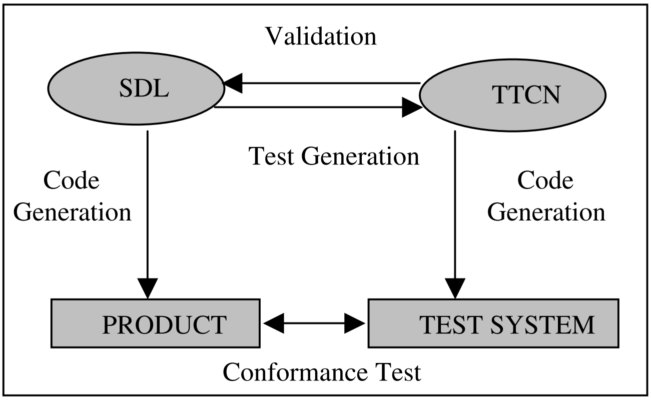

Fig. 1. Interworking between SDL and TTCN

TTCN is extensively used for conformance testing of the system. Conformance testing verifies whether an implementation performs according to the stated standard/specification/environment [10]. This involves the development of Test Suites, a collection of test cases that specify objectives, operating conditions, inputs and expected outputs, in order to evaluate an implementation.

Among the other formal description techniques available include Estelle and Lotos developed by ISO. Estelle is based on an extended finite state machine model and Lotos is based on a calculus of communicating systems [12,13].

4 Testing a 3G 1xEV-DV Stack

1xEV-DV is a CDMA based third generation standard in the wireless value chain and has been standardized within 3rd Generation Partnership Project 2 (3GPP2). It provides for high peak data rates (3.072 Mbps on the forward link and 451.2 Kbps on the reverse link) and promises the enabling of real-time applications such as video-conferencing, voice over IP, 3G-multimedia and telemedicine.

3GPP2 had taken up the 1xEV-DV Standards Development activities. Working Group (WG) 5 of the Technical Specification Group – CDMA 2000 (TSG-C) was entrusted with the responsibility of identifying a framework proposal for the standard. Within WG5, two groups (out of a total of eight) L3NQS and 1XTREME submitted framework proposals for the standard. L3NQS constituted Lucent, LG, LSI, Qualcomm, Nortel and Samsung whereas 1XTREME constituted each of Nokia, Motorola, Philips Semiconductors and Texas Instruments. A harmonized version of the proposals was sought in the light of which Nokia submitted a proposal on a way to move forward during the October 2001 TSG-C meetings. This resulted in a consensus on a harmonized framework for the 1xEV-DV Standard without objection. Thereafter the

