Formal Based Testing of ATM Signalling

Ina Schieferdecker and Axel Rennoch
GMD Fokus
tel: +49 30 254 99 {241,239}
e-mail: {ina,axr}@fokus.gmd.de

May 3, 1996

Abstract

The paper describes an ongoing work on setting up a formal-based testing methodology that covers the test suite generation from a protocol specification, the consistency check between protocol specification and test suite, the implementation of test suites, and the test result analysis. We are going to establish a tool set that allows us to automate the testing process as far as possible. The working example are ATM signalling protocols in local area networks.

1 Introduction

Communication protocols are internationally standardised in order to ensure the inter-operability of different protocol implementations that are running in heterogeneous environments. The protocol specification describes the desired functional behaviour. Two protocol implementations may inter-operate with each other provided that the functional behaviour is implemented correctly and that the timed behaviour and the performance characteristics are harmonised with each other.

Conformance testing checks the equivalence between an implementation and its specification. Due to theoretical, technical, and economical impossibilities for exhaustive tests, conformance tests check only a restricted subset of the protocol functionalities of an implementation. Even if exhaustive tests for conformance would be possible, implementations may be nevertheless incapable of inter-operating, what is mainly due to implementation aspects, which are beyond the scope of conformance testing. In particular, timing and performance issues of communication protocols are not addressed by conformance testing. Inter-operability problems that are caused by wrong timer settings or different speeds of protocol implementations, can be detected only if the testing methodology is enhanced with timing and performance aspects.

On the other hand, time and performance issues of communication protocols are as important as their functional correctness in high-speed networks and, in particular, in ATM networks . Multimedia communication that are running over ATM networks and are using the capabilities of different transmission services such as constant or variable bit rate service, impose new quality-of-service and performance requirements on network system components and on protocols in particular. Data transfer in high-speed networks must not only be fast, but must also be realized with guaranteed delivery.

For example, audio and video streams must be delivered with a low error-rate and within a well-defined delay and delay jitter. Performance testing is an extension of conformance testing with the ability to check performance requirements of a protocol implementation in normal usage or overload situations.

2 Basics of Conformance and Performance Testing

ISO/IEC and ITU-T investigated conformance testing of communication protocols, during the last decade. The results of this work is the international ISO/IEC standard 9646 "OSI Conformance Testing Methodology and Framework" (CTMF [1]). It describes a quite general testing approach for OSI communication protocols, defines the notions of conformance testing such as test architectures and test methods, and the "Tree and Tabular Combined Notation" (TTCN) for the description of test suites. CTMF offers a basis for protocol testing in industry and gains wider acceptance and dissemination in industry.

Conformance testing of communication protocols is defined as a black box testing, i.e. the internal structure of the protocol implementation is unknown to the tester. The implementation is tested by issuing external stimuli to the implementation, by awaiting the reaction of the implementation and by comparing these reactions with the expected ones. Stimuli and reactions are test events, whose ordering is described in sequences of test events, the so called test cases. Based on the reactions of the implementation, the tester can decide on the conformance of the implementation.

Due to the manual determination of test purposes and manual specification of test cases, the quality of a test in terms of fault coverage, minimality, and confidence on the testing results depends crucially on the skills of the test designers. In addition, a test suite may contain errors and may be inconsistent with the protocol specification, so that an implementation may pass a test although its behaviour contradicts the specification.

In order to improve the strength of conformance testing, there is the ISO/IEC and ITU-T activity on "Formal Methods in Conformance Testing" (FMCT [3]). It investigates the formal description of test purposes, the automated generation of test cases from the formal protocol specification, and the validation of test cases against the formal protocol specification.

Performance tests check the timed behaviour and the performance characteristics of implementations. In particular, they are ought to check for the fulfilment of quality-of-service requirements. In opposite to protocol conformance testing, performance testing of communication protocols is not an issue of standardisation bodies. Although work is already in progress to extend formal description techniques (FDTs) with timing aspects, the derivation of appropriate tests and their description and execution is not considered yet. Only some papers such as [7, 24] consider performance testing.

Formal-based performance testing includes a method to generate test purposes from QoS requirements on the basis of the protocol specification, a method to derive appropriate test cases from test purposes in order to check time-critical paths in the protocol execution, a method to execute the test cases in real-time with the possibility to monitor or trace the reactions of the protocol, and a method to backtrack the results of test execution to the protocol specification in order to identify potentially faulty parts of the protocol implementation.

