

## **Methods for Testing and Specification (MTS); Report of the CATG applications**

---



---

Reference

DTR/MTS-00030-3 (9qc00ics.PDF)

---

Keywords

Methodology, SDL, testing, TTCN, validation

**ETSI**

---

Postal address

F-06921 Sophia Antipolis Cedex - FRANCE

---

Office address

650 Route des Lucioles - Sophia Antipolis  
Valbonne - FRANCE  
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  
Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

Internet

[secretariat@etsi.fr](mailto:secretariat@etsi.fr)  
Individual copies of this ETSI deliverable  
can be downloaded from  
<http://www.etsi.org>  
If you find errors in the present document, send your  
comment to: [editor@etsi.fr](mailto:editor@etsi.fr)

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 1999.  
All rights reserved.

---

# Contents

Intellectual Property Rights.....	4
Foreword .....	4
Introduction .....	4
1 Scope.....	5
2 References.....	5
3 Abbreviations.....	6
4 INAP CS-2 .....	6
4.1 Test architecture and interfaces.....	6
4.2 Test Purposes .....	7
4.2.1 Informal TP descriptions.....	7
4.2.2 Formal MSC TPs.....	8
4.3 The development of MSC TPs.....	9
4.3.1 Preparatory Work .....	9
4.3.2 Development process .....	9
4.4 Ensuring consistency between MSC TPs and SDL model.....	10
4.5 MSC TPs for different purposes .....	10
4.5.1 MSC TP traces .....	11
4.5.2 MSC TPs for documentation purposes.....	12
4.5.3 MSC TP Autolink input .....	12
4.6 Test cases.....	13
4.7 Expenses .....	14
5 Other Experiences .....	15
5.1 Experiences of using CATG for B-ISDN test suites .....	15
History .....	18

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Methods for Testing and Specification (MTS).

---

## Introduction

The development of a test suite accompanying a specification of a standard facilitates the proof that products based on the specification actually conform to it. The test suite can be used to test the functionality of a product, its conformance and to a certain degree also the interoperability with other products. Usually the development of the test suite lags somewhat behind the development of the base standard itself. This is natural considering that tests can only be defined for functions that have already been agreed upon during the standardization process.

The use of the formal Specification and Design Language (SDL) [4] and [5] for the specification of standards allows Computer Aided Test Generation (CATG). With CATG, processing power is used to facilitate and speed up the development of a test suite. CATG also ensures the consistency between a protocol specification and its test suite.

The present document is based on the experience made with the INAP CS2 and the B-ISDN test suite developments. It reflects the state of the art.

---

# 1 Scope

The present document describes the experiences gained from the application of the CATG functionality of the SDL tool available at ETSI (SDT Autolink tool) on the SDL specification of Core INAP CS-2. The SDLs of Core INAP, attached as annex A to [3], gives the normative requirements of the protocol behaviour. The SDLs also makes use of the ASN.1 definitions of the INAP PDUs, therefore providing a complete specification of the protocol suitable for application of CATG.

---

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.
- A non-specific reference to an ETS shall also be taken to refer to later versions published as an EN with the same number.

- [1] ETS 300 771-1 (1995): "Broadband Integrated Services Digital Network (B-ISDN); Digital Subscriber Signalling System No. two (DSS2) protocol; B-ISDN user-network interface layer 3 specification for point-to-multipoint call/bearer control; Part 1: Protocol specification; [ITU-T Recommendation Q.2971 (1995), modified]".
- [2] EN 300 771-6: "Broadband Integrated Services Digital Network (B-ISDN); Digital Subscriber Signalling System No. two (DSS2) protocol; B-ISDN user-network interface layer 3 specification for point-to-multipoint call/bearer control; Part 6: Abstract Test Suite (ATS) and partial Protocol Implementation eXtra Information for Testing (PIXIT) proforma specification for the network".
- [3] EN 301 140-1 (V1.3): "Intelligent Network (IN); Intelligent Network Application protocol (INAP); Capability Set 2 (CS2); Part 1; Protocol Specification".
- [4] CCITT Recommendation Z.100 (1993): "CCITT specification and description language (SDL)".
- [5] CCITT Recommendation Z.100 Addendum 1 (1996): "CCITT specification and description language (SDL)".
- [6] ITU-T Recommendation Z.105 (1995): "SDL Combined with ASN.1 (SDL/ASN1)".
- [7] CCITT Recommendation Z.120 (1993): "Message sequence chart (MSC) note: Recommendation Z.120 is temporarily unavailable for technical reason".
- [8] ISO/IEC 9646-3 (1992): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 3: The Tree and Tabular Combined Notation (TTCN)".