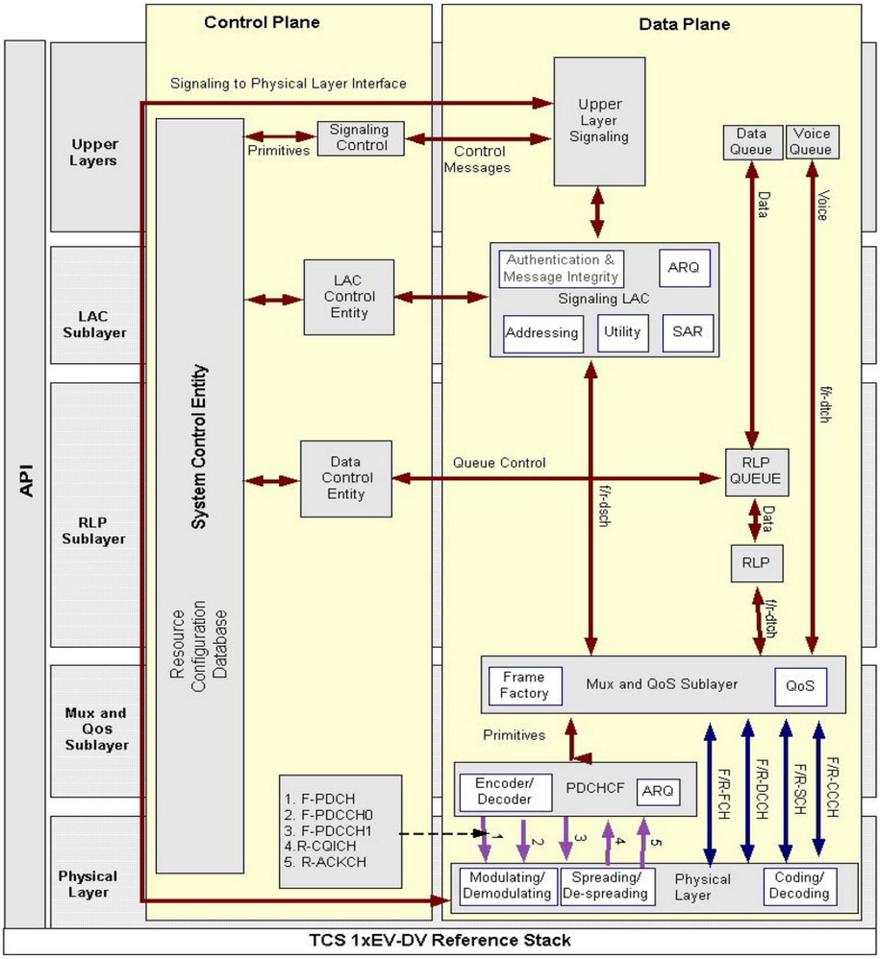

Fig. 2. 1xEV-DV Architecture

Working Groups 2 and 3 in 3GPP2 took up the text development activity and a released standard was available by the end of May 2002 as release ’C’ of IS-2000 [1-5].

The 1xEV-DV Base Station stack consists of Upper Layer Signaling, Link Access Control (LAC), Medium Access Control (MAC), Radio Link Protocol (RLP) and the Physical Layer. An additional entity called the Forward Packet Data Channel Control Function (F-PDCHCF) also exists in a 1xEV-DV Base Station stack. The architecture of the 1xEV-DV Base Station stack is depicted in Figure 2.

In order to facilitate testing of the stack, every artifact of the protocol stack was treated as an entity. The artifact may be a layer or sublayer of a stack or it may be the entire Mobile or Base Station. The definition of every artifact as an entity results in it possessing the following attributes:

- Channels: These are the communication paths between processes contained in different entities. It is through channels that entities are notified of events. The behavior of an entity is defined solely in terms of its dynamic reactions to events on its input channels, and the production of events on its output channels.

- State: The state of the entity captures its current status at a particular moment in time. An entity undergoes state change on the occurrence of events which are communicated to it through the channels
- Parameters: These are the variables whose values at any point of time form a part of the abstract state of the entity
- Processes: These are the threads of computation that act on the events arriving on the input channels of the entity, or that act upon timers. As a result of the computation performed within processes, events may be generated on the output channels.
- Components: These are again entities, which form sub-entities of the entities behavior.

The entity framework is shown in Figure 3.

