

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

Rudolf Scheurer^{*}, Jens Grabowski[†], Dieter Hogrefe[†],

^{*}University of Fribourg, Institute for Informatics, ch. du Musée 3, CH-1700 Fribourg, Switzerland, rudolf.scheurer@unifr.ch

[†]University of Lübeck, Institute for Telematics, Ratzeburger Allee 160, D-23538 Lübeck, Germany, {jens, hogrefe}@itm.mu-luebeck.de

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. Both test suites have been compared, but the results were only of restricted value because the test suites base on different test architectures. In order to achieve more significant comparison results the SAMSTAG tool has been adapted to the test method chosen by the ATM Forum, i.e., the remote test method, and the test suite has been re-generated. In this paper we present a revised comparison of various aspect of the two test suites.

Keywords

SDL, MSC, TTCN, conformance testing, test case generation, abstract test suite, remote test method, 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 [9]. 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

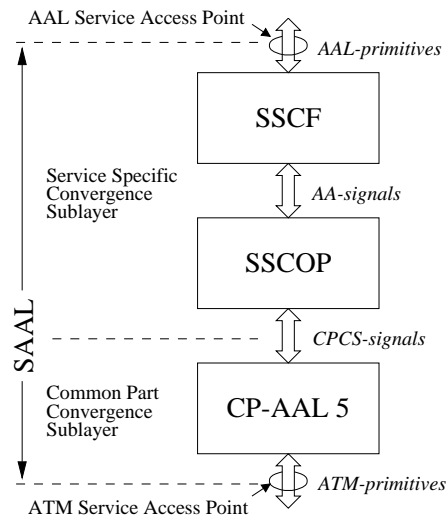


Figure 1. Structure of the Signalling ATM Adaption Layer (SAAL)

been generated automatically [5]. For another 9% of the test purposes a test case has been found but due to complexity and limitations of the SAMSTAG prototype verification could not be obtained.

At the same time the ATM Forum¹ developed a test suite for SSCOP which has been approved in December 1996 [1]. 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.

The main difference of the test suite generated with the SAMSTAG tool and the test suite from the ATM Forum was due to the use of a different abstract test method. In terms of ISO IS 9646 [7] the test suite generated by SAMSTAG makes use of the *distributed test method* whereas the ATM forum test suite is based on the *remote test method*. We compared both test suites in more detail [6], but, the results are only of restricted value, because the choice of the test architecture heavily influences the testability of test purposes and the specification of the test cases itself.

As a consequence we modified the SAMSTAG tool in such a way that it is able to cope with the remote test method and re-generated part of the test suite. We focused on the re-generation of comparable test cases, i.e., on test purposes which are common for both test suites. In this paper we present a revised comparison of the two test suites for the remote test 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, and the conclusions are given in Section 5.

2. Service Specific Connection Oriented Protocol (SSCOP)

SSCOP [9] is used in the *B-ISDN ATM Adaption Layer (AAL)* [14,15]. 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*

¹The ATM Forum is a non-profit international organization accelerating cooperation on ATM technology.

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) [8]. 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.

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 protocol data unit (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.

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 formalized before [5,13].

3. Subject of comparison

The subject of the comparison are two test suites for the SSCOP protocol developed in 1996 and 1997, respectively. 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*.

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).

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 [5,13].

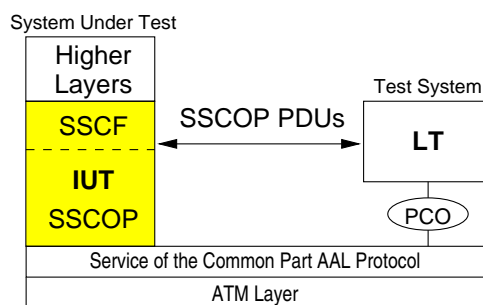


Figure 2. Test method as used of the two test suites

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 a test suite.

4.1. Test method

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.

The ATM Forum test suite and the SAMSTAG test suite are both 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. The configuration of this test method is shown in Figure 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.

The structure of the SAMSTAG 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 leaves 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 that were generated for this comparison, there are 86 in total.