## 3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation One
ASP	Abstract Service Primitive
ATS	Abstract Test Suite
B-ISDN	Broadband ISDN
CATG	Computer Aided Test Generation
CS	Capability Set
INAP	Intelligent Networks Application Protocol
ISDN	Integrated Services Digital Network
ISUP	ISDN User Part
IUT	Implementation Under Test
MSC	Message Sequence Chart
PDU	Protocol Data Unit
PTC	Parallel Test Components
SCF	Service Control Function
SDL	Specification and Description Language
SSF	Service Switching Function
SS7	Signalling System No. 7
TCAP	Transaction Capabilities Application Protocol
TP	Test Purpose
TTCN	Tree and Tabular Combined Notation

## 4 INAP CS-2

This clause describes the experiences gained from the application of the CATG functionality of the SDL tool available at ETSI (SDT Autolink tool) on the SDL specification of Core INAP CS-2. The SDLs of Core INAP, attached as annex A to [3], gives the normative requirements of the protocol behaviour. The SDLs also makes use of the ASN.1 definitions of the INAP PDUs [6], therefore providing a complete specification of the protocol suitable for application of CATG.

### 4.1 Test architecture and interfaces

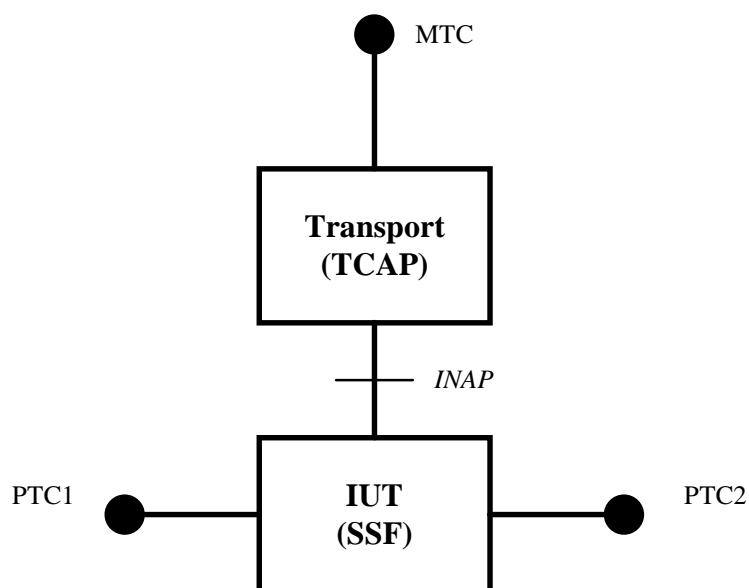


Figure 1: Test architecture for INAP CS-2

The INAP protocol enables communication between an SCF and an SSF in a public network. A Test Suite is provided only for the SSF, where the IUT is either a local exchange or a transit exchange. INAP is implemented on top of TCAP in SS7, therefore the Remote Test Method is used with the MTC serving the normative interface as depicted in figure 1. The protocol operations of INAP states normative requirements on what the SSF should do in terms of signalling, for example establish a call to a third party, release a call etc. The PASS verdicts in the test suite must therefore in most cases be based on the signalling events. However, since INAP is defined independently of the particular signalling system used (national variants of either DSS.1 or ISUP is assumed), it is not possible to provide the definitions of PTC1 (incoming signalling) and PTC2 (outgoing signalling) in the test suite. Instead the co-ordination messages between the PTCs and the MTC are provided, which are generated from the SDL specification of INAP. The SDL specification of INAP reflects the requirements on the signalling in terms of an 'abstract signalling system', which can easily be mapped to DSS.1 or ISUP. The messages of this 'abstract signalling system' are used in the test suite as the co-ordination messages.

## 4.2 Test Purposes

The method used for the development of the test purposes for INAP was adapted to cater for the specific needs of CATG. The tool used (Autolink) requires a MSC [7] as input, as explained in clause 6. The development of the TPs comprised the following two steps:

- 1) the test purposes were manually identified and documented using English text based on the protocol requirements;
- 2) using SDL simulation, MSCs were generated for each test purpose to provide input for the Autolink tool. The generated MSCs were also included in the test purpose document, giving further definition of the test purposes. This was a requirement from organizations not using TTCN [8] for testing purposes, but still required a firm definition of each test purpose.

The added value of this method is that the consistency between the informally developed test purposes and the protocol is guaranteed. A number of errors in the manually developed test purposes were detected using this method.

The rest of this subclause gives examples of test purposes and the usage of the SDL simulator to generate the MSCs.

### 4.2.1 Informal TP descriptions

Informal TP descriptions are specified by using tables which structure a more or less informal text. The table entries specify the test case, provide a textual TP description, may refer to pre- and postambles, describe the pass criteria and may provide further information. The example in table 1 describes the TP of the test case IN2\_A\_BASIC\_CO\_BV\_03.