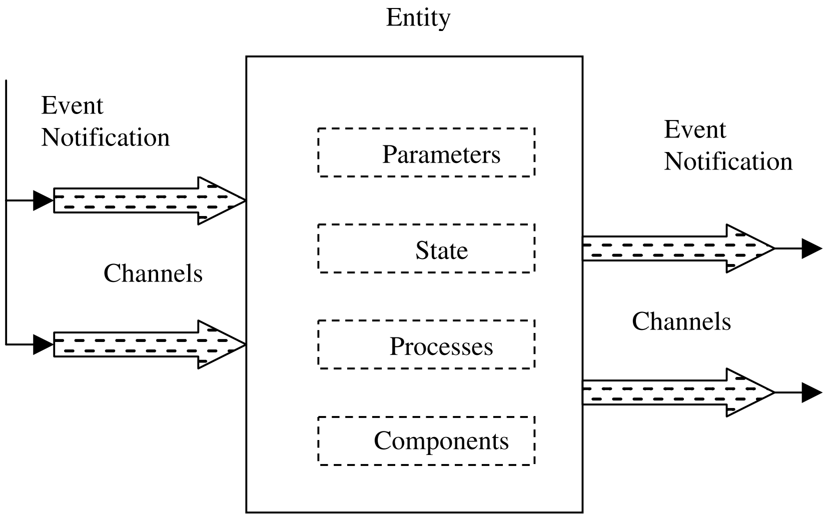

Fig. 3. The Entity Framework

With this definition of an entity SDL was chosen for the design, development and testing of the stack. SDL provides the concept of blocks, processes and channels, which merge very well, with the entity concept. SDL also has a high degree of testability owing to its formalism for parallelism, interfaces, communication and time. The powerful simulation facilities of the tool enabled early testing of the developed system. This allowed an early assessment of the performance of the system as well as that of the required resources. Decision in favor of SDL was also taken due to the following additional reasons:

- Continuously evolving specifications necessitated an easy way of modifying the design model
- It has architecture that reflects the standard
- SDL tools e.g. Telelogic Tau enable code generation for nearly all the commonly used platforms such as VxWorks, Win32, Nucleus, OSE Delta and QNX. This provides a way of dealing with the challenge of delivering portable solutions

- Studies suggested that the time for development of a call automation solution were reduced by 50% [16]

In the SDL system, Architecture is graphically represented with the help of Blocks while Processes represent Behavior. Communications between various processes are handled with the help of Signals and Channels. SDL has its own set of Abstract Data Types to handle data and functions.

The entities such as Signaling, LAC, MAC and Physical Layer of the 1xEV-DV stack were represented by blocks in the SDL system. The various sub-layers such as Authentication, Automatic Repeat Request (ARQ), Addressing, Utility, Message Integrity and Segmentation and Reassembly (SAR) within LAC were represented by processes. Each process undergoes various state transitions during its life cycle depending on the primitive being transmitted from other processes. Primitives between processes were handled in the form of signals sent over the channels in the system. State machines within processes handled the transition between different states. The system level diagram for the 1xEV-DV reference implementation is shown in Figure 4.

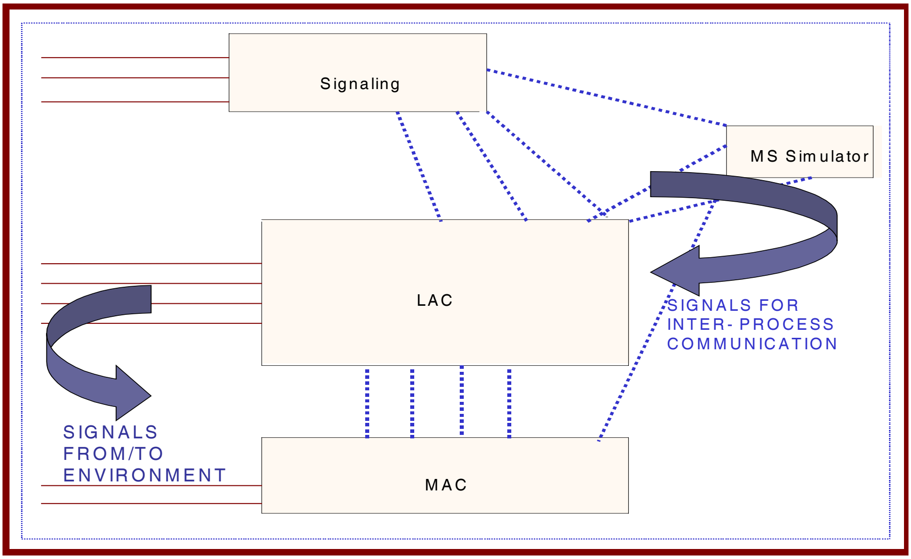
Signaling
MS Simulator
LAC
SIGNALS FOR INTER-PROCESS COMMUNICATION
SIGNALS FROM/TO ENVIRONMENT
MAC