Performance testing shall comprise the identification of the time consumption of a single entity under different load situations, the identification of unsuitable timer settings or different performance of protocol implementations to be the cause of inter-operability problems, and the identification of the efficiency of a protocol implementation e.g. in terms of the number of unnecessarily retransmitted data packets

3 The Testing Approach

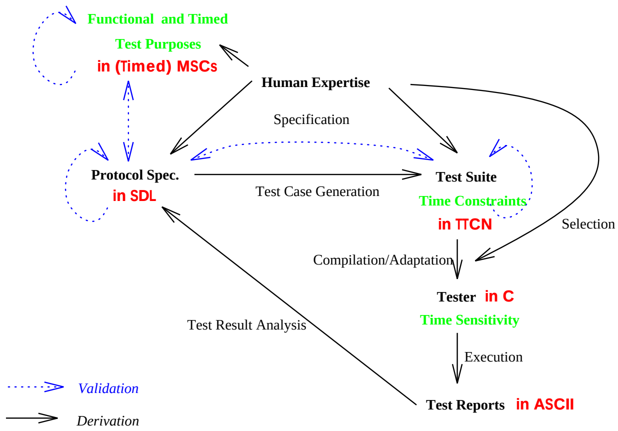

Figure 1: The Testing Approach

Our approach to performance and conformance testing is shown in Fig. 1 and consists of

1. The creation of a protocol specification.

   We use SDL — the Specification and Description Language [9] — for the protocol specification. SDL has been standardised by ISO and ITU-T (formerly CCITT) as a formal description technique for OSI communication protocols. SDL gains a wide acceptance in the telecommunication industry. Furthermore, SDL specifications of communication protocols are becoming more often an integral part of protocol standards.

2. The creation of functional and timed test purposes for the description of functional and QoS requirements[1].

   Classical MSCs — message sequence charts [10] — are used to represent functional test purposes for conformance testing, while timed MSCs are used for the description of QoS requirements, which define test purposes for performance testing.

3. The creation of a test suite, which is either designed by an expert or automatically derived from the protocol specification.

[1] We are dealing with time and performance-oriented QoS characteristics only, i.e. with delays, throughputs, and error rates.

Test suites are expressed in TTCN — the Tree and Tabular Combined Notation [2], that is the only standardised notation for the description of test suites. Test suites may be restricted to specific test cases in order to reduce the costs of testing in terms of the size of the executable tests, execution time, etc. In particular, automatically generated test suites contain are in general huge in their sizes. Expert knowledge is needed to reduce them by a designated abandonment of tests for unlikely and unessential behaviour.

4. The compilation of the test suite to executable code for a specific test equipment.

5. The test execution in a given runtime test environment.

6. The collection, analysis, and evaluation of test results. Databases should ease the access and the analysis of test results. Of particular interest are methods to identify potential faults in the implementation on the basis of the specification, whenever a test has failed.

4 The Availability of Tools

	Methods	Tool Support	Selected Tool
Validation of SDL	√	√	SDT
Validation of MSCs	√	√	SDT
Validation of MSCs versus SDL			SDT Validator
Validation of TTCN	√	+	ITEX
Validation of TTCN versus SDL			ITEX/SDT Simulator
Test Case Generation	+	−−	
Compilation of TTCN	+		
- test equipment dependent		+	HP TTCN
- test equipment independent		−	ITEX GCI (announced)

Figure 2: Methods and Selected Tool Support for Conformance Testing

Although conformance testing is under investigation for several years, we had to recognise that there are open issues and unsolved problems, in particular in the generation of test suites from formal specifications as well as in the compilation and execution of test suites. A consequence was to postpone the development of the performance testing approach and to go further with conformance testing.

There are well-established tool environments for the creation, validation, and maintenance of SDL specifications and of MSCs, that also support the consistency check between SDL and MSCs. Available TTCN tools are not that matured as SDL tools. In particular, SDT/ITEX [17, 8] is the only tool environment that is able to check the consistency between SDL and TTCN. In addition, SDT/ITEX offers means to derive TTCN test cases from SDL specifications by a semi-automated step-by-step simulation of an SDL specification (TTCNLink).

Our experiences with the current version 3.02 of the SDT/ITEX tool set differ: SDT is one of the leading tool environments for designing and prototyping distributed systems with SDL specifications.

