Comparison of an Automatically Generated and a Manually Specified Abstract Test Suite for the B-ISDN Protocol SSCOP

Jens Grabowski[†], **Rudolf Scheurer**[*], **Dieter Hogrefe**[†]

April 1997
Bericht A-97-07

[†]University of Lübeck, Institute for Telematics, Ratzeburger Allee 160, D-23552 Lübeck, Germany, {jens, hogrefe}@itm.mu-luebeck.de
[*]University of Berne, Institute for Informatics and Applied Mathematics, Neubrückstr. 10, CH-3012 Bern, Switzerland, scheurer@iam.unibe.ch

Zusammenfassung

Die Testfallgenerierungsmethode SAMSTAG (SDL and MSC based test case generation) konnte erfolgreich auf das Protokoll SSCOP (Service Specific Connection Oriented Protocol) aus der B-ISDN ATM Anpassungsschicht angewendet werden. Die erzeugte Testreihe erreicht ein fast 70-prozentige Überdeckung der zuvor identifizierten Testzwecke. Parallel zu unserer Arbeit wurde beim ATM Forum ebenfalls eine Testreihe für SSCOP erstellt. Im Gegensatz zu der von SAMSTAG automatisch erzeugten Test Suite wurde diese von Hand entwickelt. Der vorliegende Bericht beinhaltet einen Vergleich verschiedener Aspekte der beiden Testreihen.

Abstract

The test generation method SAMSTAG (SDL and MSC based test case generation) has successfully been applied to the B-ISDN ATM Adaption Layer protocol SSCOP (Service Specific Connection Oriented Protocol). A test suite has been generated covering approximately 70% of the test purposes identified. In parallel to our work the ATM Forum developed another test suite for SSCOP. Unlike the test suite generated automatically by the SAMSTAG tool, this one was specified manually. In this paper we present a comparison looking at various aspects of the two test suites.

CR Categories and Subject Descriptors:
C.2.0 [Computer-Communication Networks]: General; C.2.2 [Computer-Communication Networks]: Network Protocols; D.2.5 [Software Engineering]: Testing and Debugging; D.3.0 [Programming Languages]: General

General Terms:
Test Case Generation, SDL, MSC, TTCN, B-ISDN ATM, SSCOP

Comparison of an Automatically Generated and a Manually Specified Abstract Test Suite for the B-ISDN Protocol SSCOP

Jens Grabowski[†], Rudolf Scheurer[*], Dieter Hogrefe[†],

[†]University of Lübeck, Institute for Telematics, Ratzeburger Allee 160, D-23552 Lübeck, Germany, {jens, hogrefe}@itm.mu-luebeck.de
[*]University of Berne, Institute for Informatics and Applied Mathematics, Neubrückstr. 10, CH-3012 Bern, Switzerland, scheurer@iam.unibe.ch

Abstract
The test generation method SAMSTAG (SDL and MSC based test case generation) has successfully been applied to the B-ISDN ATM Adaption Layer protocol SSCOP (Service Specific Connection Oriented Protocol). In parallel to our work the ATM Forum developed another test suite for SSCOP. Unlike the test suite generated automatically by the SAMSTAG tool, this one was specified manually. In this paper we present a comparison looking at various aspects of the two test suites.

Keywords
SDL, MSC, TTCN, conformance testing, test case generation, abstract test suite, B-ISDN SSCOP

1. Introduction

SAMSTAG [3,4,12] is a method and a tool for the automatic generation of abstract test cases in TTCN [7, Part 3] format based on SDL [10] system specifications and MSC [11] test purposes. The method was intended to help saving time and money expenses on the one hand, and to ensure the consistence between specification and test cases on the other hand.

The SAMSTAG method is based on the bi-simulation of the SDL test system and a particular MSC test purpose. In a first step the SAMSTAG tool looks for a trace that corresponds to the given test purpose and includes pre- and postamble of the future test case. In a second step this nucleus is verified if it really gives evidence of exactly the property to be tested alone, and completed, e.g. adding inconclusive alternatives. In this manner the test purposes are validated in the same time.

Starting in 1995 we performed a case study based on the B-ISDN protocol SSCOP [8]. The choice of SSCOP was influenced by the interest of the ITU-T in a review of the SSCOP SDL specification and by the need for a test suite for SSCOP. The case study has shown that automatic test generation based on SDL specifications and MSC test purposes is feasible. For 68% of the MSC test purposes identified complete TTCN test cases have been generated automatically [5]. For another 9% a test case has been found but due to complexity and limitations of the SAMSTAG prototype verification could not be obtained.