**Table 1: Informal TP description**

IN2_A_BASIC_CO_BV_03	
<b>Purpose:</b>	Test of <b>Connect</b> procedure with supplementary services parameters
<b>Requirement ref</b>	11.6
<b>Selection Cond.</b>	M
<b>Preamble:</b>	O_OS
<b>Test description</b>	SCF sends to SSF <b>Connect</b> invoke containing parameters related to supplementary services, with: <ul style="list-style-type: none"> <li>- destinationRoutingAddress;</li> <li>- originalCalledPartyID;</li> <li>- redirectingPartyID;</li> <li>- redirectionInformation.</li> </ul> SSF sends a SetupRequest to B side.
<b>Pass criteria</b>	Check that the above parameters are mapped from Connect into the Setup request: <ul style="list-style-type: none"> <li>- destinationRoutingAddress -----&gt; calledPartyNumber</li> <li>- originalCalledPartyID-----&gt; originalCalledNumber</li> <li>- redirectingPartyID-----&gt; redirectingNumber</li> <li>- redirectionInformation-----&gt; redirectionInformation</li> </ul>
<b>Postamble:</b>	SigConA_Release_thenB

## 4.2.2 Formal MSC TPs

The informal TPs are documented in form of MSCs. The development process of the MSC TPs is described in subclause 4.3. The MSC TPs may be structured into three separate MSCs, a preamble MSC, a testbody MSC and a postamble MSC. Preamble and postamble MSCs are referred to in the testbody MSCs by using MSC references. The MSC TP in figure 2 implements the informal TP description shown in table 1. The testbody MSC refers to its preamble O\_OS and its postamble SigConA\_Release\_thenB. The pre- and postambles are also documented using MSCs. The names of testbody MSCs correspond to test case names and the names of preamble and postamble MSCs correspond to test step names.

An additional structuring of a testbody MSC by means of further MSC references is possible, but has not been used for the INAP CS-2 ATS.

