21

The Testing of BT's Intelligent Peripheral using abstract test suites from ETSI

N. Webster
Network Intelligence Engineering Centre, BT plc.
BT Laboratories, Martlesham Heath, IPSWICH, IP5 7RE, ENGLAND
Tel. 01473 645910; Email: nick@chaplin.bt.co.uk

Abstract

This paper describes the practical use of ETSI test suites in the testing of an Intelligent Peripheral, which is the Intelligent Network node responsible for in-band speech processing - playing announcements and processing any in-band responses.

Standardised test suites, which are increasingly available from ETSI, offer the potential to cut costs. This is achieved by reducing the need for expensive in-house test development whilst maintaining the testing quality through a set of approved tests which are independent of development.

This paper uses the experiences of testing the Intelligent Peripheral to examine whether these advantages are realised in practice. It describes the test suites, their test coverage, the modifications needed for test execution and the test results and metrics for the testing cycle.

Keywords

Tree and tabular combined notation, abstract test suites, intelligent peripheral.

1 INTRODUCTION

The testing of network elements to ensure successful network operation is an expensive but necessary activity. The complexity of the specifications is continually increasing and with it the costs of testing. Improved specifications have allowed the development of standardised test suites which can help reduce testing costs and provide a powerful set of protocol testing criteria which are independent of development. This helps to ensure the quality of the network elements and their successful interworking in the network.

Increasingly, the documentation package for a protocol or network element includes both a functional specification and a test specification coupled to it. A number of conformance test specifications are available which are based on the ISO 9646 [1] methodology and are written in the Tree and Tabular Combined Notation (TTCN). The availability of commercial tools for editing, compiling and executing TTCN provided the opportunity for standardised test suites to be used in testing the Intelligent Peripheral (IP).

A. Cavalli et al. (eds.), *Protocol Test Systems VIII*

The IP in its network configuration is shown in Figure 1. The role of the IP is to perform in-band speech processing for the Intelligent Network (IN). The IP is controlled from the IN Service Control Point (SCP) using the INAP protocol. The SCP manages the service provided to the caller by first requesting the IP to play selected announcements and then checking the IPs response. The latter contains the speech processed in-band response from the caller encapsulated in an INAP message. The IN Signalling Switching Point (SSP) connects the caller to the SCP and IP using the C7 National User Part (NUP) and INAP respectively. The IP processes the speech channels of the C7 NUP.

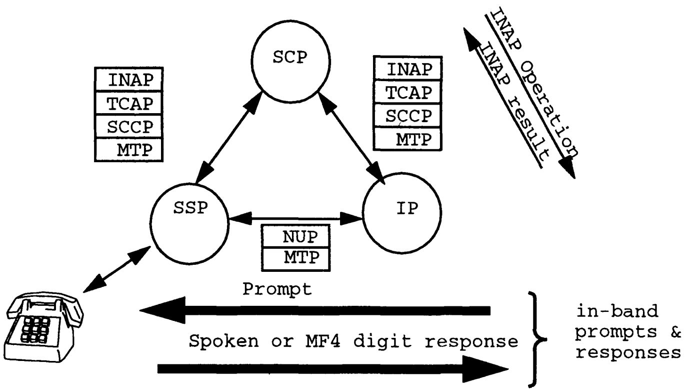

Figure 1: Overview of the Intelligent Peripheral

The project was split into development and test teams. The role of the development team was to add the Intelligent Network (IN) functionality of:-

- ITU blue book SCCP,
- ITU white book TCAP [8] and
- Core ETSI INAP [2]

to an existing speech application platform. The INAP software was developed in-house whilst the SCCP and TCAP software was procured from an external supplier.

The speech applications processing, C7 National User Part (NUP) and Message Transfer Part (MTP) had already been tested with proven in-house tests that will not be described here. The SCCP testing is still in progress and so is not described.