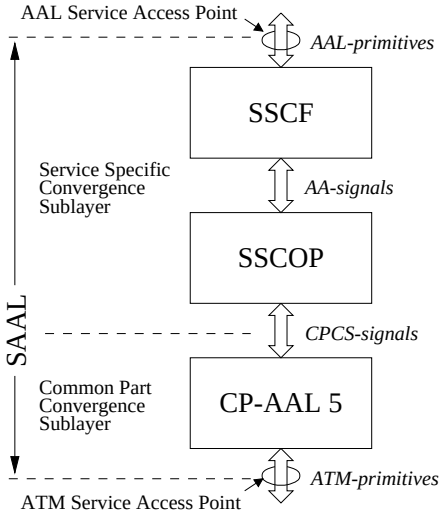


Figure 1. Structure of the Signalling ATM Adaption Layer (SAAL)

At the same time the ATM Forum[1] developed a test suite for SSCOP which has been approved in December 1996 [2]. This test suite was specified by hand. There are activities at ETSI (European Telecommunication Standards Institute) in this area too, but we will not consider it in our comparison since the work is based on a set of test purposes very similar to the ATM Forum test purpose set and further results, i.e. test cases, are not yet available.

We seize the opportunity to compare a test suite generated automatically with one specified by hand. The conclusions of this comparison shall lead to further improvements of the manual method of test case specification and, in the future, to its replacement by automatic methods like the SAMSTAG method.

The paper proceeds as follows: Section 2 introduces the SSCOP protocol, Section 3 presents the two test suites to be compared. The comparison of the different aspects of the test suites is done in Section 4. Summary and outlook are given in Section 5.

2. Service Specific Connection Oriented Protocol (SSCOP)

SSCOP [8] is used in the *B-ISDN ATM Adaption Layer* (AAL) [13,14]. The purpose of the AAL is to enhance the services provided by the ATM layer in order to meet the needs of different upper layer applications. One particular AAL type is the *signalling AAL* (SAAL). The SAAL provides communication functions for ATM entities which are responsible for signalling.

As shown in Figure 1, SSCOP can be used within the SAAL. The SAAL is divided into two sublayers, the *Common Part AAL* (CP-AAL) and the *Service Specific Convergence Sublayer* (SSCS). The SSCS comprises an SSCOP entity and a *Service Specific Coordination Function* (SSCF) [9]. The objective of SSCF is to map the services provided by the SSCOP protocol to different AAL interfaces. SSCF definitions for *User Network Interface* (UNI) and *Network Node Interface* (NNI) can be found in the ITU-T Recommendations Q.2130 and Q.2140.

[1]The ATM Forum is a non-profit international organisation accelerating cooperation on ATM technology.

2.1. Objective of SSCOP

SSCOP is a connection oriented protocol. Its main purpose is to provide the service of a generic reliable data transfer. In order to implement a reliable data transfer by using the unreliable service of the underlying ATM layer *selective retransmission* is used. This means, all data packets get a sequence number to preserve *sequence integrity*. An SSCOP entity indicates the loss of data packets by sending an USTAT PDU. Additionally, SSCOP entities exchange STAT PDUs periodically. This is done for keeping track of lost data packets in the special case of lost USTAT PDUs. Further characteristics of SSCOP are:

Flow Control. An SSCOP receiver is able to control the rate at which the peer is allowed to send data packets (windowing).

Error Reporting to Layer Management. SSCOP informs the layer management about specific errors such as protocol errors, resynchronization of the connection, or lost data packets.

Keep Connection Alive. SSCOP maintains connections even over periods in which no data transfer is performed. By using a set of timers a connection is partitioned into a *connection control phase*, an *active phase*, a *transient phase*, and an *idle phase*. The status of a connection is communicated between protocol entities by using POLL and STAT PDUs.

Local Data Retrieval. The SSCOP user is able to retrieve data packets which have not yet been released by the transmitting entity. Different access schemes are provided (full, partial, or selective retrieval).

Protocol Error Detection and Recovery. During operation SSCOP detects errors and triggers a recovery mechanism by exchanging ER and ERAK PDUs with the peer entity.

Connection Control. Connection control is related to establishment, release, and resynchronization of an SSCOP connection. A timer is set to protect against PDU loss during the connection control phase.

2.2. The SSCOP recommendation

The ITU-T Recommendation Q.2110 for SSCOP contains a section with a formal specification of SSCOP by means of SDL diagrams. This specification covers about 50 pages and defines the procedures of an SSCOP protocol entity. An SSCOP entity comprises 10 different states with approximately 300 state transitions. Each SSCOP state reflects a particular connection control state.

The SDL specification also contains several informal parts, e.g., textual references to queues and buffers, tables listing default parameter and field values to signals. In order to use the specification as input for the SAMSTAG tool these parts had to be formalised before [5].

3. Subject of comparison

The subject of the comparison are two test suites for the SSCOP protocol developed in 1996. The first one has been specified manually by experts of the ATM Forum, the second one represents the results of our case study in connection with SAMSTAG. We will refer to them using the terms *ATM Forum test suite* and *SAMSTAG test suite*.