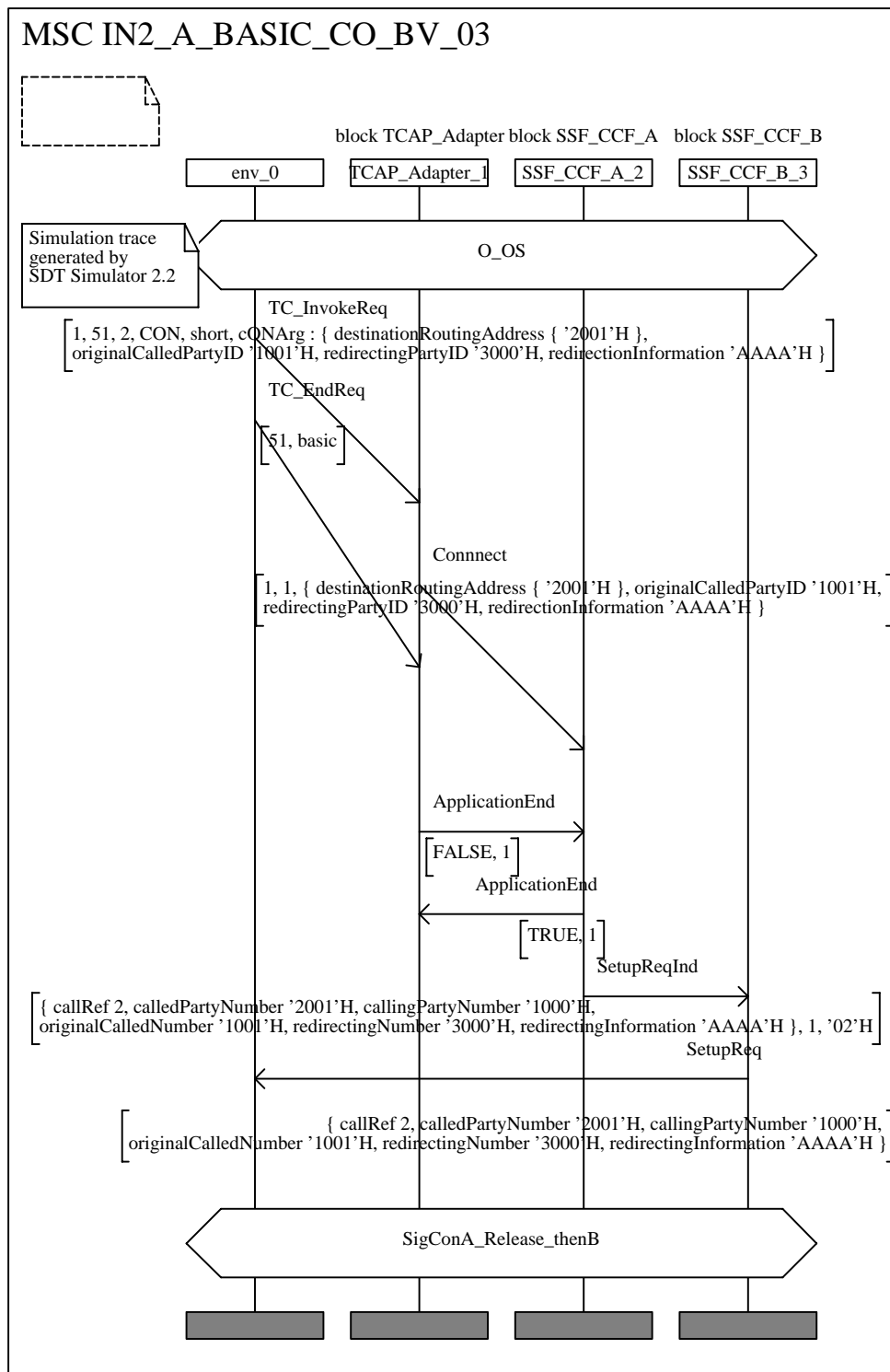


Figure 2: Formal MSC TP



## 4.3 The development of MSC TPs

### 4.3.1 Preparatory Work

The MSCs are generated by means of simulation. In order to facilitate their generation the user interface of the SDL simulator used at ETSI was adapted to the specific needs of the MSC TP generation for the INAP CS 2 protocol. Several buttons for the settings of different simulator options and for the automatic execution of pre- and postambles have been added.

Different simulator options are needed due to the fact that the TP development process also comprises a validation of the TPs and the corresponding INAP CS 2 SDL (cf. next section). For the validation a more detailed view on the protocol behaviour is needed than for the MSC TP development itself. The different views are provided by means of different simulator options.

The same pre- and postambles are used in several test cases. In order to ease their execution they have been implemented as buttons. Pressing a button forces the simulator to perform the actions to execute a corresponding pre- or postamble.

### 4.3.2 Development process

The development of the MSC TPs is a two phase process. It comprises a validation phase and a development phase.

In the first phase the MSC TP to be produced is executed in order to validate the TP and SDL model. If the expected behaviour and the validation run are not in accordance, the TPs, the SDL model or both may have to be changed or corrected. The validation step has to be repeated until the TP and the SDL model are in accordance.

In the second phase the TP is simulated again by performing the following steps:

- 1) start the simulator;
- 2) set the simulator options according to the needs of the MSC TP generation. This is done by using the appropriate predefined button;
- 3) if required, execute a preamble by using a predefined button;
- 4) start the MSC trace. As a result the rest of the simulation will be displayed and recorded as a MSC;
- 5) if a preamble has been performed, introduce a MSC reference referring to a preamble MSC into the displayed MSC trace. This has to be done manually;
- 6) simulate the TP;
- 7) if necessary, include a MSC reference referring to a postamble MSC into the displayed MSC trace. This has to be done manually;
- 8) rename the MSC trace to the test case name;
- 9) store the MSC trace in MSC PR form. This has to be done because the PR form is the basis for the further processing of the MSC Tps.

Pre- and postamble MSCs (steps 5 and 7) have to be generated like normal MSC TPs. But, this has to be done only once for the whole testsuite. The postamble needs only to be referred to, but not to be executed. During test case generation its correct execution will be checked by the Autolink tool. With some practise the performance of the actions' 1 to 5 and 7 to 9 only take a few seconds. Action 6 may take a little bit longer.

Although the development phase of the MSC TP development process looks more complicated, our experience has shown that it takes less time than the validation phase.

## 4.4 Ensuring consistency between MSC TPs and SDL model

During the validation phase in the MSC TP development process the SDL model may be corrected or modified. This may change the behaviour of the model and the already developed MSC TPs may become invalid. In order to detect invalid MSCs after each change of the SDL model the MSC TPs have to be re-validated against the SDL model. This can be done automatically by writing a script.

For each MSC TP this script has to start the SDT validator in the command mode, i.e., without user interface, and to input the MSC into the validator. Depending on the complexity of the SDL model the validation of all MSC TPs may take some time. To reduce the time for the re-validation the MSC TPs may be re-validated in parallel on several computers.

## 4.5 MSC TPs for different purposes

As mentioned the MSC TPs are developed by means of simulation. The simulator traces are recorded and afterwards stored as MSCs, which then form the basis for the MSC TPs. They are only the basis because:

- references to pre- and postamble MSCs have to be added manually;
- the MSCs do not present the information needed for documentation purposes in an adequate manner; and
- references to pre- and postamble MSCs should be added during the simulation itself.