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 cases.

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.

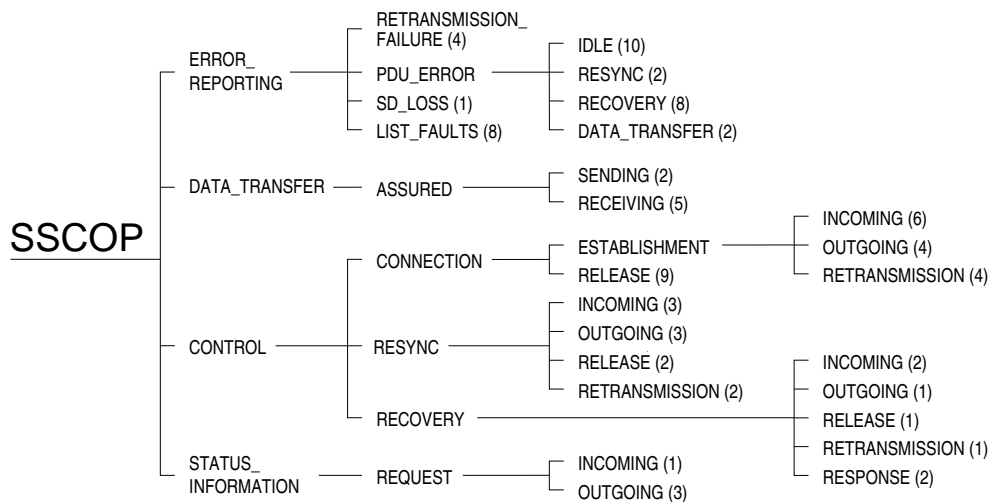


Figure 3. SAMSTAG test suite structure

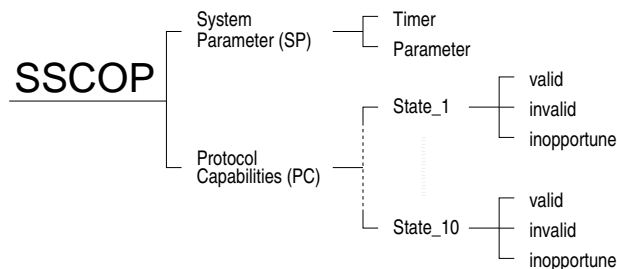


Figure 4. ATM Forum test suite structure

4.3.1. Coverage

Most of the test purposes for the ATM Forum test suite cover state transitions, but there are some exceptions in state `Data_Transfer_Ready` (state 10) with test purposes being function-oriented. As the 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. Nevertheless one test purpose for each of these state transitions could be specified (describing the default behavior), but they are missing in the ATM Forum test suite.

The specification of the test purposes for the SAMSTAG test suite has been done independently of any abstract test method. Based on the SSCOP specification, 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 a total of 281 test purposes had been identified for the original test suite for the distributed test method, covering *all* SSCOP states and transitions between.

For the remote test method only 151 test purposes could be considered, since almost all the other test purposes relate to the instable states mentioned above and are therefore not applicable for this test method. For this comparison we focused on the 82 test purposes

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 5. SAMSTAG test suite: informal test purpose description

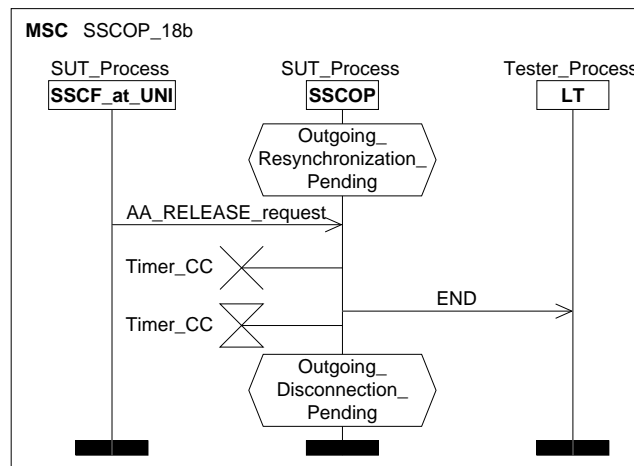


Figure 6. SAMSTAG test suite: formal test purpose specified as MSC

that are in common with the ATM Forum test suite, while adding the missing four test cases mentioned above. In the end the SAMSTAG tool was able to generate verified test cases for 84% of the 86 test purposes in the SAMSTAG test suite.