3.1. ATM Forum test suite

The ATM Forum test suite is the result of the joint work of several experts contributing to the *Testing Technical Committee*. The documentation of the test suite states that "a testing matrix has been developed after study of the SSCOP specification and a selection of the appropriate test groups". Subsequently, the test cases were generated manually based on the selected test method (cf. Section 4.1.2).

3.2. SAMSTAG test suite

The test cases of the SAMSTAG test suite were generated applying the SAMSTAG tool [3–5,12]. Based on a SDL specification of SSCOP and test purposes specified by means of MSCs the tool automatically generates the test cases. Along with the test cases the appropriate data definitions are generated too. Further details on the generation procedure are given in [5].

4. Comparison

In this section we are going to compare the two test suites mentioned above. The comparison is split up into sections, each one covering a particular aspect of an abstract test suite.

4.1. Test methods

ISO/IEC IS 9646 [7] recommends different test methods to be used for protocol conformance testing. These methods mainly differ in the interfaces between tester processes and IUT, and the possibilities to stimulate and observe the IUT during the test.

4.1.1. SAMSTAG: distributed test method

The definition of the test method for the SAMSTAG case study was guided by the *distributed* test method of ISO/IEC IS 9646. The concrete test method as used is shown in Figure 2 (a).

There are an upper and a lower interface to the IUT. The upper interface is a *point of control and observation* (PCO) which is connected to an upper tester (UT). The UT exchanges AA-signals with the IUT. The lower interface is served by a lower tester (LT). The LT exchanges SSCOP PDUs with the IUT, i.e., as allowed by ISO/IEC IS 9646 we abstract from the encoding of PDUs within service primitives to be exchanged at the lower PCO. Generally UT and LT coordinate themselves by using *test coordination procedures* (TCPs). We do not model TCPs, because during test case implementation they follow indirectly from the sequence of AA-signals and SSCOP PDUs to be send to and received from the IUT during the test run.

Figure 2 (a) does not exactly correspond to the distributed test method as defined in ISO/IEC IS 9646. The PCO between IUT and UT is not standardised, i.e., it is not a service access point (SAP). Due to the non-existence of a standardised SAP between IUT and UT it may be more appropriate to use the *remote test method*[2], but currently, the SAMSTAG tool is not able to deal with the remote test method.

[2] A detailed discussion on appropriate test methods for ATM AAL conformance testing can be found in [15].

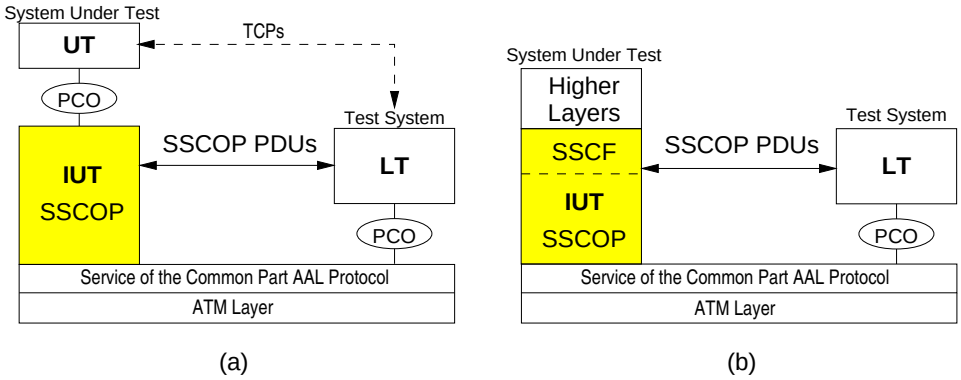


Figure 2. Test methods used: SAMSTAG (a) and ATM Forum (b)

4.1.2. ATM Forum: remote test method

The test suite of the ATM Forum is based on the *remote single layer embedded* (RSE) test method. In this test method the IUT is stimulated and observed at only one PCO. The *system under test* includes the SSCOP of the IUT, the SSCF and a higher layer, such as Q.2931. The LT communicates with the IUT via PDUs, as done in the SAMSTAG test method. The configuration as used by the ATM Forum test suite is shown in Figure 2 (b).

The most important difference of the remote test method comparing it to the SAMSTAG test method is the arrangement of the PCOs available. The missing PCO above the IUT implies that there needs to exist some other means to steer the IUT, in addition to the limited possibilities to achieve this from the point of view of the LT. These means are given by the use of the TTCN construct *implicit send events* (Section 4.4.2).

4.2. Structure and complexity of the test suites

In this section we compare the test suites with respect to their structure and complexity.

4.2.1. SAMSTAG test suite

The structure of the SSCOP test suite is shown in Figure 3. It is a tree structure and reflects the SSCOP functionality. The root of the tree represents the whole test suite. Nodes and leafs represent test groups and refer to functions or aspects of SSCOP functions. The test cases in one group should focus on a specific aspect to be tested. The numbers in round brackets following the leaves denote the number of test cases attached to this leaf. The test suite contains a total of 281 test purposes.