One solution for the latter two problems is to write a script that transforms the recorded MSC traces into the appropriate form. Such a script has the advantage that the MSCs needed for documentation and as Autolink input can be repeated automatically if a MSC TP has to be adapted or re-simulated due to changes in the SDL model. An example may clarify the character of the different MSC TPs.

### 4.5.1 MSC TP traces

A MSC trace resulting from a simulation run with manually added references to pre- and postamble MSCs is shown in figure 3.

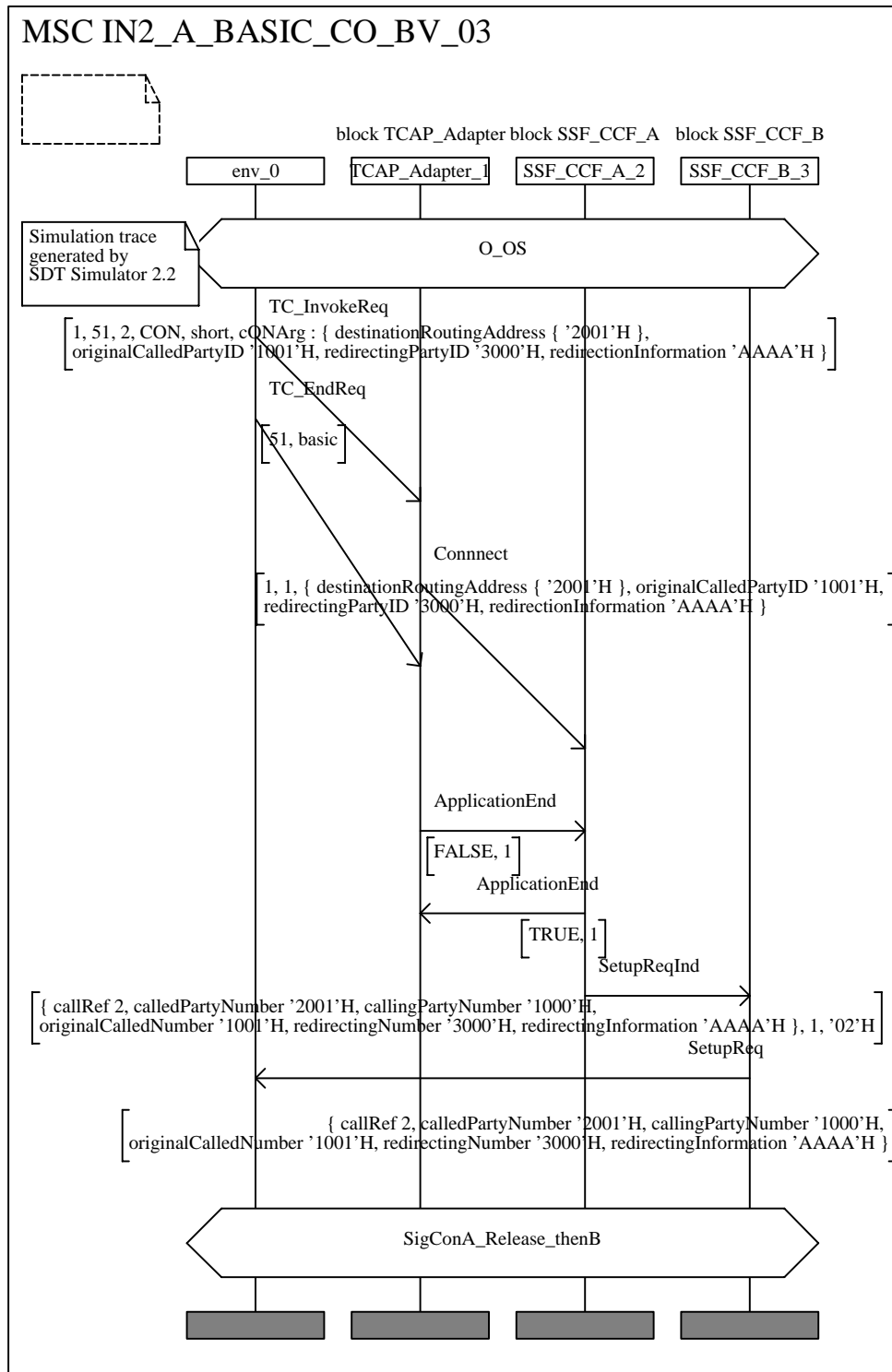


Figure 3: MSC TP Trace

## 4.5.2 MSC TPs for documentation purposes

For INAP CS-2 documentation purposes it is necessary to present the different interfaces of the system and not only one environment axis. Furthermore, the internal signal exchange among the two SSF-SCF axis should be hidden whereas the internal signal exchange between the TCAP adapter axis and the rest of the system should be visible. In this case this is due to the fact that the TCAP adapter does not belong to the system to be tested itself. It provides only a standardized interface. The TP descriptions however are related to the signal exchange between TCAP adapter and SSF-CCF processes. In order to ease the understanding of the TP descriptions it was decided to keep TCAP Adapter and SSF-CCF system in the MSC TP description. Figure 2 shows the MSC TP that is used for the documentation of the MSC TP trace shown in figure 3.