It has valuable components like the simulator, validator, or coverage viewer. Unfortunately, the actual version does not support ASN.1 data types, i.e. optional parameters in signals are not allowed. ITEX is on an earlier stage of development and lacks in a number of bugs. A number of software performance reports have been submitted to the tool provider which often result in interim patch release distributions. In addition, ITEX is not well documented. On the other hand, ITEX offers a valuable feature for the cross simulation of a TTCN specification with an SDL specification. This feature helps to detect dynamic errors in the test suite behaviour in a comfortable way.

Besides that, only proprietary tools and prototypes are available for the automated derivation of TTCN test cases. For example, the British Telecom stopped the development of GTS and STG [5] and does not release them, since they are in a prototype form. Instead, they are using the semi-automated test generation capabilities of SDT/ITEX. Likewise, the tools TVEDA and TELVIS [5] are prototypes. They are intended for internal use at CNET (France Telecom) only and are not sold nor given to external organisations.

TTCN compilers such as the TTCN compiler for the Siemens K1297 or the TTCN compiler for HP 75000 are most often dedicated to a specific, vendor-dependent test environments. However, there is a tendency to define a vendor-independent interface — the Generic Code Interface — between a TTCN compiler and the test equipment, so that any GCI conformant TTCN code will be executable on a GCI conformant test equipment. A first implementation of this interface is announced as a new (beta-) release of the ITEX test suite compiler. Therefore, we delayed or work with the test suite compiler.

5 Generating Test Cases

Concluding from the considerations given above, the most vacant area in formal-based conformance testing is the automated generation of test cases from a protocol specification.

Test cases assume to adhere to certain test purposes. A test purpose may describe issues of control flow, data flow, signal flow, and execution time of a given protocol. Such a test purpose can be oriented towards a transition (a single step) in the communication protocol or towards a path (a sequence of steps) that covers a single protocol functionality, e.g. connection setup or data transfer. Although transition testing is a special case of path testing, it simplifies the generation of test cases.

Transition based test case generation methods assume that the protocol is given by an (extended) finite state machine (FSM). FSM based test case generation methods [15] are the methods of transition tours, distinguishing sequences, characterising sets, and of unique input/output sequences. They require certain properties of the FSM that are often not fulfilled by real communication protocols.

A quit different approach to test generation is taken by path based test cases that check a whole protocol functionality that the protocol is expected to realize, and not only a single transition. These methods rely on the model of (structured) labelled transition systems (LTS). Implementation relations identify the protocol conformant implementations. [21] developed the basic approach for deriving a tester for an LTS — the so called canonical tester that is based on the implementation relation **conf** [6]. Only recently, [20] defined the implementation relation **ios** for a dedicated subclass of LTS — the input-output transition systems. **Ios** reflects more adequately the needs of conformance testing than **conf** and other implementation relations. A prototype tool for a test generation algorithm based on the work in [19, 20] has been implemented. It is written in C++ and uses the data types and algorithms library LEDA, Release 3.3b from the University of Saarbruecken.

Currently, we extend it with a method to restrict the resulting tester to test purposes that are defined by MSCs. In path based testing taking into account test purposes, the number of test cases is by far smaller than those derived by transition based test generation methods. Although the coverage of the test cases is restricted to certain protocol functionalities, the quality in terms of size and coverage of the test suite (the set of all test cases) might be better than that of transition based test suites (presumed that the paths cover the main and critical functionalities of the protocol under study). Please notice that the development of meaningful test purposes, i.e. the identification of the protocol paths to be tested, requires an in-depth knowledge of the protocol under study. This is in opposite to transition based test case generation, which requires no additional knowledge but uses the raw protocol specification only.

6 The Example — ATM Signalling Protocols

We exemplarily apply formal-based testing to a real communication protocol in order to validate its strength and shortcomings. We selected ATM signalling protocols [18] due to their importance in ATM networks for the administration of virtual connections. The goals of testing ATM signalling are twofold:

- Testing pure ATM signalling allows us to check that a virtual connection is set up in real-time. That requires not only the correct behaviour of signalling mechanisms, but also the correct behaviour of timers.
- ATM signalling supports the negotiation of different traffic classes such as VBR and CBR with varying QoS characteristics. It has to be checked whether the obtained QoS coincides with the contracted ones. Thus, the established AAL connections are a good candidate to apply performance testing.