The challenge for the test team was to test the new protocols of the IP, so that the IP was suitable for integration testing with a real SCP and SSP. The use of test case generation tools as described in [9] was considered but not adopted for the following reasons:-

- The tools did not handle large systems,
- The tools were focused towards SDL coverage not data - a vital part of TCAP/INAP,
- "Formal" specifications in SDL were not available

Unfortunately, the tight timescales for the IP project gave little time to develop in-house test suites, so there was no other option but to use the ETSI Abstract Test Suites (ATS). These were at different stages of development:-

- SCCP ATS - stable and mature,
- TCAP ATS [6] - being extensively modified,
- INAP ATS [7] - stable and close to maturity.

An assessment was made of the available TTCN tools based on the criteria of cost, tool support, testing capability and the ability to support the protocols of the IP. We chose Telelogics ITEX editor, and the Siemen's K1197 tester [5]. Together these comprised:-

- protocol encoders and decoders for MTP,NUP,SCCP,TCAP and INAP,
- a TTCN IS compiler,
- a menu driven message building system and
- a proven test environment.

The processes of development and testing were kept independent within the project, so that the tests did not re-use any development software. Therefore, any assumptions made in the development did not percolate into the tests thus weakening them.

2 TESTING METHODOLOGY

This section describes the methodology for the creation of test suites by standardisation bodies and the correction and verification of the test suites for IP testing.

2.1 Test Suite Creation

A necessary pre-cursor to the writing of a protocol test suite is the specification of the protocol in textual and graphical notation. The specification should be complete and correct (although not necessarily formal) to enable developers and network managers to implement the protocol in real equipment.

The specification of the tests then follows, and may take several man-years to complete. The first and most important activity is the creation of a set of test purposes which explain the proposed test method, architecture and give a precise definition of the purpose of each test. The definition is in english text and cross references the protocol specification. The test purpose document is then used as the high-level test design document during the creation of tests in TTCN. Each test purpose is expanded into detailed TTCN test behaviour in the test suite. The writing of the test suite is a skilled manual activity. The test suite is usually the last document to be written and is often several hundred pages long.

2.2 Review and Transformation

Before the test suite could be used in the IP project it was analysed using ITEX and then reviewed for accuracy and completeness. Enhancements to the test suite to make it suitable for IP testing were identified.

The restrictions of the TTCN compiler supplied with the tester were compared against the range of TTCN constructs used in the test suite and the protocol layers supported on the tester. The un-supported constructs were assessed in order to define a strategy for the transformation of the test suite into a form which would execute on the target tester.

2.3 Test Verification

The transformed test suite was compiled on the K1197 for a more rigorous semantic analysis. Errors identified at this stage were removed before recompilation.

The suite was verified against both:-

- a simulated system under test (SUT) and
- the IP.

The simulated SUT was run on one C7 port of the tester and sent protocol messages to the C7 port running the TTCN, via an external looped-back connection. This approach conveniently allowed TTCN events to be checked. It would have been possible to have verified the tests against more comprehensive simulations on the K1197, but the timescales did not permit their development. The test suite was further exercised by running it against the IP implementation. At this stage, internal and external message traces were available to be used as verification tools.

The tester conveniently displayed execution traces of the test steps and constraints which was invaluable in test verification. The test suite parameters could be modified after compilation to configure the tests to match the configuration of the implementation under test.

The running test suite (in this case TCAP) was connected by an internal software interface to the run-time operating system. The interface was comprised of a user-defined *test manager* and *special file*. The *test manager* connected the automatic MTP layers to the *special file*, and could filter or respond to specific MTP messages. The *special file* was the interface between the TCAP test suite and the *test manager*, and converted TTCN events into protocol messages for the *test manager* and visi versa. The *special file* could be easily modified to add new protocol primitives to the test suite.

With the architecture as shown in Figure 2 the *test manager* could provide automatic SCCP, TCAP or INAP layers if so desired. However, since the IP only used connectionless SCCP, an automatic SCCP was not required and therefore the sent and received Protocol Data Units (PDUs) were defined explicitly in the test suite.