Fig. 4. System Level Diagram of 1xEV-DV Reference Stack

A Reference Implementation (RI) of the 1xEV-DV Base Station stack was developed which included Layer 3 and Layer 2 of the stack [11]. In the absence of actual hardware, Layer 1 was simulated using a simulator tool. The reference implementation was tested independently. The various stages in the workflow of testing of the reference stack are described in the following sections. Also discussed are some of the distinctive techniques adopted to test the system.

Task Testing: At its heart a mobile communication system is a collection of tasks/processes working together to accomplish system function. Each task is often a deterministic program able to execute separately from, and concurrently with, other tasks. In the SDL system each process, analogous to a task, was tested independently using a number of techniques such as simulation and logging. In the absence of implementations for the adjacent layers, the simulation facilities were exploited. Input signals to the process were provided through the connecting channels.

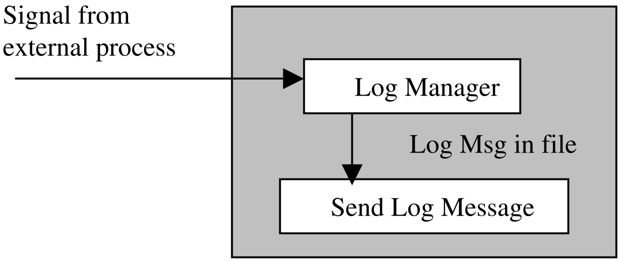

Fig. 5. Logging Framework

Logging was extensively used for testing at various stages. A Log Manager entity in the SDL system was responsible for logging all data received in the form of signals to a file. The output in the file was then examined to verify the functional correctness of that layer/sublayer. The logging framework is depicted in Figure 5.

The Log Manager collected all the parameters to be logged in the form of a signal from the external entity. The Log Manager then directed the message with all the parameters to the 'sendLogMsg' function, which would put the information in the appropriate log file. For instance, if the message to be logged were the error message then it would go in the error log file and if the message contained debug information it would be logged to the debug log file. The logging in SDL is depicted by in Figure 6.

Behavior Testing: Test scenarios covering the different kinds of input signals to a process were designed in order to test the behavior of the various entities. Environment functions for interaction between the outside world and the SDL system were handled in the SDL system. Test cases corresponding to input signals from the environment were executed to verify the correctness of the functionality of entities that received inputs from environment.

Finite State machines (FSM) represented the behavior of each process. The FSMs were tested using Message Sequence Charts (MSC) in SDL. MSC was also used to show interactions between components. MSC's provided a clear description of system communication in the form of message flows. The behavior of the process was expressed in terms of a number of states and inputs, and the outputs and state transitions produced by the arrival of a given input in a given state.

Inter-task Testing: A technique used to carry out inter-task testing, was to store the output from the intermediate processing stages i.e. the outputs from the intermediate entities of the stack were stored into a file and made available to the next entity in sequence. This enabled use of the same parent test data for all of the entities in the stack. This mechanism of testing is depicted in Figure 7.