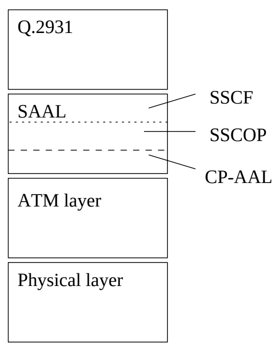

Figure 3: ATM Signalling Protocol Stack

ATM signalling is defined by ITU-T in the standard Q.2931 [14] (both by an SDL specification and English text) and by the ATM Forum in the UNI 3.1 [4] specification (by English text). Although, both specifications describe not completely identical protocol functionalities, they claim to coincide in the common parts. ATM signalling uses a protocol stack, which is represented in Fig. 3. Q. 2931 resides on top of the signalling AAL (SAAL). The purpose of SAAL is to provide a reliable communication between Q.2931 entities over the ATM layer. SAAL is divided into two parts — the common part (CP-AAL) [11] — and the service specific part. The service specific part consists of two sub-layers — the Service Specific Coordination Function (SSCF) [13] and the Service Specific Connection Oriented Protocol (SSCOP) [12].

First conformance test suites exist already for UNI 3.1. [23] and [22] contain abstract test suites for the UNI 3.1 signalling for the user-side on general outgoing and incoming call tests and on general call clearing and general error tests, respectively.

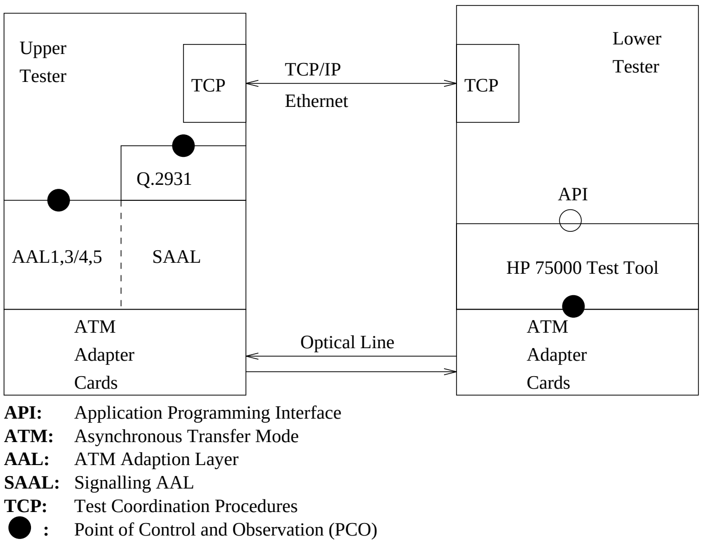

Figure 4: Test Configuration for AAL and Signalling Testing

The testing scenario is given in Fig. 4. Since we test pure ATM signalling and the inter-working between ATM signalling and AAL, two PCOs at the upper tester side must be available. The signalling and AAL protocols on the lower tester side are emulated by the HP 75000 testing tool. It offers a programmable interface, which is used to adapt the executable TTCN code of the ITEX compiler, as well as an HP specific TTCN compiler, which generates directly executable code.

The project on formal-based testing cooperates closely with the ATC — the ATM Test and Conformance Center at GMD Fokus. The main activities of the ATC are inter-operability tests of multi-vendor ATM equipment, performance tests of ATM adapter cards, routers, and ATM switches up to the TCP/IP layer, and conformance tests according to the ATM Forum UNI 3.0/3.1 and the ITU-T Q.2931 recommendations. The ATM equipment at the ATC includes ATM switches such as Fore Systems ASX200/ASX100, Siemens HicomXpress switches, Cisco A100, and Synoptics 10124S; ATM adapter cards such as Fore Systems SBA100/SBA200, Efficient Networks ENI-155s-MF, and SBus Adapter cards; and ATM routers such as Cisco 7000 router and Fore Systems LAX-20 router. The test systems that are currently used at the ATC are HP 75000 Broadband Series Test System and the Tekelec Airtronic Chameleon Open.

7 First Results

First results were achieved in course of creating machine processable protocol specification and test suites, of semi-automatically deriving test cases, and of compiling and executing some tests.