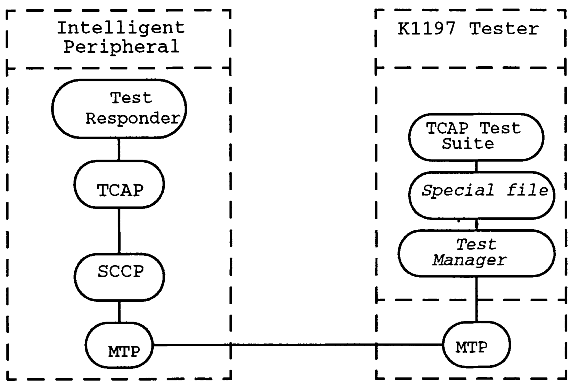

Figure 2: TCAP Test Architecture

3 TESTING THE TCAP LAYER

ETSI supplied early versions of their test suite [6], which comprised 140 test cases and proposed the use of a test responder as the upper tester, replacing the TCAP user during testing. This was a remote coordinated test method [1]. The purpose of the test responder was to send and receive primitives from TCAP, under the control of the test suite. Test management messages for the responder were encapsulated in normal TCAP messages.

3.1 Test Architecture

The TCAP test architecture is shown in Figure 2. Test management messages to the responder were defined as TTCN implicit sends, but this approach was under review by ETSI, and it was not clear precisely how the responder commands would be defined. Thus, owing to the demanding timescales for the IP development, it was decided to design our own responder and responder interface in SDL [3].

We decided to encapsulate the responder commands and responses in the user-defined data within TCAP un-structured dialogues, since the IP only made use of TCAP structured dialogues.This conveniently separated the facilities of TCAP under test from those used in testing. The main assumption made in testing TCAP was that the operation of structured and unstructured dialogues were independent.

The specific responder test sequences were defined for each test, encoded on a SUN workstation and then added to the ATS in TTCN test suite parameters.

Figure 3 shows a typical TTCN test case. Test steps PRE_ID and SEND_BEGIN_RI2 cause test management messages to be sent to the responder to reset it and then send TC-INVOKE and TC-BEGIN primitives to TCAP. This causes TCAP to send a BEGIN message, which is verified by the receive constraint BEGIN_R_I to give a test pass. The message sequence for the test including both TCAP and responder messages is shown in Figure 4.

Line	Behaviour	Constraint	Verdict	Comments
1	+PRE_ID			Reset Test Responder
2	+SEND_BEGIN_RI2			Send Test Responder command
3	START T_WAIT			Initialise Test Timer
4	?BEGIN	BEGIN	PASS	Receive BEGIN - test PASS
5	+POST_TEST			request responder test log
6	TIMEOUT		FAIL	Timer expiry: test FAIL
7	POST_TEST			request responder test log
8	?OTHERWISE		FAIL	Other receive event: test FAIL
9	POST_TEST			request responder test log