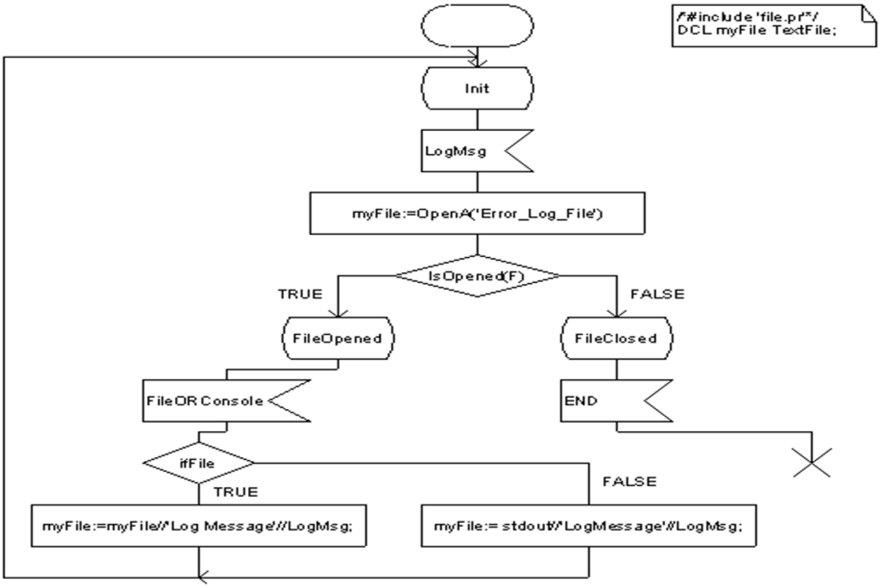

Fig. 6. Logging in SDL

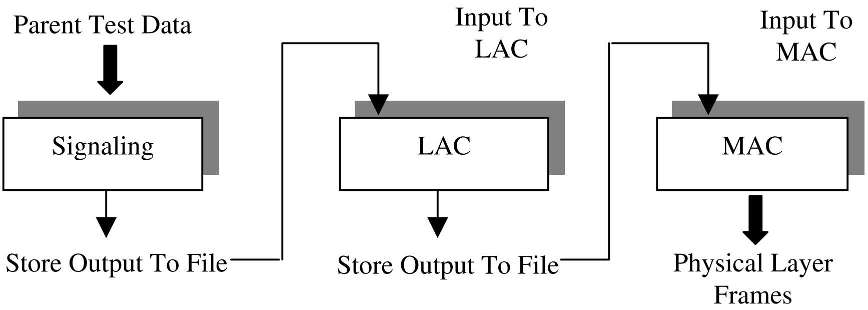

Fig. 7. Inter-task Testing

Tasks that communicate via FIFO queues were tested to uncover errors in the associated processes. Mobile communication system designed with multitasking approach use various mechanism such as semaphores, mailboxes, message systems and FIFO queues for inter task communication. In the stack, queues were used at the boundaries of tasks such as LAC and MAC. The queues were tested extensively with different conditions and inputs to ensure their correctness.

Once errors in individual tasks and in system behavior were isolated, time-related testing was performed. The expiry of timers such as those indicating the sending of the next Physical Layer frame (every 20 ms) was simulated by the manual sending of signals from the environment. These were used to trigger the sending of a Physical Layer frame from the MAC layer.

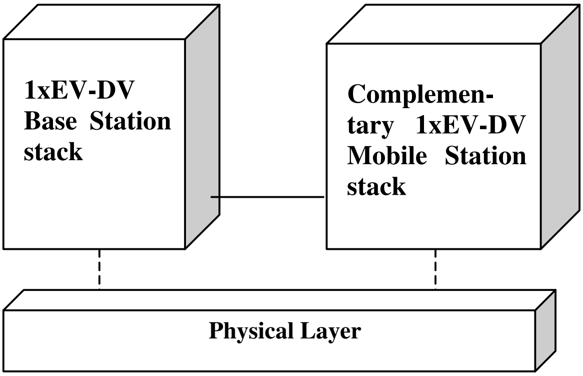
1xEV-DV Base Station stack
Complementary 1xEV-DV Mobile Station stack
Physical Layer

Fig. 8. Complementary Stack for 1xEV-DV

System Testing: SDL provides the feature of simulation using MSCs through which the flow of data can be easily traced. The simulation feature proved to be an invaluable tool, allowing easy verification of message interactions via MSCs. COM files (File containing all the input signals) were generated for the 1xEV-DV system and various scenarios were tested. The excitation in the SDL simulator was done by explicitly feeding test values into system.