4.2.2. ATM Forum test suite

The structure of the ATM Forum test suite is mainly state oriented. The test suite structure comprises two groups, one for test purposes related to Protocol Capabilities (PC) and one for System Parameters (SP). The PC group contains 10 subgroups, one for each state of SSCOP, each of these groups having a subgroup for valid (expected PDU in correct state), invalid (syntactically incorrect PDU), and inopportune (valid PDU, but considered irrelevant for the particular state) signals. The structure of the test suite is depicted in Figure 4. The test suite comprises 317 test purposes.

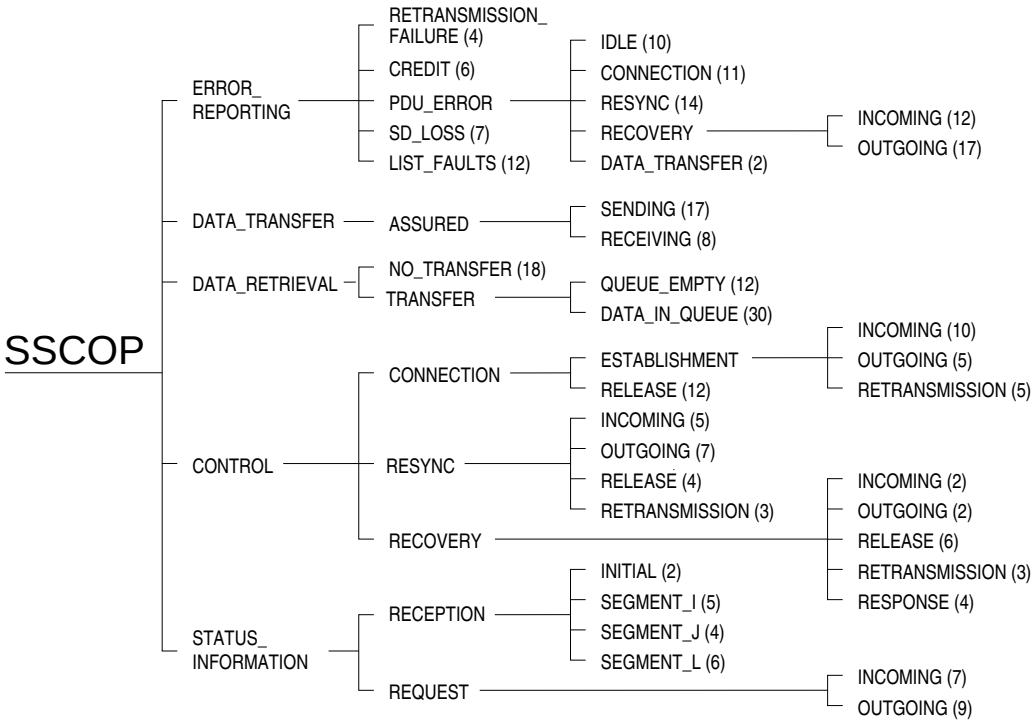


Figure 3. SAMSTAG test suite structure

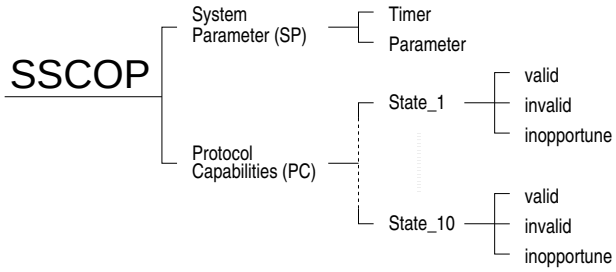


Figure 4. ATM Forum test suite structure

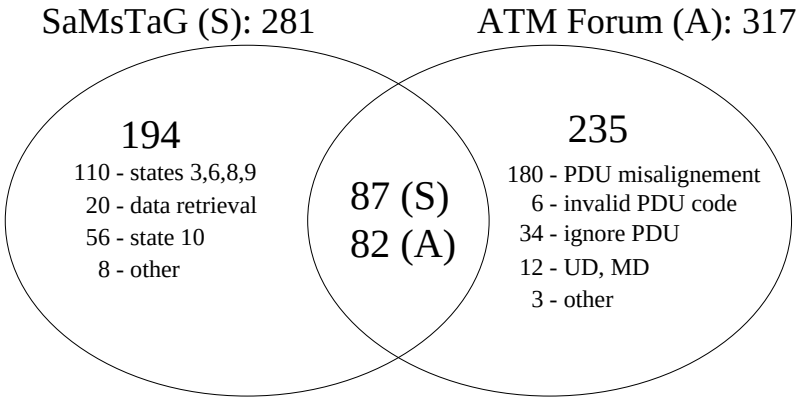
4.3. Test purposes

A test case checks a property of the specification. The test purposes describe the properties that the test cases should verify. In order to give some confidence that an IUT conforms to its specification, a test suite should cover as much properties of the specification as possible. Thus the set of test purposes should be as complete as possible.

