Final draft ETSI EN 301 931-4 V1.1.2 (2001-07)

European Standard (Telecommunications series)

Intelligent Network (IN); Intelligent Network Capability Set 3 (CS3); Intelligent Network Application Protocol (INAP); Protocol specification; Part 4: SDLs for SCF-SSF interface

Reference
DEN/SPAN-03063/1-4

Keywords
CS3, CTM, IN, INAP, protocol, UPT

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - 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

Contents

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 ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available 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 ETSI 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 European Standard (Telecommunications series) has been produced by ETSI Technical Committee Services and Protocols for Advanced Networks (SPAN), and is now submitted for the Vote phase of the ETSI standards Two-step Approval Procedure.

The present document is part 4 of a multi-part deliverable covering Intelligent Network (IN); Intelligent Network Application Protocol (INAP); Capability Set 3 (CS3); Protocol specification as identified below:

Part 1: "Common aspects";

Part 2: "SCF-SSF interface";

Part 3: "SCF-SRF interface";

Part 4: "SDLs for SCF-SSF interface".

The present document and parts 1 to 3 define the Intelligent Network (IN) Application Protocol (INAP) for IN Capability Set 3 (IN CS-3). The present document and parts 1 to 3 define the INAP for IN CS-3 based upon ETSI Core INAP CS-2 (EN 301 140-1) and ITU-T IN CS3 Recommendation Q.1238 (1999).

The structure of the present document and parts 1 to 3 follows the ITU-T Recommendation Q.1238 rather than that usual for an ETSI deliverable.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

1 Scope

The present document is part 4 of the multi-part deliverable for IN Capability Set-3. The present document specifies the protocol on the SSF-SCF interface in SDL, and provides an overview of the structure of the SDL model.

2 References

All documents referred to in the present document are identified in EN 301 931-1.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in EN 301 931-1 apply.

3.2 Abbreviations

For the purposes of the present document, the abbreviations and acronyms given in EN 301 931-1 apply.

4 Introduction to the INAP CS3 SDL model

4.1 Scope of the CS-3 model

The SDL model developed for CoreINAP CS-3 is based upon that produced for CS-2. However, there have been extensive revisions to the internals of the CS-2 model:

- the CS-1 and CS-2 models have been combined;
- the InterfaceHandler process has been removed, replaced by the SSME-Control which only handles SSF-SCF communication;
- the BCP has been added to manage all signalling to/from the BCSMs (previously passed through the InterfaceHandler);
- as a consequence, the CS process is updated based on the DPs received from the BCSMs, not on the signalling;
- the Trigger Table has been re-designed, and effort has gone into using RemoteProcedures, refining data structures and ensuring correct initialization and use of data structures, all with the objective of making the SDL model simulate more efficiently;
- the FIMController and FIMAgent have been added to model Multiple Point of Control;
- functionality has been extended to model new behaviour in CS-3.

The normative information provided by the SDL diagrams is the dynamic behaviour at the external interfaces. The internal behaviour and structure have been introduced for modelling purposes. No SDL specification of the SCF is provided; the behaviour of the SCF is specified in clause 9 of EN 301 931-2.

The SDL model specifies precisely and unambiguously the external behaviour and the interworking between the different functional entities: SSME, CSA, CS, SSF-FSM, BCSM. However, the internal behaviour is only for information and the SDL model does not impose any requirements on the internal structure of an implementation. The data structures used are imported from clause 14 of EN 301 931-2 and from clause 12 of EN 301 931-3 (ASN.1 definitions).

The SDLs are a full data model, and therefore can be used for simulation purposes. There are however, some limitations with the SDL model. The model only concerns the SCF-SSF interface. The model s intended to cover call party handling only, it is not a full model of the SCF-SSF interface. It is intended to illustrate the CSCV transitions on signalling events, Detection Points and SCF operations. Where the model does go beyond CPH, in making reference to charging operations etc., then these are not fully modelled, in particular if they have no impact on CPH and on CSCV transitions.

The conformance test suite for Core INAP CS3 has been automatically derived from the model. The model can also be used as a platform for service emulation etc.