In the absence of a Mobile Station implementation, a complementary protocol stack was developed. The complementary protocol stack bypasses the Physical Layer completely and eliminates hardware dependencies while testing the protocol stack and the application layers. It could test the adequacy of system timer values while sending and receiving messages. It could analyze the messages and point out non-conformance to protocol standards.

The complementary stack and 1xEV-DV stack runs on same system. The SW on the system has an alternate module, which hides the routing information from the upper layers and route the messages/signals to the complementary stack. During a normal product run, these messages are to be routed to the Physical layer through the interface provided by the hardware. The complementary stack is shown in Figure 8.

Conformance Testing: In order to verify the functional correctness of each entity in the stack against its protocol specification generic protocol conformance test architecture was designed. TTCN was used extensively to carry out the conformance testing activity.

The Telelogic Tau TTCN tool was employed to create the Test System. Each entity to be tested was configured as the IUT. The test architecture is depicted in Figure 9.

The LT and the UT are the Lower and Upper Testers for the implementation under test (IUT). The Points of Control of Observation (PCO's) are the points in the abstract interface where the IUT can be stimulated and the responses can be inspected. The communication between the LT/UT and the IUT is through Abstract Service Primitives (ASP's). An ASP may or may not contain a Protocol Data Unit (PDU).

The SDL system was simulated using TTCN-SDL co-simulator. The simulation allowed execution of a TTCN test suite in the host environment. For the conformance testing at the system level, the IUT was the 1xEV-DV system. The TTCN-SDL co-simulator was connected with the SDL simulator using the '*start-itex*' command in the SDL simulator window followed by a '*go-forever*' command to execute the SDL system. Then the test cases were executed. The logs were also generated after the tests were carried out.

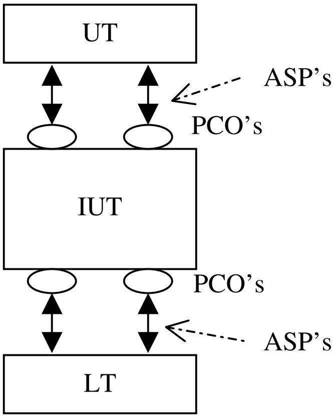

Fig. 9. General Protocol Conformance Test Architecture

Protocol standards bodies such as 3GPP and 3GPP2 have defined standardized conformance test suites, which usually consist of a large number of test cases. The successful execution of these test cases provides a reasonable assurance that the tested implementation follows all the rules defined by the protocol. The TTCN scripts were developed using Signaling Conformance Tests for CDMA 2000 Spread Spectrum Systems [17], since 3GPP2 conformance test cases for 1xEV-DV were not released. Specific tests were designed for testing the F-PDCHCF and the Message Integrity sublayer, which are new in the 1xEV-DV standard. The message flow between the Mobile and Base Station was verified by using the logging framework and MSC's in SDL. A sample MSC for a Mobile Originated 1xEV-DV call is shown in Figure 10 and 11.

In the absence of a Physical Layer implementation, the 1xEV-DV stack was integrated with a tool providing Layer 1 simulation. The tool provided a solution for designing, modeling and simulating analog and digital signals. All the features of physical layer such as modulation, demodulation, encoding, decoding etc. were implemented in the form of blocks. The PDU from the MAC sub layer was presented as an input to the simulator and the simulated data was visualized with the help of various charts.

5 Usage of Testing Tools

In order to provide for test automation, the testing tool SmarTEST was used to automate the functional and regression testing of the stack.