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 formalized by means of MSC diagrams. These MSCs then served as input for the SAMSTAG tool. An example of an informal description produced for a transition path is shown in Figure 5. 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 behavior. In case of restrictions on time and money the informal descriptions may be used for the selection of the most important test cases. The formalization of the test purpose in Figure 5 is provided by the MSC in Figure 6.

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 5 is shown in Figure 7. The focus of the description is more on the function, and there is less information given about the behavior of the IUT.

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

Figure 7. ATM Forum: informal test purpose description

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		<IUT!BGN>			
2		PRLT?BGN	BGN_111_Y		
3		PRLT!BGAK	BGAK_880_N		
4		<IUT!RS>	SD_340_L		
5		PRLT?RS	RS_352_S		
6		<IUT!END>	END_230_Q		
7		PRLT?END	END_230_Q		
8		PRLT!END	END_230_Q		
9		PRLT?ENDAK	ENDAK_0_M	PASS	
10		PRLT?END	END_230_Q	INCONC	
11		PRLT?POLL	POLL_100_S	INCONC	
12		PRLT?END	END_001_D	INCONC	
Detailed Comments :					

Figure 8. SAMSTAG test suite: sample test case

4.4. Test cases

In this section we have a closer look at the test cases themselves. We compare the test cases with respect to aspects such as *structure*, *dynamic behavior* and *verdicts*. The sample test cases being compared are shown in Figure 8 (SAMSTAG) and Figure 9 (ATM Forum). They are related to the test purposes shown in figures 5 and 7, 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 ATM Forum test suite (lines 1 and 6, Figure 9). Therefore the final verdict may be assigned in the test case at the end of the postamble (Line 9, Figure 8).

Beside the event sequences representing preamble, test body, and postamble, there are further alternatives listed that lead to an inconclusive verdict (**INCONC**, lines 10–12). These alternatives comprise events allowed by the protocol specification but which are not conclusive for the test purpose. A default behavior **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 9) is

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 9. ATM Forum test suite: sample test case

different. Test steps are used to structure the TTCN dynamic behavior 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 behavior, 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 behavior description

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

Assignment

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

Test Case Dynamic Behaviour					
Test Case Name : SSCOP_09b_PPO2					
Group : CONTROL/CONNECTION/RELEASE					
Purpose : Verify that the IUT generates the END PDU on demand in state 2					
Configuration :					
Default :					
Comments : corresponds to S2_V_A3 in ATM Forum test suite					
Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		<IUT!BGN>	BGN_111_Y		test body
2		PRLT?BGN	BGN_111_Y		
3		<IUT!END>	END_230_Q		
4		PRLT?END	END_230_Q		
5		PRLT!BGN	BGN_110_M	INCONC	test step <i>S4_VERIFY</i>
6		PRLT?END	END_230_Q		
7		PRLT?BGAK	BGAK_0_R		
8		PRLT?END	END_230_Q	PASS	test step <i>postamble</i>
9		PRLT!BGREJ	BGREJ_010_L		
10		PRLT!END	END_230_Q		
11		PRLT?ENDAK	ENDAK_0_M	INCONC	
12		PRLT?END	END_230_Q		
Detailed Comments :					

Figure 10. SAMSTAG test case corresponding to S2_V_A3 (test purpose SSCOP_09b)

to an inconclusive verdict. Since SAMSTAG does not use test steps the whole structure and all the verdicts are contained right in the dynamic behavior 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 inconclusive cases only. Pass verdicts are assigned in the preliminary form, the first time after the test body (Line 4, Figure 9), and an optional second time inside a test step that acts as postamble to verify a particular state. At the very end of the test case, i.e. in test step *postamble*, a final verdict is assigned referencing *R*, the standard variable used to store the preliminary verdicts.