4.2 The Information Model used in the CS-3 SDL model

Figure 1 shows the IN CS-3 information model. The various objects in the information model are defined in clause 6.6.2.1 of EN 301 931-2. There are zero, one or many Call Segments in one Call Segment Association. There is a one to one correspondence between Call Segment and Connection Point; a Call Segment contains 0 or 1 ControllingLeg, and 0 or more Passive Legs, and so on. There are two objects of type BCSM, the OriginatingBCSM and the TerminatingBCSM.

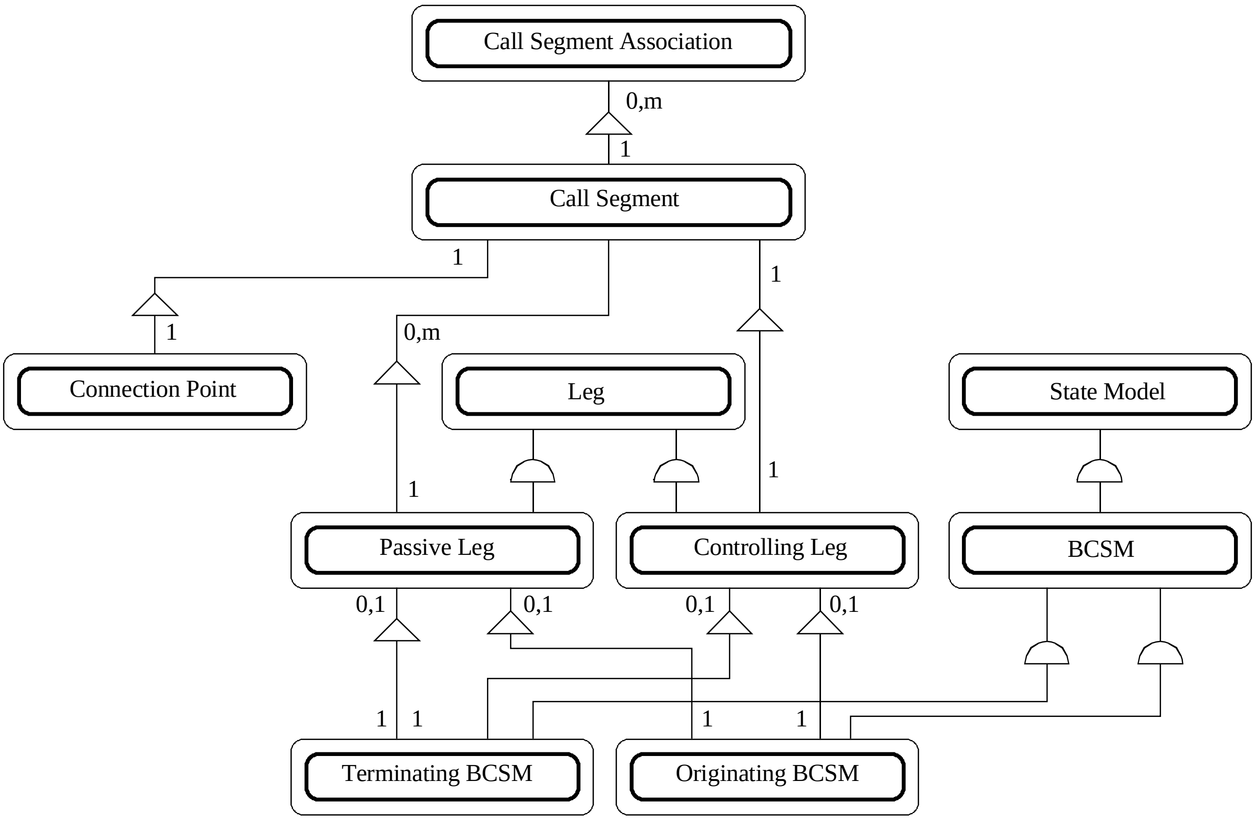

Figure 1: IN CS-3 information model

Figure 2 shows the SDL model of CS3 SSF/CCF. The objects in the information model of figure 1 map to the SDL model of figure 2 in the following way.

The Call Segment Association object is modelled by a process type in the SDL model. The CallSegmentAssociation manages:

- the creation of CallSegments; and
- the dialogue with the SCF (via the SSME Controller and TCAP).

The Call Segment object is modelled using two process types in the SDL model: CallSegment and SSF-FSM.