## 4.5.3 MSC TP Autolink input

For the Autolink tool another MSC input is required. Autolink uses system level MSCs as input. A system level MSC only includes one system axis and one or more environment axes. The environment axes represent the interfaces of the SDL system with its environment. These interfaces are the points where SDL channels access the environment. In case of one environment axis all interfaces are matched on the same environment axis. In case of several axes each axis describes one interface. The different environment axis are named after the SDL channels to which the interfaces belong. The MSC TP in figure 4 describes the Autolink input for the MSCs shown in figures 2 and 3.

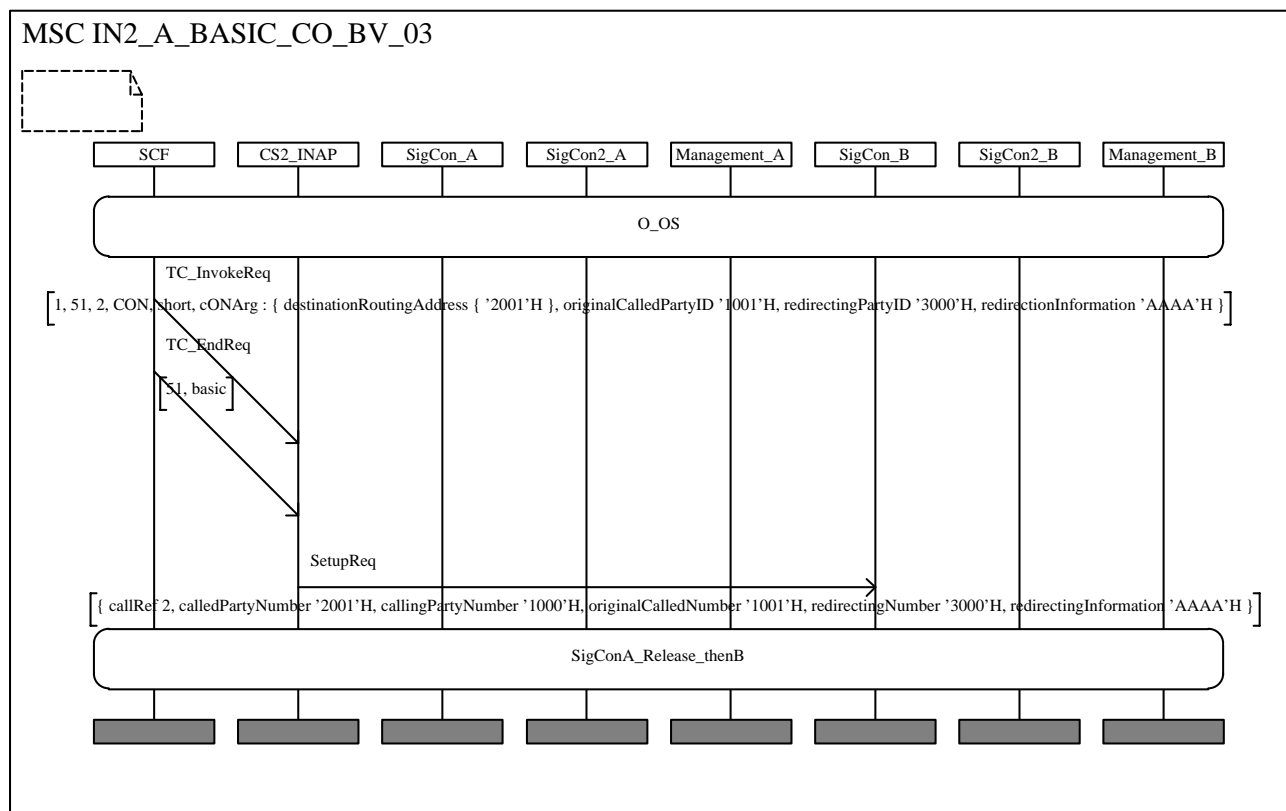


Figure 4: Autolink input

NOTE: It is also possible to produce the required system level MSCs by applying the filtering function of the SDT MSC editor. But this has to be redone manually if due to changes of the SDL system a TP has to be adapted or re-simulated.