Figure 3: TCAP test case

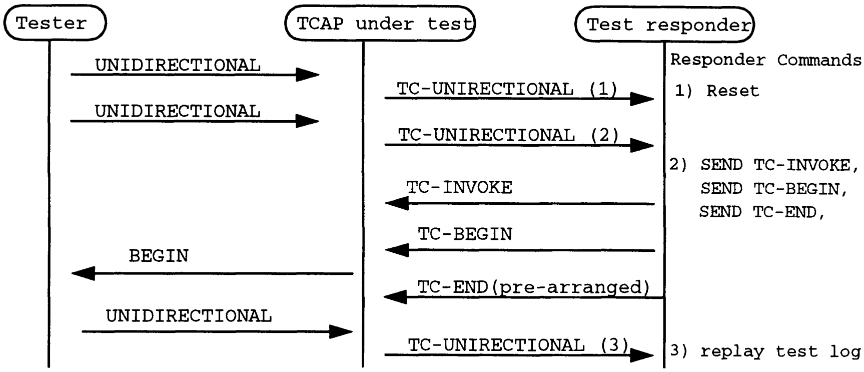

Figure 4: Message Sequence Chart for TCAP test case

The responder was enhanced to extract and store a TCAP dialogue identity from a received primitive and subsequently use it in primitive requests. The responder was also enhanced to improve the verification of the primitive interface to TCAP by recording the sequence of incoming and outgoing TCAP primitives. The recorded sequence could then be replayed at the request of the tester which is the function of test step POST_TEST in Figure 3.

3.2 TCAP ATS Enhancements

Naming Conventions

There was no easily comprehensible naming convention in the ATS. Several overlapping conventions were present which together with the scale of the ATS made it difficult to understand.

Problems

There were a number of unresolved issues within the ATS:-

- whether to use structured or unstructured dialogues for responder communication,
- the responder sequences were not all defined,
- where to place the test responder command within a TCAP message,
- some tests did not assign verdicts,
- some tests did not shut down active dialogues if the test failed,
- sometimes the active dialogues were not returned to the idle state,
- the ATS did not test white book facilities.

An important issue was the tests only tested the blue book facilities supported by white book TCAP. Thus we made sure that the tests we ran tested white book facilities of white book TCAP. We ensured that test verdicts were assigned for each test and the correct state of any dialogue was verified at the end of each test. The tests were made robust, so that if the tests failed, active dialogues were terminated.

The ATS used many different datatypes, which were transformed into three types: integer, boolean and octet string.

ASN.1 on the K1197

The K1197 supported the majority of the ISO 9646 IS constructs but did not support TTCN Abstract Syntax Notation 1 (ASN.1) [4] Protocol Data Unit (PDU) and constraint definitions. Unfortunately ASN.1 was used extensively throughout the test suite. Possible solutions were:-

- TTCN structured constraints,
- octet string constants,
- TTCN test suite operations.

Structured constraints were not feasible for the variable length ASN.1 fields which were present in TCAP. Structured constraints could have been used if the fields had been of fixed length.

Manually encoded octet string constants were viable but would have been of fixed length and thus would not support variable length fields. Additionally software maintenance would have been expensive.

TTCN test suite operations were selected. The approach needed manual coding of the test suite operations, but enabled software re-use in other constraints and allowed easy adaptation to other ASN.1 encoding formats.

3.3 TCAP test results

After running 140 tests, the problems discovered with the TCAP implementation were:-

- large TCAP PDUs and primitives were truncated,
- abnormal TCAP primitive sequences were incorrectly handled,
- the Dialogue Portion specification was easy to misinterpret,
- transactions remained active after termination.

These are described in detail below.

The development of the responder messages was initially slow since the first two problems prevented the transfer of responder messages through TCAP. TCAP was monitored internally to pinpoint the error and to help identify the problems we created protocol messages using the K1197 message definition facility.

When the responder log was replayed to the K1197 at the end of a test, a loss of messages occurred. The reason was that a TCAP invoke state machine was run for each unstructured dialogue and was still active when the same dialogue was re-used, subsequently blocking the sending of a unidirectional PDU. This highlighted how practical issues can impact unexpectedly on the ATS.

The specification of the TCAP dialogue portion was complex and took some time to understand before the TCAP dialogues could be successfully initiated. Once these problems were resolved the test suite was executed and in only a few days it was possible to run all the tests.

Overall the identified faults gave a clear perspective on the suppliers development process and showed that the lower interface of TCAP was well checked for normal and abnormal sequences. However the TCAP primitive interface had been checked only for a limited set of normal sequences.

The metrics for TCAP testing are shown in Table 1.