The SSF-FSM process type corresponds with the FSM for Call Segment described in clause 8.2.2 of EN 301 931-2.

The process type CallSegment manages the:

- IDs of the connected legs (connection view). This data structure models the Connection Point object;
- creation of the SSF-FSM;
- filtering of DPs;
- processing of IN operations changing the connection view (CPH-operations, CON, ICA).

The CallSegment process defines states for the CS. This approach is different from the CVS approach in ITU-T Recommendation Q.1224: the CVS approach explicitly names combinations of associated CSs in a particular state. The number of such combinations is infinite, since the number of CSs in a CSA has no upper bound. Instead, in the SDLs the Call Segment states are given a name (e.g. Stable-2-Party) and the CSA is represented as a set of associated CSs, possibly in different states.

The process type SSF-FSM manages the:

- processing of IN operations;
- handling of DPs (EDPs and TDPs).

The Connection Point object is modelled by a data structure as described above.

The Leg object is also a data structure within the process type CallSegment. It is identical to the data structures of Passive Leg and Controlling Leg. The data structure contains the status of the leg and the type of BCSM associated with the leg. The LegID is used by another process (the FIMAgent) to associate the leg with a particular BCSM.

T-BCSM and O-BCSM are both objects of the class BCSM in the information model. The SDL model is not exactly constructed this way. Since the O-BCSM and T-BCSM significantly differ, they are modelled in two completely independent process types.

The SDL model contains additional process types, more or less related to the various entities described in EN 301 931-2.

The SSME-Control provides:

- Processing of the operations MoveCallSegments and CreateCSA.
- Handling of the dialogue with the SCF.
- The passing of primitives between the SSF-SCF Interface and the CSA.
- The processing of IN CS-3 management operations.
- The handling of static arming of TDPs.
- The "Master" TDP table.
- The processing of ManageTriggerData.

The FIMController provides:

- supervises the creation of FIMAgents.

NOTE 1: Only one FIM_Manager can exist per switch, i.e. two instances of this half process, which are closely linked (not shown).

The FIM Agent process manages:

- The feature interaction between different SCFs.
- The passing of DPs and PIC primitives between the BCSMs and the CallSegmentAssociation.
- The handling of TDPs.
- Creation/removal of BCSM chains.
- One FIMAgent exists per call.

The BCP process type manages:

- The creation of O-BCSMs and T-BCSMs.
- The passing of Signalling Control primitives between the SigCon and the BCSMs.
- The passing of DPs and PICs between the BCSMs and the FIM.

This Process only models half of the BCP.

NOTE 2: Only one BCP can exist per switch, i.e. two instances of this half process, which are closely linked (not shown).