4.3.1. Coverage

The specification of the SAMSTAG test purposes was based on the SSCOP SDL specification, looking at all state transitions. For each state transition from one state to a next state a number of transition paths exist which can be seen as properties or test purposes to be tested. Following this strategy we identified 281 test purposes covering all SSCOP states and transitions between.

Most of the test purposes for the ATM Forum test suite also cover state transitions, but there are some exceptions in state 10 with test purposes being function-oriented. As the



Note: some ATM Forum test purposes correspond to more than one SaMsTaG test purposes

Figure 5. Survey of test purpose sets

Identifier:	SSCOP_18b
Description:	If SSCOP is in state **Outgoing_Resynchronization_Pending** and gets an **AA_RELEASE_request** signal from the SSCOP user, then SSCOP should cancel **Timer_CC**, send an **END** PDU to its peer entity, set **Timer_CC** again, and change into the new state **Outgoing_Disconnection_Pending**.

Figure 6. SAMSTAG test suite: informal test purpose description

remote test method does not designate a PCO right above the IUT, some of the SSCOP states can not be built, i.e., they are not stable. Therefore, the ATM Forum test suite does not include test purposes for the states *Incoming Connection Pending* (3), *Incoming Resynchronization Pending* (6), *Recovery Response Pending* (8), and *Incoming Recovery Pending* (9). These states are left upon receipt of a response of the SSCF entity, and there is no possibility to have an influence on that with this test method.

Figure 5 gives a survey of how many test purposes the two test suites have in common. 87 test purposes of the SAMSTAG test suite relate to 82 test purposes of the ATM Forum test suite. For 87% of those test purposes SAMSTAG was able to generate verified test cases. The other 235 test purposes of the ATM Forum concern the alignment, length or code of PDUs (186), the ignoring of PDUs in a particular state (34), `UD/MD` PDUs (12) and some alternatives to existing test purposes of the SAMSTAG test suite (3). On the other hand there are 194 test purposes in the SAMSTAG test suite either concerning states not covered by the ATM Forum test suite (110), data retrieval (20), or some extra test purposes (64).

4.3.2. Test purpose specification

The identification and specification of the test purposes for the SAMSTAG test suite follows directly from the coverage criterion. For each transition path a test purpose was specified. This was done in two steps. In a first step for each test purpose an informal description was produced. In a second step the informal test purposes were formalised by means of MSC diagrams. These MSCs then served as input for the SAMSTAG tool. An

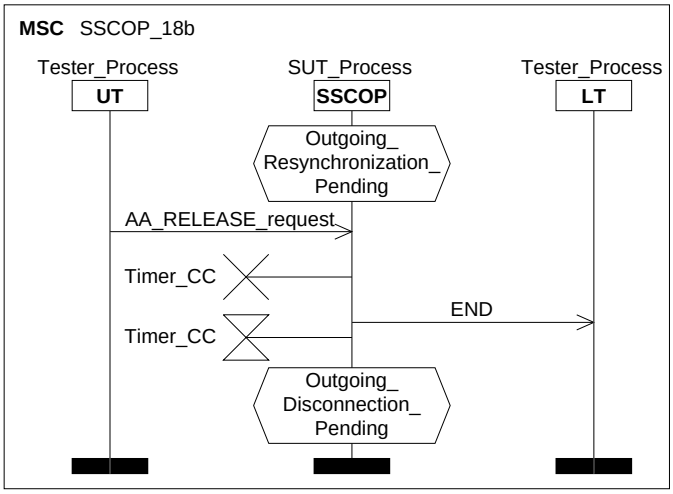


Figure 7. SAMSTAG test suite: formal test purpose specified as MSC

Identifier:	S5_V_A3
Description:	Verify that the IUT generates the **END** PDU on demand at state 5.

Figure 8. ATM Forum: informal test purpose description

example of an informal description produced for a transition path is shown in Figure 6. The informal description is very close to the SDL specification. But, its goal is to clarify the purpose of a test case and not to specify the entire system behaviour. In case of restrictions on time and money the informal descriptions may be used for the selection of the most important test cases. The formalisation of the test purpose in Figure 6 is provided by the MSC in Figure 7.

In the ATM Forum test suite the test purposes are described in an informal manner only. The description of the test purpose for the same example as in Figure 6 is shown in Figure 8. The focus of the description is more on the function, and there is less information given about the behaviour of the IUT.

4.4. Test cases

In this section we want to have a closer look at the test cases themselves. We compare the test cases with respect to aspects such as *structure*, *dynamic behaviour* and *verdicts*. The sample test cases being compared are shown in Figure 9 (SAMSTAG) and Figure 10 (ATM Forum). They are related to the test purposes shown in figures 6 and 8, respectively.

4.4.1. Structure

The SAMSTAG tool automatically generates the preamble and postamble for the test cases. They are included in the test cases directly, i.e., they are not referenced as test steps like it is done in the test suite of the ATM Forum (lines 1 and 6, Figure 10). Therefore the final verdict may be assigned in the test case at the end of the postamble (Line 11, Figure 9).