	TCAP	INAP	additional INAP
ETSI ATS used	yes	yes	no
Status of ETSI ATS	draft (not mature)	draft (mature)	-
TTCN compiler	yes	yes	yes
Test Suite execution	auto	semi-auto	semi-auto
Number of Tests	140 tests	50 tests	20 tests
ATS enhancement	60 days (40%)	15 days (60%)	5 days (70%)
Coding of test suite operations	20 days (13%)	5 days (20%)	2 days (10%)
Responder/Harness specification	10 days (6%)	-	-
Responder/Harness coding	15 days (10%)	-	-
Test Execution including fault removal	30 days (20%)	10 days	5 days
Final test execution	3 days	0.5 day	0.5 day
TOTAL man power	140 days	30 days	12 days
Man power per test	1.0 days	0.6days	0.6 days

Table 1: Metrics for Testing

4 TESTING THE INAP-SRF

The ETSI INAP test suite [7] contained over 50 IP tests and made extensive use of ASN.1. The ETSI test suite used the multi-party [1] remote test method. The INAP test suite also contained tests for the IN SSP which were not used as they were not relevant to the IP testing.

4.1 Test Architecture

The ATS was written to use the services of an automatic TCAP layer running under the ATS, which unfortunately was not available on the K1197. The two solutions were:-

- write a TCAP emulation with an Abstract Service Primitive (ASP) interface,
- transform the TCAP TTCN ASPs to TTCN PDUs (as in the TCAP ATS).

Siemens offered to provide a TCAP emulation but it was decided to transform the ATS since this approach would focus all of the test control within the test suite.

The K1197 emulated the NUP and INAP protocols and a digital telephone was connected to the NUP link to listen to in-band recorded announcements and to make in-band voice or keyed digit responses. The test architecture is shown in Figure 5.

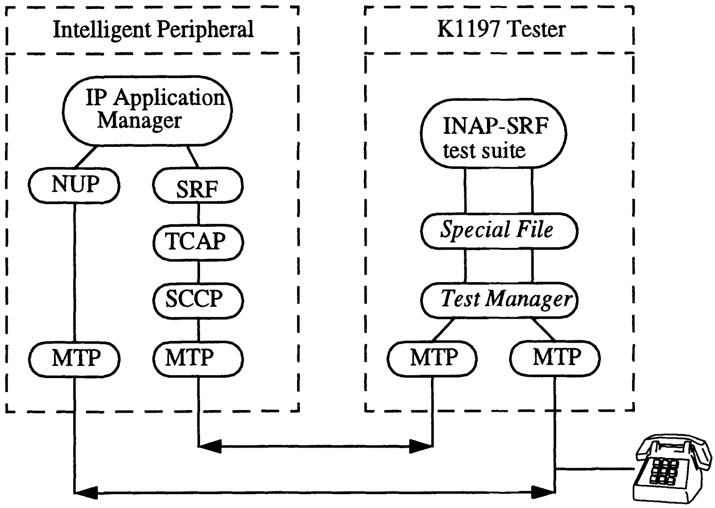

Figure 5: Test Architecture for the SRF

A typical message sequence for an INAP-SRF test is shown in Figure 6. The test begins with a NUP Initial and Final Address Message (IFAM) to the IP, which then sends an INAP Assist Request Instruction (ARI) in a TCAP BEGIN. The INAP Play Announcement is then sent in a TCAP CONTINUE, which replies with an INAP specialised resource report in a TCAP CONTINUE. The IP also sends the Address Complete Message (ACM) and Answer message (ANS) and the call clears with a release (REL).