## 4.6 Test cases

An example of a generated test case and a generated constraint is shown below in tables 2 and 3 respectively.

The postprocessing of the test case involves replacing the values of the constraint parameters with PIXIT parameters and the addition of the timer tSSF. Both tasks are performed automatically with a UNIX script operating on the TTCN MP file, and the result is shown in table 4. The result is then manually inspected for consistency, especially concerning the timer handling and necessary modifications are performed.

**Table 2: Generated Test Case**

Test Case Dynamic Behaviour					
Test Case Name : IN2_A_BASIC_CO_BV_03					
Group :					
Purpose :					
Configuration :					
Default : OtherwiseFail					
Comments :					
NR	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+O_OS	CIR_Connect_006		
2		SCF!TC_InvokeReq	CIR_Connect_006 (1,51)		
3		SCF!TC_EndReq	C_TC_EndReqBasic (51, basic)		
4		SigConB?SetupReq	C_SetupReq({ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1000'H, originalCalledNumber '1001'H, redirectingNumber '3000'H, redirectingInformation 'AAAA'H })	(PASS)	
5		+SigConA_Release_thenB			
6		SCF?TC_AbortInd	C_TC_AbortInd(51)	INCONC	
7		SigCon_B?SetupReq	C_SetupReq({ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1001'H })	INCONC	
8		SCF?TCAPFailureInd	C_TCAPFailureInd_003	INCONC	
Detailed comments.					

**Table 3: Generated Constraint**

ASN.1 ASP Constraining Declaration	
Constraint Name	: C_SetupReq (callRef : SetupIRType)
ASP Type	: SetupReq
Purpose	:
Derivation Path	:
Comments	: OtherwiseFail
Constraint Value	
{ setupIRType 1 callRef }	
Detailed Comments	

Table 4: Test Case after post processing

Test Case Dynamic Behaviour					
Test Case Name : IN2_A_BASIC_CO_BV_03					
Group :					
Purpose :					
Configuration :					
Default : OtherwiseFail					
Comments :					
NR	Label	Behaviour Description	Constraints Ref	Verdict	Comments
1		+O_OS			
2		SCF!TC_InvokeReq	CIR_Connect_006 (1,51)		
3		SCF!TC_EndReq	C_TC_EndReqBasic (51, basic)		
		START ssfT			
4		SigConB?SetupReq	C_SetupReq({ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1000'H, originalCalledNumber '1001'H, redirectingNumber '3000'H, redirectingInfomration 'AAAA'H })	(PASS)	
		CANCEL ssfT			
5		+SigConA_Release_thenB			
Test Case Dynamic Behaviour					
6		SCF?TC_AbortInd	C_TC_AbortInd(51)		
		CANCEL ssfT		INCONC	
7		SigCon_B?SetupReq	C_SetupReq({ callRef 2, calledPartyNumber '2001'H, callingPartyNumber '1001'H })		
		CANCEL ssfT		INCONC	
8		SCF?TCAPFailureInd	C_TCAPFailureInd_003	INCONC	
		CANCEL ssfT		INCONC	
		?TIMEOUT ssfT		INCONC	
Detailed Comments.					

## 4.7 Expenses

Considering the experiences gained with the application of CATG for INAP (performed by STF 100), this subclause contains a summary of the resources required for the different tasks.

Initially, some considerable resource was spent to install and refine the tools and the CATG methodology. Resources were also spent to update the SDL model and already developed test purposes due to changes to the protocol made by the responsible TB (SPS3). Note that the development of the test suite was started before the completion of the protocol, allowing errors detected during the TP development to be corrected in the protocol before approval.

At the time of writing this report, STF 100 has developed 264 test purposes and test cases. 188 test purposes and test cases was developed using CATG, and 75 test purposes and test cases was developed manually following the 'traditional' ISO 9646 method. The reason for developing test purposes and test cases manually was that the SDL specification does not include error handling. Consequently, the tests for error handling had to be developed manually. The resources spent for the different tasks are the following (the figures for PT 53 is extrapolated from the Monthly Administrative Reports):

**Table 5: Resources required**

Task	Hours spent for CS-2 ATS (CATG)	PT 53 (Manual development of CS-1 ATS)
Validation and changes made to the SDL model, deployment/refinement of the methodology, preparatory work.	520	N/A
Development of test purposes (264)	1 480 5 hours/TP	3 hours/TP
Automatic generation of test cases	few minutes to start the tool	N/A
Post processing of automatically generated test suite (188 test cases)	40 (10 mins/TC)	N/A
Manual development of test cases (75 test cases)	80 (1 hour/TC)	1.5 hours/TC
Documentation	464 (1.8 hours/TP&TC)	2 hours/TP&TC
<i>Total</i>	<i>2 584</i>	-

When compared to manual development (PT 53), the CATG reduces the effort to produce the test cases with approximately 85 % (10 mins as compared to 1,5 hours per test case). The time required to produce the test purposes were 2 hours longer per test purpose for CATG as compared to manual development. The reason is the additional effort required for generating MSCs (approx. 10 mins/TP), in applicable cases correct the test purpose if not in accordance with the protocol (SDL model), but also due to the much increased complexity of the protocol itself.