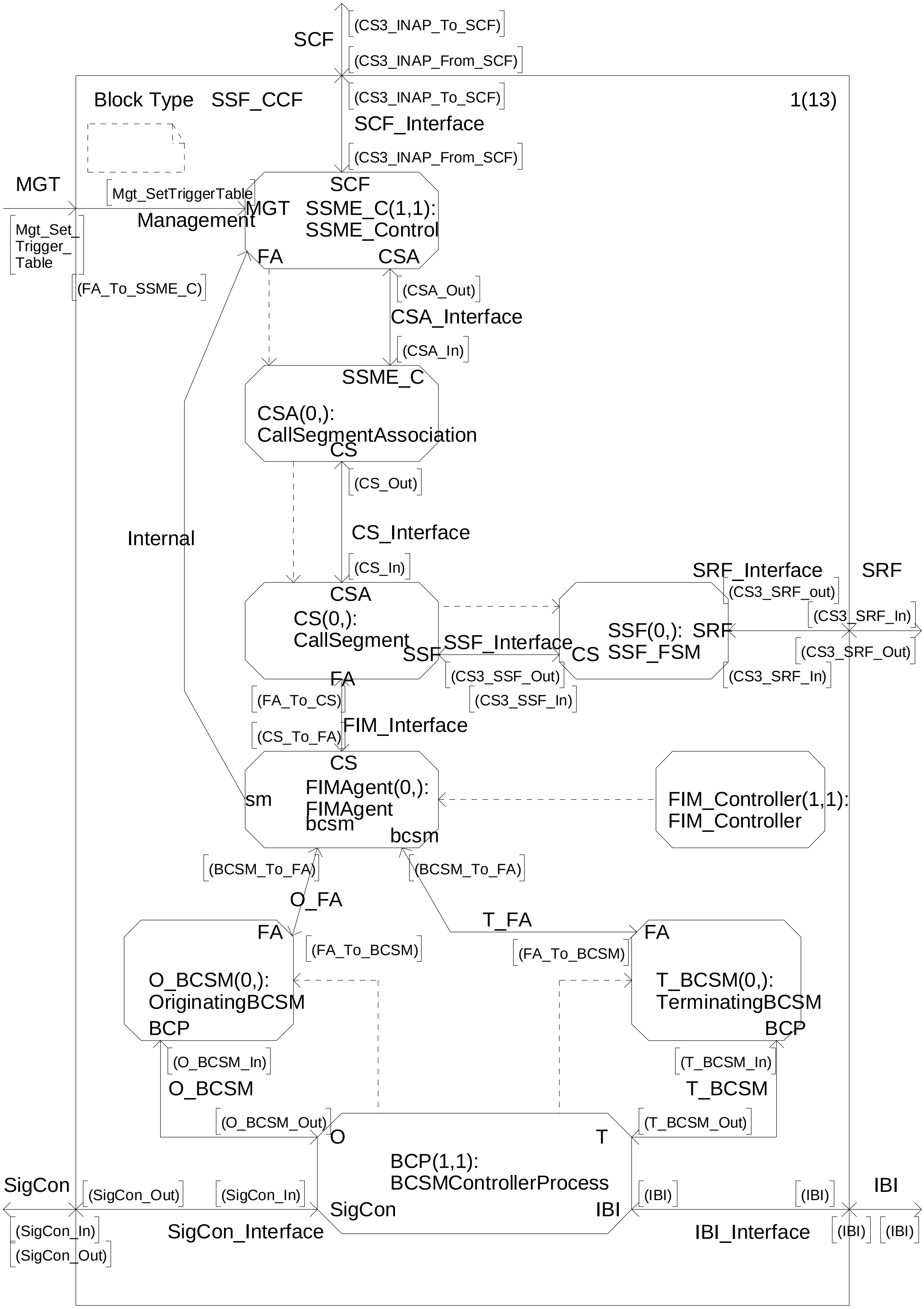

Figure 2: The INAP IN CS3 SDL model, half-call view

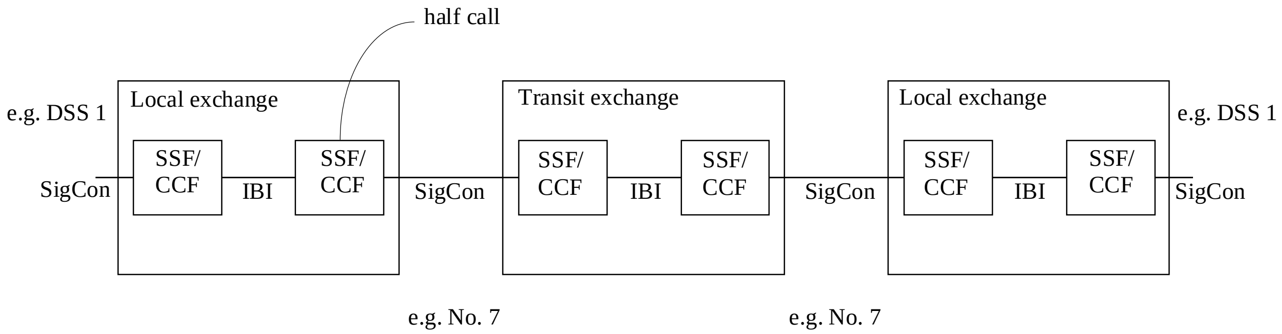

Figure 3: General example scenario for INAP showing the SSF/CCF only

Figure 3 shows a general example scenario for IN related to the SDL model. The Intra-BCSM Interface (IBI) interface is an internal intra BCSM interface between two half calls.

The full SDL model consists of two half calls as indicated in figure 4. Figure 2 is a refinement of the SSF/CCF block. It shows one half call under control of the SCF, including the service switching and CCFs. In order to operate the model, the Block SSF_CCF needs to be doubled to get two half calls. In this way the full functionality of the interworking between the O-BCSM and T-BCSM can be simulated.