Beside the event sequences representing preamble, test body, and postamble, there

Test Case Dynamic Behaviour

Test Case Name : SSCOP_18b
Group : CONTROL/RESYNC/RELEASE/
Purpose : cf. Figures 6 and 7
Configuration :
Default : stddefault
Comments :

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		PRUT!AA_ESTABLISH_request	AA_ESTABLISH_request_11F_Y		
2		PRLT?BGN	BGN_111_Y		
3		PRLT!BGAK	BGAK_880_N		
4		PRUT?AA_ESTABLISH_confirm	AA_ESTABLISH_confirm_88_E		
5		PRUT!AA_RESYNC_request	AA_RESYNC_request_35_J		
6		PRLT?RS	RS_352_S		
7		PRUT!AA_RELEASE_request	AA_RELEASE_request_23_G		
8		PRLT?END	END_230_Q		
9		PRLT!END	END_230_Q		
10		PRLT?ENDAK	ENDAK_0_M		
11		PRUT?AA_RELEASE_confirm		PASS	
12		PRLT?POLL	POLL_100_S	INCONC	
13		PRLT?END	END_001_D	INCONC	
14		PRLT?BGN	BGN_111_Y	INCONC	
15		PRUT?MAA_ERROR_indication	MAA_ERROR_indication_P_L	INCONC	
16		PRLT?POLL	POLL_100_S	INCONC	
17		PRLT?END	END_001_D	INCONC	
18		PRLT?BGN	BGN_111_Y	INCONC	
19		PRUT?MAA_ERROR_indication	MAA_ERROR_indication_O_K	INCONC	

Detailed Comments :

Figure 9. SAMSTAG test suite: sample test case

Test Case Dynamic Behaviour

Test Case Name : S5_V_A3
Group : PC/STATE_5/VAL/
Purpose : Verify that the IUT generates the END PDU on demand at state 5.
Configuration :
Default :
Comments : Ref. 5.0 g, Fig. 20(18 of 51)/PICS PC8

Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+S5_PREAMBLE			
2		<IUT!END>	END_R_USER		
3		START T_Opr			
4	LB1	LT_PCO?END	END_R_USER	(P)	
5		+S4_VERIFY			
6		+postamble			
7		LT_PCO?MD	MD_R_GEN		
8		GOTO LB1			
9		LT_PCO?UD	UD_R_GEN		
10		GOTO LB1			
11		+TS_Opr			

Detailed Comments :

Figure 10. ATM Forum test suite: sample test case

are further alternatives listed that lead to an inconclusive verdict (`INCONC`, lines 12–19). These alternatives comprise events allowed by the protocol specification but which are not conclusive for the test purpose. A default behaviour `stddefault` is declared in the test case header. It is used to catch all other events that could occur during the execution of the test case, assigning a `FAIL` verdict to them.

The structure of the sample test case from the ATM Forum test suite (Figure 10) is different. Test steps are used to structure the TTCN dynamic behaviour descriptions. A test step called `S5_PREAMBLE` (Line 1) drives the IUT in the particular start state as required by the test purpose. Test step `S4_VERIFY` (Line 5) is used to verify the state reached after the test body. Test step `postamble` (Line 6) is used to place the IUT at state 1 (Idle). Instead of declaring a default behaviour, a test step called `TS_Opr` (Line 11) is used to catch all other events occurring at the end of the test body. Labels are used to create a loop for the reception of signals that should be ignored. In the sample test case on Line 4 the LT is waiting for an END PDU, ignoring MD and UD PDUs (lines 7 and 9) and jumping back to the location of label `LB1`.

4.4.2. Test events within the dynamic behaviour description

The ATM Forum test suite and the SAMSTAG test suite differ with respect to the use of implicit send events and timers.

Implicit send events. Within the test architecture of the ATM Forum test suite there is no explicit PCO above the IUT. But, for some of the test cases it is necessary to have a means of forcing the IUT to initiate a particular PDU. Therefore, *implicit send* events[3] are used to achieve this purpose (Line 2, Fig 10).

The test method used for the SAMSTAG test suite includes a PCO above the IUT. Thus there is no need of implicit send events. The test cases explicitly describe how the IUT gets forced to send a particular PDU by means of PDUs sent by the testers.

Timer. The ATM Forum test suite makes use of two timers. The first one, called `T_Wait`, is used to limit the test time waiting for "no response" from the IUT. The second one, called `T_Opr`, is used to allow sufficient time for a test operator to initiate some test action, i.e., it is used in conjunction with the implicit send events for test coordination. These timers are not used to verify the exact timing of an implementation, but to limit the time which the test should wait for a PDU, or to limit the total duration of the test.

SAMSTAG allows the use of timers within MSC test purposes. This feature was used for the IUT process. For the sake of generating an *abstract* test suite we renounced from using it on the testers side.