The framework in which SmarTEST can be used for test automation is depicted in Figure 12. Scripts executed the logic of a test case. The execution of a test case

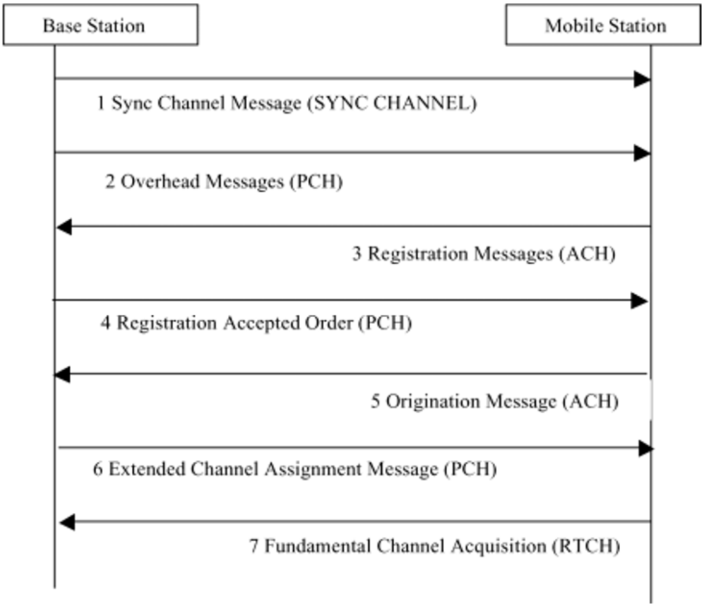
Base Station
Mobile Station
1 Sync Channel Message (SYNC CHANNEL)
2 Overhead Messages (PCH)
3 Registration Messages (ACH)
4 Registration Accepted Order (PCH)
5 Origination Message (ACH)
6 Extended Channel Assignment Message (PCH)
7 Fundamental Channel Acquisition (RTCH)

Fig. 10. MSC for a Mobile Originated Call–1

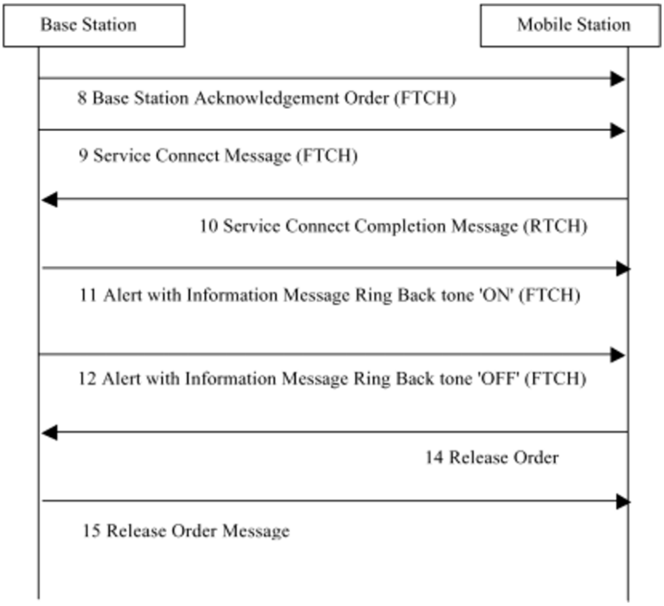
Base Station
Mobile Station
8 Base Station Acknowledgement Order (FTCH)
9 Service Connect Message (FTCH)
10 Service Connect Completion Message (RTCH)
11 Alert with Information Message Ring Back tone 'ON' (FTCH)
12 Alert with Information Message Ring Back tone 'OFF' (FTCH)
14 Release Order
15 Release Order Message

Fig. 11. MSC for a Mobile Originated Call–2

resulted in the generation of Mobile Originated (MO) or Mobile Terminated (MT) calls as desired. Each test case contained the logic required to initiate MO or MT calls with different scenarios. Based on the results of call execution, the script decided the