The model has five interfaces. The SigCon interface could, for example, be a DSS1 interface. The Messages on SigCon are abstract and have to be mapped to a real protocol, e.g. DSS1, in a concrete case. IBI is completely internal to a switch. The INAP interface to the SCF is the one with the standardized INAP messages. The MGT interface is an interface internal to the model for performing management functions, such as static trigger arming. The SRF interface is an interface internal to the model for communication with an associated SRF. Annex D of EN 301 931-2 shows examples of the abstract primitive signals, their interfaces and possible mappings to a real protocol such as ISUP and DSS1.

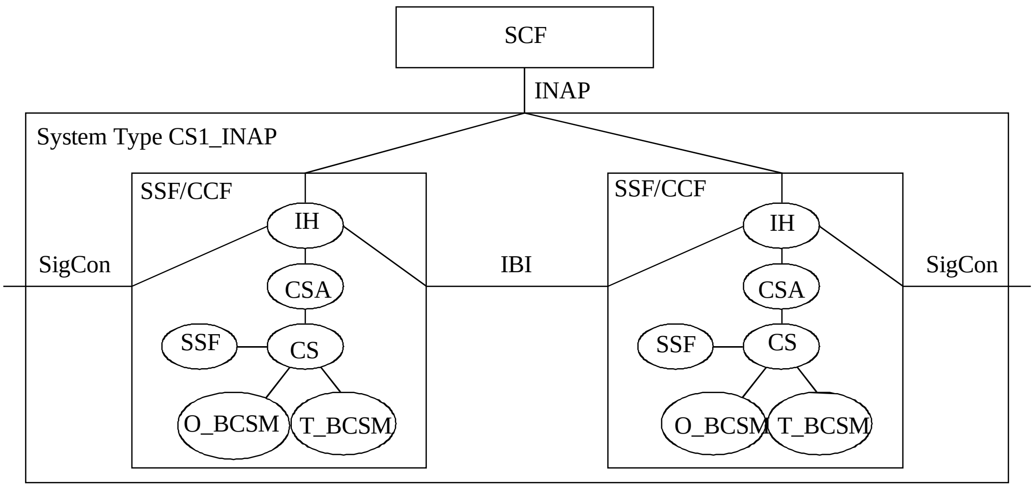

Figure 4: Two half calls (with simplified objects)

4.3 Example for the interworking of the SSF/CCF SDL processes

The message sequence charts (MSC) in the attached files shows the interworking between the various process instances in the SDL model after a SetupInd is received through the SigCon interface.

The first MSC, in file Process_Interworking_1BCSM.pdf demonstrates the interworking of the various processes in the SDL model on establishment of a call and TDP triggering, when the SDL model is initialized to model only Single Point of Control.

The second MSC, in file Process_Interworking_3BCSMs.pdf demonstrates the interworking of the various processes in the SDL model on establishment of a call and TDP triggering for 2 services, when the SDL model is initialized to model Multiple Point of Control.

5 Connection View State Transitions

A description of the Connection View State Transitions can be found in EN 301 931-2, clause 6.6.

6 Basic Call Controller Process

A description of the Basic Call Controller Process can be found in EN 301 931-2, clause 6.2.1.

7 Modelling of the FIM and Multiple Point of Control

7.1 Overview of the FIM

The functionality of the Feature Interactions Manager (FIM) is described in CoreINAP CS-3 in less detail than for other objects in the INAP model. The SDL model of the FIM therefore does not correspond exactly with that described in EN 301 931-2. In particular, the SDL model of the FIM is only concerned with interactions between IN services; there is no facility to model interactions between IN and non-IN services.

As explained in clause 4, the FIM has been split into 2 for the purposes of modelling: the FIM Controller, permanently present, and the FIM Agent, an instance of which is created in the half call model for each incoming or outgoing call. The FIM Controller handles the creation of FIMAgent process instances. The Master TDP Table is held by the SSME-Control, and a copy is retrieved by each FIM Agent process instance when it is created. This enables the modelling of call unrelated trigger management (ManageTriggerData and CreateOrRemoveTriggerData operations) in the SSME-Control, and all TDP processing, including call related trigger management (SetServiceProfile) in the FIMAgent.