4.4.3. Test verdicts

The SAMSTAG procedure for the test case generation not only comprises the generation of preamble, test body, and postamble, but also the calculation of all the events leading to an inconclusive verdict. Since SAMSTAG does not use test steps the whole structure and all the verdicts are contained right in the dynamic behaviour description table of the test case. The verdicts assigned are final.

By way of contrast the test cases of the ATM Forum test suite assign final verdicts for

[3]The semantics of an implicit send event is that the SUT shall be controlled as necessary (to be specified in the PIXIT) in order to cause the initiation of the PDU (or ASP) referenced.

PDU Constraint Declaration

Constraint Name	: BGN_R_RET(parN_SQ,parN_MR:INTEGER)
PDU Type	: BGN
Derivation Path	:
Encoding Rule Name	:
Encoding Variation	:
Comments	: constraint for retransmitted BGN PDU

Field Name	Field Value	Field Encoding	Comments
UU	*		
PAD	*		
RESERVED	'000000'O		
N_SQ	INT_TO_BIT(parN_SQ,8)		
PL	?		
RSVD	'00'B		
PDU_Type	'0001'B		
N_MR	INT_TO_BIT(parN_MR,24)		

Detailed Comments :

Figure 11. ATM Forum test suite: sample constraint for the `BGN` PDU

inconclusive cases only. Pass verdicts are always assigned in the preliminary form, the first time after the test body (Line 4, Figure 10), and an optional second time inside a test step that acts as postamble to verify a particular state.

4.5. Declarations and constraints

The test suites comprise declaration and constraint parts that differ considerably.

4.5.1. Declarations part

One difference concerning the declarations part is the use of ASN.1 [1] to define data types. The ATM Forum test suite makes use of it, e.g., to define a list of unknown length using `SEQUENCE OF List_element_type`. The SAMSTAG tool, however, is restricted to the use of SDL data types and maps them to TTCN data types. Due to the prototype character it is not yet able to automatically generate data declarations for abstract data types being too complex. These and some general declarations (e.g., the declarations of protocol parameters) have to be added manually. However, the declarations of PDU/ASP constraints, PDU/ASP types, simple data types, and PCOs are generated automatically.

The ATM Forum test suite makes use of test suite and test case variables, the SAMSTAG test suite not at all. This is due to the different generation methods of the test suites. The SAMSTAG tool generates the test cases separately, each time restarting the simulation of the specification. Thus, repeatedly all state variables of the SSCOP entity contained in this specification are reset to their initial values. During test case generation, the actual parameters for each signal are calculated and saved into the constraints declarations. Therefore the test cases of the SAMSTAG test suite use constraints containing specific parameter values, instead of values depending on test suite or test case variables like in the ATM Forum test suite.

Phase / Subphase		Expenses	to be performed
Completion of SDL specification		1 month	manually
Preparatory work	Specification of test method and test suite structure	1 month	manually
	Identification and Specification of test purposes	2 months	manually
	Specification of different tester models	2 months	manually
Test suite generation		1 month	automatically

Figure 12. SAMSTAG test suite development expenses

4.5.2. Constraints part

The constraint part of the SAMSTAG test suite contains a huge amount of constraints declarations. The automatic generation procedure SAMSTAG is based on does not yet make use of parameterisation, whereas the ATM Forum test suite does. Therefore, a lot of different constraints may be declared for only one PDU type. An example constraint of the ATM Forum test suite making use of parameterisation is depicted in Figure 11.

Another point in this context is, as mentioned before, that the SAMSTAG tool generates the test cases for each test purpose separately. The test cases, each one with its own data declarations part, have to be merged into one test suite. Depending of the quality of the merging process implemented, redundant declarations are sorted out more or less.

For the declaration of constraints, e.g., the constraint in Figure 11, the ATM Forum test suite makes use of TTCN matching mechanisms such as "?" (any value), "*" (any or omit), or "-" (omit). The SAMSTAG tool itself does not generate constraints containing matching mechanisms. SAMSTAG calculates the concrete values. However, if matching mechanism were used in signal parameters in the test purpose MSCs, then they would appear in the generated constraints. Without this measure the SAMSTAG tool is not able to estimate the relevance of the parameters (in order to decide where to use wildcards).

4.6. Development Expenditure

The goal of SAMSTAG is to improve the conformance testing process in a twofold manner. On the one hand it should save time and money expenses, and on the other hand the application of SAMSTAG should ensure the consistence between specification and test cases. It is obvious that the latter goal has been achieved. For judging time and money savings a comparison of both test suites with respect to the development expenses is required.

Since there was no data on the ATM Forum test suite available at the time of this writing we are just able to present our expenses. This is done in Figure 12. The development process is structured into the main phases *Completion of SDL specification*, *Preparatory work* and *Test suite generation*. The Preparatory work phase itself is divided into the subphases as they were passed through during the development of SAMSTAG test suite. Further details on this Preparatory work phase are given in [5]. The expenses for test case generation not only include the mere generation time, but also the work of relating test purposes and test models and the experimentation on the application of different

SAMSTAG heuristics [6].