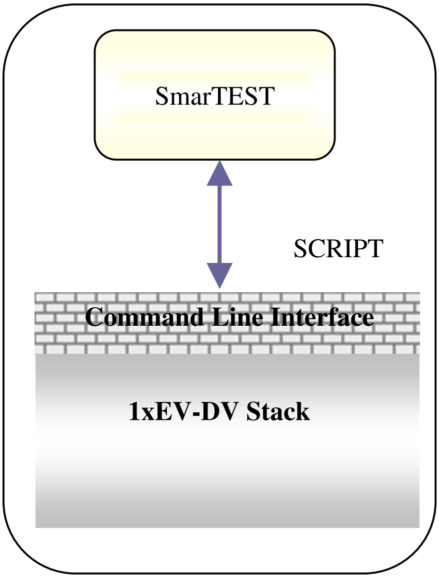

Fig. 12. Test Automation Framework

success or failure of the test. The necessary parameters required for executing the calls were set-up using the SmarTEST GUI, and routed to the stack with the help of scripts. The test cases are managed and automatically executed through the SmarTEST engine. APIs provided on top of the 1xEV-DV stack, enabled the test cases to be driven through its Command Line Interface. Scripts have the ability to execute the logic of the test case and decide the status of success or failure of the test case. The system test cases were managed and automatically executed through the SmarTEST engine. APIs provided on top of the 1xEV-DV stack, enabled the system test cases to be driven through its CLI.

6 Conclusion

Testing of mobile communication protocols is a very complex exercise. With time-to-market becoming a very important business success factor for vendors of mobile communication solutions, the testing activity becomes all the more critical. This challenge is magnified in the absence of hardware. The testing strategy needs to be planned and worked out carefully while the stack development is under progress. Usage of techniques such as complementary protocol stacks and logging can provide a low-cost and flexible solution in the testing of mobile communication protocols. SDL tools enable faster, portable, testable and maintainable protocol stack development.

References

1. C.S0001 Introduction to cdma2000 Standards for Spread Spectrum Systems, Release C 1.0, 3GPP2, May 2002

2. C.S0002 Physical Layer Standard for cdma2000 Spread Spectrum Systems, Release C 1.0, 3GPP2, May 2002
3. C.S0003 Medium Access Control Standard for cdma2000 Spread Spectrum Systems, Release C 1.0, 3GPP2, May 2002
4. C.S0004 Signaling Link Access Control Standard for cdma2000 Spread Spectrum Systems, Release C 1.0, 3GPP2, May 2002
5. C.S0005 Upper Layer Signaling Standard for cdma2000 Spread Spectrum Systems, Release C 1.0, 3GPP2, May 2002
6. Specification and Description Language – 2000, Z.100, ITU-T, November 2000
7. ITU-T Recommendation X.292, OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – The Tree and Tabular Combined Notation (TTCN), 1998
8. ISO/IEC 9646-3, Information technology – Open systems interconnection – Conformance testing methodology and framework – Part 3: The Tree and Tabular Combined Notation (TTCN), 1998
9. Performance Analysis of Communication Systems formally specified in SDL - Martin Steppler, ACM Publication, 1998
10. Improving conformance and interoperability testing- Jame D. Kindrick, John A. Sauter, Robert S. Matthews, StandardView Vol. 4, No. 1, March/1996
11. On Testing Hierarchies for Protocols, Deepinder P. Sidhu, Senior Member, IEEE, Howard Motteler, A4ember, IEEE, and Raghu Vallurupalli, 590, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 1, NO. 5, OCTOBER 1993
12. Formal Specification Based Conformance Testing, Behqet Sarikaya, Concordia University, Gregor u. Bochmonn, Michel Maksud, Jean-Marc Serre, Universitd de Montreal, ACM Publication, 1986
13. A Test Design Methodology for Protocol Testing, B. Sarikaya. G. Bochmann. and E. Cerny, IEEE Transactions on Software Engineeritlg, Vol. 13, No. 5, March 1987.
14. Roger S. Pressman: Software Engineering – A practitioner's approach
15. Protocol testing: Review of methods and relevance for software testing, Gregor v. Bochmann and Alexandre Petrenko,109, ACM 1994
16. Case Study by Vocalis Group plc. & published by Telelogic
17. Signaling Conformance Tests for cdma2000 Spread Spectrum Systems,3GPP2 TSG-C, OCT 2001.