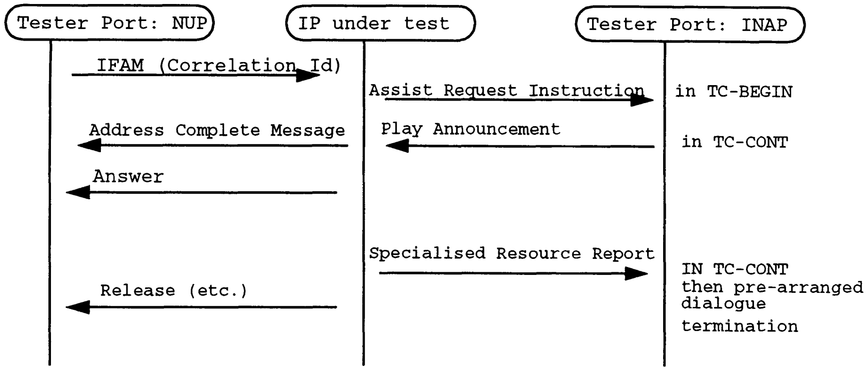

Figure 6: Message Flow for INAP test

4.2 INAP-SRF ATS enhancement

The INAP test suite used a generic bearer call control signalling interface. Thus the ATS was enhanced for the specific SSP-IP NUP interface of the IP by the addition of the associated TTCN PDUs and constraints.

The review of the test suite found a number of errors:-

- verdicts were not always assigned,
- some of the message flows were incorrect,
- call disconnection in some tests was incorrect,
- the use of *ElementarymessageID* was wrong in some tests.

The transformation of the ASP constraints to PDU constraints was straight forward, and the TCAP ATS PDU definitions were re-used in the INAP ATS. The TTCN ASP fields which defined the INAP operations in ASN.1 were each replaced by a test suite operation. It was written so as to use any test suite parameters, constants or variables in precisely the same form as in the ASP definition. Thus the operation of the transformed and original ATS were identical.

The TTCN datatypes were converted into integer, boolean or octet strings.

4.3 Supplementary INAP-SRF Tests

20 new tests were written in TTCN to test:-

- BT's IN extensions and
- call disconnection procedures.

The metrics for these tests are shown in Table 1.

4.4 INAP-SRF Test Results

The metrics for INAP testing are shown in Table 1.

The testing showed the benefits of test suite parameters, which could be modified interactively to match the values supported by the IP. The response of the IP system to un-supported values was not tested explicitly within the ETSI ATS, but was an important feature which was easily checked.

5 DISCUSSION OF THE USE OF ABSTRACT TEST SUITES

Table 1 shows a summary of the testing for the different protocols. It shows the time taken in man-days for the different stages of the testing. The percentage time for each activity is shown in parenthesis.

The INAP ATS has a metric of 0.6 days per test. The 20 supplementary INAP tests have a similar metric of 0.6 days, but this includes the creation of new tests which is more efficient. The TCAP metric of 1.0 days was twice that of INAP, because of the time to correct the draft test suite (60 days) and to clear faults in the IP (30 days). If the ATS had been more mature, less time would have been taken and the overall metric would have improved. The metric could have been further improved if the tester had supported TTCN ASN.1.

TCAP has been successfully tested using an in-house version of the ETSI test suite. A large number of tests have been run and problems highlighted. An equivalent in-house ATS would probably have taken several man-years to develop. The main limitation with the test suite was that only normal sequences were sent by the TCAP test responder. The use of abnormal sequences and large ASPs (including maximum length) needs to be considered to enhance the test suite. This would only be possible during TCAP testing and not during testing of the TCAP user.

The ability to modify the test suite to match the implementation under test is important. This includes the selection of test suite parameters and changes to the TTCN.

5.1 K1197 Tester

The protocol encoders and decoders were invaluable in decoding protocol messages and in the creation of messages during test suite verification. Test suite parameters could be changed interactively and the tests re-run without recompilation.

The support and in-depth technical help from the supplier was good and if any problems were encountered a quick and accurate response was forthcoming. The importance of good support in using new technology cannot be overstated.

5.2 TTCN

The notation of TTCN is powerful and comprehensive, however without the use of naming conventions and design information, the ATS' are difficult to understand and the impact of proposed changes unnecessarily time-consuming to assess. Large TTCN test suites are rather intimidating.