The SDL specification of ATM signalling by ITU-T given in Q.2931 is not a completely formalised specification containing even informal descriptions. We enhanced it significantly to be processable by SDT. We found not only inconsistencies in the specification (especially the informal descriptions are imprecise), but also open issues in the protocol mechanisms (e.g. Restart procedure).

Likewise, the ATM Forum test suites were found to be erroneous and incomplete. However, the most interesting results were results on the inconsistency between the SDL specification and the TTCN test suite. Indeed, we expected inconsistencies because the specification and the test suites are developed by different people (and by different organisations). The results on inconsistencies were obtained only after extending the SDL specification, since the specification and the test suite were initially not comparable. We had to introduce an adaptation process SAAL in order to specify the relation between SAAL service access point primitives (used in Q.2931) and the pure signalling message exchange (used in the ATM Forum test suites). Also, an application process AP has been specified to serve the upper boundary interface of Q.2931. Last but not least, we took into account the slight differences between Q.2931 (defining public UNI) and the UNI 3.1 based test cases (addressing private UNI), e.g. the optional (not mandatory) transmission of a CALL PROCEEDING message from the network to the user side. In such cases, we followed the intention to modify our SDL system in order to validate the original ATM Forum test suites.

The experiment of cross-simulating the extended Q.2931 SDL specification with ATM Forum test suites comprises the successful simulation of 55 test cases addressing general outgoing and incoming call tests [22] and general call clearing tests [23]. The latter document also includes test cases with message encoding errors (repeated information elements, wrong message length fields etc.), which we do not run since the corresponding error detection procedure are not part of the SDL specification of ITU and requires a set of modified SDL specifications due to the missing support of ASN.1 within SDT.

We discovered 62 defects in the ATM-Forum tests, a lot of typos, but also inconsistencies with TTCN (syntax) and wrong or missing type declarations. In the behaviour part we found missing initialisation of test suite variables, wrong indentations of test statements, incompatible message parameter values (call reference flag parameter) and even wrong direction of message exchanges.

Figure 5 has been generated with the coverage viewer tool of the SDT validator. It illustrates the validation coverage of transitions in the SDL system due to the cross-simulation results (stored in MSCs) obtained from our experiments with the ATM test cases. For simplification, the figure is reduced to the SDL system states which include transitions executed at least once (it is also possible to

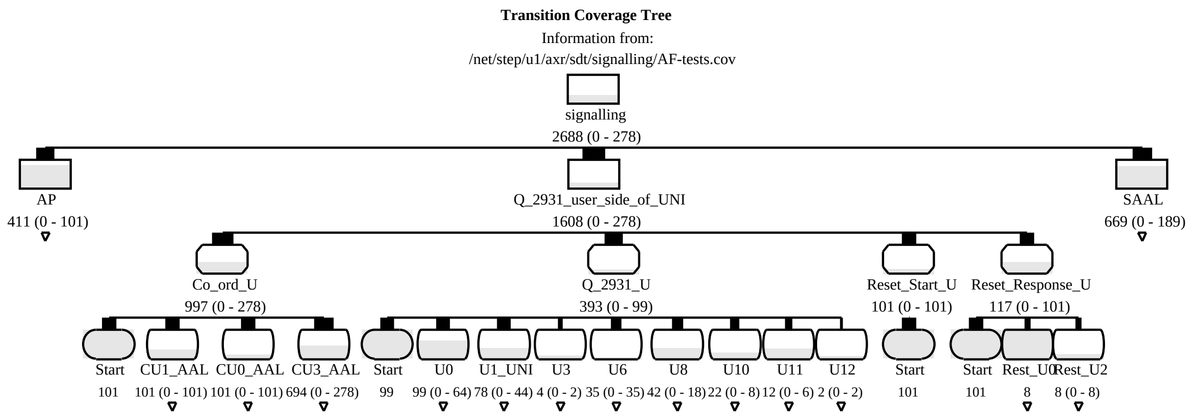

Figure 5: Coverage of ATM Forum Test Suites for Q.2931

display the information for each individual transition). The number of executed transitions associated with the system state is presented below the node, a range is also shown indicating the number of times the least and most executed transition within the system state has been executed. In each node, the number of transitions that have been executed in relation to the total number of transitions is also displayed (filling level). From the figure we can learn which system state have not been touched by the test cases (e.g. CU2_AAL or all states within process Reset_Start_U).