In total the expenses for our case study comprises 7 months. We believe that this is a very good result and that due to increasing experience the expenses may decease further.

It should also be noted that SAMSTAG generates test cases for only 70 % of the identified test purposes. The expenses for the manual completion of the test suite and for getting familiar with SSCOP and SAMSTAG have not been estimated.

5. Conclusions, summary and outlook

The ATM Forum test suite takes advantage of several TTCN language concepts to improve the (human) readability. The structuring by using test steps, timer guards, matching mechanisms, test suite/test case variables, and the parametrization of constraints ease the survey of the abstract test suite (ATS).

However, considering a subsequent automatical processing of the ATS readability is less important than the validation of the test purposes and the verification of the test cases with respect to the specification. This is done automatically within the SAMSTAG method. In this connection the fact that the SAMSTAG tool succeeds for 86% of the test purposes in common with the ATM Forum test suite receives its importance.

Most differences between the test suites are due to the different test methods they are based on. The remote method implies the use of implicit send events and it is assumed that there are no test purposes existing for some of the SSCOP states. For the present, the SAMSTAG tool, but not the method, is not able to cope with the remote test method. Thus future work has to focus on the influence of different test generation methods on the test case generation process.

Feedback regarding the results of this comparison will be given to the ATM Forum and ETSI.

REFERENCES

1. CCITT. Recommendations X.208 (ISO/IEC IS 8824 and 8824/AD1): Information Processing Systems - Open Systems Interconnection - *Specification of Abstract Syntax Notation One (ASN.1)* and Addendum 1: ASN.1 Extensions. The International Telegraph and Telephone Consultative Committee (CCITT), Geneva, November 1989.
2. ATM Forum. *Conformance Abstract Test Suite for the SSCOP for UNI 3.1.* Technical Committee ATM Forum, September 1996. af-test-0067.000.
3. J. Grabowski. *Test Case Generation and Test Case Specification with Message Sequence Charts.* PhD thesis, University of Berne, Institute for Informatics and Applied Mathematics, February 1994.
4. J. Grabowski, D. Hogrefe, and R. Nahm. *Test Case Generation with Test Purpose Specification by MSCs.* In O. Faergemand and A. Sarma, editors, *SDL'93 - Using Objects.* North-Holland, October 1993.
5. J. Grabowski, R. Scheurer, Z. Dai, and D. Hogrefe. *Applying SAMSTAG to the B-ISDN Protocol SSCOP - Technical Description and TTCN Testsuite.* Schriftenreihe der Institute für Mathematik/Informatik, Technical Report A-97-01, Medical Univ. of Lübeck, Germany, January 1997. (submitted for IWTCS 1997).

6. J. Grabowski, R. Scheurer, D. Toggweiler, and D. Hogrefe. *Dealing with the Complexity of State Space Exploration Algorithms.* In Proceedings of the 6th. GI/ITG technical meeting on 'Formal Description Techniques for Distributed Systems', University of Erlangen, June 1996.
7. ISO/IEC. Information Technology - Open Systems Interconnection - *Conformance Testing Methodology and Framework.* International multipart standard 9646, ISO/IEC, 1994.
8. ITU Telecommunication Standards Sector. ITU-T Recommendation Q.2110: B-ISDN ATM Adaption Layer - *Service Specific Connection Oriented Protocol (SSCOP).* ITU, Geneva, July 1994.
9. ITU Telecommunication Standards Sector. ITU-T Recommendation Q.2130: B-ISDN ATM Adaption Layer - *Service Specific Coordination Function (SSCF).* ITU, Geneva, July 1994.
10. ITU Telecommunication Standards Sector SG 10. ITU-T Recommendation Z.100: *Specification and Description Language (SDL).* ITU, Geneva, 1996.
11. ITU Telecommunication Standards Sector SG 10. ITU-T Recommendation Z.120: *Message Sequence Chart (MSC).* ITU, Geneva, 1996.
12. R. Nahm. *Conformance Testing Based on Formal Description Techniques and Message Sequence Charts.* PhD thesis, University of Berne, Institute for Informatics and Applied Mathematics, February 1994.
13. ITU Telecommunication Standards. ITU-T Rec. I.362: *B-ISDN ATM Adaption Layer (AAL) - Functional Description.* ITU, Geneva, March 1993.
14. ITU Telecommunication Standards. ITU-T Rec. I.363: *B-ISDN ATM Adaption Layer (AAL) - Specification.* ITU, Geneva, March 1993.
15. S.J. Yoo, L. Collica, and T.W. Jeong. *Conformance testing of ATM Adaption Layer protocol.* In B. Baumgarten, H.-J. Burkhardt, and A. Giessler, editors, *Testing of Communicating Systems.* Chapman & Hall, 1996.