Due to the improved handling of the signalling, organizations with an interest in INAP has decided not to make any further use of the manually developed test suite and instead use the automatically generated test suite.

NOTE: The manually developed test suite disregarded the signalling events and assigned PASS based on time-out events at the INAP interface.

---

## 5 Other Experiences

### 5.1 Experiences of using CATG for B-ISDN test suites

STF 87 has produced over 1 400 test cases in two Abstract Test Suites for testing B-ISDN [1] point-multipoint connections. Both test suites were produced manually. The ATS for the network side (EN 300 771-6 [2]) comprises over 800 test cases.

As part of its task STF 87 evaluated various CATG tools including the Autolink tool. Autolink was evaluated by writing several test purposes according to the methodology described in the present document and by attempting to generate test cases from these test purposes. The results were compared with the manually produced test cases.

Table 6 summarizes the experiences of this evaluation (the unit of time used is one man-hour). The figures in the "CATG" column are estimates extrapolated from the case study (except for item 1 which is the actual time spent adapting the SDL model in order to do the evaluation). The figures in the "Manual" column are the actual man-hours obtained from STF 87 project statistics.

**Table 6: Experiences with CATG for B-ISDN**

	Task	CATG	Manual
1	Adaption/validation of SDL model for CATG	160	-
2	Development of Test Purposes	140 (800 informal TPs) 40 (200 formal TPs as MSCs) (10 mins/TP)	140 (800 informal TPs) (10 mins/TP)
3	Validation of TPs against SDL model by simulation	-	140 (10 mins/TP)
4	Production of ATS (TTCN)	16 (5 mins/TC)	320 (20 - 30 mins/TC)
5	Post processing	100 (30mins/TC for 200 TCs)	-
6	Manual development of additional TCs	220 (additional 600 invalid TCs)	-
	Total	676 man-hours	600 man-hours

As can be seen, it is estimated that using Autolink would have taken slightly longer. This is due to the additional effort in adding detail necessary for CATG to the SDL model (row 1).

However, the following points should not be overlooked:

Out of a total of 800 test cases about 600 are for invalid behaviour - these cannot be generated automatically from the SDL (unless invalid behaviour is added to the model, which in this case was not feasible) and should be produced manually. Thus in row 4 only 200 of the test cases were produced using Autolink and in row 5 the 220 hours is the manual effort needed to produce the additional invalid test cases.

Note that items 1 and 7 in the list below are necessary because it was considered infeasible to make the SDL model more complex purely for the sake of CATG. If an even more detailed model was used in other contexts then the additional work might be considered a good investment.

The estimated overhead for the manual post-processing that needs to be done after the automatic test case generation is probably conservative. In some cases it would not be unreasonable to estimate 1hr/test case. In the B-ISDN example the typical postprocessing involved would be:

#### Major tasks

- 1) splitting of the non-concurrent (i.e. interleaved) test cases produced by the tool into parallel TTCN behaviour trees and the addition of Configuration tables;
- 2) addition of missing information in the Constraints (certain parameters are not modelled in the SDL specification);
- 3) addition of several pre-ambles (the AAL was not fully modelled in the SDL specification);
- 4) addition of Test Suite Parameters.

#### Minor tasks

- 1) re-formatting and re-structuring of the ASN.1 constraints for readability;
- 2) addition of test steps (not strictly necessary, but reduces the size of the test suite and helps readability);
- 3) addition of comments.

By using specially-written scripts to do some of the post-processing it is possible that the overhead could be reduced. There are no figures currently available to indicate what improvements might be expected.



The performance aspects of the Autolink were disappointing. A single test case of medium complexity took over 40 hours to generate. Simpler test cases took anything from ½ - 2hrs. The reason for this has not been investigated. It is probable that with more powerful hardware (top of the range Sparc) that these times can be significantly reduced. The hardware used in this evaluation was a Sun Sparc 20 with 128 MB RAM and 256 MB swap.

Finally, the quality of the CATG test cases with respect to execution in a real test system against a proper IUT has not been determined.

---

## History

<b>Document history</b>		
V1.1.1	February 1999	Publication