The FIMAgent maintains a data structure which relates each BCSM to a corresponding CSAID/legID pair and vice versa. All TDP processing is performed in the FIMAgent, and it also assigns the legID for TDPs (always 1 or 2). In this way, the BCSM processes are unaware of their legIDs.

7.2 Approach to Multiple Point of Control

In clause 6.5.1.2.2 of EN 301 931-2, various rules are provided to cover TDP and EDP processing in a Multiple Point of Control Scenario. These rules do not cover all possibilities regarding behaviour in an MPC scenario. There is, however, a general objective:

If there are more than one SSF - SCF control relationships acting on the same Originating BCSM or Terminating BCSM, then the event detection point processing requested by any of the involved service logics (SLPIs) shall be performed in the same way as if triggering had occurred in different Originating BCSMs respectively in different Terminating BCSM's, which are separated by a Network Node interface.

It was decided that the most efficient and flexible way to model MPC would be to provide a chain of BCSMs. The BCSMs are chained, so the first BCSM is controlled by the first service, the second by the second service, etc. The idea is that in this way, the general objective of MPC is reached in a very natural way: *the event detection point processing requested by any of the involved service logics (SLPIs) shall be performed in the same way as if triggering had occurred in different Originating BCSMs respectively in different Terminating BCSM's*. the only difference being that the BCSMs are not in different network nodes, but in the same network node.

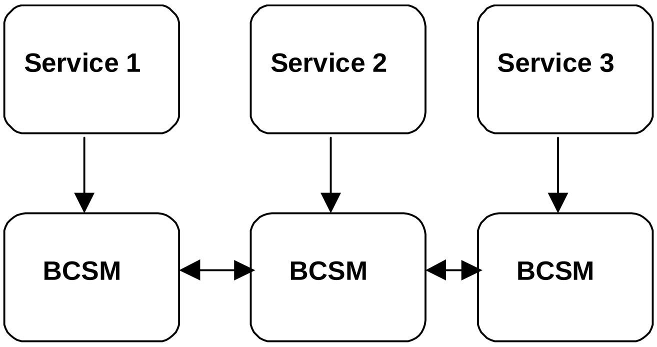

Figure 5: BCSM Chaining

This scenario does not automatically apply the EDP reporting rule whereby EDPs are reported in a forward/backward direction according to the order of triggering of the SSF instance - it is necessary to trigger the first service from the first BSCM in the chain, the second from the second BCSM etc. But this scenario does provide the flexibility to modify the order of triggering, and forces those using the SDL model to carefully consider how their services are to be deployed. This increases the usefulness of the SDL model as an evaluation tool. Depending on whether Single Point of Control or Multiple Point of Control is to be modelled, the SDL model can be initialized with 1 or more BCSMs in each chain. The maximum number of BCSMs is simply a Constant defined in the ASN.1 - easily modified by a user if necessary.

No additional primitives have been created to allow the BCSMs to communicate along a chain - the BCP performs any necessary primitive translation. A number of procedures and internal primitives have been added between the BCP, the FIMAgent and the CS to manage the creation and maintenance of the BCSM chain. Extensive use of SDL Remote Procedures has been used, to enable the FIM Agent to control the creation and removal of the chain of BCSMs.

The BCP needs to maintain a connection point data structure, in order to correctly process the routeing of primitives from the BCSMs, to either the IBI, the Signalling Termination, to another BCSM in the chain, or another BCSM if in the CSCV states Stable_Multi_Party, Stable_Multi_Passive_Party or Transfer. The BCP will maintain this data structure based on signalling to/from the BCSMs. The CS will inform the FIM of certain CPH operations such as SplitLeg, DisconnectLeg and MoveLeg/MergeCallSegments, the FIM uses this information to update the BCPs connection point data structures using remote procedures.

8 SDLs

The SDL diagrams are contained in an Adobe Portable Document Format™ file (CS3_SDL_v81.pdf contained in archive en_30193104v010102v0.ZIP) (CS3_SDL_v81.pr also provided) which accompanies the present document.

History

Document history				
V1.1.1	August 2000	Public Enquiry	PE 20001215:	2000-08-16 to 2000-12-15
V1.1.2	July 2001	Vote	V 20010831:	2001-07-02 to 2001-08-31