From this coverage analysis it is apparent that the ATM Forum test suites do not cover all protocol functionalities that should be conformance tested. Here, we expect useful results from an automated test case generation by applying our test derivation tool.

Finally, first automatically generated executable tests from the corrected test suite were run successfully and showed that the formal-based testing approach leads to results of practical relevance.

References

[1] ISO/IEC JTC 1/SC 21. Information technology – open systems interconnection – conformance testing methodology and framework. International Multipart Standard 9646, 1992.

[2] ISO/IEC JTC 1/SC 21. *IS 9646 Part3: The Tree and Tabular Combined Notation (TTCN)*, 1992.

[3] ISO/IEC JTC 1/SC 21. Information technology – open systems interconnection – formal methods in conformance testing. Working Draft, 1995.

[4] ATM Forum. *ATM User-Network Interface Specification, Version 3.1*, September 1994.

[5] L. Boullier, B. Kelly, M. Phalippou, A. Rouger, and N. Webster. Evaluation of some test generation tools on a real protocol example. In Mizuno et al. [16], pages 135–148.

[6] E. Brinksma. A theory for the derivation of tests. In S. Aggarwal and K. Sabnani, editors, *8th Intern. Symposium on Protocol Specification, Testing and Verification, Atlantic City, New Jersey, U.S.A.*, IFIP Transactions, pages 63–74. North-Holland, 1988.

[7] D.D. Clark. Protocol design and performance. Tutorial at IEEE INFOCOM'95, April 1995.

[8] ITEX. Manuals version 3.02. Telelogic AB, 1996.

[9] ITU-T. Recommendation z.100: Specification and description language (sdl), 1992. Blue Book, Vol. X.

[10] ITU-T. *Recommendation Z.120: Message Sequence Charts (MSC)*. International Telecommunication Union ITU-T, 1993.

[11] ITU-T Document. *TD-XVIII/10: AAL Type 5, Draft Recommendation Text for Section 6 of I.363 (Note)*, January 1993.

[12] ITU-T Recommendation. *Q. 2110: BISDN - ATM Adaptation Layer - Service Specific Connection Oriented Protocol (SSCOP).*

[13] ITU-T Recommendation. *Q. 2130: BISDN Signalling ATM Adaptation Layer - Service Specific Coordination Function for Support of Signalling at User-to-Network Interfaces (SSCF at UNI).*

[14] ITU-T Recommendation. *Q. 2931: Broadband Integrated Services Digital Network (B-ISDN). Digital Subscriber Signalling System No. 2 (DSS 2). User-Network Interface (UNI) Layer 3 Specification for Basic Call/Connection Control*, February 1995.

[15] R. Lai and W. Leung. Industrial and academic protocol testing: the gap and the means of convergence. *Computer Networks and ISDN Systems*, 27:537–547, 1995.

[16] T. Mizuno, T. Higashino, and N. Shiratori, editors. *7th IFIP WG6.1 Intern. Workshop on Protocol Test Systems, Tokyo, Japan*, 1994.

[17] SDT. Manuals version 3.02. Telelogic AB, 1996.

[18] B. Stiller. A survey of UNI signalling systems and protocols for ATM networks. *ACM Computer Communications Review*, 25(2):21–33, 1995.

[19] J. Tretmans. Testing labelled transition systems with inputs and outputs. In *Participant's Proc. of IWPTS'95, Evry, France*, pages 461–476, 1995.

[20] J. Tretmans. Implementation relations for transition systems specification. COST 247 Meeting, Madrid, Spain, 1996.

[21] C.D. Wezeman. The CO-OP method for compositional derivation of conformance testers. In E. Brinksma, G. Scollo, and C.A. Vissers, editors, *9th Intern. Symposium on Protocol Specification, Testing and Verification, Enschede, The Netherlands*, IFIP Transactions, pages 145–160. North-Holland, 1989.

[22] S. Yoo and L. Collica (eds.). *ATM Forum/95-0584R1: Abstract Test Suite for the UNI 3.1 Signalling for the User Side: General Outgoing Call and Incoming Call Tests*, August 1995.

[23] S. Yoo and L. Collica (eds.). *ATM Forum/95-0858: Abstract Test Suite for the UNI 3.1 Signalling for the User Side: General Clearing and Error (general) Test Section*, August 1995.

[24] S. Zhang and S.T.Chanson. On transition time testing based on extended finite state machines. In Mizuno et al. [16], pages 75–89.