Inconclusives

The ATM Forum test suite uses test steps consisting of Unique-Input-Output (UIO) sequences to verify that a particular state has been reached after execution of the test body. Furthermore, a generic test step called *postamble* is used to return to a stable initial state. But, the fact that the verification test steps are independent from the test body they are attached to may cause some problems concerning the specification of inconclusive verdicts.

An example is given in Figure 10. The test step *S4_VERIFY* does not consider the context of the preceding test body. The alternative on Line 6 is left out in test step *S4_VERIFY*, same as the alternative on Line 12. These alternatives represent signal repetitions of lines 4 and 8, respectively, caused by timer timeouts. Thus the occurrence of these signals would lead to a **FAIL** verdict when applying the ATM Forum test suite.

The SAMSTAG tool however calculates a postamble suited for the particular test body and verifies the UIO property of the test case as a whole. Hence the test cases generated by SAMSTAG are most often shorter than the corresponding ones of the ATM Forum test suite. Looking at the sample test case mentioned above, this may be seen in Figure 11.

Test Case Dynamic Behaviour					
Test Case Name : SSCOP_09b					
Group : CONTROL/CONNECTION/RELEASE					
Purpose :					
Configuration :					
Default : stddefault					
Comments :					
Nr	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		<IUT!BGN>			
2		PRLT?BGN	BGN_111_Y		
3		<IUT!END>			
4		PRLT?END	END_230_Q		
5		PRLT!END	END_230_Q		
6		PRLT?ENDAK	ENDAK_0_M	PASS	
7		PRLT?END	END_230_Q	INCONC	
Detailed Comments :					

Figure 11. Test case for test purpose SSCOP_09b as generated by SAMSTAG

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 [2] 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.

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 parameterization, 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 parameterization is depicted in Figure 12.

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

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 12. ATM Forum test suite: sample constraint for the BGN PDU

merging process implemented, redundant declarations are sorted out more or less.

For the declaration of constraints, e.g., the constraint in Figure 12, 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).

5. Conclusions

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 and test case variables, and the parameterization of constraints ease the understanding of the abstract test suite (ATS). Our future work will focus on the readability of the TTCN code generated by the SAMSTAG tool.

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, thus assuring the quality of the test cases, as shown on the example of the completeness of inconclusive verdicts. In this connection the fact that the SAMSTAG tool succeeds for 84% of the test purposes in common with the ATM Forum test suite receives its full importance.

REFERENCES

1. ATM Forum. Conformance Abstract Test Suite for the SSCOP for UNI 3.1. Technical Committee ATM Forum, September 1996. af-test-0067.000.
2. 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.
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. Færgemand 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, and D. Hogrefe. Comparison of an Automatically Generated and a Manually Specified Abstract Test Suite for the B-ISDN Protocol SSCOP. Technical Report A-97-07, University of Lübeck, Germany, April 1997.
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.2130: B-ISDN ATM Adaption Layer - Service Specific Coordination Function (SSCF). ITU, Geneva, July 1994.
9. ITU-TS. ITU-T Recommendation Q.2110: B-ISDN ATM Adaption Layer - Service Specific Connection Oriented Protocol (SSCOP). ITU Telecommunication Standards Sector, Geneva, July 1994.
10. ITU-TS. ITU-T Recommendation Z.100: Specification and Description Language (SDL). ITU Telecommunication Standards Sector SG 10, Geneva, 1996.
11. ITU-TS. ITU-T Recommendation Z.120: Message Sequence Chart (MSC). ITU Telecommunication Standards Sector SG 10, 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. R. Scheurer. *Demonstrating the Applicability of Automatic Test Case Generation Methods*. PhD thesis, University of Berne (Switzerland), Institute for Informatics, June 1997.
14. ITU Telecommunication Standards. ITU-T Rec. I.362: B-ISDN ATM Adaption Layer (AAL) - Functional Description. ITU, Geneva, March 1993.
15. ITU Telecommunication Standards. ITU-T Rec. I.363: B-ISDN ATM Adaption Layer (AAL) - Specification. ITU, Geneva, March 1993.