Test engineers would adopt TTCN more readily if there was a simple guide to its use. TTCN is very difficult to review by those who are not fluent in it. Training in TTCN is essential.

The use of a TTCN editor was invaluable to modify and check the TTCN. However because of TTCNs spread-out nature it was sometimes easier to modify the TTCN using a text editor.

6 CONCLUSION

Abstract test suites from ETSI are being used successfully to test an Intelligent Peripheral. They have provided large numbers of effective tests, without the need for expensive in-house test development and have helped to reduce the costs of testing. The use of standardised tests ensured independence from the development process and so BTs Network Interconnection Responsible Officer was able to approve the tests.

The ETSI protocol documentation, encompassing functional and test specifications, was a sound basis for protocol testing. Unfortunately, for state-of-the-art protocols the availability of the test specification was somewhat delayed behind the specifications. Thus draft test suites were used for TCAP and INAP. It was time-consuming to resolve the TCAP problems. Some practical issues had not been considered, which we felt necessary to improve, such as the emphasis of the white book TCAP tests being to test blue book facilities of white book TCAP. It was important that the functionality to be used by the IP was actually tested. The INAP ATS was much easier to update.

The abstract test suites were modified to correct errors and updated to support the specific implementation and architecture under test. The TTCN tools were not as well developed as equivalent specification tools and need enhancement to make them more usable, so that large and complex test suites can be handled effectively.

The execution of the abstract test suite was achieved successfully, using practical techniques made possible by the flexibility and power of the K1197. The absence of ASN.1 support in the TTCN compiler caused some difficulties for the TCAP and INAP testing but was overcome by the use of TTCN test suite operations.

The testing of the IP using ETSI test suites was successful and was probably more efficient than equivalent in-house test suite developments. However at the moment, standardised test suites are not the solution for testing all network elements. For each network element, a fresh assessment must be made of the protocols and architecture to be tested, the maturity and applicability of the test suites and the facilities of the testers which will run the test suites.

In the future, the standardised test suites are likely to be more widely adopted as the number of available test suites increase and the quality of the TTCN support tools improves.

7 REFERENCES

[1] ISO/IEC: *Conformance Testing Methodology and Framework- Part 3: The Tree and Tabular Combined Notation.* ISO 9646, ISO/IEC, 1992.

[2] ETSI: *Core Intelligent Network Application Protocol Specification.* PrETS 300, 374-1,1994.

[3] CCITT: *Specification and Description Language (SDL).* CCITT Recommendations Z.100, CCITT/ITU 1988.

[4] CCITT: *Abstract Syntax Notation No 1 (ASN.1) Encoding Rules.* Recommendations X.209 CCITT/ITU 1989.

[5] Siemens AG: *Manuals for the K1197* Tester.

[6] ETSI: *TCAP Abstract Test Suite.* Draft version from ETSI SPS2, January 1994.

[7] ETSI: *INAP Abstract Test Suite.* Draft version from ETSI SPS3, July 1994.

[8] CCITT: *Transaction Capabilities.* CCITT Recommendations Q.771-Q.775, CCITT/ITU 1992.

[9] L. Boullier et Al.: Evaluation of some test generation tools on a real protocol example. IWPTS VII - IFIP WG6.1 Proceedings Nov. 1994.

8 ACKNOWLEDGEMENTS

The author wishes to thank his colleagues Mr. P.McGuinness and Mr. M.Duffell for their help with TCAP, and also Hans Neuendorf of Siemens for guidance and technical help in the use of the K1197.

9 BIOGRAPHY

Nick Webster graduated in physics from the University of Bristol in 1973 and has spent most of his working life with British Telecom. He began his career in semi-conductor research, then changed his work-area, and for the last 12 years has been responsible for the design and testing of protocol systems including System X, ISDN and IN. He has concentrated on the application of modern testing technology, such as TTCN, to telecommunications systems.